UNIVERSITY OF CALIFORNIA,
IRVINE

Neural Networks in Economics

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Economics

by

Alexander Parret

Dissertation Committee:
Professor Matthew Harding, Chair
Professor Matthew Freedman
Associate Professor Yingying Dong
Assistant Professor Ying-Ying Lee

2020

DEDICATION

To my parents Charles and Victoria Parret for their unending love and support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

ACKNOWLEDGMENTS

I would like to thank Matthew Harding for his support and guidance over the past few years. I was very fortunate to enter my second year at UC Irvine at the same time he joined the faculty. Our mutual interest in neural networks and deep learning is niche in econometrics and it was incredibly valuable to have someone equally passionate about these estimators as an advisor. In addition, his creation of the Deep Data Lab provided an incubator for many intellectually stimulating discussions.

I am grateful to Ying-Ying Lee for her guidance and always pointing me in a useful direction. I owe many thanks to Yingying Dong for pushing me to consider the practicalities of my work and to always consider the applied researcher. I thank Matt Freedman for his insight and encouragement.

I am forever grateful to my colleagues at UC Irvine for their friendship and encouragement. Our weekly gatherings were a highlight of my week and a huge source of rejuvenation. A special thanks to Elizabeth Maloney for always listening to me prattle on about my work and providing me with invaluable feedback.

I am incredibly thankful to the Economics faculty at the University of San Francisco who embraced me and provided an amazing learning environment. I thank Bruce Wydick for inspiring me to change my major to Economics, Sunny Wong for his boundless energy and enthusiasm, and Man-Liu Lau for his encouragement. A special thanks to Michael Jonas who introduced me to Econometrics and fostered my passion for the subject.

I would like to thank the many people who are responsible for my personal and intellectual development. Last, but far from least, I wish to thank my parents and grandparents for their love and support over the years.

CURRICULUM VITAE

Alexander Parret

EDUCATION

Doctor of Philosophy in Economics	**2020**
University of California, Irvine	*Irvine, CA*
Masters of Arts in Economics	**2013**
University of San Francisco	*San Francisco, CA*
Bachelor of Arts in Economics	**2012**
University of San Francisco	*San Francisco, CA*

ABSTRACT OF THE DISSERTATION

Neural Networks in Economics

By

Alexander Parret

Doctor of Philosophy in Economics

University of California, Irvine, 2020

Professor Matthew Harding, Chair

The chapters of this dissertation explore the theoretical and empirical potential of neural networks and deep learning as estimation techniques in economics. The first chapter provides a novel approximation result for two hidden layer neural networks that makes clear the trade-off between width and depth. I leverage this result to provide consistency and $o_p(n^{-1/4})$ convergence rates for this estimator and demonstrate its flexibility in finite samples. In addition, I introduce a new algorithm called cross-training that allows construction of asymptotic confidence intervals for linear functionals. In the second chapter I provide a new neural network designed for a panel data setting with a common index. I allow for unobserved heterogeneity to enter in the form of additive cross-sectional fixed effects and provide a correction for the incidental parameter bias by re-centering the score. I apply this estimator to the demand for cigarettes in the United States. In the third and final chapter I explore the role of autoencoders as a dimensionality reduction technique when outcomes are binary. The autoencoder outperforms other dimensionality reduction techniques, like principal components analysis, in uncovering latent choice probabilities. This estimator is applied in a consumer segmentation exercise.

Chapter 1

Consistency and Convergence Rates for Extended Neural Networks

This chapter establishes a new approximation result for two hidden layer (extended) neural networks. I leverage this approximation result to provide new consistency and $o_p(n^{-1/4})$ convergence rate results for this estimator under a least squares objective. In addition, I propose a novel and computationally feasible inference procedure for neural networks and deep learning designed to construct pointwise valid confidence bands for linear functionals of the parameters.

1.1 Introduction

Neural networks and deep[1] learning have increasingly become attractive estimators for economists. While the current state of theoretical results is limited, what we do know suggests particular neural networks share many desirable properties with commonly used

[1] Neural networks with more than one hidden layer.

estimators, e.g., sieve estimators[2] (Chen, 2007). Neural networks use nonlinear transformations of affine combinations to construct functions from covariates. These nonlinear transformations are referred to as activation functions and play a large role in both theoretical and practical use. The current theoretical literature provides results for single hidden layer neural networks with smooth[3] activation functions (White (1990), Chen and White (1999), Chen (2007)) and very recently deep neural networks with piecewise linear activation functions[4] (Farrell, Liang, and Misra, 2018). However, practitioners have experimented with a variety of neural network architectures and found that increasing depth provides benefits in accuracy and generalizability (Safran and Shamir (2017), Telgarsky (2016)).

This work extends the known theoretical results for single hidden layers with smooth activation functions to those with a second hidden layer. The first major hurdle to overcome is an approximation result that depends on the width of the first and the second layer. I provide an approximation theorem when the underlying function admits a suitable representation. Heuristically, I separate the approximation into two components. The first approximates a sequence of adaptive basis vectors, while the second projects the observed outcome onto these vectors. This decomposition allows for the combination of known approximation results from the single hidden layer neural network literature with those used in series estimation. In addition to allowing for more flexibility in the approximation, the benefits of the proposed decomposition can be intuitively justified as further breaking up the function into a sequence of much simpler functions. In the single hidden layer network, one must rely on a single sequence of nonlinear transformations to reconstruct the unknown function. A second hidden layer allows the network to construct much simpler functions in this first layer and combine them in the second.

Once an approximation result that depends directly on the width of each layer is available

[2] Some of the most commonly used sieve estimators are series and splines.

[3] These functions typically belong to the sigmoid family, but may also include radial basis functions (Chen, Racine, and Swanson, 2001) or Fourier series (Gallant and White, 1988)

[4] In the literature these are referred to as rectified linear units or ReLU.

it is possible to utilize general results from the sieve literature to establish convergence rates. I explicitly provide consistency of the extended neural network and convergence rates comparable to the single layer case. The key component to achieving fast enough rates is restricting the growth rate of the parameter space. The final rate is achieved through a delicate balance between the complexity of the estimator, measured by metric entropy, and the approximation error.

The final contribution of this chapter is the introduction of cross-training. This algorithm, in conjunction with, sample splitting facilitates the construction of pointwise confidence bands for linear functionals. This modification uses ideas from other computational approaches like the jackknife and cross-fitting to stabilize the local behavior of the optima. The underlying idea is to augment the parameter space by auxiliary parameters that depend only on a subset of the data. These parameters are then invariant to local perturbations with respect to the left-out observations. Cross-training is shown, in simulations, to allow asymptotic variance calculations for both classical and extended neural networks. In contrast estimating these models without cross-training results in solutions where plug-in estimates of the variance are unstable in finite samples. In addition, because cross-fitting utilizes a fixed number of splits, the asymptotic properties related to consistency and convergence rate are unaffected.

The remaining of the chapter proceeds as follows. The first section (1.2) gives a brief overview of the relevant literature. The next section (1.3) presents the model and assumptions along with defining neural networks. Section 1.4 presents and discusses the approximation result for extended neural networks. The consistency and rate results are presented in section 1.5. I then discuss asymptotic normality for linear functionals in section 1.6 and introduce cross-training in section 1.6.4. The remaining sections (1.7 and 1.8) provide simulation results and conclude.

1.2 Literature Review

Much of the success statistics and econometrics has in describing the behavior of neural networks, and other adaptive estimators, lies in the method of sieves (Grenander, 1981). However, this literature has primarily focused on what is termed "linear" sieves while neural networks belong to the class of "nonlinear" sieves (Chen, 2007). A highly attractive property of linear sieves is the simplicity of their construction. Linear sieves utilize a fixed transformation of the covariates space, e.g., power series estimation, where the order of the polynomial grows with the sample size. These methods have been shown to work very well in practice and are well understood in the theoretical literature, e.g., Newey (1994) and Newey (1997). The nonlinear sieve literature is much less developed, and the construction is more intricate. In contrast to linear sieves, the nonlinear sieve is data adaptive, e.g., free-knot splines[5]. Neural networks fall into the class of nonlinear sieves as they operate in a similar way to a series estimator but construct a data-driven transformation of the covariates space rather than a fixed one.

The most relevant general framework for sieve and nonlinear sieve estimation is the sequence of papers Shen and Wong (1994), Chen and Shen (1998) and Shen (1997) which provide the conditions for calculating convergence rates under a variety of sampling assumptions. In the particular case of neural networks I build upon Chen and White (1999) who[6] established $o_p(n^{-1/4})$ convergence rates for classical neural networks. Their results hinge on one's ability to impose sufficient control on the complexity of the network while taking advantage of approximation results delivered by Makovoz (1996).

In more recent work Farrell, Liang, and Misra (2018) established convergence rates for deep

[5] These are splines where the location of the knots is a parameter of the model rather than a pre-determined value.

[6] These rate results were the culmination of a decade of work on the statistical properties of neural networks largely beginning with White (1990), Hornik, Stinchcombe, and White (1989), and Hornik, Stinchcombe, and White (1990).

neural networks with piecewise linear activation functions. I view this work as complimentary as the activation function plays an important role in empirical work. There are clear trade-offs between the approaches. If one believes the true functional form is smooth, then using smooth activation functions in finite samples will have better approximation capabilities. However, in very high dimensions smooth activation functions can become computationally difficult to work with. I discuss this comparison further in Appendix A.2.

1.3 Model

I consider the nonparametric regression model where: $\{z_i\}_{i=1}^n = \{y_i, x_i'\}_{i=1}^n$ is a sequence of random vectors. The outcome of interest y_i is sampled from:

$$y_i = g_0(x_i) + e_i, \qquad \mathbb{E}[e_i|x_i] = 0, \qquad \mathbb{E}[e_i^2|x_i] = \sigma^2(x_i) < \infty \tag{1.1}$$

The dependent variable $y_i \in \mathcal{Y} \subset \mathbb{R}$ is explained by some unknown function $g_0(\cdot)$ of observed covariates $x_i \in \mathcal{X} \subset \mathbb{R}^p$. I impose the following restrictions on the function space $\mathcal{G}$ in addition to the sampling properties of $\{z_i\}$:

A1.3.1 The random vectors $\{z_i\}_{i=1}^n = \{y_i, x_i'\}_{i=1}^n$ are i.i.d.. In addition, $y_i \in \mathcal{Y} \subset \mathbb{R}$ and $x_i \in \mathcal{X} \subset \mathbb{R}^p$ where $\mathcal{X}$ and $\mathcal{Y}$ are compactly supported.

A1.3.2 The unknown function g_0 admits the representation: $g_0(x_i) = \sum_{k=1}^{\infty} \beta_k \psi_k(x_i)$ where $\Psi = \{\psi_1, \psi_2, \dots\}$ is a bounded sequence. The elements of β_k are ordered such that: $|\beta_k| \leq Ck^{-a}$ for some constant C and $a > 1$.

A1.3.3 Each $\psi_k \in \mathcal{W}_2^m(\mathcal{X})$ where $\mathcal{W}$ is a Sobelev space with m weak derivatives[7] and has a

[7] By application of Morrey's embedding theorem this space can be embedded into a Hölder space Λ^γ ($\mathcal{W}_2^m \hookrightarrow \Lambda^\gamma$) where $\gamma = m - p/2$ so long as the number of covariates $p < 2m$ where m is the number of weak derivatives

Fourier representation:

$$\psi_k(x_i) = \int \exp(i\delta' x_i) d\sigma_\psi(\delta) \tag{1.2}$$

where σ_ψ is a complex measure on $\mathbb{R}^p$ satisfying:

$$\int \max\left\{|\delta|, 1\right\}^{m+1} d\left|\sigma_\psi\right|(\delta) < \infty \tag{1.3}$$

The i.i.d. assumption in **A1.3.1** will be convenient for the inference approach outlined in section 1.6, but can be relaxed to weak dependence for the consistency and convergence rate results with some modifications. The conditions on the covariate support $\mathcal{X}$ are made in most of the nonparametric literature. However, the bound on y_i is somewhat irregular. This bound is typically present in the theory of neural networks and can essentially[8] be imposed by proper control over higher order moments as mentioned in Farrell, Liang, and Misra (2018). Without loss of generality I take $\mathcal{X} = [0,1]^p$ and $\mathcal{Y} = [0,1]$ to be the unit intervals.

Assumption **A1.3.2** is a standard assumption in the series estimation literature. Functions admitting such an expansion are commonly assumed in sieve estimation. Without loss of generality I take $\psi_k \in [0,1]$ $\forall k$. In addition, this form of coefficient decay is standard in the series estimation literature and is necessary for the approximation error to vanish asymptotically Newey (1997). I impose the condition A**1.3.3** to utilize approximation results from Makovoz (1996) previously used in the neural network literature. In previous work, e.g., Chen and White (1999) this assumption is made on the function itself. Here I impose weaker assumptions by allowing the function to be a composition of functions satisfying this property. This difference allows for increased flexibility in high-dimensional settings where some or many elements of x_i can be omitted from estimation[9] of each ψ_k. It is important

[8] Alternatively one may introduce truncation arguments as in Shen (1997).

[9] I do not take advantage of this case in the theory as I focus on the fully connected model. The results that follow can then be thought of as upper bounds as the entropy of this estimator will decrease with fewer connections.

to note that the δ in **A1.3.2** refer to the same parameters from equations 1.4 and 1.5, introduced in the next section. An extensive list of the properties of such functions can be found in Barron (1993).

1.3.1 Neural Networks

I consider two specific neural networks. The first is a single hidden layer network:

$$g_n^{(nn)}(x_i) = \sum_{j=1}^{d} s(\tilde{x}_i'\delta_j)\gamma_j \tag{1.4}$$

where $\tilde{x}_i = (1, x_i')'$, $\gamma_j \in \mathbb{R}$ and $\delta_j \in \mathbb{R}^{p+1}$. The second is a deep network, the extended neural network, which appends an additional hidden layer to the classical case:

$$\begin{align} g_n^{(dnn)}(x_i) &= \sum_{k=1}^{K} s\left(\sum_{j=1}^{d} s(\tilde{x}_i'\delta_j)\gamma_{jk}\right)\beta_k \tag{1.5} \\ &= \sum_{k=1}^{K} p_{d,k}(x_i)\beta_k \tag{1.6} \end{align}$$

where $\beta_k \in \mathbb{R}$, $\delta_j \in \mathbb{R}^{p+1}$, and $\gamma_{jk} \in \mathbb{R}$. For notation and intuition I define the scaled inner sum $s\left(\sum_{j=1}^{d} s(\tilde{x}_i'\delta_j)\gamma_{jk}\right) \equiv p_{d,k}(x_i)$. This component is almost identical to the single layer network in equation 1.4, but γ_j is now a vector in $\mathbb{R}^K$. The notation for this component is labeled $p_{d,k}(\cdot)$ to resemble series estimation terminology used in Andrews (1991) and Newey (1994). The width in the first layer is determined by the subscript d. The function $s(\cdot)$ is the so-called activation function and is a priori specified[10]. As discussed in the introduction I focus on smooth activation functions $s(\cdot)$ belonging to the sigmoid class of functions:

C1.3.1 The function $s(t)$ is a sigmoid function if it is bounded on an interval $[a, b]$ for some

[10] This is in contrast to the projection pursuit regression estimator (Friedman and Stuetzle, 1981) where $s(\cdot)$ is estimated.

$a, b \in \mathbb{R}$ and satisfies the Lipschitz condition $|s(t) - s(t')| \leq L|t - t'|$

This choice has both theoretical and practical implications and was the dominant choice for both theoretical and empirical work prior to the introduction of ReLU, one of the most commonly used piecewise linear activation functions[11]. The theoretical advantages of sigmoid functions follow from being Lipschitz and uniformly bounded $\sup_{x \in \mathcal{X}} |s(x)| \leq 1$. These properties are useful in establishing consistency and fast enough convergence rate results for neural networks.

The practical implications for the choice of activation functions comes down to computation or beliefs about the unknown function of interest. If the researcher believes the true function is smooth, then using an activation function that is itself smooth will facilitate a more efficient approximation in finite samples. However, in very large networks the computational advantage from using the ReLU function can be quite large.

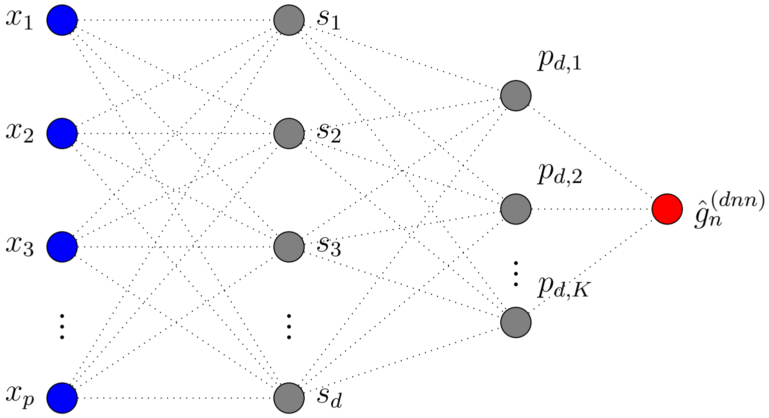


Figure 1.1: Graphical depiction of the extended neural network.

The primary difference between classical neural networks and deep learning is expanding the number of hidden layers. There has been much work attempting to justify the empirical

[11] It is important to note that the choice of smooth activation functions has gone out of favor in modern deep learning research. Most current work focuses on the rectified linear unit (ReLU) defined as $s(t) = 1_{t>0}t$. This switch is largely attributed to the difficulty of estimating networks with smooth activation functions. The gradients of such networks are more complicated than piecewise linear functions. I fully acknowledge that these issues exist but can be largely avoided through the usual nonparametric practice of attempting different parameter initializations.

success of adding additional layers both through simulations and theory. Unfortunately, most of the approximation theorems for deep learning are not functions of the model parameters with the notable exception Yarotsky (2017) leveraged by Farrell, Liang, and Misra (2018). This approximation result was a huge step forward, but only applies to networks utilizing the ReLU activation. The result presented in this chapter allows neural networks defined with smooth activation functions to take advantage of an additional layer.

To ensure the sieve space $\mathcal{G}_n^{dnn}$ is compact I make the following assumptions on the magnitude of the parameters:

A1.3.4 The parameters of the sieve space $\mathcal{G}_n^{dnn}$ satisfy the following bounds:

$$||\beta||_1 \leq \Delta_n, \quad \sum_{j=1}^{d_n} ||\delta_j||_1 \leq d_n \Delta_n, \quad \sum_{k=1}^{K_n} ||\gamma_k||_1 \leq K_n \Delta_n \tag{1.7}$$

where $\Delta_n, K_n, d_n \to \infty$ slowly with n

These bounds can be enforced by proper control of the width, K_n and d_n, or imposing constraints or penalties directly on the objective. These assumptions are the same as in White (1990) and Chen and White (1999) but include the additional bounds on $||\gamma_k||_1$. Heuristically, these conditions force the complexity of the estimator to be bounded for fixed values of n. The parameters, and thus the sieve space $\mathcal{G}_n$, will be allowed to grow with n such that $\mathcal{G}_n$ becomes dense in the original space $\mathcal{G}$.

1.4 Approximation

Under assumption **A1.3.3** Chen and White (1999) showed the approximation rate for $g_n^{(nn)}$ to functions satisfying **A1.3.3** is $O(d^{-1/2-1/(p+1)})$ under a weighted Sobolev norm by applying results from Makovoz (1996) to sigmoid type activation functions. I provide a new approximation theorem for $g_n^{(dnn)}$ that leverages the approximation results from Makovoz (1996) along with results from the series literature.

The unknown function can be represented by an infinite series of basis functions by assumption **A1.3.2**. Under assumption **A1.3.3** each term in the series can be approximated by a neural network. An approximation to the basis functions in the series are $\{p_{d,k}\}_{k=1}^{K}$ from the equation 1.5. However, unlike the classical neural network case there will now be two sources of approximation error. The first source is the approximation error due to estimating the basis vectors and the second is induced by truncating the series at K.

Theorem 1.1. *Suppose assumptions* **A1.3.2** *and* **A1.3.3** *are satisfied. The approximation error of $g_n^{(dnn)}$ to the target function g_0 satisfies:*

$$\left|\left|g_0 - \sum_{k=1}^{K} p_{d,k}\beta_k\right|\right| = O(d^{-1/2-1/(p+1)}\log K) + O(K^{-a}) \tag{1.8}$$

Proof. Under assumption **A1.3.2** one has:

$$\begin{aligned}\left|\left|g_0 - \sum_{k=1}^{K} p_{d,k}\beta_k\right|\right| &= \left|\left|\sum_{k=1}^{\infty}\psi_k\beta_k - \sum_{k=1}^{K} p_{d,k}\beta_k\right|\right| && (1.9)\\ &= \left|\left|\sum_{k=1}^{K}(\psi_k - p_{d,k})\beta_k + \sum_{k=K+1}^{\infty}\psi_k\beta_k\right|\right| && (1.10)\end{aligned}$$

Then by Minkowski's inequality:

$$\begin{aligned}\left|\left|\sum_{k=1}^{K}(\psi_k - p_{d,k})\beta_k + \sum_{k=K+1}^{\infty}\psi_k\beta_k\right|\right| &\leq ||(\psi_1 - p_{d,1})\beta_1|| + ||(\psi_2 - p_{d,2})\beta_2|| + \cdots && (1.11)\\ &+ ||(\psi_k - p_{d,k})\beta_k|| + \left|\left|\sum_{k=K+1}^{\infty}\psi_k\beta_k\right|\right| && (1.12)\end{aligned}$$

Each of the first K terms can be decomposed as:

$$||(\psi_k - p_{d,k})\beta_k|| \leq ||\psi_k - p_{d,k}||\,||\beta_k|| \tag{1.13}$$

Now note that $p_{d,k} = s\left(\sum_{j=1}^{d} s(\tilde{x}_i'\delta_j)\gamma_{jk}\right)$ which is $f_n^{(nn)}$ scaled to output units in $[0,1]$. Since $\psi_k \in [0,1]$ for any $k \in \{1,2,\ldots,K\}$, the approximation error of $||\psi_k - p_{d,k}|| = O(d^{-1/2-1/(p+1)})$ as in Makovoz (1996). This holds for each term in the sequence where for some constant $c_1 > 0$:

$$\left|\left|\sum_{k=1}^{K}(\psi_k - p_{d,k})\beta_k\right|\right| \leq \frac{c_1}{d^{1+2/(p+1)}}\sum_{k=1}^{K}||\beta_k|| \tag{1.14}$$

Then under **A1.3.2** the coefficients are bounded by:

$$\sum_{k=1}^{K}||\beta_k|| \leq c_2\sum_{k=1}^{K}k^{-a} \leq c_2\sum_{k=1}^{K}\frac{1}{k} \leq c_2\log K \tag{1.15}$$

with some constant $c_2 > 0$. The second inequality holds by $a > 1$ and the last inequality follows from the partial sum of a Harmonic series being $O(\log K)$. Finally, the truncation error can be bounded to depend on a with some constant $c_3 > 0$.

$$\left|\left|\sum_{k=K+1}^{\infty}\psi_k\beta_k\right|\right| \leq \sum_{K+1}^{\infty}||\beta_k|| \tag{1.16}$$

$$\leq \frac{c_3}{a-1}K^{-a} \tag{1.17}$$

□

1.4.1 Discussion

This approximation partitions the original problem into many simpler ones. The target function is approximated by the composition of many simpler ones as in Barron (1993) and Makovoz (1996). However, the inclusion of an additional layer allows for these functions to play the role of adaptive bases. Instead of learning the entire function[12] in a single layer,

[12] It is possible to consider the case where $g_0(x_i)$ cannot be completely characterized by a composition of functions satisfying **A1.3.3**. In this case the inclusion of ψ_k rather than simply f_{nn} may allow for tighter

the basis is learned by the sequence ψ_k. Furthermore, the trade-off between K and d is fully characterized. Achieving a good approximation to the underlying function requires balancing the approximation error of the sequence with the approximation error of each basis function:

$$K_n \asymp d_n^{\frac{1}{2a}+\frac{1}{a(1+p)}} \tag{1.18}$$

This will ensure the truncation error vanishes proportionally to the error in estimating each basis function. This choice has clear theoretical implications for how to choose K_n and d_n given values of p and a. Figure 1.2 shows how K_n and d_n vary a for fixed $p = 1$. An important property of equation 1.18 is that when p is large the growth rate of d_n is much faster than K_n. This suggests that in high-dimensional problems one may want to include many more terms in the first hidden layer relative to the second.[13]

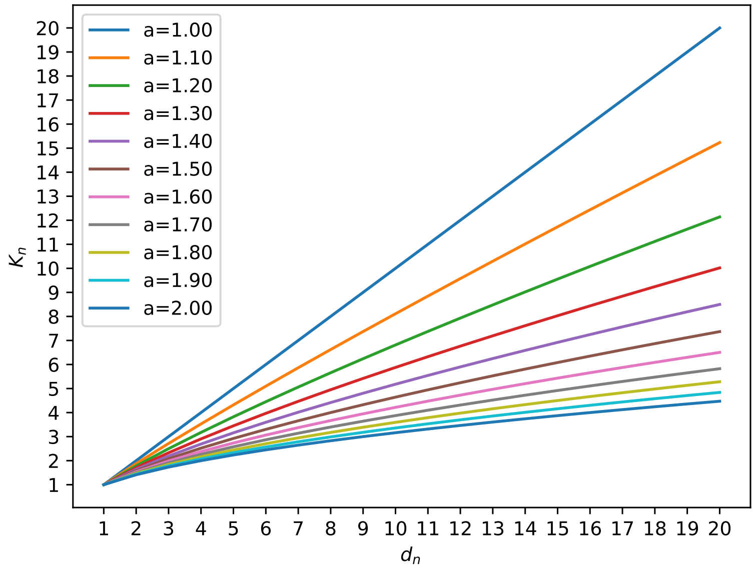


Figure 1.2: Growth rates for neural network widths K_n and d_n over various choices of a with a single covariate ($p = 1$).

control of the approximation error. I leave this investigation to future work.

[13] In the case where each adaptive basis was a function of a strict subset of the covariate space one can find tighter entropy bounds. This may be one justification for why deeper networks work in practice as methods like dropout (Srivastava, Hinton, Krizhevsky, Sutskever, and Salakhutdinov, 2014) perform a similar operation and are nearly always used.

1.5 Asymptotic Theory

Consider the sieve least squares problem:

$$\sup_{g \in \mathcal{G}_n} Q_n(g) = \sup_{g \in \mathcal{G}_n} -\frac{1}{n} \sum_{i=1}^{n} \ell_n(g, z_i) \tag{1.19}$$

The choice of sieve space, $\mathcal{G}_n$, is equation (1.5) satisfying the constraints in **A1.3.4** and $\ell_n(g, z_i) = (y_i - g(x_i))^2 = e_i^2$. I make the following assumptions on the error term e_i and properties of the objective:

A1.5.1 The second unconditional moment of the error term exists and is finite $\mathbb{E}[e_i^2] < \infty$. In addition $\mathbb{E}(|e_i|^{2+\gamma}) < \infty$ for some $\gamma > 0$

A1.5.2 The sieve spaces $\mathcal{G}_n$ are compact.

A1.5.3 The population objective $Q(g)$ is continuous at g_0 and for any $\varepsilon > 0$:

$$Q(g_0) - \sup_{g \in \mathcal{G}_n : ||g, g_0|| > \varepsilon} Q(g, z_i) > 0 \tag{1.20}$$

A1.5.4 For some constant C_1 and any $\varepsilon > 0$

$$\sup_{g \in \mathcal{G}_n : ||g_0 - g|| < \varepsilon} var(\ell_n(g) - \ell_n(g_0)) \leq C_1 \varepsilon^2 \tag{1.21}$$

A1.5.5 For some constant C_2 and any $\varrho > 0$ there exists some $s \in (0, 2)$ such that

$$\sup_{g \in \mathcal{G}_n : ||g_0 - g|| < \varrho} |\ell_n(g) - \ell_n(g_0)| \leq \varrho^s U(z_i) \tag{1.22}$$

$$E[U(z_i)]^{2+\gamma} \leq C_2 \text{ for some } \gamma \geq 0 \tag{1.23}$$

A1.5.6 Let $\mathcal{F}_n = \{\ell_n(g) - \ell_n(g_0) : \rho(g, g_0) \leq \varrho, g \in \mathcal{G}_n\}$ and for some constant C_3 then there exists:

$$\varrho = \inf \left\{ \varrho \in (0,1) : \frac{1}{\sqrt{n}\varrho^2} \int_{\varrho^2}^{\varrho} \sqrt{\mathbf{H}_{[]}(\nu, \mathcal{F}_n, || \cdot ||_2)} d\epsilon \leq C_3 \right\} \tag{1.24}$$

The first condition is quite weak, ruling out only pathological examples like the Cauchy and is made in most parametric and nonparametric literature. The second condition is also standard in the nonparametric literature, e.g., Chen and Shen (1998), Shen and Wong (1994), or Chen (2007). Compactness follows directly from the definition of the neural network spaces $\mathcal{G}_n$. The identification condition, **A1.5.3** is a standard regularity condition and ensures the population criterion has sufficient curvature to identify the population parameter. The assumptions **A1.5.4**, **A1.5.5**, and **A1.5.6** follow from Chen and Shen (1998) and control the local behavior of the estimator close to the population parameter. The key condition is **A1.5.6** which is a bound on the bracketing metric entropy. Verifying this assumption for the extended neural network is key for determining the final convergence rate.

1.5.1 Consistency

Under **A1.5.3** and the approximation result from Theorem 1.1 consistency of g_n follows from a uniform convergence result for the sieve objective Q_n to the population criterion Q.

Theorem 1.2. *If the sampling and function form assumptions from section 1.3 hold along with* **A1.5.1**, **A1.5.2**, **A1.5.3**, $\Delta_n = o(n^{1/4})$ *and* $K_n d_n \Delta_n^4 \log \Delta_n d_n K_n = o(n)$ *then:*

$$\lim_{n\to\infty} \sup_{g\in\mathcal{G}_n} |Q_n(g) - Q(g)| = 0 \tag{1.25}$$

and $g_n \xrightarrow{p} g_0$

Proof. Using boundedness and the sampling assumptions from **A1.3.1**, it suffices to verify the conditions from theorem 2.5 in White and Wooldridge (1991). Let the uniform bound be defined as $m(g_n, z_i) = \sup_{g \in \mathcal{G}_n} |y_i - g(x_i)|$ and define upper bounds:

$$\sup_{z} \sup_{g \in \mathcal{G}_n} |\ell(g_n, z_i)| \leq \bar{\ell}_n \tag{1.26}$$

$$\sup_{z} \sup_{g \in \mathcal{G}_n} |y_i - g(x_i)| \leq \bar{m}_n \tag{1.27}$$

First note that one can use Bernstein's Inequality for independent random variables to bound deviations from the expectation of the objective function:

$$\Pr\left[\left|\sum_i \ell(g_n, z_i) - \mathbb{E}\left[\ell(g_n, z_i)\right]\right| > \epsilon\right] \leq 2\exp\left[\frac{-\epsilon^2}{2\bar{\ell}_n^2(n + 2\epsilon/3)}\right] \tag{1.28}$$

for any $\epsilon > 0$. In addition, one can use the Lipschitz property of ℓ_n to define a bound on differences between the objective for any $g_n \in G_n$:

$$|\ell(g_n, z_i) - \ell(g_0, z_i)| \leq \sup_{g \in G_n} |y_i - g(x_i)| \leq \bar{m}_n \tag{1.29}$$

Using 1.29 and Bernstein's Inequality one can bound deviations of this difference from its expectation:

$$\Pr\left[\left|\sum_i m(g_n, z_i) - \mathbb{E}\left[m(g_n, z_i)\right]\right| > \epsilon\right] \leq 2\exp\left[\frac{-\epsilon^2}{2n\bar{m}_n^2 + 4\epsilon\bar{m}_n/3}\right] \tag{1.30}$$

Putting these bounds together one has the desired maximal inequality:

$$\Pr\left[\sup_{g\in\mathcal{G}_n}\left|n^{-1}\sum_{i=1}^{n}[\ell(g_n,z_i)-\mathbb{E}\left(\ell(g_n,z_i)\right)]\right|>\epsilon\right] \tag{1.31}$$

$$\leq 2\exp\mathbf{H}_n\left(\left[\frac{\epsilon}{6\bar{m}_n}\right],\mathcal{G}_m,||\cdot||_\infty\right)\left[\exp\left[\frac{-\epsilon^2n^2}{18n\bar{\ell}_n^2+4\epsilon n\bar{\ell}_n^2}\right]\right. \tag{1.32}$$

$$\left.+\exp\left[\frac{-4n^2\bar{m}_n^2}{2n\bar{m}_n^2+8n\bar{m}_n^2/3}\right]\right] \tag{1.33}$$

$$\leq 2\exp\mathbf{H}_n\left(\left[\frac{\epsilon}{6\bar{m}_n}\right],\mathcal{G}_m,||\cdot||_\infty\right)\left[\exp\left[\frac{-\epsilon^2n}{\bar{\ell}_n^2(18+4\epsilon)}\right]+\exp\left[-6n/7\right]\right] \tag{1.34}$$

where $\mathbf{H}_n$ is the metric entropy. It suffices to show $n^{-1}\bar{\ell}_n^2\to 0$ as $n\to\infty$ and $\forall\epsilon>0$

$$(\bar{\ell}_n^2/n)\mathbf{H}_n([\epsilon/6\bar{m}_n],\mathcal{G}_n,||\cdot||_\infty)\to 0 \quad \text{as } n\to\infty \tag{1.35}$$

Let $\sup_{g\in\mathcal{G}_n}|g|\leq\Delta_n$ and $\Delta_n>1$, then it can be shown $\bar{\ell}_n=4\Delta_n^2$ and $\bar{m}_n=4\Delta_n$. The first condition is satisfied with $\Delta_n=o(n^{1/4})$. In addition by the calculations in Appendix A.1:

$$\mathbf{H}_n(\epsilon,\mathcal{G}_n,||\cdot||_\infty)\leq\omega_n\left[\log\frac{16}{\epsilon}+\log\left(\Delta_n(1+\Delta_n+p\Delta_n^2)\right)+\log d_nK_n\right] \tag{1.36}$$

where $\omega_n=1+K_n(d_n+2)+d_n(p_n+1)$ is the number of parameters to characterize g_n. Using this bound and our definition of $\bar{m}_n$:

$$\mathbf{H}_n(\epsilon/6\bar{m}_n,\mathcal{G}_n,||\cdot||_\infty)=\mathbf{H}_n(\epsilon/24\Delta_n,\mathcal{G}_n,||\cdot||_\infty) \tag{1.37}$$

$$\leq\omega_n\log\frac{384\Delta_n}{\epsilon}+\omega_n\log\left(\Delta_n(1+\Delta_n+p\Delta_n^2)\right)+\omega_n\log d_nK_n \tag{1.38}$$

Then $\exists$ an $n \in \mathbb{N}$ s.th. $\forall \epsilon > 0$ $\Delta_n^3 \geq 384/\epsilon$ and $\Delta_n^3 \geq \Delta_n^2 \geq \Delta_n$.

$$\begin{aligned}
\mathbf{H}_n(\epsilon/24\Delta_n, \mathcal{G}_n, ||\cdot||_\infty) &\leq \omega_n \log \Delta_n^3 + \omega_n \log \Delta_n^3 \, (p+2) + \omega_n \log d_n K_n && (1.39)\\
&\leq \omega_n \left[6 \log \Delta_n + \log(p+2) + \log d_n K_n \right] && (1.40)\\
&\leq \omega_n 6 \log \Delta_n (p+2) d_n K_n && (1.41)
\end{aligned}$$

Putting this together with $\bar{\ell}^2/n$:

$$(\bar{\ell}^2/n)\mathbf{H}_n(\epsilon/6\bar{m}_n, \mathcal{G}_n, ||\cdot||_\infty) \leq n^{-1} 96 \Delta_n^4 \omega_n \log \Delta_n (p+2) d_n K_n \tag{1.42}$$

Also note that $\omega_n = O(K_n d_n)$ therefore $K_n d_n \Delta_n^4 \log \Delta_n d_n K_n = o(n)$ □

It is worth noting that the requirement $\Delta_n = o(n^{1/4})$ is not strong here given the coefficient decay in **A1.3.2**. One may take $\Delta_n = O(\log n)$ which leaves the growth of d_n and K_n fairly flexible. Without any other considerations: $d_n = O(n^\alpha)$ and $K_n = O(n^\beta)$ for some $0 < \alpha + \beta < 1$. This result follows from:

$$\begin{aligned}
n^\alpha n^\beta (\log n)^4 \log\left((\log n) n^\alpha n^\beta\right) &= n^\alpha n^\beta (\log n)^4 \left[\log(\log n) + \alpha \log n + \beta \log n\right] && (1.43)\\
&\leq n^\alpha n^\beta (\log n)^4 \left[\log n + \alpha \log n + \beta \log n\right] && (1.44)\\
&\leq 3 n^{\alpha+\beta} (\log n)^5 = o(n) && (1.45)
\end{aligned}$$

Combining this result with the balance condition in 1.18 gives us an optimal trade-off:

$$n^\beta \asymp n^{\alpha\left(\frac{1}{2a} + \frac{1}{a(1+p)}\right)} \tag{1.46}$$

As noted in section 1.4.1, the number of basis functions K_n grows with n, but the rates are much slower when p or a are larger. This follows from the increased complexity of approximating ψ_k when p is large and the faster decay rates for the coefficients β_k.

1.5.2 Convergence Rates

The convergence rate result follows from verification of the conditions in Chen and Shen (1998). The verification of **A1.5.6** is specific to the extended neural network and is entirely new in the literature.

Theorem 1.3. *If* **A1.5.1**, **A1.5.4**, **A1.5.5**, *and* **A1.5.6** *are satisfied and equation (1.18) holds with decay parameter* $a > (p+1)/2$ *then the convergence rate for an extended neural network in a sieve least squares problem is:*

$$||g_n - g_0||_{L2} = o_p(n^{-1/4}) \tag{1.47}$$

Proof. The verification of **A1.5.4** and **A1.5.5** are shown in Chen and Shen (1998) and Chen and White (1999) for $g_n^{(nn)}$ and the least squares objective. The verification of **A1.5.6** is entirely new and is the key condition that determines the convergence rate.

First note that one can write the difference between the objective evaluated at any $g \in \mathcal{G}_n$ and g_0 as:

$$\begin{aligned}
\ell_n(g; z_i) - \ell_n(g_0; z_i) &= -\frac{1}{2}(y_i - g)^2 + \frac{1}{2}(y_i - g_0)^2 && (1.48)\\
&= (g - g_0)y_i - \frac{1}{2}(g^2 - g_0^2) + (g - g_0)g_0 - (g - g_0)g_0 && (1.49)\\
&= (g - g_0)\left(e_i + \frac{1}{2}(g_0 - g)\right) && (1.50)
\end{aligned}$$

Now to verify **A1.5.4**, the variance term is bounded by Minkowski's and Cauchy-Schwarz inequalities.

$$\begin{aligned}
\mathbb{E}[\ell_n(g; z_i) - \ell_n(g_0; z_i)]^2 &\leq \mathbb{E}[((g - g_0)e_i)^2] + \frac{1}{2}\mathbb{E}[(g_0 - g)^4] && (1.51)\\
&\leq \mathbb{E}[e_i^2]||g - g_0||^2 + \frac{1}{2}\mathbb{E}[(g_0 - g)^4] && (1.52)
\end{aligned}$$

The first term is taken care of by **A1.5.1** and the second is handled by **A1.3.1**:

$$\frac{1}{2}\mathbb{E}[(g_0 - g)^4] \leq \sup_x (g_0 - g)^2 ||g_0 - g||^2 \tag{1.53}$$

$$\leq (\sup_x ||g_0||^2 + \sup_x ||g||^2)||g_0 - g||^2 \tag{1.54}$$

Without loss of generality $\sup_x ||g_0|| = \sup_x ||g|| = 1$. Therefore:

$$\mathbb{E}[\ell_n(g; z_i) - \ell_n(g_0; z_i)]^2 \leq const. ||g - g_0||^2 \tag{1.55}$$

To verify **A1.5.5** I use the same uniform bound as above only now to both terms.

$$|\ell_n(g; z_i) - \ell_n(z_i, g_0)| = \left|(g - g_0)\left(e_i - \frac{1}{2}(g - g_0)\right)\right| \tag{1.56}$$

$$\leq ||g - g_0||_\infty \left(|e_i| + \frac{1}{2}\left(||g||_\infty + ||g_0||_\infty\right)\right) \tag{1.57}$$

The first term requires an interpolation result between $|| \cdot ||$ and $|| \cdot ||_\infty$. Using Lemma 2.1 Chen and Shen (1998) $||g - g_0||_\infty \leq ||g - g_0||^{\frac{2}{2+p}}$:

$$||g - g_0||_\infty \left(|e_i| + \frac{1}{2}\left(||g||_\infty + ||g_0||_\infty\right)\right) \leq ||g - g_0||^{\frac{2}{2+p}} \left(|e_i| + 1\right) \tag{1.58}$$

the desired result is achieved with $U(z_i) = (|e_i| + 1)$ which is finite by **A1.5.1**.

The most important component is the verification of **A1.5.6**. This condition controls the complexity of the estimator and ensures the entropy integral is finite. I examine a bound on:

$$\frac{1}{\sqrt{n}\varrho^2} \int_{b\varrho^2}^{\varrho} \sqrt{\mathbf{H}_{[]}(\epsilon, \mathcal{F}_n, ||\cdot||_2)} d\epsilon \tag{1.59}$$

First note that $\mathbf{H}_{[]}(\nu, \mathcal{F}_n, ||\cdot||_2) \leq \mathbf{H}(\nu^{1/s}, \mathcal{G}_n, ||\cdot||)$. It sufficies to examine the bound on the metric entropy for the sieve space $\mathcal{G}_n$. Let $\nu^{1/s} = \epsilon$ and plugging in the calculations from appendix A.1:

$$\int_{\varrho^2}^{\varrho} \sqrt{K_n d_n \Delta_n \log \frac{K_n d_n \Delta_n}{\epsilon}} d\epsilon \tag{1.60}$$

Further let $\vartheta_n = K_n d_n$ and note that since n is large and $\Delta_n = O(\log n)$ one can treat Δ_n as a constant. Substituting $K_n d_n$ for ϑ_n:

$$\int_{\varrho^2}^{\varrho} \sqrt{\vartheta_n \log \frac{\vartheta_n}{\epsilon}} d\epsilon \leq \sqrt{\vartheta_n} \int_{\varrho^2}^{\varrho} \sqrt{\log \frac{\vartheta_n}{\epsilon}} d\epsilon \tag{1.61}$$

Now let $t = \sqrt{\log(\vartheta_n/\epsilon)}$ then $\epsilon = \vartheta_n \exp\{-t^2\}$ and $d\epsilon = -2t\vartheta_n \exp\{-t^2\}\, dt$. The limits of integration are then $u \equiv \sqrt{\log(\vartheta_n/\varrho)}$ and $l \equiv \sqrt{\log(\vartheta_n/\varrho^2)}$. After the change of variables:

$$\vartheta_n^{(3/2)} \int_l^u -2t^2 \exp\{-t^2\} dt \tag{1.62}$$

Integration by parts yields:

$$\begin{aligned} \vartheta_n^{(3/2)} \left(t \exp\left\{-t^2\right\}\Big|_l^u - \int_l^u \exp\left\{-t^2\right\} dt \right) &\leq (\vartheta_n)^{3/2} t \exp\left\{-t^2\right\}\Big|_l^u & (1.63) \\ &= \varrho\sqrt{\vartheta_n \log(\vartheta_n/\varrho)} - \varrho^2\sqrt{\vartheta_n \log(\vartheta_n/\varrho^2)} & (1.64) \\ &\leq \varrho\sqrt{\vartheta_n \log(\vartheta_n/\varrho)} & (1.65) \end{aligned}$$

and $\varrho = n^{-1/2}\sqrt{\vartheta_n \log(\vartheta_n)}$. Since $K_n \asymp d_n^{\frac{1}{2a}+\frac{1}{a(1+p)}}$ the final convergence rate is obtained by balancing the approximation error with ϱ.

$$
\begin{aligned}
d_n^{-1/2-1/(p+1)} \log(K_n) &= n^{-1/2}\sqrt{d_n K_n \log(d_n K_n)} && (1.66)\\
\sqrt{n} &= d^{1/2+1/(p+1)}\sqrt{d_n K_n \log(d_n K_n)}\frac{1}{\log K_n} && (1.67)\\
O(n) &= d^{1+2/(p+1)} \left(d_n^{1+\frac{1}{2a}+\frac{1}{a(1+p)}} \log(d_n^{1+\frac{1}{2a}+\frac{1}{a(1+p)}} \right) / \log(d_n^{\frac{1}{2a}+\frac{1}{a(1+p)}}) && \\
& && (1.68)\\
&= d_n^{2+2/(p+1)+(1+p+2a)/(2a(1+p))} \log(d_n) && (1.69)
\end{aligned}
$$

$$
\begin{aligned}
||\hat{g}_n - g_0|| &= O_p\left[(n/\log(n))^{-((1/2)+(1/(p+1)))/(2+2/(1+p)+(1+p+2a)/(2a(1+p)))}\right] && (1.70)\\
&= O_p\left[(n/\log(n))^{-a(p+3)/(2a(2p+5)+p+1)}\right] = o_p(n^{-1/4}) && (1.71)
\end{aligned}
$$

which holds so long as:

$$
\begin{aligned}
-a(p+3)/(2a(2p+5)+p+1) &< -1/4 && (1.72)\\
a &> (p+1)/2 && (1.73)
\end{aligned}
$$

□

The convergence rate here is similar to the result in Chen and White (1999) as the same machinery is used. However, it is important to recognize this rate will never be faster than the classical case. The rate for the classical neural network is governed by:

$$
\left(1+\frac{2}{p+1}\right)/4\left(1+\frac{1}{p+1}\right) = \frac{p+3}{4p+8} \qquad (1.74)
$$

In the case of the extended neural network the presence of a makes direct comparison difficult. However, using the condition $a > (p+1)/2$ for any $\epsilon > 0$ the extended neural network rate is governed:

$$\frac{a(p+3)}{2a(2p+5)+p+1} = \frac{(p+3)(p+1)+2\epsilon(p+3)}{4(p+3)(p+1)+4\epsilon(2p+5)} \tag{1.75}$$

One can see that when $\epsilon \to 0$ the extended neural network rate approaches $1/4$ which is never faster than the single layer rate (1.74). However, as p gets larger the rates will be the same. This conclusion is not surprising as I am introducing additional complexity by adding a layer, increasing the entropy number, while at the same time adding another level of approximation error. This reduced rate comes at the benefit of additional flexibility in the approximation, reducing the burden placed on any individual ψ_k to approximate the entire function.

1.6 Inference

In this section I discuss an approach to obtaining pointwise confidence bands for linear functionals of g_n. I focus on the 'evaluation functional', i.e., $g_n(\bar{x})$ for some $\bar{x} \in \mathcal{X}$. This functional is linear, but irregular in the sense that it is not estimable at the $\sqrt{n}$ rate (Chen, Liao, and Sun, 2014). I abstract away from the theoretical difficulties of establishing normality and focus on the computational aspect of estimating the sieve variance[14]. To this end I utilize sample splitting and introduce a new algorithm called cross-training to construct the confidence bands. In simulations the proposed plug-in estimator performs well and attains conservative, but reasonable coverage.

[14] I assume the estimator is asymptotically linear and a central limit theorem holds. The assumptions necessary for this to be true are outlined in A.3, but left unverified.

1.6.1 Asymptotic Normality

Suppose $\ell(g, z_i) - \ell(g_0, z_i)$ can be approximated by $\Delta_\ell(g_0, z_i)[g - g_0]$ which is the pathwise derivative of $\ell(g_0, z_i)$ in the direction of $[g - g_0]$ defined:

$$\Delta_\ell(g_0, z_i)[g - g_0] = \lim_{\tau \to 0} \frac{\ell(g_0 + \tau[g - g_0], z_i) - \ell(g_0, z_i)}{\tau} \tag{1.76}$$

In order to explicitly write down a normality result for functionals of the neural network estimator I require some additional notation. Define the norm on $\mathcal{G}_n$:

$$||g - g_0||_\ell^2 = \lim_{\tau \to 0} -\frac{\partial \mathbb{E}\left[\Delta_\ell(g_0 + \tau[g - g_0], z_i)[g - g_0]\right]}{\partial \tau} \tag{1.77}$$

Let $\mathcal{V}$ be the closed linear span of $\mathcal{G}_n - \{g_0\}$ under $||\cdot||_\ell$ with inner product defined as:

$$\langle v_{g_1}, v_{g_2} \rangle_\ell = \lim_{\tau \to 0} -\frac{\partial \mathbb{E}\left[\Delta_\ell(g_0 + \tau v_{g_2}, z_i)[v_{g_1}]\right]}{\partial \tau} \tag{1.78}$$

for any $v_{g_1}, v_{g_2} \in \mathcal{V}$. The functional under consideration is the evaluation functional $h(g) = g(\bar{x})$. The path derivative of this functional in the direction of $v = g - g_0$ is:

$$\frac{\partial h(g_0)}{\partial g}[v] = \lim_{\tau \to 0} \frac{h(g_0 + \tau[v]) - h(g_0)}{\tau} \tag{1.79}$$

which is linear in v. Then by the Riesz representation theorem there is a $v_n^\star$ that satisfies the following conditions:

$$\frac{\partial h(g_0)}{\partial g}[v] = \langle v_n^\star, v \rangle_\ell \tag{1.80}$$

$$\frac{\partial h(g_0)}{\partial g}[v_n^\star] = ||v_n^\star||_\ell^2 = \sup_{v \in \mathcal{V}, ||v|| \neq 0} \frac{\left| \frac{\partial h(g_0)}{\partial g}[v] \right|^2}{||v||_\ell^2} \tag{1.81}$$

If the functional is regular, the sieve representer under the norm $||\cdot||_\ell^2$ will be finite. However, this object diverges in the irregular case. Fortunately, the recent work Chen, Liao, and Sun (2014) and Chen and Liao (2014) show that one can still construct valid inference in this case. Now define:

$$||v_n^\star||_{sd}^2 = \text{Var}\left(\frac{1}{\sqrt{n}}\sum_{i=1}^{n}\Delta_\ell(g_0, z_i)[v_n^\star]\right) \tag{1.82}$$

This object characterizes the sieve variance so long as $||v_n^\star||_\ell \, / \, ||v_n^\star||_{sd} = O(1)$. Let $u_n^\star = v^\star / \, ||v_n^\star||$ be the normalized representer and one has:

$$\sqrt{n}\left(g_n(\bar{x}) - g_0(\bar{x})\right) / \, ||v_n^\star||_{sd} = \sqrt{n}\mu_n\left[\Delta(g_0, z_i)[u_n^\star]\right] + o_p(1) \to N(0,1) \tag{1.83}$$

where μ_n is the centered empirical process. I do not prove this result here but assume that it holds. I construct estimates of $||v_n||_{sd}^\star$ in the subsequent section and verify that this approach works well in simulations.

1.6.2 Variance Estimation

The empirical sieve representer for the functional $h(g_0)$ solves the following optimization problem:

$$||v_n^\star||_\ell^2 = \sup_{v \in \mathcal{V}, ||v|| \neq 0} \frac{\left|\frac{\partial h(g_0)}{\partial g}[v]\right|^2}{||v||_\ell^2} \tag{1.84}$$

$$= \sup_{\lambda = (v_g) \in \mathbb{R}^\omega, \lambda \neq 0} \frac{\lambda' s_n s_n' \lambda}{\lambda' \mathbb{E}\left[-H_n\right] \lambda} \tag{1.85}$$

where $\omega = 1 + K_n(d_n + 2) + d_n(p_n + 1)$ refers to the neural network dimension. The term s_n is the path derivative of the functional $h(g_0)$ and $H_n = \partial^2 \ell(g_n, z_i)/\partial\theta\partial\theta'$. Furthermore by equation 1.80:

$$\frac{\partial h(g_0)}{\partial g}[v] = s_n'\lambda \tag{1.86}$$

$$= \langle v_n^\star, v\rangle_\ell \tag{1.87}$$

$$= \lambda_n^{\star\prime}\mathbb{E}\left[-H_n\right]\lambda \tag{1.88}$$

such that $v_n^\star = \lambda_n^\star = \mathbb{E}[-H_n]^{-1}s_n$. Now plugging this into the score for our particular form of $\ell(g_0, z_i)$:

$$\Delta_\ell(g_0, z_i)[v_n^\star] = -2\mathbb{E}\left[-H_n\right]^{-1} s_n e_i \tag{1.89}$$

$$||v_n^\star||_{sd}^2 = \mathrm{Var}(\Delta_\ell(g_0, z_i)[v_n^\star]) = 4\mathbb{E}\left[-H_n\right]^{-1}\mathbb{E}\left[s_n e_i e_i' s_n'\right]\mathbb{E}\left[-H_n\right]^{-1} \tag{1.90}$$

In practice one needs explicit analytical forms for s_n and H_n. This can be done by treating the problem 'as-if' it was fully parametric. This concept is not new to the literature for linear sieves, e.g., Newey (1997) or Hahn, Liao, and Ridder (2018). However, it has yet to be verified for use in the case of nonlinear sieves (Chen, Liao, and Sun, 2014). Recall the form of $g_n^{(dnn)}$ from 1.5:

$$g_n^{(dnn)}(x_i) = \sum_{k=1}^{K} s\left(\sum_{j=1}^{d} s(\tilde{x}_i'\delta_j)\gamma_{jk}\right)\beta_k \tag{1.91}$$

$$= \sum_{k=1}^{K} p_{d,k}(x_i)\beta_k \tag{1.92}$$

If the form of $p_{d,k}(x_i)$ was a fixed transformation of x_i this would fall into the class of series estimators. However, the neural network adds a further layer of adaptation and thus complication to the construction of $||v_n^\star||_{sd}$. The score function of $g_n^{(dnn)}$ with respect to each parameter set $\{\beta_k\}, \{\gamma_{jk}\}, \{\delta_j\}$ can be compactly written:

$$\frac{\partial \ell(g_n, z_i)}{\partial \beta_k} = -2p_{d,k}(x_i)e_i \tag{1.93}$$

$$\frac{\partial \ell(g_n, z_i)}{\partial \gamma_{jk}} = -2\frac{\partial p_{d,k}}{\partial \gamma_{jk}}\beta_k e_i \tag{1.94}$$

$$\frac{\partial \ell(g_n, z_i)}{\partial \delta_k} = -2\frac{\partial p_{d,k}}{\partial \delta_k}\beta_k e_i \tag{1.95}$$

where the expressions of $\partial p_{d,k}/\partial \gamma_{jk}$ and $\partial p_{d,k}/\partial \gamma_{jk}$ are the partial derivatives of the generated basis functions. Let the parameters be collected into $\theta = [\text{vec}(\delta)', \text{vec}(\gamma)', \text{vec}(\beta)']'$ and the scores be stacked as:

$$\partial \ell(g_n, z_i)/\partial \theta = [\text{vec}(\partial \ell(g_n, z_i)/\partial \delta)', \text{vec}(\partial \ell(g_n, z_i)/\partial \gamma)', \text{vec}(\partial \ell(g_n, z_i)/\partial \beta)']' \tag{1.96}$$

The hessian terms can then be written as $\partial^2 \ell(g_n, z_i)/\partial \theta \partial \theta'$ and the complete characterization of $\text{Var}(g_n)$ can be written in the familiar sandwich form:

$$\begin{bmatrix} \frac{\partial h(g_0)}{\partial \delta} \\ \frac{\partial h(g_0)}{\partial \gamma} \\ \frac{\partial h(g_0)}{\partial \beta} \end{bmatrix}' \mathbb{E}\left[\frac{\partial^2 \ell(g_n, z_i)}{\partial \theta \partial \theta'}\right]^{-1} \mathbb{E}\left(\frac{\partial \ell(g_n, z_i)}{\partial \theta}\frac{\partial \ell(g_n, z_i)}{\partial \theta}'\right) \mathbb{E}\left[\frac{\partial^2 \ell(g_n, z_i)}{\partial \theta \partial \theta'}\right]^{-1} \begin{bmatrix} \frac{\partial h(g_0)}{\partial \delta} \\ \frac{\partial h(g_0)}{\partial \gamma} \\ \frac{\partial h(g_0)}{\partial \beta} \end{bmatrix} \tag{1.97}$$

This expression acts as a ‘delta-method’ plug-in estimator for equation 1.89. However, these variance estimates tend to be quite poor in practice. Instead I alter the procedure by introducing a sample splitting scheme and a new algorithm called cross-training.

1.6.3 Sample Splitting

Consider a shuffled partition of the design matrix $\mathbf{Z} = \left\{\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}\right\}$ where $\mathbf{Z}^{(1)} = \{z_i\}_{i=1}^{\lceil n/2 \rceil}$ and $\mathbf{Z}^{(2)} = \{z_i\}_{i=\lceil n/2 \rceil+1}^{n}$. Furthermore define $n_1 = \lceil n/2 \rceil$ and $n_2 = n - \lceil n/2 \rceil$. The conditional mean function is identified using either $\mathbf{Z^{(1)}}$ or $\mathbf{Z^{(2)}}$ by assumption **A1.3.1**.

The sieve least squares estimate on the first half of the data gives:

$$\hat{g}_{n_1} = \sum_{k=1}^{K} \hat{f}_{d,k}(x_i) = \sup_{g \in \mathcal{G}_{n_1}} -\frac{1}{2n_1} \sum_{i=1}^{n_1} \ell(g_{n_1}, z_i) \tag{1.98}$$

Now for observations $z_i \in \mathbf{Z}^{(2)}$ let $f_d^K = (f_{d,1}, f_{d,2}, \ldots, f_{d,K})'$ evaluated at z_i we find optimal weights $\hat{\alpha}$ that solve the sample objective:

$$\hat{\alpha} = \arg\min_{\alpha} \frac{1}{n_2} \sum_{i=1}^{n_2} \left(y_i - f_d^K(x_i)'\alpha\right)^2 \tag{1.99}$$

$$= \left(\frac{1}{n_2} \sum_{i=1}^{n_2} f_d^K(x_i) f_d^K(x_i)' \right)^{\dagger} \frac{1}{n_2} \sum_{i=1}^{n_2} f_d^K(x_i) y_i \tag{1.100}$$

Clearly $\hat{\alpha}$ is an adjustment term to the estimate of β obtained in 1.98. The final estimate for $g_{n_2}(x_i) = \sum_{k=1}^{K} \hat{f}_{d,k}(x_i)\alpha_k = \sum_{k=1}^{K} p_{d,k}(x_i)\tilde{\beta}_k$

Constructing the variance of g_{n_2} adds the additional parameter α, but all other parameters θ depend only on observations in n_1. Let the score vector be partitioned into $s_i = \partial\ell(g_n, z_i)/\partial\alpha$ and $\zeta_i = \partial\ell(g_n, z_i)/\partial\theta$ and the partitioned hessian matrix is:

$$\begin{bmatrix} \frac{\partial \zeta_i}{\partial \theta} & 0 \\ \frac{\partial s_i}{\partial \theta} & \frac{\partial s_i}{\partial \alpha} \end{bmatrix} \tag{1.101}$$

where the upper right element is zero as α is fixed when estimating θ. An influence function term for the parameters pertaining to θ is defined $\phi_i = -\mathbb{E}\left[\frac{\partial \zeta_i}{\partial \theta}\right]^{-1} \hat{\zeta}_i$ and an estimate of the variance of the evaluation functional $g_n(\bar{x})$ is:

$$f_d^K(x_i)'\mathbb{E}\left[\frac{\partial s_i}{\partial \alpha}\right]^{-1} \left[n^{-1} \sum_{i=1}^{n} \left(s_i + \mathbb{E}\left[\frac{\partial s_i}{\partial \theta}\right] \phi_i \right) \left(s_i + \mathbb{E}\left[\frac{\partial s_i}{\partial \theta}\right] \phi_i \right)' \right] \mathbb{E}\left[\frac{\partial s_i}{\partial \alpha}\right]^{-1} f_d^K(x_i) \tag{1.102}$$

This expression[15] has an interesting form as it is similar to the consistent estimate of $||v_n^\star||_{sd}^2$ for series estimation of the evaluation functional in the case where $p_d^K(x_i)$ is a fixed transformation (Chen, Liao, and Sun, 2014). The additional terms are necessary given the adaptive construction of $p_{d,k}$ in the extended[16] neural network.

1.6.4 Cross-training

Implementation of neural networks and deep learning can be difficult due to the presence of many local minima or even more problematic, saddle points. To ensure stability of the optimization and reach minima that will satisfy the normality conditions I augment estimation by including additional parameters that depend on only a subset of the observations. The justification for this approach is to augment the moment conditions implied by optimizing the neural network sieve least squares problem. The modification is:

$$g_n^{(xnn)}(x_i) = \sum_{k=1}^{K} s((x_i' \otimes \Xi_i')\delta_k + \delta_{0k})\beta_k \tag{1.103}$$

where $\Xi_i = \left(2, -\frac{1}{r-1}\mathbf{e}'_{(i \mod r)}\right)' \in \mathbb{R}^{r+1}$ and $I_r = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_{r-1}, \mathbf{e}_0]$ is an r dimensional identity matrix. This vector expands the original covariate space by judiciously adding and subtracting each covariate.

The completely general approach to cross-training is discussed in algorithm 1. However, to get a feel for this method consider the case where the number of splits r is two and there is only one covariate ($p = 1$). Let the indicator functions $\iota_j(i) = \mathbf{1}\{(i \mod r) = j\}$ for each $j \in \{0, 1, \dots, r-1\}$. In the case with only two splits ι_0 is equal to one for all even

[15] This formulation would make a two-step procedure like those found in Hahn, Liao, and Ridder (2018) incredibly useful. Unfortunately, such a construction requires an explicit approximation result for $p_{d,k}$ separately from g_n. The approximation theorem in 1.4 has this in mind, but ultimately constructs one sieve estimate rather than two.

[16] The form of the variance for the classical neural network is the same as in 1.102, but the elements of each term change to reflect the construction in 1.4

observations and zero otherwise. Likewise, ι_1 is one for odd observations and zero otherwise. The first four observations are transformed as:

$$\begin{bmatrix} x_1 \otimes \Xi_1 \\ x_2 \otimes \Xi_2 \\ x_3 \otimes \Xi_3 \\ x_4 \otimes \Xi_4 \end{bmatrix}' = \begin{bmatrix} 2x_1 & 0 & -x_1 \\ 2x_2 & -x_2 & 0 \\ 2x_3 & 0 & -x_3 \\ 2x_4 & -x_4 & 0 \end{bmatrix} \tag{1.104}$$

For each observation the operation $\otimes \Xi_i$ takes the p dimensional covariate vector x_i and for each $j \in \{1, 2, \ldots, p\}$ multiplies multiples[17] the first element by two and the remaining elements by either zero or $-1/(r-1)$. Plugging $x_i \otimes \Xi_i$ into the first four observations of $g_n^{(nn)}$:

$$g^{(xnn)}(x_{(1,3)}) = \sum_{k=1}^{K} s\left((x_{1,3} \otimes \Xi_{1,3})'\delta_k + \delta_{0k}\right)\beta_k \tag{1.105}$$

$$= \sum_{k=1}^{K} s\left(\begin{bmatrix} 2x_{(1,3)} & 0 & -x_{(1,3)} \end{bmatrix}\delta_k + \delta_{0k}\right)\beta_k \tag{1.106}$$

$$g^{(xnn)}(x_{(2,4)}) = \sum_{k=1}^{K} s\left((x_{(2,4)} \otimes \Xi_{(2,4)})'\delta_k + \delta_{0k}\right)\beta_k \tag{1.107}$$

$$= \sum_{k=1}^{K} s\left(\begin{bmatrix} 2x_{(2,4)} & -x_{(2,4)} & 0 \end{bmatrix}\delta_k + \delta_{0k}\right)\beta_k \tag{1.108}$$

When the number of splits is two ($r = 2$) there are effectively two estimates for $g_n^{(xnn)}$, the estimates generated from equation (1.105) and (1.107). The first estimate is invariant to any changes in the observations indexed by an even number, likewise for the second estimate with respect to the odd index. The implied moment conditions for the cross-trained parameters

[17] In the case where $p = 1$ each x_i is a scalar thus rendering the transposition and kronecker product unnecessary, but included for the general cases.

δ_{2k} and δ_{3k} are respectively:

$$\frac{1}{n_0}\sum_{i\in\iota_0} x_i s_{\delta_{0k}}\left(y_i - \sum_{k=1}^{K} s\left(x_i'\delta_{1k} + x_i'\left(\delta_{1k}-\delta_{3k}\right)+\delta_{0k}\right)\beta_k\right) = 0 \tag{1.109}$$

$$\frac{1}{n_1}\sum_{i\in\iota_1} x_i s_{\delta_{1k}}\left(y_i - \sum_{k=1}^{K} s\left(x_i'\delta_{1k} + x_i'\left(\delta_{1k}-\delta_{2k}\right)+\delta_{0k}\right)\beta_k\right) = 0 \tag{1.110}$$

where $n_j = \sum_{i=1}^{n}\iota_j(i)$ for $j = \{0,1\}$ and $s_{\delta_{mk}} = s'\left(x_i'\delta_{1k} + x_i'\left(\delta_{1k}-\delta_{mk}\right)\right)$ for $m = \{2,3\}$ is the derivative of the activation function $s(z)$ evaluated at z. The moment condition for δ_1 is the sum of 1.109 and 1.110. If the underlying function is exactly approximated by this neural network and z_i is truly i.i.d. all of the parameters would be equivalent, i.e., $\delta_{1k} = \delta_{2k} = \delta_{3k}$. However, in practice the neural network is only an approximation to the true function and so a final estimate of $g_n^{(xnn)}(x_i$ can be constructed by averaging over all r models. A straightforward way of achieving this is to replace Ξ_i with $\tilde{\Xi}_i = (2, -\frac{1}{r}\iota_r')'$ such that $g_n^{(xnn)}(x_i)$ is:

$$g_n^{(xnn)}(x_i) = \sum_{k=1}^{K} s((x_i \otimes \tilde{\Xi}_i)'\delta_k + \delta_{0k})\beta_k \tag{1.111}$$

which will average over all the data to produce one estimate of g_0.

In the general case for arbitrary p and r the algorithm can be summarized entirely by the construction of Ξ. Once the number of splits is given the construction of Ξ informs the dimensionality of the input space. The choice of r is a tuning parameter but does not appear to be too important so long as the values are reasonable, at least in our simulations. I summarize the procedure in Algorithm 1. A nice feature of cross-training is that it does not require more than one pass through the data for optimization. The optimization problem has higher complexity, as the number of parameters to optimize grows, but is solved only once.

Algorithm 1 Cross-Training

```
1: Shuffle the paired data $\{y_i, x_i\}$. Let the index of the shuffled data be $i = \{1, 2, \dots, n\}$.
2: Generate the $r$ dimensional identity matrix $I_r$ with columns $I_r = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_{r-1}, \mathbf{e}_0]$.
3: Construct the splitting matrix $\Xi$ where each row can be written as an $\mathbb{R}^{r+1}$ vector:

   $$\Xi_i = \left(2, -\frac{1}{r-1}\mathbf{e}'_{(i \mod r)}\right)' \tag{1.112}$$

   This matrix expands the parameter space by $r$. The solution to $(i \mod r)$ determines the
   expanded parameter to be omitted[18].
4: Define the set $R_m = \{j \in \mathbb{Z} : 0 \leq j \leq r-1, j \neq m\}$ and the indicator function
   $\iota_j(i) = \mathbf{1}\{(i \mod r) = j\}$. Let the sample size for any $\iota_j$ be defined $n_j = \sum_{i=1}^n \iota_j(i)$.
5: Let the parameters $\delta$ and $\beta$ be collected into $\theta$. The cross-trained neural network solves the
   sieve least squares problem:

   $$\max_\theta -\frac{1}{n}\sum_{i=1}^n \left(y_i - \sum_{k=1}^K s((x_i \otimes \Xi_i)'\delta_k + \delta_{0k})\beta_k\right)^2 \tag{1.113}$$

   The moment conditions for the cross-trained parameters $\delta_{mk}$ for any $m \in \{0, 1, \dots, r-1\}$ are:

   $$\frac{1}{n_m}\sum_{i\in\iota_m} x_i s_{\delta_{mk}} \left(y_i - \sum_{k=1}^K s\left(x_i'\delta_{1k} + x_i'\left(\delta_{1k} - \frac{1}{r-1}\sum_{j\in R_m}\iota_j\delta_{j+2,k}\right) + \delta_{0k}\right)\beta_k\right) = 0 \tag{1.114}$$

   where $s_{\delta_{mk}}$ is the first derivative of the activation function evaluated at the index
   $x_i'\delta_{1k} + x_i'\left(\delta_{1k} - \sum_{j\in R_m}\iota_j\delta_{j+2,k}\right)$. The parameter that uses the full sample, $\delta_1$, satisfies the sum of all
   these conditions:

   $$\sum_{m=0}^{r-1}\frac{1}{n_m}\sum_{i\in\iota_m} x_i s_{\delta_{mk}} \left(y_i - \sum_{k=1}^K s\left(x_i'\delta_{1k} + x_i'\left(\delta_{1k} - \frac{1}{r-1}\sum_{j\in R_m}\iota_j\delta_{j+2,k}\right) + \delta_{0k}\right)\beta_k\right) = 0 \tag{1.115}$$

   One uses these moment conditions to update the parameters iteratively until convergence is
   achieved.
6: The cross-training algorithm finds $r$ different neural network estimates. This occurs because
   $g_n^{(xnn)}(x_i)$ depends on what partition $i$ belongs to. An average of these estimates can be constructed by using:

   $$g_n^{(xnn)}(x_i) = \sum_{k=1}^K s((x_i' \otimes \tilde{\Xi}_i')\delta_k + \delta_{0k})\beta_k \tag{1.116}$$

   where the matrix $\tilde{\Xi}_i = \left(2, -\frac{1}{r}\iota_r'\right)$. This constructs a single estimate averaged over all $r$ splits.
```

1.7 Monte Carlo

To compare the most used approaches in econometrics to the extended neural network it is important to first consider the univariate case ($p = 1$). This is a useful exercise as in smooth univariate settings, linear sieves and kernel estimation perform well and have been used across a myriad of empirical applications.

1.7.1 Univariate Simulation Design

Let the data generating process be defined by:

$$x_i \sim Tri[-1, 0, 1] \tag{1.117}$$

$$g(x_i) = \sin(3\pi x_i/2)(1 + 18x^2[\text{sgn}(x) + 1])^{-1} \tag{1.118}$$

$$y_i = g(x_i) + e_i \tag{1.119}$$

$$e_i \sim N(0, 1) \tag{1.120}$$

I chose this function as it was recently used in Calonico, Cattaneo, and Farrell (2018). However, I modify the covariate distribution to be symmetric triangular rather than uniform. I report results across various choices of $d_n = K_n$ for $n = 500$ and $n = 2500$ with $r \in \{0, 5, 10\}$.

The nominal level for confidence intervals is 95%. In addition, I report the percentage of simulations (out of 500) for which the asymptotic variance produced unstable estimates without cross-training. For comparison I consider local linear and local constant kernels of various order where the bandwidth is chosen via cross-validation.

As noted in section 1.6.4, estimation without the cross-training extension begins to become unstable once $K_n = 10$ for $n = 500$ and is never stable for our choices of K_n when $n = 2500$.

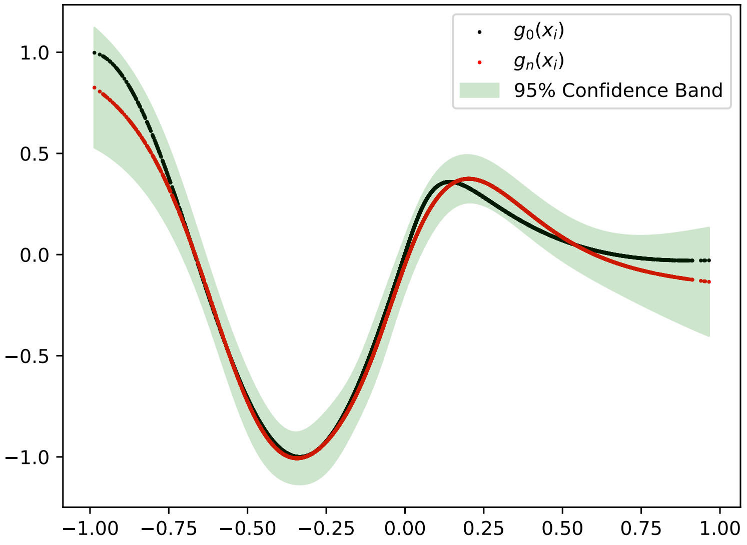


Figure 1.3: An illustration of the extended neural network output with a 95% confidence band in the univariate simulation design. The width of both layers in the extended neural network pictures is ten.

It is apparent that for inference standard estimation (without cross-training) does not work well. In addition, the deep extension performs slightly better relative to *xnn* when the sample size is larger. In addition, the cross-trained extended neural network outperforms kernel estimation in RMSE across all choices of $K_n = d_n$ when $r = 10$ and nearly all choices when $r = 5$.

Method	Kernel-Ord.	RMISE	Abs. Bias	ISD	SE	Med Len	Avg. Coverage	Med. CV bw
Const.	Epa-2^{nd}	0.1317	0.0478	0.1174	0.1105	0.4176	0.9180	0.0852
	Epa-4^{th}	0.1285	0.0426	0.1168	0.1052	0.3996	0.8970	0.1882
	Epa-6^{th}	0.1299	0.0415	0.1188	0.1044	0.3925	0.8930	0.2883
Linear	Epa-2^{nd}	0.1305	0.0485	0.1148	0.1037	0.3972	0.8920	0.0966

Neural Net K	Est.	RMISE	Abs. Bias	ISD	SE	Med Len	Avg. Coverage	Fail-pct
6	NN	0.1220	0.0321	0.1041	0.1521	0.5711	0.9839	<1%
	X-NN-5	0.1197	0.0416	0.1023	0.1166	0.4543	0.9600	0%
	X-NN-10	0.1229	0.0400	0.1070	0.1178	0.4587	0.9520	0%
	X-DNN-5	0.1344	0.0328	0.1165	0.1272	0.4924	0.9560	0%
	X-DNN-10	0.1361	0.0385	0.1182	0.1285	0.4974	0.9480	0%
10	NN	0.1426	0.0181	0.1172	0.1784	0.6787	0.9925	47%
	X-NN-5	0.1207	0.0388	0.1032	0.1241	0.4856	0.9700	0%
	X-NN-10	0.1202	0.0353	0.1038	0.1246	0.4868	0.9699	<1%
	X-DNN-5	0.1291	0.0293	0.1140	0.1433	0.5542	0.9780	0%
	X-DNN-10	0.1257	0.0272	0.1128	0.1451	0.5617	0.9840	0%
12	NN	0.1504	0.0158	0.1216	0.1912	0.7198	0.9915	76%
	X-NN-5	0.1208	0.0355	0.1043	0.1279	0.5001	0.9740	0%
	X-NN-10	0.1188	0.0323	0.1042	0.1268	0.4956	0.9760	<1%
	X-DNN-5	0.1282	0.0274	0.1148	0.1495	0.5783	0.9850	0%
	X-DNN-10	0.1245	0.0217	0.1126	0.1491	0.5763	0.9880	<1%
14	NN	0.1545	0.0147	0.1258	0.1846	0.6984	0.9886	82%
	X-NN-5	0.1210	0.0324	0.1021	0.1311	0.5111	0.9800	0%
	X-NN-10	0.1197	0.0305	0.1047	0.1295	0.5041	0.9780	0%
	X-DNN-5	0.1262	0.0234	0.1146	0.1548	0.5952	0.9900	0%
	X-DNN-10	0.1259	0.0214	0.1138	0.1550	0.5982	0.9900	0%
16	NN	0.1557	0.0145	0.1260	0.1950	0.7343	1.0000	83%
	X-NN-5	0.1250	0.0297	0.1052	0.1340	0.5200	0.9820	<1%
	X-NN-10	0.1185	0.0281	0.1050	0.1316	0.5115	0.9800	0%
	X-DNN-5	0.1286	0.0191	0.1173	0.1590	0.6105	0.9900	0%
	X-DNN-10	0.1278	0.0175	0.1159	0.1563	0.6056	0.9900	<1%

Table 1.1: Simulation results for the first design with $n = 500$ over 500 replications.

Method	Kernel-Ord.	RMISE	Abs. Bias	ISD	SE	Med Len	Avg. Coverage	Med. CV bw
Local Const	Epa-2nd	0.0682	0.0229	0.0613	0.0594	0.2274	0.9240	0.0567
	Epa-4th	0.0632	0.0170	0.0587	0.0555	0.2126	0.9220	0.1322
	Epa-6th	0.0650	0.0172	0.0610	0.0552	0.2110	0.9120	0.2038
Local Lin.	Epa-2nd	0.0668	0.0245	0.0588	0.0563	0.2148	0.9120	0.0645

Neural Net K	Est.	RMISE	Abs. Bias	ISD	SE	Med Len	Avg. Coverage	Fail-pct
10	NN	0.0665	0.0178	0.0528	0.0780	0.2976	0.9872	53%
	X-NN-5	0.0684	0.0394	0.0452	0.0572	0.2271	0.9598	<1%
	X-NN-10	0.0671	0.0413	0.0446	0.0538	0.2134	0.9360	0%
	X-DNN-5	0.0618	0.0279	0.0495	0.0617	0.2409	0.9660	0%
	X-DNN-10	0.0599	0.0269	0.0473	0.0628	0.2446	0.9780	0%
14	NN	0.0689	0.0125	0.0550	0.0893	0.3362	0.9865	85%
	X-NN-5	0.0684	0.0394	0.0452	0.0572	0.2271	0.9598	<1%
	X-NN-10	0.0660	0.0393	0.0446	0.0557	0.2216	0.9559	<1%
	X-DNN-5	0.0604	0.0240	0.0482	0.0650	0.2538	0.9760	<1%
	X-DNN-10	0.0596	0.0219	0.0496	0.0652	0.2549	0.9780	0%
18	NN	0.0701	0.0117	0.0566	0.0931	0.3458	0.9861	85%
	X-NN-5	0.0715	0.0334	0.0499	0.0619	0.2394	0.9718	<1%
	X-NN-10	0.0644	0.0362	0.0443	0.0575	0.2275	0.9679	<1%
	X-DNN-5	0.0605	0.0189	0.0515	0.0684	0.2666	0.9820	<1%
	X-DNN-10	0.0582	0.0192	0.0494	0.0669	0.2610	0.9860	<1%
22	NN	0.0703	0.0104	0.0570	0.0981	0.3509	0.9615	89%
	X-NN-5	0.0996	0.0301	0.0785	0.0813	0.2676	0.9792	4%
	X-NN-10	0.0634	0.0336	0.0448	0.0587	0.2313	0.9717	1%
	X-DNN-5	0.0626	0.0145	0.0554	0.0730	0.2807	0.9920	<1%
	X-DNN-10	0.0587	0.0171	0.0508	0.0686	0.2678	0.9880	<1%
26	NN	0.0705	0.0110	0.0566	0.1347	0.4184	1.0000	86%
	X-NN-5	0.3016	0.0384	0.2526	0.1965	0.4830	0.9902	18%
	X-NN-10	0.0629	0.0314	0.0446	0.0600	0.2355	0.9799	<1%
	X-DNN-5	0.0736	0.0129	0.0647	0.0819	0.2996	0.9899	1%
	X-DNN-10	0.0586	0.0154	0.0505	0.0697	0.2717	0.9920	<1%

Table 1.2: Simulation results for the first design with $n = 2,500$ over 500 replications.

1.7.2 Bivariate Simulation Design

In the second design the dimension of the input is increased to $p = 2$. The true function is a non-additive bivariate function of x_1 and x_2. This setting is of much greater interest as standard sieve or kernel estimators start performing poorly in practice for larger p. The case where $p = 2$ is already a non-trivial exercise. Let the data be generated as:

$$x_{ij} \sim Tri[0, 0.5, 1] \quad j \in \{1, 2\} \tag{1.121}$$

$$g_0(x_i) = \frac{40 \exp\{8((x_1 - .5)^2 + (x_2 - .5)^2)\}}{(\exp\{(8((x_1 - .2)^2 + (x_2 - .7)^2)\} \exp\{(8((x_1 - .7)^2 + (x_2 - .2)^2)))\}} \tag{1.122}$$

$$y_i = g_0(x_i) + e_i \tag{1.123}$$

$$e_i \sim N(0, 1) \tag{1.124}$$

The function is difficult to visualize solely from the formula, but is depicted in figure 1.4.

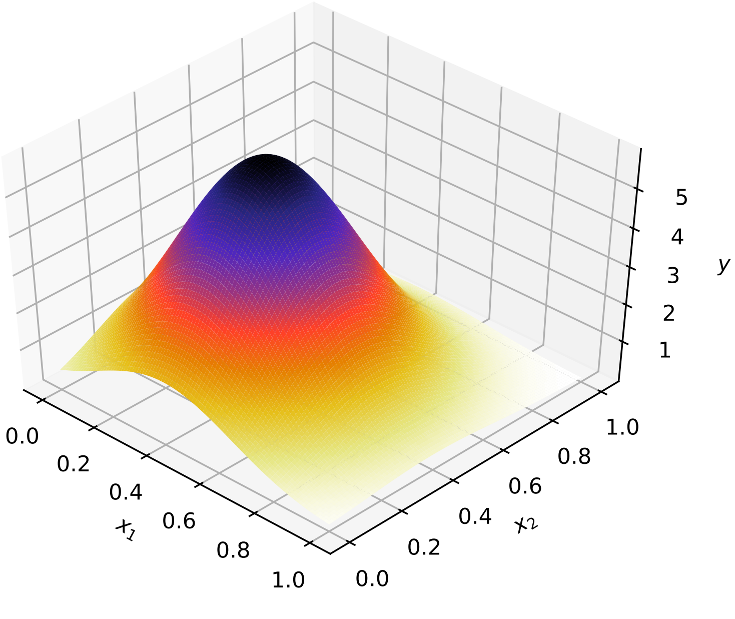


Figure 1.4: Visualization of the true function g_0 in the bivariate simulation design.

Method	Kernel-Ord.	RMISE	Abs. Bias	ISD	SE	Med Len	Avg. Coverage	Med. CV bw
Local Const	Epa-2^{nd}	0.2796	0.1436	0.2234	0.2227	0.8649	0.8920	0.0536
	Epa-4^{th}	0.2679	0.1371	0.2095	0.1846	0.7168	0.8140	0.1215
	Epa-6^{th}	0.2809	0.1457	0.2189	0.1766	0.6889	0.7500	0.1745
Local Lin	Epa-2^{nd}	0.2516	0.0985	0.2070	0.1939	0.7534	0.9090	0.0660
Neural Net K	Est.	RMISE	Abs. Bias	ISD	SE	Med Len	Avg. Coverage	Fail-pct
10	NN	0.2502	0.0628	0.2120	0.2503	0.9088	0.9404	22%
	X-NN-5	0.2280	0.0405	0.1963	0.1632	0.6194	0.8830	0%
	X-NN-10	0.2258	0.0457	0.1904	0.1651	0.6254	0.8960	0%
	X-DNN-5	0.3054	0.0717	0.2727	0.1865	0.7102	0.8060	0%
	X-DNN-10	0.2316	0.0322	0.2120	0.1812	0.6906	0.8948	<1%
14	NN	0.2383	0.0495	0.2014	0.3007	1.0770	0.9732	70%
	X-NN-5	0.2214	0.0396	0.1908	0.1748	0.6624	0.9200	0%
	X-NN-10	0.2186	0.0448	0.1859	0.1754	0.6631	0.9279	<1%
	X-DNN-5	0.2985	0.0599	0.2673	0.2088	0.7967	0.8660	0%
	X-DNN-10	0.2179	0.0249	0.2030	0.2012	0.7674	0.9420	0%
18	NN	0.2296	0.0323	0.1932	0.4045	1.3256	0.9825	88%
	X-NN-5	0.2169	0.0430	0.1862	0.1835	0.6938	0.9400	0%
	X-NN-10	0.2122	0.0438	0.1792	0.1831	0.6913	0.9458	<1%
	X-DNN-5	0.2751	0.0490	0.2482	0.2229	0.8488	0.9120	0%
	X-DNN-10	0.2082	0.0221	0.1880	0.2126	0.8092	0.9699	<1%
22	NN	0.2291	0.0263	0.1882	6.9869	15.7343	1.0000	92%
	X-NN-5	0.2167	0.0443	0.1831	0.1916	0.7226	0.9509	<1%
	X-NN-10	0.2127	0.0419	0.1788	0.1877	0.7078	0.9498	<1%
	X-DNN-5	0.2546	0.0372	0.2306	0.2327	0.8893	0.9479	<1%
	X-DNN-10	0.2080	0.0210	0.1867	0.2212	0.8404	0.9780	0%

Table 1.3: Simulation results for the second design with $n = 500$ over 500 replications.

In the multivariate setting the performance of *xnn* and *xdnn* is noticeably better than the bivariate kernel across all metrics for both sample sizes. In addition, the inclusion of cross-training is essential if inference is desired. The asymptotic variance for the minima found without this approach is almost never finite. The performance of the cross-trained classical and extended neural networks are similar with respect to RMSE for the smaller sample size with the slight edge going to the extended model in the larger n case. The extended neural network exhibits reduced bias from the increased complexity but an increased monte carlo variance as well as the asymptotic estimates.

Method	Kernel-Ord.	RMISE	Abs. Bias	ISD	SE	Med Len	Avg. Coverage	Med. CV bw
Local Const	Epa-2^{nd}	0.1626	0.0871	0.1256	0.1298	0.5029	0.9100	0.0405
	Epa-4^{th}	0.1567	0.0810	0.1205	0.1067	0.4145	0.8460	0.0975
	Epa-6^{th}	0.1657	0.0888	0.1254	0.1030	0.4015	0.7660	0.1478
Local Lin	Epa-2^{nd}	0.1393	0.0596	0.1108	0.1080	0.4189	0.9140	0.0528

Neural Net K	Est.	RMISE	Abs. Bias	ISD	SE	Med Len	Avg. Coverage	Fail-pct
18	NN	0.1473	0.0378	0.1240	0.1473	0.5316	0.9216	89%
	X-NN-5	0.1070	0.0195	0.0919	0.0791	0.2964	0.8980	0%
	X-NN-10	0.1062	0.0213	0.0932	0.0795	0.2986	0.8920	0%
	X-DNN-5	0.1020	0.0095	0.0918	0.0902	0.3442	0.9440	0%
	X-DNN-10	0.1004	0.0103	0.0895	0.0935	0.3558	0.9540	0%
24	NN	0.1252	0.0263	0.1034	4.4848	16.1341	1.0000	95%
	X-NN-5	0.1042	0.0204	0.0878	0.0861	0.3221	0.9357	<1%
	X-NN-10	0.1053	0.0207	0.0897	0.0836	0.3136	0.9217	<1%
	X-DNN-5	0.1025	0.0103	0.0909	0.0969	0.3690	0.9639	<1%
	X-DNN-10	0.1002	0.0101	0.0888	0.0994	0.3776	0.9699	<1%
30	NN	0.1175	0.0212	0.0943	7.4350	28.2281	1.0000	94%
	X-NN-5	0.1041	0.0195	0.0879	0.0912	0.3396	0.9513	1%
	X-NN-10	0.1053	0.0222	0.0886	0.0881	0.3291	0.9398	<1%
	X-DNN-5	0.1017	0.0106	0.0895	0.1028	0.3899	0.9720	0%
	X-DNN-10	0.1002	0.0106	0.0882	0.1034	0.3917	0.9780	<1%
36	NN	0.1166	0.0195	0.0940	13.0976	45.0650	1.0000	95%
	X-NN-5	0.1045	0.0180	0.0881	0.0957	0.3544	0.9635	1%
	X-NN-10	0.1041	0.0202	0.0881	0.0923	0.3442	0.9554	1%
	X-DNN-5	0.1009	0.0100	0.0881	0.1072	0.4053	0.9819	<1%
	X-DNN-10	0.1003	0.0100	0.0875	0.1070	0.4048	0.9819	<1%

Table 1.4: Simulation results for the second design with $n = 2,500$ over 500 replications.

1.8 Conclusion and Future Work

I have shown consistency and relatively fast convergence rates for neural networks with two hidden layers. This estimator has good finite sample properties and may improve on the single layer counterpart in large n settings depending on the dimensionality of x_i and the smoothness properties of the underlying function. I developed a straightforward approach to constructing asymptotically valid pointwise confidence intervals. This construction relies on standard sieve results for inference but utilizes cross-training to improve empirical performance.

There are many avenues to consider extensions for this estimator and variants thereof. The two directions that seem most promising and potentially fruitful are a rigorous analysis of cross-training and extending the approximation results. In the former case a rigorous analysis of cross-training with respect to the finite sample optimization problem as well as the implications for the stochastic equicontinuity conditions is necessary to determine the impact on asymptotic normality. In addition, this analysis may illuminate directions for more general cases of data dependence.

In terms of the approximation extensions the obvious one is to iterate to many hidden layers. However, it is not clear that doing so will result in an improvement for fully connected networks without extending the underlying function space. It seems more useful to elaborate on comments made in section 1.4.1 where each adaptive basis is a function of subsets of the parameter space. This would allow more flexibility in the construction while decreasing the entropy cost for adding layers.

Chapter 2

Estimation of Panel Data with Neural Networks

This chapter introduces a novel formulation of neural networks for estimation under panel data structures. I propose an estimator consistent with a common cross-sectional index and allow for unobserved heterogeneity to be correlated with this index in the form of cross-sectional fixed effects. The inclusion of fixed effects results in a non-negligible asymptotic bias in the limiting distribution of the index. I correct this bias by re-centering the scores and demonstrate its effectiveness in finite samples. I apply the panel neural network to the demand for cigarettes in the United States and compare the results to commonly used alternative models. I find that own-price elasticity for cigarettes is more elastic than is typically cited in the literature and argue this relationship is consistent with the curvature of aggregate demand. In addition, I empirically verify the regressive nature of cigarette taxes by examining the conditional average own-price elasticities with respect to varying total expenditure quantiles.

2.1 Introduction

Neural networks and deep learning have had excellent empirical success in both time series and cross-sectional settings. However, economic data is often available in a panel or pseudo-panel format. The current literature on neural networks does not have much to say on how to leverage panel structure in estimation. One may ignore the panel structure by slicing the data in either cross-sectional or time dimensions and treating them as separate models. The former estimator would be consistent with cross-sectional heterogeneity while the latter with time heterogeneity. However, nonparametric estimation[1] of individual time series or repeated cross-sections is not possible when the number of observations, in the time or cross-sectional dimension respectively, is small. This chapter fills this gap in the literature by introducing the panel neural network. This estimator assumes a common index across both time and cross-sectional units taking advantage of both T and n. I will focus on continuous outcomes and the classical neural network[2] as to avoid confounding the extension to panel data.

One of the most important benefits of panel data is the ability to incorporate unobserved heterogeneity in some form. In economics this is typically through fixed effects or common correlated effects. In this chapter, I focus on the former and allow for the inclusion of additive cross-sectional effects. These parameters can be estimated directly alongside the common index as cross-sectional specific intercepts. However, estimation of these effects induces an additional bias term that does not vanish asymptotically. This bias is present due to the dimension of the fixed effect parameters[3] which grows linearly with the cross-sectional dimension of the panel.

In the nonparametric panel data literature with continuous outcomes, the inclusion of fixed effects is typically straightforward. In the linear sieve case, e.g., power series, the transfor-

[1] In addition, this choice should be informed by the underlying model and assumptions one is willing to make rather than a gap in the literature.

[2] Neural networks with a single hidden layer

[3] The fixed effects are often referred to as incidental parameters in this context.

mations are fixed in finite samples and standard transformations can eliminate the need to estimate the fixed effects directly. The panel neural network, along with other nonlinear sieves, does not admit such straightforward approach. However, taking inspiration from the nonlinear panel literature this problem can be alleviated by correcting the first order conditions of the objective with respect to the function estimates. This correction eliminates the first order bias induced by estimating the fixed effects. I provide this correction and demonstrate its effectiveness in finite samples.

In an application, I focus on cigarette demand in the United States using the Nielsen consumer panel data spanning ten years from 2007-2017. To maximize the number of households I break up the panel into eight overlapping four-year periods. I compare the panel neural network to a power series and linear model all including household level fixed effects. The primary object of interest is the average and conditional average own-price elasticity.

I find that own-price elasticity for cigarettes is slightly more elastic than is typically cited in the literature, but argue this relationship is consistent with the curvature of aggregate demand. In particular, the elasticities are larger in absolute value in the initial years when substantial tax changes shifted consumers into a more elastic region of demand. As price changes leveled out in subsequent years, the elasticities returned to levels more in line with the literature. In addition, I empirically verify the regressive nature of cigarette taxes by examining the conditional average own-price elasticities by total expenditure quantiles. These elasticities are monotonically decreasing in total expenditure suggesting a larger burden of the tax falls on lower income households.

The remaining of the chapter proceeds as follows. The first section (2.2) gives a brief overview of the relevant literature. The next section (2.3) presents the model and assumptions. Section 2.4 presents the consistency result for the panel neural network along with score corrections for the incidental parameters problem. The finite sample performance of the estimator is assessed in 2.6. I then focus on the application to cigarette demand in sections 2.7 through sections 2.7.4 and conclude in 2.8.

2.2 Literature Review

The relevant literature pertaining to neural networks is the same as in the previous chapter, but with heavy emphasis on the consistency results from White (1990) and $o_p(n^{-1/4})$ convergence rates in Chen and White (1999). I utilize these results but focus on panel structure and allow for inclusion of incidental parameters.

The problem of incidental parameters was first examined by Neyman and Scott (1948), who observed the inconsistency of the parameters of interest[4] in the presence of incidental parameters. Under fixed T asymptotics, common parameter estimation is typically restricted to specific parametric models and even then, only a subset of these parameters can be consistently estimated. For example, the conditional Logit model allows estimation of the effects of an observable on the log odds, but average partial effects (APEs) are not available. However, if one is willing to allow $n, T \to \infty$ where $n/T \to \kappa$ for some $0 < \kappa < \infty$, more general results can be obtained. The nonparametric panel literature has examined continuous outcome models with fixed effects[5], but the solutions are not directly applicable to nonlinear sieves. The primary complication is that the transformations of the covariates are not fixed, but data dependent.

The most relevant literature for this chapter is that of nonlinear panels. Although I consider only continuous outcomes, which would fall under the linear model in a fully parametric case, the insights from the nonlinear panel literature can be extended to more general models. This literature has developed a few alternative approaches to dealing with the bias[6]. The most

[4] These parameters are also referred to as common parameters in the literature.

[5] See Sun, Zhang, and Li (2015) for a recent review.

[6] A comprehensive historical review is available in Arellano, Hahn, et al. (2007) and more recent results in Fernández-Val and Weidner (2018).

pertinent for this chapter is the correction based on re-centering the score (Bester and Hansen (2009), Fernández-Val and Weidner (2016)). Heuristically, the score for the neural network parameters is adjusted to eliminate the first order impact of estimating the fixed effects. The score is then invariant to small perturbations in the incidental parameter space. This approach lends itself particularly well to neural networks as first order optimization methods are almost exclusively used to estimate these models. At each iteration of the optimization procedure, the score can be updated with a new correction based on the current solution. The score correction terms are derived from a functional second order expansion but have simple expressions when treating the models 'as-if' they were parametric as in Hahn, Liao, and Ridder (2018).

It is worth noting there are other ways that may work equally well for eliminating the first order bias in these models. Simulation based techniques like the jackknife in Hahn and Newey (2004), Hahn and Kuersteiner (2002), Carro (2007), Fernández-Val (2009), Fernández-Val and Vella (2011), and Fernández-Val and Weidner (2016) have been shown to work well in parametric models.[7] However, these procedures require repeated passes of the data, increasing computational complexities, and are typically associated with a decrease in precision.

2.3 Model

I consider the nonparametric regression model where one observes $n \times T$ realizations from the random vector $\left\{\{z_{it}\}_{t=1}^{T}\right\}_{i=1}^{n}$ where $z_{it} = (y_{it}, x'_{it})'$. Each cross-sectional outcome y_{it} is sampled from:

$$y_{it} = \alpha_i + g_0(x_{it}) + e_{it}, \qquad \mathbb{E}[e_{it}|x_{it}] = 0, \qquad \mathbb{E}[e_{it}^2|x_{it}] = \sigma^2(x_{it}) \tag{2.1}$$

[7] Some initial work suggests these can work in this setting as well, but with much higher computation cost relative to the analytical correction.

A2.3.1 For each $t \in \{1, 2, \ldots, T\}$, the random vectors $\{z_i\}_{i=1}^n = \{y_i, x_i'\}_{i=1}^n$ are independent and identically distributed conditional on α_i. In addition, $y_{it} \in \mathcal{Y} \subset \mathbb{R}$ and $x_{it} \in \mathcal{X} \subset \mathbb{R}^p$ where $\mathcal{X}$ and $\mathcal{Y}$ are compactly supported.

A2.3.2 The vectors $\{z_t\}_{t=1}^T$ are stationary ϕ-mixing sequences with $\phi(k) = \phi_0 \zeta^k$, $\zeta \in (0,1)$, and $k > 0$ where:

$$\phi(k) \equiv \sup_{t \in \mathbb{N}} \sup_{\Pr(G)>0, G \in \{z\}_{-\infty}^t, H \in \{z\}_{t+k}^\infty} |\Pr(H|G) - \Pr(H)| \tag{2.2}$$

A2.3.3 The unknown index $g_0 \in \mathcal{W}_2^q(\mathcal{X})$ where $\mathcal{W}$ is a Sobelev space with q weak derivatives and has a Fourier representation:

$$g_0(x_{it}) = \int \exp(i\delta' x_{it}) d\sigma_g(\delta) \tag{2.3}$$

where σ_g is a complex measure on $\mathbb{R}^p$ satisfying:

$$\int \max\{|\delta|, 1\}^{q+1} d\left|\sigma_g\right|(\delta) < \infty \tag{2.4}$$

A2.3.4 Let $m = nT$. The parameters of the sieve space $\mathcal{G}_m$ satisfy the following bounds:

$$||\gamma||_1 \leq \Delta_m, \quad \sum_{j=1}^{d_m} ||\delta_j||_1 \leq d_m \Delta_m \tag{2.5}$$

where $\Delta_m, d_m \to \infty$ slowly with m

Here I utilize common sampling assumptions in the panel literature where cross-sectional units are independent across observations but temporally dependent. The assumptions on functional form **A2.3.3** and parameter restrictions **A2.3.4** follow from White (1990) and Chen and White (1999). I utilize these results rather than those established in the previous chapter to avoid additional complications in deriving the score corrections.

2.4 Asymptotic Theory

The consistency and convergence rates for neural networks under weakly dependent time series is shown in White (1990) and Chen and White (1999) respectively. In this chapter I extend these results to a panel data setting. In addition, the probability limit of the panel estimator is not centered at the correct value and will be asymptotically biased. This bias occurs when one has to estimate the fixed effects, inducing the incidental parameter bias discussed in 2.1. Fortunately, this bias can be analytically corrected by examining the discrepancy between the score of the infeasible and plug-in estimators.

I make the following assumptions on the stochastic error term e_{it} and properties of the objective:

A2.4.1 The second unconditional moment of the error term exists and is finite $\mathbb{E}[e_{it}^2] < \infty$.

A2.4.2 The sieve spaces $\mathcal{G}_m$ are compact.

A2.4.3 The population objective $Q(g, \alpha_i)$ is continuous at g_0 and for any $\varepsilon > 0$:

$$Q(g_0, \alpha_i) - \sup_{g \in \mathcal{G}_m : ||g, g_0|| > \varepsilon} Q(g, \alpha_i, z_i) > 0$$

As mentioned in chapter 1 section 1.5, the first condition is quite weak, ruling out only pathological examples like the Cauchy and is made in the vast majority of both parametric and nonparametric literature. The latter conditions are also the same as the previous chapter and are standard in the nonparametric literature, e.g., Chen and Shen (1998), Shen and Wong (1994), or Chen (2007). Compactness follows directly from the definition of the neural network spaces $\mathcal{G}_m$. The identification condition, **A2.4.3** is a standard regularity condition and ensures a first order condition identifies an optimal parameter conditional on the value of α_i. This assumption is critical as it allows us to utilize the score corrections provided in the subsequent section 2.4.2.

2.4.1 Consistency

Consider the sieve least squares problem:

$$\sup_{g\in\mathcal{G}_m} Q_m(g,\alpha_i) = \sup_{g\in\mathcal{G}_m} -\frac{1}{m}\sum_{i=1}^{n}\sum_{t=1}^{T}\ell_{it}(g,\alpha_i,z_{it}) \tag{2.6}$$

where $m = nT$ and $\ell_{it}(g,\alpha_i,z_{it}) = (y_{it}-\alpha_i-g(x_{it}))^2$. The sieve space I consider is the classical neural network defined as:

$$g_m(x_{it}) = \sum_{j=1}^{d} s(\tilde{x}_{it}'\delta_j)\gamma_j \tag{2.7}$$

where $\tilde{x}_{it} = (1, x_{it}')'$ as in chapter 1. The first key result is to establish uniform consistency of the objective for a fixed α_i. This consistency result for panel neural networks is new as it pertains to a panel data setting rather than cross-sectional or time series.

Theorem 2.1. *If the sampling and function form assumptions from section 2.4 hold along with* **A2.4.1**, **A2.4.2**, **A2.4.3**, $\Delta_m = o(m^{1/4})$ *and* $d_m\Delta_m^2\log\Delta_m d_m = o(m^{1/2})$ *then:*

$$\lim_{m\to\infty}\sup_{g\in\mathcal{G}_m}|Q_m(g,\hat{\alpha}_i)-Q(g,\hat{\alpha}_i)| = 0$$

and $g(\hat{\alpha}_i)_m \xrightarrow{p} g_0(\hat{\alpha}_i)$

Proof. Using boundedness and the sampling assumptions from **A2.3.1** and **A2.3.2** one can stack the time series vectors and use the fact that independence implies ϕ-mixing. The resulting vector ℓ_m is a bounded stationary ϕ-mixing sequence. It suffices to verify the conditions from theorem 2.5 in White and Wooldridge (1991).

First note that one can use Bernstein's Inequality for ϕ-mixing data to bound deviations from the expectation of the objective function:

$$\Pr\left[\left|\sum_{i,t}\ell(g_m, z_{it}, \hat{\alpha}_i) - \mathbb{E}\left[\ell(g_m, z_{it}, \hat{\alpha}_i)\right]\right| > \epsilon\right] \leq c_1 \exp\left[\frac{-c_2\epsilon m^{-1/2}}{2\Delta_m^2}\right] \tag{2.8}$$

where $c_1, c_2 \in (0, \infty)$. In addition, one can use the Lipschitz property of ℓ_m to define a bound on differences between the objective for any $g_m \in G_m$:

$$|\ell_m(g_m) - \ell_m(g_0)| \leq \sup_{g \in G_m} |y_{it} - g - \hat{\alpha}_i| = \tilde{\ell}_m \tag{2.9}$$

Using 2.9 and Bernstein's Inequality one can bound deviations of this difference from its expectation:

$$\Pr\left[\left|\sum_{i,t}\tilde{\ell}(g_m, z_{it}, \hat{\alpha}_i) - \mathbb{E}\left[\tilde{\ell}(g_m, z_{it}, \hat{\alpha}_i)\right]\right| > \epsilon\right] \leq c_3 \exp\left[\frac{-c_4\epsilon m^{-1/2}}{2\Delta_m}\right] \tag{2.10}$$

where, as before, $c_3, c_4 \in (0, \infty)$. Putting these bounds together one has the desired maximal inequality:

$$\Pr\left[\sup_{g \in \mathcal{G}_m}\left|m^{-1}\sum_{i=1}^{n}\sum_{t=1}^{T}[\ell(g_m, z_{it}, \hat{\alpha}_i) - \mathbb{E}\left(\ell(g_m, z_{it}, \hat{\alpha}_i)\right)]\right| > \epsilon\right] \tag{2.11}$$

$$\leq C_1 \exp \mathbf{H}_m\left(\left[\frac{m\epsilon}{6W_m}\right], \mathcal{G}_m, ||\cdot||_\infty\right)\left[\exp\left[\frac{-c_2\epsilon W_m m^{-1/2}}{\Delta_m}\right] + \exp\left[\frac{-c_4\epsilon m^{1/2}}{6\Delta_m^2}\right]\right] \tag{2.12}$$

where $C_1 = \max\{c_1, c_3\}$, $W_m \geq \sup_{g_m \in G_m}\sum_{i,t}\mathbb{E}\tilde{\ell}_m$, and $\mathbf{H}_m$ is the metric entropy. Now note that W_m can be taken to be $2m\Delta_m$ such that the second term is dominant as $\Delta_m \to \infty$ with m. It suffices to show $\Delta_m^2/\sqrt{m} \to 0$ as $m \to \infty$ and $\forall\epsilon > 0$

$$\mathbf{H}_m\left(\epsilon/\left(12\Delta_m\right), \mathcal{G}_m, ||\cdot||_\infty\right)\left[\Delta_m^2/\sqrt{m}\right] \to 0 \quad \text{as } m \to \infty \tag{2.13}$$

The first condition is satisfied with $\Delta_n = o(m^{1/4})$. In addition, the metric entropy of $\mathcal{G}_m$ can be shown to be:

$$\mathbf{H}_m(\epsilon, \mathcal{G}_m, ||\cdot||_\infty) \leq \omega_m \left[\log\frac{8}{\epsilon} + \log\left(\Delta_m + p\Delta_m^2\right) + \log d_m\right] \tag{2.14}$$

where $\omega_m = d_m(p+2)$ is the number of parameters to characterize g_m. Plugging in 2.14 to the entropy component of 2.13:

$$\mathbf{H}_m(\epsilon/12\Delta_m, \mathcal{G}_m, ||\cdot||_\infty) \leq \omega_m \log\frac{96\Delta_m}{\epsilon} + \omega_m \log\left(\Delta_m + p\Delta_m^2\right) + \omega_m \log d_m \tag{2.15}$$

Then $\exists$ an $m \in \mathbb{N}$ s.th. $\forall \epsilon > 0$ $\Delta_m \geq 96/\epsilon$ and $\Delta_n^2 \geq \Delta_n$.

$$\begin{aligned}\mathbf{H}_m(\epsilon/12\Delta_m, \mathcal{G}_m, ||\cdot||_\infty) &\leq \omega_m \log\Delta_m^2 + \omega_m \log\Delta_m^2(p+1) + \omega_m \log d_m \\ &\leq \omega_n 4\log\Delta_n(p+1)d_n\end{aligned}$$

Plugging this result into the full equation 2.13:

$$\mathbf{H}_m\left(\epsilon/\left(12\Delta_m\right), \mathcal{G}_m, ||\cdot||_\infty\right)\left[\Delta_m^2/\sqrt{m}\right] \leq m^{-1/2} 16\Delta_m^2 \omega_m \log\Delta_m(p+1)d_m$$

Also note that $\omega_m = O(d_m)$ therefore $d_m\Delta_m^2 \log\Delta_m d_m = o(m^{1/2})$ is sufficient. □

This result is useful by itself for the cases where $\hat{\alpha}_i$ can be absorbed into the stochastic error component, often referred to as a random effects model. However, in the fixed effects framework theorem 2.1 is not enough. It will be the case that $g_m(\hat{\alpha}_i) \xrightarrow{p} g_0(\hat{\alpha}_i)$, but unless there is either no estimation error for $\hat{\alpha}_i$, one has access to the true values, or estimation of $\hat{\alpha}_i$ has no impact on estimation of g_m, the estimates will be biased. Furthermore, this bias does not vanish asymptotically as the number of parameters to be estimated grows linearly with n.

2.4.2 Score Corrections

Determining the order of the bias depends on the second directional derivative of the loss function with respect to g_0 followed by an expansion around the incidental parameters. The pathwise derivative of ℓ_{it} in the direction of v_g is defined as $\Delta_\ell[v_g]$. The most convenient way to approach this expansion is to first consider the orthogonal reparameterization as in Fernández-Val and Weidner (2018):

$$\alpha_i^\star = \alpha_i - \left(\mathbb{E}_T\left[\frac{1}{m}\sum_{i,t}\frac{\partial \ell_{it}}{\partial \alpha_i \partial \alpha_i'}\right]\right)^{-1}\left(\mathbb{E}_T\left[\frac{1}{m}\sum_{i,t}\frac{\partial \Delta_\ell(g,\hat{\alpha}_i,z_{it})[v_g]}{\partial \alpha_i}\right]\right)' g_0 \tag{2.16}$$

with the corresponding modified objective $\ell_{it}^\star = \ell_{it}(g,\alpha_i^\star,z_{it})$ where α_i is replaced by $\alpha_i^\star$. The supremum of ℓ_{it} and $\ell_{it}^\star$ with respect to g are identical but this modification induces information orthogonality between the incidental parameters and the estimates of the unknown function. A first order approximation of the modified score in the direction of v_g and around α_i:

$$\Delta_{\ell^\star}(g,\hat{\alpha}_i,z_{it})[v_g] = \Delta_{\ell^\star}(g,\alpha_i,z_{it})[v_g] + \frac{\partial \Delta_{\ell^\star}(g,\alpha_i,z_{it})[v_g]}{\partial \alpha_i}(\hat{\alpha}_i^\star - \alpha_i) \tag{2.17}$$

$$= \Delta_{\ell^\star}(g,\alpha_i,z_{it})[v_g] + \frac{\partial \Delta_{\ell^\star}(g,\alpha_i,z_{it})[v_g]}{\partial \alpha_i}\frac{1}{T}\sum_{t=1}^{T}\psi_{it} \tag{2.18}$$

The second line follows[8] from substituting $\hat{\alpha}_i^\star - \alpha_i$ with the first order approximation $\psi_{it} = \mathbb{E}_T\left[\frac{\partial^2 \ell_{it}^\star}{\partial \alpha_i \partial \alpha_i'}\right]^{-1}\frac{\partial \ell_{it}^\star}{\partial \alpha_i}$. The first term is the score, which identifies the population parameter, and the second represents the asymptotic bias. Utilizing the specific form of the objective

[8] This expansion has no higher order terms as the form of ℓ_{it} admits no higher order derivatives with respect to α_i.

and collecting the orthogonalization coefficients of g into $g^{\star}$, the second term is:

$$\begin{aligned}\frac{\partial \Delta_{\ell}^{\star}(g_0, \alpha_i, z_{it})[v_g]}{\partial \alpha_i} &= \frac{\partial}{\partial \alpha_i}\left[\lim_{t \to 0} \frac{\left(y_{it} - \alpha_i - g_0^{\star} - \tau[v_g^{\star}]\right)^2 - \left(y_{it} - \alpha_i - g_0^{\star}\right)^2}{\tau}\right] && (2.19)\\ &= -\frac{\partial 2\left(y_{it} - \alpha_i - g_0^{\star}\right)\left[v_g^{\star}\right]}{\partial \alpha_i} && (2.20)\\ &= 2\left(v_g - \Xi_{it} v_g\right) && (2.21)\end{aligned}$$

where $\Xi_{it} = \left(\mathbb{E}_T\left[\frac{1}{m}\sum_{i,t}\frac{\partial \ell_{it}}{\partial \alpha_i \partial \alpha_i'}\right]\right)^{-1}\left(\mathbb{E}_T\left[\frac{1}{m}\sum_{i,t}\frac{\partial \Delta_{\ell}(g, \hat{\alpha}_i, z_{it})[v_g]}{\partial \alpha_i}\right]\right)'$ and $v_g^{\star} = v_g - \Xi_{it} v_g$. This component can be viewed as the residual from a projection of v_g onto the space spanned by the incidental parameters. Finally, adding the influence function term and taking expectations, the bias in the score can be characterized by:

$$\mathbb{E}\left[\Delta_{\ell}(g, \hat{\alpha}_i, z_{it})[v_g]\right] = \mathbb{E}\left[2\left(v_g - \Xi_{it} v_g\right)\psi_{it}\right] \tag{2.22}$$

It is possible to utilize a higher order expansion for $\alpha_i^{\star} - \alpha_0$. Doing so illuminates a further source of bias as outlined in Hahn and Newey (2004). The source of the additional bias is related to correlation induced by estimating α_i and g_m on the same data. However, this source of bias can be eliminated by leave-t-out estimators of the fixed effects. Construction of such an estimator is outlined in algorithm 2.

2.5 Estimation

In practice one needs plug-in estimates for the bias. The estimate for v_g has two distinct components as the form of g_m has two distinct sets of parameters, δ and γ. The remaining components are standard as they consist of a projection and well-known estimates of the influence function. Let e_{it} be estimated residuals and s_{β} be the derivative of the activation function $s(\cdot)$ with respect to β. The plug-in corrections to the scores for δ_j and γ_j are:

$$B(\delta_j) = \frac{1}{nT} \sum_{i=1} \sum_{t=1} e_{it}(\tilde{x}_{it} s_{\beta}(x'_{it}\delta_j)\gamma_j - \alpha_i^{\star}(\delta_j)) \tag{2.23}$$

$$B(\gamma_j) = \frac{1}{nT} \sum_{i=1} \sum_{t=1} e_{it}(s(\tilde{x}'_{it}\delta_j) - \alpha_i^{\star}(\gamma_j)) \tag{2.24}$$

where $\alpha_i^{\star}$ is the solution to a least squares regression of the scores on the space spanned by the incidental parameters.

The neural network parameters are updated with first order methods while the incidental parameters have closed form solutions conditional on the current values of $\hat{g}_m$. At each iteration the scores for δ and γ are corrected by the current values of $B(\delta_j)$ and $B(\gamma_j)$. If elimination of the own-observation bias is desired one can utilize leave-t-out estimates for $\hat{\alpha}_i$ and $\hat{\alpha}_i^{\star}$. The number of observations that is left out will be dependent on the mixing properties of the time series and will be a tuning parameter in practice[9]. The complete updating scheme is outlined in algorithm 2.

[9] It is likely possible to determine an optimal bandwidth using ideas from Politis and White (2004).

Algorithm 2 Split Score Corrections

```
Set the desired tolerance ($tol$) for convergence.
Let $\theta^{(k)} = (\delta^{(k)}, \gamma^{(k)})$ and $B^{(k)} = \big(B(\delta^{(k)}), B(\gamma^{(k)})\big)$ be the estimates and score corrections
  respectively at iteration $k$.
The function estimate $g_m$ and residuals for iteration $k$ are defined as:
      $g_m(\theta^{(k)}, x_{it}) = \sum_{j=1}^{d} s(\tilde{x}_{it}'\delta_j^{(k)})\gamma_j^{(k)}$
      $e_{it} = y_{it} - \hat{\alpha}_i^{(k)} - g_m(\theta^{(k)}, x_{it})$
Construct the binary bandwidth matrix $W = [w_1, w_2, \ldots, w_T]$ where the vectors $w_j \in W$ have
  elements $w_{jt} = \min\left\{1, \left\lfloor \frac{|j-t|}{h} \right\rfloor\right\}$ for some bandwidth parameter $h \in \mathbb{N}$.
while $\epsilon > tol$ do
    Given current values of $\theta^{(k)}$, $\hat{\alpha}_i^{(k)}$, and residuals set $\hat{\alpha}_i^{(k+1)} = \frac{\sum_{j=1}^{T} e_{ij} w_{jt}}{\sum_{j=1}^{T} w_{jt}}$.
    Construct the score $\Delta_\theta^{(k)}$ and $B^{(k)}$ using the new values $\hat{\alpha}_i^{(k+1)}$.
    Given an update rule $m(\cdot)$ the new value is: $\theta^{(k+1)} = \theta^{(k)} + m(\Delta_\ell^{(k)} - B^{(k)})$
    update $\epsilon$ based on the desired convergence criterion.
end while
```

2.6 Monte Carlo

To assess the finite sample performance of the estimator and score corrections I consider a simulation exercise designed to mimic demand estimation for a product that depends only on its own price p_1 and the price of a substitute p_2. In the simulation the covariates are exogenous, but pre-determined and correlated with unobserved cross-sectional effects. To maintain monotonicity and convexity[10] I focus on relatively simple functional forms that nevertheless allow for interesting substitution patterns.

Let the data be generated from:

$$
\begin{align}
\alpha_i, \varepsilon_{it} &\overset{iid}{\sim} N(0,1) \tag{2.25}\\
\nu_{1it}, \nu_{2it}, &\overset{iid}{\sim} U[0,1] \tag{2.26}\\
p_{1it} &= \beta_{p_1}\alpha_i + \rho_{p_1} p_{1i,t-1} + \nu_{1it} \tag{2.27}\\
p_{2it} &= \beta_{p_1}\alpha_i + \rho_{p_1} p_{1i,t-1} + \nu_{2it} \tag{2.28}\\
q_{it} &= \alpha_i + f(p_{1,it}, p_{2,it}) + \varepsilon_{it} \tag{2.29}\\
f(p_{1,it}, p_{2,it}) &= \exp\left\{\cos(0.5 p_{1,it} p_{2,it}\right\} \tag{2.30}
\end{align}
$$

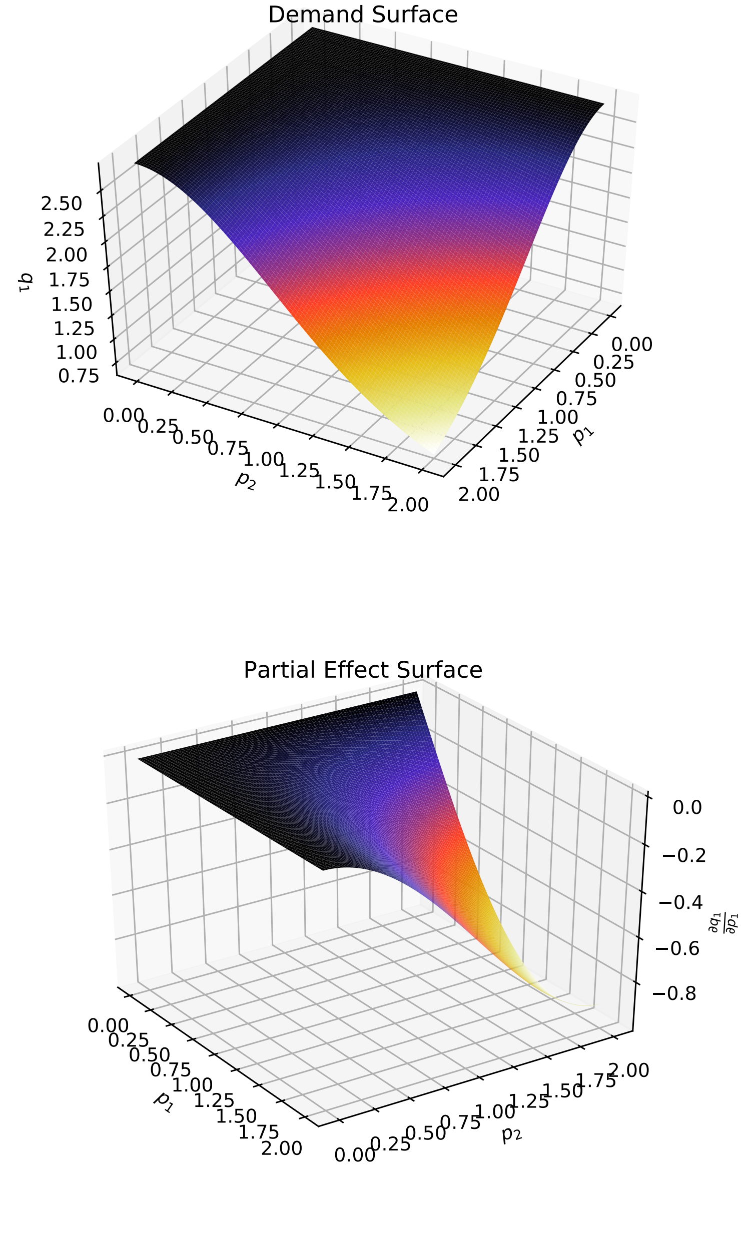
I set $\beta_{p_1} = \beta_{p_2} = 0.2$ and $\rho_{p_1} = \rho_{p_2} = 0.5$.

I will focus on estimation of the demand function and the own-price elasticity. The latter will be the primary object of interest in the empirical application. In this design the analytical partial effects with respect to $p_{1,it}$ is:

$$
\frac{\partial q_{it}}{\partial p_{1,it}} = -0.5 p_{2,it} \sin(0.5 p_{1,it} p_{2,it}) f(p_{1,it}, p_{2,it}) \tag{2.31}
$$

which is estimated by the derivative of the neural network estimator with respect to the first

[10] This conditions hold locally for the support I simulate, but are not global properties of this function.


Demand Surface

Partial Effect Surface

Figure 2.1: Visualization of the simulated demand and own-price partial effect surface.

price:

$$\frac{\partial f(p_{1,it}, p_{2,it})}{\partial p_{1,it}} = \sum_{j=1}^{K} \delta_{j1} s_{p_{1,it}}(\tilde{x}'_{it}\delta_j)\gamma_j \tag{2.32}$$

		Function Estimation				Partial Effects		
n	d	RMSE	Bias	% Bias	ISD	RMSE	Bias	SD
100	8	0.5702	0.4722	0.2293	0.3126	0.2112	0.0076	0.1399
		0.1621	-0.0798	-0.0289	0.0948	0.2527	0.0029	0.1792
	10	0.5574	0.4497	0.2185	0.3228	0.2056	0.0094	0.1331
		0.1669	-0.0824	-0.0301	0.0998	0.2606	0.0031	0.1890
	12	0.5465	0.4322	0.2100	0.3285	0.1986	0.0082	0.1273
		0.1614	-0.0816	-0.0302	0.0970	0.2545	0.0043	0.1862
200	8	0.6026	0.5170	0.2664	0.3010	0.2020	0.0103	0.1205
		0.1526	-0.0350	-0.0015	0.0945	0.2514	0.0085	0.1713
	10	0.5665	0.4792	0.2472	0.2940	0.1923	0.0120	0.1087
		0.1485	-0.0385	-0.0048	0.0903	0.2474	0.0036	0.1701
	12	0.5356	0.4293	0.2222	0.3128	0.1890	0.0121	0.1055
		0.1502	-0.0358	-0.0024	0.0918	0.2482	0.0067	0.1676
300	8	0.5804	0.4980	0.2537	0.2883	0.2013	0.0166	0.1097
		0.1471	-0.0363	-0.0032	0.0885	0.2494	0.0103	0.1643
	10	0.5414	0.4514	0.2305	0.2896	0.1948	0.0160	0.1027
		0.1463	-0.0422	-0.0063	0.0858	0.2451	0.0104	0.1631
	12	0.5193	0.4199	0.2150	0.2964	0.1923	0.0161	0.0984
		0.1466	-0.0378	-0.0038	0.0861	0.2486	0.0119	0.1669
400	8	0.6021	0.5238	0.2686	0.2865	0.1984	0.0127	0.1067
		0.1484	-0.0309	-0.0000	0.0870	0.2488	0.0103	0.1625
	10	0.5735	0.4905	0.2521	0.2869	0.1942	0.0145	0.1003
		0.1520	-0.0315	0.0001	0.0914	0.2481	0.0121	0.1614
	12	0.5489	0.4653	0.2394	0.2812	0.1890	0.0142	0.0938
		0.1562	-0.0280	0.0021	0.0922	0.2573	0.0098	0.1655

Table 2.1: Estimation of demand function and average partial effects across various network dimensions and sample sizes for $t = 50$ over 500 replications.

As expected, the bias is quite substantial for function estimates without the score correction. This bias does not vanish with larger T. The corrected scores eliminate nearly all the bias, particularly for larger sample sizes. The choice of width d for the neural networks does not

		Function Estimation				Partial Effects		
n	d	RMSE	Bias	% Bias	ISD	RMSE	Bias	SD
100	8	0.5906	0.5008	0.2423	0.3057	0.1969	0.0165	0.1161
		0.1589	-0.0790	-0.0280	0.0906	0.2502	0.0099	0.1729
	10	0.5542	0.4499	0.2182	0.3169	0.1901	0.0175	0.1110
		0.1572	-0.0802	-0.0284	0.0863	0.2510	0.0111	0.1736
	12	0.5042	0.3935	0.1918	0.3083	0.1862	0.0182	0.1047
		0.1609	-0.0812	-0.0287	0.0886	0.2526	0.0110	0.1720
200	8	0.5721	0.4943	0.2556	0.2780	0.1924	0.0120	0.1017
		0.1529	-0.0332	0.0007	0.0872	0.2526	0.0109	0.1641
	10	0.5602	0.4725	0.2447	0.2913	0.1915	0.0127	0.0986
		0.1501	-0.0339	-0.0007	0.0875	0.2455	0.0101	0.1604
	12	0.5179	0.4227	0.2196	0.2901	0.1866	0.0134	0.0931
		0.1494	-0.0362	-0.0014	0.0836	0.2484	0.0098	0.1612
300	8	0.5819	0.5112	0.2603	0.2663	0.2003	0.0143	0.1011
		0.1477	-0.0348	-0.0020	0.0846	0.2494	0.0134	0.1591
	10	0.5325	0.4568	0.2334	0.2625	0.1940	0.0160	0.0952
		0.1482	-0.0350	-0.0022	0.0850	0.2495	0.0124	0.1575
	12	0.5272	0.4198	0.2150	0.3098	0.1899	0.0159	0.0898
		0.1454	-0.0354	-0.0025	0.0834	0.2447	0.0130	0.1572
400	8	0.6109	0.5280	0.2707	0.2974	0.1936	0.0141	0.0990
		0.1468	-0.0304	0.0003	0.0849	0.2431	0.0128	0.1558
	10	0.5734	0.4871	0.2504	0.2927	0.1893	0.0158	0.0937
		0.1459	-0.0308	0.0000	0.0866	0.2411	0.0135	0.1572
	12	0.5184	0.4264	0.2199	0.2859	0.1809	0.0160	0.0872
		0.1507	-0.0275	0.0031	0.0842	0.2522	0.0150	0.1582

Table 2.2: Estimation of demand function and average partial effects across various network dimensions and sample sizes for $t = 100$ over 500 replications.

appear to matter much for the chosen values of $\{8, 10, 12\}$. In terms of the average partial effects the bias is negligible as has been well documented in the literature for fully parametric models (Fernández-Val, 2009). The bias is slightly lower for the corrected estimator at the cost of a variance increase.

2.7 Application: Cigarette Demand

Cigarettes are one of the most heavily taxed products in the United States. The revenue from tobacco taxes averaged 13.8 billion between 2007 and 2017, accounting for 17.4 percent of all excise tax revenue over that period, the third largest source behind highway and aviation[11]. This policy choice has been cited for unintended externalities related to the negative correlation between smoking and income, i.e., cigarette taxes are highly regressive (Harding, Leibtag, and Lovenheim, 2012). A rigorous empirical estimate of this phenomenon should allow for a flexible specification with respect to income and cigarette prices. This is an ideal setting for the panel neural network as the functional form can be left unspecified without the need to impose additive separability. I will focus my analysis on average and conditional average own-price elasticities, where the latter will be averages over total expenditure groups[12]. The focus on own-price elasticity is important as it is the primary component in getting at welfare estimates of these taxes (Hausman and Newey, 2017).

2.7.1 Data

I utilize the Nielsen consumer panel and scanner data spanning a ten-year period from 2007-2017. This source is collected by the Nielsen marketing group and managed by the Kilts Center for Marketing at the University of Chicago[13]. The consumer panel consists of 162,767 unique households over this period. On average a household will stay in the panel for four years. The scanner data comes from approximately 35,000 stores across 55 (Metropolitan

[11] Source: The Office of Management and Budget historical tables

[12] I work with total expenditure as a proxy for income as income is based on coarse survey data and does not vary within a year. Appendix B.2 shows these are highly correlated. In addition, there is the additional benefit of working with revealed rather than stated income levels.

[13] "Researcher(s) own analyses calculated (or derived) based in part on data from The Nielsen Company (US), LLC and marketing databases provided through the Nielsen Datasets at the Kilts Center for Marketing Data Center at The University of Chicago Booth School of Business. The conclusions drawn from the Nielsen data are those of the researcher(s) and do not reflect the views of Nielsen. Nielsen is not responsible for, had no role in, and was not involved in analyzing and preparing the results reported herein."

Statistical Areas) MSAs in the U.S. The combination of scanner and consumer panel data contains over 3.2 million unique UPC codes.

I focus my analysis on a subset of the consumer panel consisting of 'active' panelists who smoke cigarettes. The definition of 'active' is important as it defines which households can be considered for estimation and ultimately defines a sample for which the estimated elasticities are valid. I define 'active' as the set of households that do not fail to report cigarette purchases in consecutive months. Therefore an 'active' household may miss reporting cigarette purchases in June and September of 2007, but not June and July. This definition is justified by the well-known addictive property of cigarettes. It seems extremely unlikely that a smoking household would go two consecutive months without any purchases. Therefore, the lack of purchases likely reflects an omission of reporting rather than the absence of purchases. There are two plausible cases where this may not be true. The first is that households may stockpile cigarettes as is the case for other durable goods (Hendel and Nevo, 2002). However, the patterns do not suggest this behavior in the active cigarette sample[14]. Another possible explanation is that the household may be attempting to or has quit smoking. I posit these households have fundamentally different behavior, and likewise different elasticities, than those without intent to quit and should be omitted from the analysis.

To maximize the number of households available for estimation I focus on a rolling window of panels. Each sub-panel consists of four years, e.g., 2007-2010 and 2008-2011, measured in monthly increments. The number of available users is tabulated in table 2.3. This table reports the full Nielsen sample for these years along with cigarette users[15] and 'active' cigarette users. Unfortunately, the 'active' set is much smaller than the total sample or cigarette sample. In order to check the representativeness of this sample I conduct chi-squared tests across various demographics and report the most relevant in table (2.4). I find these tests fail to reject differences for household income, race, or education levels. The main

[14] A further discussion is in appendix B.3

[15] I limit this to users who purchase cigarettes in at least 50% of months with non-zero expenditure.

differences appear to be in household composition. The 'active' sample tends to have older heads of household with no children or children over eighteen years of age[16]. Tables 2.5 and 2.6 report Kolmogorov-Smirnov tests of empirical distribution equality and various sample moments. Average expenditure across these samples is comparable and the K-S test fails to reject equality. However, the sample mean of average quantity purchased in the 'active' set is considerably larger, closer to one pack per day, than the overall cigarette user set. The other moments are generally different and the K-S test rejects across all panel blocks. If one believes this data is reported accurately then the larger sample average suggests the following analysis is more relevant to heavy, rather than casual, cigarette users.

The greatest challenge with utilizing this data for estimation is the presence of missing purchases and prices, even when aggregated to the monthly level. The lack of reported prices is straightforward to deal with by augmenting the consumer panel data with the scanner data and matching the missing prices at the month, zip code or DMA level. The issue of missing purchases is much more problematic. In a general demand estimation setting one typically considers the 'zero' problem as a censoring issue. There is a myriad of solutions to this problem that can tied back to Heckman (1976) and Heckman (1979), e.g., Lewbel and Pendakur (2009). However, in the case of cigarettes it is not reasonable to believe missing purchases are zero. I conjecture the missing quantities are failures to report rather than

[16] Additional summary statistics and tabulations are available in appendix B.1

Years	Full Sample	Cigarette Sample	Cigarette Sub-Sample
07-10	97,305	9,270	950
08-11	97,919	8,482	873
09-12	94,682	7,544	831
10-13	92,044	6,848	842
11-14	91,008	6,377	902
12-15	90,634	5,986	835
13-16	92,070	5,725	796
14-17	92,756	5,374	725

Table 2.3: Sample sizes of Nielsen and cigarette user subpanels for each rolling window.

zero. However, this assumption does not lend itself well to a selection model as it is unclear what the mechanism behind failure to report is. Instead I make a stronger assumption and follow Chernozhukov, Hausman, and Newey (2019) in assuming these purchases are missing at random. Under this assumption the estimates are unaffected by various lengths of T_i. This assumption is not directly testable, but one can attempt to falsify this claim through various regressions of missingness on observed covariates. Overall, there are no demographic observables that have significant correlation with missingness across all panel block, a further discussion is available in appendix B.3.

Household Income

Panel Years	χ^2	p-value
07-10	4.34	0.36
08-11	3.17	0.53
09-12	2.48	0.65
10-13	2.77	0.60
11-14	8.29	0.08
12-15	6.54	0.16
13-16	6.73	0.15
14-17	4.22	0.38

Head of Household Race

Panel Years	χ^2	p-value
07-10	7.78	0.05
08-11	3.87	0.28
09-12	9.40	0.02
10-13	4.56	0.21
11-14	4.90	0.18
12-15	3.13	0.37
13-16	7.43	0.06
14-17	4.06	0.26

Highest Education Level

Panel Years	χ^2	p-value
07-10	6.75	0.24
08-11	2.34	0.80
09-12	3.59	0.61
10-13	4.94	0.42
11-14	3.87	0.57
12-15	4.87	0.43
13-16	0.25	1.00
14-17	2.81	0.73

Household Composition

Panel Years	χ^2	p-value
07-10	37.44	0.00
08-11	23.45	0.00
09-12	25.67	0.00
10-13	26.39	0.00
11-14	23.37	0.00
12-15	25.77	0.00
13-16	19.07	0.00
14-17	17.08	0.01

Table 2.4: χ^2 tests of independence for various household characteristics across the sample of cigarette users and the ‘active’ subsample.

Panel Years	K-S Test		Mean	StD	Skew	Kurtosis
07-10	0.2761 (0.0)	Cigarettes	25.47	22.10	2.05	7.06
		Active	37.60	25.01	1.73	5.71
08-11	0.2619 (0.0)	Cigarettes	25.06	23.32	3.40	28.77
		Active	36.29	23.67	1.47	3.94
09-12	0.2468 (0.0)	Cigarettes	24.28	23.21	3.39	25.22
		Active	35.48	26.20	2.71	15.65
10-13	0.2571 (0.0)	Cigarettes	23.76	24.18	4.05	34.28
		Active	34.59	25.44	2.70	17.56
11-14	0.2630 (0.0)	Cigarettes	23.20	24.34	4.28	38.16
		Active	35.06	28.73	3.57	25.63
12-15	0.2853 (0.0)	Cigarettes	22.04	23.35	3.76	26.29
		Active	34.81	28.52	3.41	20.91
13-16	0.3060 (0.0)	Cigarettes	21.43	23.75	4.96	52.82
		Active	35.14	28.76	3.97	34.76
14-17	0.3115 (0.0)	Cigarettes	20.79	22.51	4.36	39.42
		Active	34.85	29.85	4.34	35.32

Table 2.5: Comparing observed household average monthly purchases of cigarette packs across samples. I report Kolmogorov-Smirnov (K-S) tests and various sample moments for the set of cigarette users and the 'active' set across panel year blocks.

Panel Years	K-S Test		Mean	StD	Skew	Kurtosis
07-10	0.0438 (1.0)	Cigarettes	10193.17	9786.03	3.21	20.01
		Active	9375.24	9043.62	3.53	23.42
08-11	0.0378 (1.0)	Cigarettes	10438.97	10126.01	3.25	20.01
		Active	10164.67	10643.38	3.47	19.42
09-12	0.0347 (1.0)	Cigarettes	10760.56	10610.79	3.29	19.63
		Active	10484.32	11010.05	3.53	19.69
10-13	0.0235 (1.0)	Cigarettes	11029.55	10831.04	3.28	19.54
		Active	11121.98	11986.56	3.72	20.41
11-14	0.0338 (1.0)	Cigarettes	11319.04	11408.10	3.74	27.54
		Active	11995.27	12388.61	3.75	23.79
12-15	0.0210 (1.0)	Cigarettes	11612.64	12110.95	4.55	47.37
		Active	11876.87	12792.24	4.54	34.67
13-16	0.0210 (0.92)	Cigarettes	11677.12	12656.17	5.92	85.39
		Active	11674.30	12737.35	4.77	41.28
14-17	0.0240 (0.86)	Cigarettes	11525.89	13077.52	9.59	232.15
		Active	11996.69	19899.51	13.94	271.27

Table 2.6: Comparing observed household average monthly expenditure across samples. I report Kolmogorov-Smirnov (K-S) tests and various sample moments for the set of cigarette users and the 'active' set across panel year blocks.

2.7.2 Demand Estimation

One has many choices when estimating demand as the econometric community has been investigating this problem for many decades. In more recent years there has been a push to estimating flexible reduced form models of quantity on prices, total expenditure, and other, potentially high-dimensional, controls, e.g., Chernozhukov, Goldman, Semenova, and Taddy (2017) and Bajari, Nekipelov, Ryan, and Yang (2015). I follow this strand of the literature and focus on estimating demand and own-price elasticity for cigarettes where the demand is modeled:

$$\ln q_{it} = \alpha_i + g(p_{it}, y_{it}) + \epsilon_{it} \tag{2.33}$$

where p_{it} and y_{it} are the price of a pack of cigarettes and total expenditure for household i at time t. It is likely the case that prices and total expenditure are endogenous and must be instrumented for. I utilize a control function approach and compare the neural network estimator to a series estimator as well as a fully linear specification.

2.7.3 Control Functions

The most common approach to dealing with endogeneity in the nonparametric literature is through control functions Newey and Powell (2003). This approach is equivalent to instrumental variables estimation under linearity, but will also hold in more general settings.[17]. The instruments for cigarette prices and total expenditure are their own lags[18]. The control

[17] See Blundell and Matzkin (2010) for conditions on the existence of control functions in nonseparable models

[18] I use four month lags as in Chernozhukov, Hausman, and Newey (2019)

function for prices solve the first stage problem:

$$\hat{h}_m = \sup_{h_m \in \mathcal{H}_m} -\frac{1}{n}\sum_{i=1}^{n}\frac{1}{T_i}\sum_{t=1}^{T_i}\left(p_{it} - h_m\left(p_{i,t-m}, X_{it}\right)\right)^2 \tag{2.34}$$

$$\nu_{it} = p_{it} - \hat{h}_m\left(p_{i,t-m}, X_{it}\right) \tag{2.35}$$

The control function for expenditure is the same, but replaces $p_{i,t-m}$ with $y_{i,t-m}$. The vector X_{it} are the other included exogeneous covariates. The residuals ν_{it} from these regressions are included in the second stage and account for the potential endogeneity of cigarette prices.

2.7.4 Results

All specifications include cigarette prices, total expenditure, and cross-sectional fixed effects. The models are specified as in equation 2.33 where g is specified in the following three ways:

$$g_1(p_{it}, y_{it}) = \beta \ln p_{it} + \lambda \ln y_{it} + \nu_{it}'\gamma \tag{2.36}$$

$$g_2(p_{it}, y_{it}) = \sum_{k=1}^{K}\beta_k \ln p_{it}^k + \sum_{k=1}^{K}\lambda_k \ln y_{it}^k + \nu_{it}'\gamma \tag{2.37}$$

$$g_3(p_{it}, y_{it}) = g_{nn}(p_{it}, \ln y_{it}) + \nu_{it}'\gamma \tag{2.38}$$

where the elements of γ are set to zero when the respective control function(s) is(are) not included[19]. The choice of $\hat{h}_m$ in 2.34 is taken to be linear such that the first stage is completely parametric. The neural network includes prices in levels and log total expenditure as inputs. In addition, the estimates for the neural network models are the median solutions over 50 randomly initialized models to avoid problems associated with poor optimization.

[19] The control function enters additively as in Newey, Powell, and Vella (1999), Newey and Powell (2003).

Cigarette Own Price Elasticity

07-10	NN	-0.7779	-0.7981	-0.7005	-0.6566
	Series	-0.6247	-0.6216	-0.6242	-0.6217
		(0.066)	(0.0706)	(0.0658)	(0.0706)
	Linear	-0.519	-0.5527	-0.518	-0.5526
		(0.0401)	(0.0505)	(0.0402)	(0.0505)
08-11	NN	-0.7414	-0.7114	-0.6204	-0.4132
	Series	-0.5252	-0.5148	-0.5253	-0.5145
		(0.0781)	(0.0827)	(0.078)	(0.0827)
	Linear	-0.384	-0.4155	-0.3836	-0.4154
		(0.0448)	(0.0511)	(0.0449)	(0.0511)
09-12	NN	-0.9088	-1.2011	-0.5881	-0.4700
	Series	-0.653	-0.7038	-0.6525	-0.7047
		(0.1148)	(0.1318)	(0.1147)	(0.1313)
	Linear	-0.3917	-0.4682	-0.3921	-0.4681
		(0.0729)	(0.0938)	(0.0725)	(0.0946)
10-13	NN	-0.9213	-0.9108	-0.4872	-0.5500
	Series	-0.5854	-0.7792	-0.5834	-0.7772
		(0.1171)	(0.1508)	(0.1174)	(0.1499)
	Linear	-0.4488	-0.6215	-0.4474	-0.6198
		(0.0624)	(0.0894)	(0.0627)	(0.0886)
	Ctrl. Prices	No	Yes	No	Yes
	Ctrl. Expend	No	No	Yes	Yes

Table 2.7: Own-price elasticities for the three specifications. The neural network results are medians associated with fifty randomly initialized networks.

Cigarette Own Price Elasticity

11-14	NN	-0.8146	-0.9094	-0.6621	-0.4320
	Series	-0.5793	-0.7349	-0.5828	-0.7353
		(0.1031)	(0.1447)	(0.1029)	(0.1443)
	Linear	-0.4894	-0.6815	-0.4915	-0.6818
		(0.0609)	(0.0645)	(0.061)	(0.0643)
12-15	NN	-0.8640	-0.8289	-0.5685	-0.6206
	Series	-0.5776	-0.6075	-0.5785	-0.6079
		(0.0969)	(0.1104)	(0.0966)	(0.1083)
	Linear	-0.4852	-0.654	-0.4856	-0.6541
		(0.057)	(0.0892)	(0.0573)	(0.09)
13-16	NN	-0.6042	-0.8934	-0.3827	-0.4385
	Series	-0.4387	-0.4842	-0.4389	-0.4814
		(0.0988)	(0.115)	(0.099)	(0.1129)
	Linear	-0.3732	-0.508	-0.3754	-0.5097
		(0.0631)	(0.1087)	(0.063)	(0.1079)
14-17	NN	-0.6067	-0.5285	-0.4910	-0.3319
	Series	-0.3926	-0.5038	-0.3885	-0.5407
		(0.0924)	(0.1232)	(0.0918)	(0.1125)
	Linear	-0.3178	-0.4565	-0.3184	-0.4909
		(0.0557)	(0.0924)	(0.0552)	(0.076)
	Ctrl. Prices	No	Yes	No	Yes
	Ctrl. Expend	No	No	Yes	Yes

Table 2.8: Own-price elasticities for the three specifications. The neural network results are medians associated with fifty randomly initialized networks.

The elasticities across all estimators are similar in magnitude for all panel blocks. The direction of the endogeneity bias, for the neural network estimator, is in line with the literature which typically suggests prices are negatively correlated with preferences. Controlling for the price endogeneity reduces the elasticity magnitudes in the neural network estimator across all year blocks. In addition, there is a noticeable downward, towards zero, trend in elasticity estimates over time. This phenomenon is consistent with the sharp increase in prices observed in the first panel block which stabilizes over time as seen in 2.2. This is the case for both the level prices and the relative prices.[20]. I conjecture that the federal tax hike in 2009

[20] I construct the relative price as the price of a pack of cigarettes divided by a weighted average of all

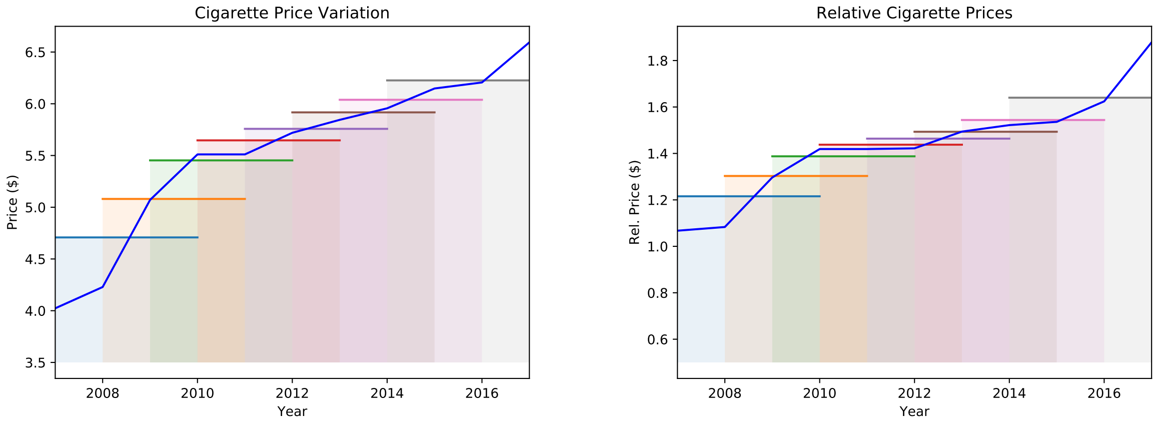


Figure 2.2: Averages for annual cigarette price time series compared with averages in each panel block. The figure on the left depicts prices in levels while the right depicts relative prices.

contributed to a temporary increase in own price elasticities that returned to more 'normal' levels as households adjusted to the new, higher, prices.

To assess the sensitivity of elasticities to various income levels I consider conditional average elasticities grouped by total expenditure quartiles. These results are reported in table 2.9 and are in line with the literature on the regressive nature of cigarette taxes. Higher total expenditure households in the sample are much less sensitive to price changes relative to the lower expenditure groups. These elasticities decrease in overall magnitude over time, but the strict ordering across expenditure groups remains.

	$[q_{.00}, q_{.25})$	$[q_{.25}, q_{.50})$	$[q_{.50}, q_{.75})$	$[q_{.75}, q_{1.00}]$
07-10	-0.7393	-0.6828	-0.6370	-0.5699
08-11	-0.4642	-0.4285	-0.3998	-0.3615
09-12	-0.5093	-0.4813	-0.4670	-0.4225
10-13	-0.6196	-0.5828	-0.5685	-0.4307
11-14	-0.4678	-0.4509	-0.4373	-0.3723
12-15	-0.6934	-0.6325	-0.6194	-0.5359
13-16	-0.4799	-0.4287	-0.4353	-0.4099
14-17	-0.3700	-0.3451	-0.3315	-0.2851

Table 2.9: Conditional average elasticities over total expenditure quantiles.

other prices in the scanner data.

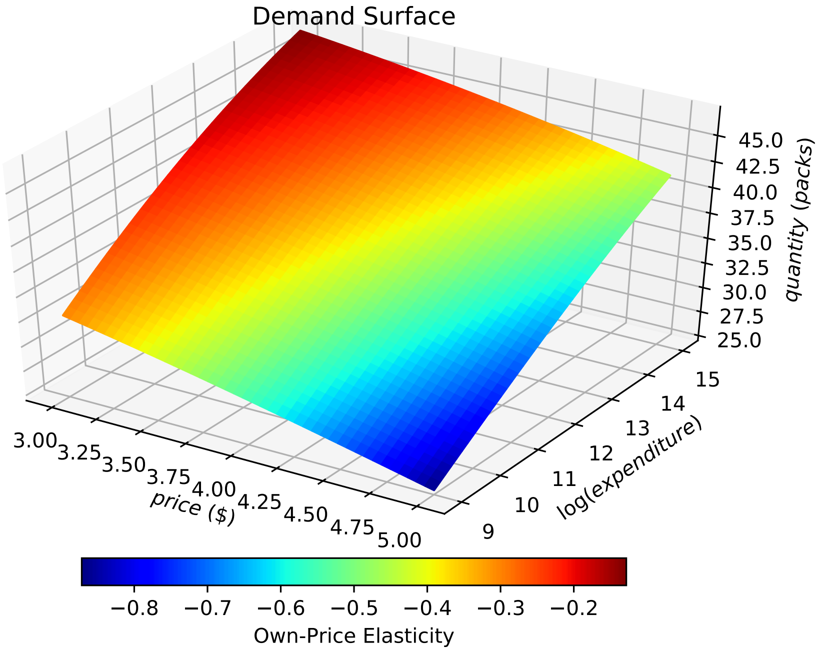


Figure 2.3: Demand surface estimated by the panel neural network for the first panel block 2007-2010 over a grid of log expenditure and prices.

2.8 Conclusion and Future Work

I provide a new estimator for panel data utilizing a neural network that is consistent with a common index and heterogeneity in the form of cross-sectional fixed effects. I provide corrections for the incidental parameter bias induced by the presence of the fixed effects and show its effectiveness in finite samples. In an application I examine demand for cigarettes in the United States using the Nielsen consumer panel data. I empirically verify the regressive nature of cigarette taxes through a fully flexible specification with respect to prices and total expenditure.

There are several clear extensions to the theoretical analysis in this chapter. The consistency result does not take advantage the independence between cross-sections. The conditions on the parameter magnitude can likely be weakened by utilizing a tighter bound on the large deviation inequality. It is also possible to extend the results from the previous chapter to be used here albeit with additional bias corrections and the inclusion of only one-way fixed effects could be extended to two-way as in Fernández-Val and Weidner (2018) with some additional modifications.

Chapter 3

Informative Dimensionality Reduction Using a Deep Autoencoder

This chapter investigates the potential of a deep neural network to perform informative non-linear dimensionality reduction of high-dimensional binary data. I show that the autoencoder, a type of neural network, learns a representation of the data on a low-dimensional manifold while simultaneously learning an inverse transformation back to the original space. This dual formulation, learning an encoding and decoding, distinguishes the autoencoder from other non-linear dimensionality reduction methods as it allows the researcher to take any point on the low-dimensional surface and map it back to the original space. I explore the performance of the deep autoencoder in a series of simulation experiments to recover the data generating process and learn the number of clusters from the data. I then apply the deep autoencoder to analysis consumer preference clusters in a dataset of purchase receipts.

3.1 Introduction

This chapter introduces a new non-linear dimensionality reduction technique with the dual aims of outperforming existing linear and nonlinear methods while aiding the interpretability of the resulting low-dimensional embedding. There are many dimensionality reduction techniques to choose from. However, the majority fail to provide an informative low-dimensional embedding of the original high-dimensional features. This deficiency means that researchers often struggle to interpret what can be learned from the low-dimensional representation. I demonstrate that the use of a specific neural network, an autoencoder, holds great promise in this area. The autoencoder can uncover latent features of the original data more effectively than other more traditional approaches. In addition, each point in the latent space can be characterized in terms of a probability distribution over the original high-dimensional input space. This can be used to improve our understanding of heterogeneous consumer preferences by inferring the latent structure from observed purchase behavior.

The availability of Big Data generates new and unprecedented challenges, for a recent review see Fan, Han, and Liu (2014). An increasingly common problem is how to take a "first look" at prohibitively large and previously unseen datasets. This first step at examining the data occurs in exploratory settings, but also when a research question is firmly established. In either case, the high-dimensional nature of the data makes traditional visualization impossible and summary statistics become overwhelming. In an exploratory setting the problem is further confounded as it is unclear how one might develop hypotheses from underlying patterns and trends. If left unaided by statistical tools and machine learning algorithms, humans have difficulty comprehending high-dimensional data and often reach misleading conclusions when attempting to do so. However, the curse of dimensionality prevents clustering algorithms like k-means or gaussian mixture models (GMM) from providing meaningful results.

An important strategy for engaging with high-dimensional datasets is to reduce their dimension by projecting the data onto a lower dimensional space (Carreira-Perpinán, 1997). Dimensionality reduction is a tool to convert high-dimensional data into lower dimensions, while preserving intrinsic properties of the data, e.g., connectivity and continuity. Techniques of dimensionality reduction can be grouped into two categories, linear and nonlinear methods. Linear dimensionality reduction methods, as the name implies, involve only linear transformations in the conversion; among them, the best-known are PCA (Pearson (1901), Hotelling (1933), Jolliffe (2002) and Independent Component Analysis (ICA) (Hyvärinen and Oja, 2000). In contrast, non-linear dimensionality reduction relaxes the linearity of the transformation. The objective becomes finding a low-dimensional manifold that the original high-dimensional data resides on. Over the last twenty years, many non-linear dimensionality reduction methods have been developed, including Isomap (Tenenbaum, De Silva, and Langford, 2000) and Locally Linear Embedding (LLE) (Roweis and Saul, 2000). Non-linear dimensionality reduction methods generally outperform linear methods when applied to complex, real-world datasets. This paper introduces a neural network strategy for performing non-linear dimensionality reduction. In a neural network the input data is transformed repeatedly through several layers and a non-linear function is applied at each layer of the data.

After the data have been mapped onto a low-dimensional space, one can readily perform other tasks on the data, e.g., visualization, clustering, or classification. These tasks are important for exploring the structure of the data and for developing new hypotheses. For example, a common challenge in economics and marketing involves "customer segmentation", clustering consumers into distinct groups using a variety of socio-demographic and transactional information on them. These customer segments are widely used in determining a range of economic decisions from marketing expenditures to direct mail targeting.

Once the data has been projected onto a lower dimensional space another crucial challenge emerges. If the latent space uncovered by the dimensionality reduction technique is not interpretable then tools like clustering provide little meaning. When using linear methods, the outcomes can be interpreted as linear combinations[1] of the original variables. However, once linearity is relaxed this interpretation no longer holds. Furthermore, non-linear methods often outperform linear methods. Therefore, one is typically forced into a trade-off between performance and interpretability.

The autoencoder provides a solution to this lack of interpretability by explicitly modeling the inverse transformation between the lower dimensional space and the original data space. This mapping provides a connection between any point, including out-of-sample, in the embedding space and the original input space. This kind of inverse-mapping information is extremely valuable to social scientists. Patterns in the embedding space should be comprehensible; for example, if the original feature space represents customer features, differences between customers in different segments are relevant.

In this chapter I provide the first econometric use of the deep autoencoder neural network (Hinton and Salakhutdinov, 2006) architecture as a non-linear dimensionality reduction method for high-dimensional binary input data. It is a neural network with an input layer, an output layer and one or more hidden layers. The network is trained with conventional backpropagation but paired with a layer-by-layer pre-training procedure. The optimization objective is to minimize the difference between the input and the output, and therefore the output is often seen as a reconstruction of the input data. While projecting the data to a low dimensional manifold, it can learn an inverse transformation which allows us to interpret every point in the embedding space in terms of a probability distribution over the original feature space of the data. The reconstruction functionality of the deep autoencoder is useful for analyzing the embedded data. Because it is trained with the entire dataset, the reconstruction recovers the underlying data generating model.

[1] It is often not obvious how to interpret the results of PCA in an insightful fashion, something that practitioners are often painfully aware of.

The remaining of the chapter proceeds as follows. The first section (3.2) presents the autoencoder and its construction. Section 3.3 discusses an interpretation of the low dimensional representation generated by the model. I discuss some computational difficulties in 3.4 and assess the model performance in 3.5. I present an application of the autoencoder to consumer segmentation in 3.6 and conclude in 3.7.

3.2 Model

I consider n iid realizations from the binary random vector $y_i \in \left\{\{0,1\}^J\right\}_{i=1}^n$. The observable outcomes y_i are governed by a latent process:

$$y_i = \begin{cases} 1 & g_0(F_i) + e_i > 0 \\ 0 & g_0(F_i) + e_i \leq 0 \end{cases} \tag{3.1}$$

where $F_i \in \mathbb{R}^d$ is a latent vector associated with y_i. The autoencoder seeks construction of the dual maps $F_i : \{0,1\}^J \to \mathbb{R}^d$ and $g_0 : \mathbb{R}^d \to [0,1]^J$. The autoencoder achieves this by finding a function that maps y_i onto itself. In most commonly used autoencoders, the dimension of the model is restricted, e.g. $d \ll J$, “encoding” the input into F_i and then “reconstructing” back to $[0,1]^J$. This approach is typically referred to as a contractive[2] autoencoder as the width of each successive layer is reduced until the encoding portion is complete. The heuristic for why this works lies in the so-called “information bottleneck”, the important characteristics of y_i are preserved while eliminating superfluous artifacts or noise.

Autoencoders and their deep counterparts are trained to reconstruct the input data at the top layer, after a series of non-linear transformations. Therefore, the optimization objective

[2] There are alternative specifications of autoencoders, e.g., overcomplete or sparse, which do not follow this pattern.

is to minimize the error between the input and the output. Let the number of total layers in the network be $2k+1$ with corresponding width d_k. The middle layer represents a projection of the input data onto a lower-dimensional embedding space. The sample objective is:

$$\mathcal{L}_n(p, y) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(p_i, y_i) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{d} y_{ij}\ln(p_{ij}) + (1 - y_{ij})\ln(1 - p_{ij}) \tag{3.2}$$

where p_{ij} is parameterized by the autoencoder. For clarity it is useful to first define the "decoder" and "encoder" portions of the network separately. Consider the case where each object has two hidden layers. The encoder and decoder are defined respectively:

$$F_i(y_i) = \sum_{j=1}^{d} s\left(\sum_{k=1}^{K} s(\tilde{y}_i'\beta_k)\gamma_{kj}\right)\omega_d \tag{3.3}$$

$$g_n(F_i) = \Lambda\left(\sum_{k=1}^{K} s\left(\sum_{j=1}^{d} s(\tilde{F}_i'\omega_j)\gamma_{jk}\right)\beta_k\right) \tag{3.4}$$

where Λ is the logistic function and $s(\cdot)$ is a suitable 'activation' function. Putting these together the autoencoder objective becomes:

$$\hat{g}_n, \hat{F}_i \in \underset{g_n \in \mathcal{G}_n, F_i \in \mathcal{F}_n}{\arg\max} \sum_{i=1}^{n}\sum_{j=1}^{d} y_{ij}\ln g_n\left(F_i\right) + (1 - y_{ij})\ln\left(1 - g_n\left(F_i\right)\right) \tag{3.5}$$

I shall refer to the coordinates F_i as factors and suppress dependence on y_i for convenience.

3.3 Embedding Space

The ability to interpret autoencoders comes primarily from the construction of F_i. This vector, often referred to as the 'embedding space', projects a nonlinear transformation of the input data onto a lower-dimensional space $\mathbb{R}^d$. In this exposition I consider an application where the input data is household grocery-purchase records over a certain period. A household's monthly purchases are organized into a binary vector, with each element indicating whether a certain type of product has been bought or not. If one restricts d to be two or three, then one can visualize the households in a two or three-dimensional space. This representation has the potential to uncover clusters of 'like" consumers or notice temporal changes in some household's purchases over the months. One potential goal is to understand what sort of shopping behaviors those patterns reflect.

Among non-linear dimensionality reduction methods, autoencoders are unique in that they provide, by construction, an "inverse-mapping". One has for any $F_i \in \mathcal{F}_i$ a reconstruction of the input. This reconstruction produces the set of potential outcomes $g_0(F_i)$, e.g., how would the probabilities of purchasing good j change if F_i is perturbed by a random vector $\epsilon_i \in \mathbb{R}^d$.

3.4 Computational Considerations

Training autoencoders, particularly large ones, is a highly non-convex optimization problem. One way of avoiding poor local minima is to initialize the weights of the network to be within some neighborhood of an optimal minima. To address this issue, Hinton and Salakhutdinov (2006) propose the use of an undirected graphical model in the form of a Restricted Boltzmann Machine (RBM) to initialize the weights.

To understand why this works, it is informative to first discuss the properties of undirected graphical models or Markov Random Fields (MRF). Firstly, the probability distribution $\Pr(x)$ of any MRF can be characterized by the Boltzmann distribution: $\Pr(x) \propto \exp\{-\varepsilon(x)\}$ where $\varepsilon(x)$ is referred to as the energy function containing connectivity information about the graph. In the case of the RBM one decomposes the observed outcomes y_i into latent or hidden units h and visible units v. This structure allows for complex distributions over the observables while maintaining mutual independence of the visible units conditional on h.

Consider a RBM with m visible nodes and k hidden nodes, all of which are stochastic, binary units. Then $v = \{(v_j)_{j=1}^m : v_j \in \{0,1\} \; \forall j\}$ and $h = \{(h_l)_{l=1}^k : h_l \in \{0,1\} \; \forall l\}$. Let edges between h_l and v_j for any pair (l, j) have a corresponding weight $w_{lj} \in \mathbb{R}$ as well as intercept terms $b_j \in \mathbb{R}$ and $c_l \in \mathbb{R}$ associated with each visible and hidden unit respectively. Then the energy function is:

$$-\varepsilon(v,h) = \sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij}h_i v_j - \sum_{j=1}^{m} b_j v_j - \sum_{i=1}^{n} c_i h_i \tag{3.6}$$

In general, for any MRF with known graph structure and parametric family, one can estimate the parameters of the energy function through maximum likelihood:

$$\ln p(v,\theta) = \ln \sum_{h} \exp\{-\varepsilon(v,h)\} - \ln \sum_{v,h} \exp\{-\varepsilon(v,h)\} \tag{3.7}$$

Plugging in the energy function for the RBM one can write down the gradients with respect to each parameter w_{ij}, b_j, and c_i:

$$\frac{\partial \ln p(\theta, v)}{\partial w_{ij}} = p(h_i = 1|v)v_j - \sum_{v} p(v)p(h_i = 1|v)v_j \tag{3.8}$$

$$\frac{\partial \ln p(\theta, v)}{\partial b_j} = v_j - \sum_{v} p(v)v_j \tag{3.9}$$

$$\frac{\partial \ln p(\theta, v)}{\partial b_j} = p(h_i = 1|v) - \sum_{v} p(v)p(h_i = 1|v) \tag{3.10}$$

Furthermore, under further parametric assumptions the conditional distributions of v and h

are symmetric and equal to:

$$\Pr(v_j = 1|h) = \Lambda(b_j + \sum_i w_{ij} h_i) \tag{3.11}$$

$$\Pr(h_i = 1|v) = \Lambda(c_i + \sum_j w_{ij} v_j) \tag{3.12}$$

where $\Lambda(\cdot)$ is the logistic function. However, the second term in each of the gradients is not analytically available. One can approximate the expectation by sampling from the model distribution. In practice, the sampling procedure is often approximated with a single Gibbs step. The so called contrastive-divergence algorithm (Hinton, 2002), works remarkably well in practice.

The advantage of training the RBM and its application to deep Neural Networks lies in the ability to vertically stack them. The general idea is that once the parameters of the RBM are found one can treat the hidden units as visible and train a new RBM in the same way as the original one. Stacking the RBMs in this way can then be viewed as training a stochastic Neural Network.

The full procedure can then be characterized in two stages of training. The pre-training stage consists of training and stacking RBMs with layer dimension corresponding to the desired neural network structure. At the fine-tuning stage, the autoencoder is trained with the standard backpropagation algorithm, as described in the last section. However, the parameters are set to the RBM parameters learned at the pre-training stage.

It is important to note that for the final embedding layer the autoencoder is linear. Here the pre-training stage utilizes a Gaussian energy function:

$$-\varepsilon(v, h) = \sum_{i=1}^{n} \sum_{j=1}^{m} w_{ij} h_i \frac{v_j}{\sigma_j^2} - \sum_{j=1}^{m} \frac{(v_j - b_j)^2}{2\sigma_j^2} + \sum_{j=1}^{n} c_i h_i \tag{3.13}$$

where σ_i^2 is the variance of Gaussian noise applied to visible unit j.

3.5 Monte Carlo

In these simulations I demonstrate the potential of autoencoders as informative dimensionality reduction tools in high-dimensional discrete choice data. In the first set I focus on recovery of the latent signals p_i and in the latter the ability to cluster patterns of choice behavior.

Consider the following data generating process for choice behavior x_i:

$$\{y_{ij}\}_{i=1}^{n} \sim B(n, p_{jl}),\ j = 1, 2, \ldots, J \tag{3.14}$$

$$p_{lj} \sim Claw(M_l) = \left(1 + \exp\left\{ - \left[\frac{1}{2}\phi(z) + \sum_{j=1}^{5} \frac{1}{10}\phi\left(\frac{z + m_{jl}}{0.1}\right)\right]\right\}\right)^{-1} \tag{3.15}$$

Each individual purchases $\sum_j x_{ij}$ goods from a fixed number of choices J. To simulate heterogeneity, I randomly assign the simulated consumers to a preference group l. Within a group the purchase probability p_{jl} is fixed but varies across groups. The probabilities are drawn from scaled claw functions, Marron and Wand (1992) with varying modes: $M_l = \{m_{1,l},\ m_{2,l},\ \ldots, m_{5,l}\}$.

I examine the case where the number of partitions $l = 4$ for various modes. The dataset is used to train a deep autoencoder with each hidden layer having 100, 50, 2, 50, 100 nodes. Figure 3.1 shows the true distribution is recovered with a high degree of accuracy.

One may also want to examine whether the embedding space separates individuals with different underlying distributions. Linear methods like PCA work well when individuals have either homogeneous or sufficiently heterogeneous distributions. I consider the case where the underlying probability of individual i purchasing good j is very similar to the probability of individual r purchasing the same good. A visualization of such a case is shown in figure 3.2. The deep Autoencoder can disentangle the choice probabilities even when underlying distributions have a high degree of overlap. For the following simulations, each partition is governed by an underlying DGP such that:

$$\{y_{ij}\}_{i=1}^{n} \sim B(n, p_{jl}), j = 1, 2, \ldots, J \tag{3.16}$$

$$p_{jl} \sim Beta(\alpha_l, \beta_l) \tag{3.17}$$

I consider four cases, all of which follow the same underlying distribution, but differ in parameters α and β as well as the number of partitions l. To gauge relative performance, I compare the embedding space of the Autoencoder to PCA and Isomap.

For the first simulation consider $l = 8$ where:

$$(\alpha_1, \beta_1) = \Big\{(6,6), (6,5), (5,6), (5,5), (5,4), (4,5), (2,3), (3,2)\Big\} \tag{3.18}$$

It is clear from figure 3.3 that nonlinear methods do a better job at disentangling the underlying distributions. PCA struggles to separate the six heavily overlapping distributions. Isomap does reasonably well, but I observe much more well-defined clusters with the Autoencoder.

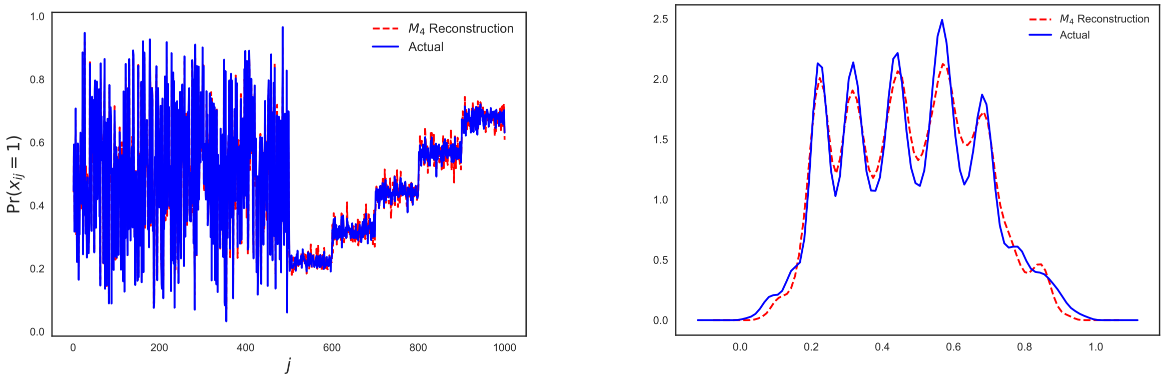


Figure 3.1: Recovery of the underlying probability distribution. Results are from the centroid corresponding to the standard claw parameterization with location parameters M_4.

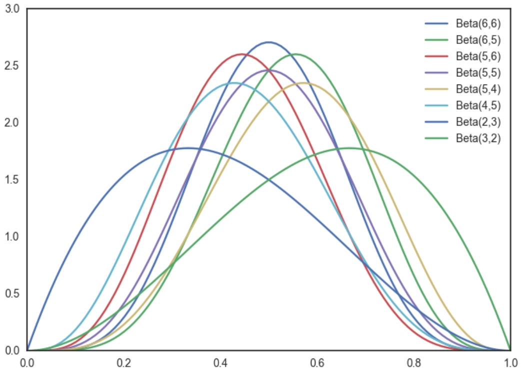


Figure 3.2: PDF of the Beta distributions in the first simulation. Most parameterizations overlap heavily.

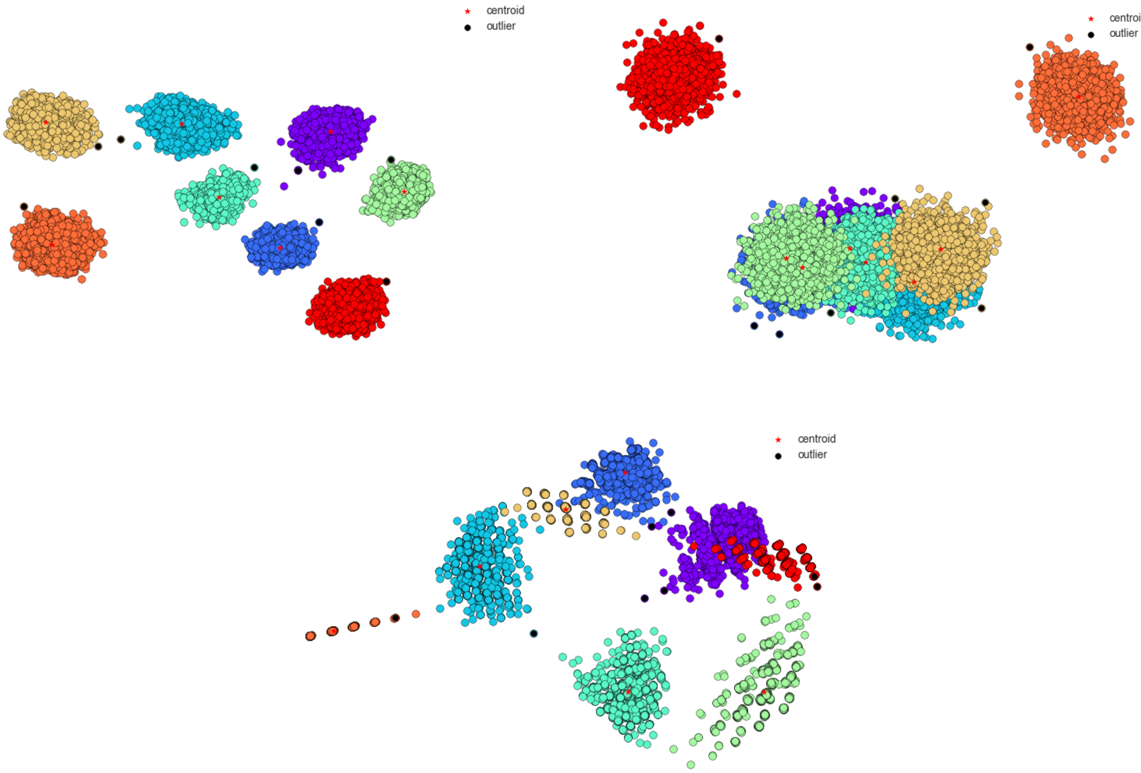


Figure 3.3: Embedding space from Autoencoder, PCA, and Isomap from left to right respectively. Figures are from the first simulation ($n = 20,000$) and colors respond to true labels.

I further consider the case where $l = 10$ and $l = 20$ such that:

$$\{\alpha_2, \beta_2\} = \{\{\alpha_1, \beta_1\}, (4,3), (3,4)\} \tag{3.19}$$

$$\{\alpha_3, \beta_3\} = \{\{\alpha_2, \beta_2\}, \{\alpha_2 + \frac{1}{2}, \beta_2 + \frac{1}{2}\}\} \tag{3.20}$$

The last case reflects a common scenario observed in real shopping data. One often observes the number of items available to be very large, but the actual number of purchases for any given shopping trip is a small subset. To reflect this scenario consider $l = 10$, $\alpha_4 = 1$, and $\beta_4 = \{30, 29, \ldots, 22\}$.

Table(3.1) displays metrics evaluating clustering in the embedding space. I use two measures of cluster separation, the Davies-Bouldin (DB) index and Silhouette. The former measures an average of similarity measures between clusters, while the later measures pairwise distances between and within clusters. Formally DB and Silhouette are respectively defined:

$$DB(k) = \frac{1}{K}\sum_{i=1}^{K} \max_{i \neq j} \left\{ \frac{\Delta(c_i) + \Delta(c_j)}{\delta(c_i, c_j)} \right\} \tag{3.21}$$

$$S(k) = \frac{1}{K}\sum_{i=1}^{K} \sum_{j \in k} \left(\frac{b_j - a_j}{\max\{a_j, b_j\}} \right) \tag{3.22}$$

where $\Delta(c_i)$ is the intra-cluster distance for i and $\delta(c_i, c_j)$ is the inter-cluster distance between i and j. Further, a_j is the average distance within a cluster and b_j is the average distance between each element in the cluster and elements in the closest neighboring cluster. A smaller value of DB indicates less similarity between clusters and a Silhouette value closer to one indicates separation.

In addition to cluster tightness and separation one may want to see how well each method is discriminating between the partitions. I report a measure of classification error: $1 - \frac{1}{n}\sum_k c_k$, where c_k is the number of individuals within a cluster that belong to the majority partition.

From the results one observes Isomap and the Autoencoder do well in classification error and cluster tightness, with a slight edge given to the deep Autoencoder when n is small and the data is not sparse. However, as the sample size increases the deep Autoencoder outshines Isomap in all four simulations across metrics. I further note, in the presence of sparsity, Isomap performs very poorly regardless of the sample size.

			$(\alpha_1,\ \beta_1)$			(α_2,β_2)	
n	Metric	AE	PCA	Isomap	AE	PCA	Isomap
$1,000$	Davies–Bouldin	0.3749	1.1418	0.8535	1.0309	1.1689	0.8682
	Silhouette	0.7196	0.3399	0.6885	0.3639	0.2467	0.3220
	Classification Error	0.0070	0.2640	0.0440	0.2460	0.4170	0.1830
$5,000$	Davies–Bouldin	0.3050	1.2196	0.2719	0.8852	1.2093	1.1879
	Silhouette	0.7732	0.3582	0.8126	0.5032	0.2524	0.3694
	Classification Error	0.0000	0.2928	0.0008	0.1758	0.4052	0.2028
$10,000$	Davies–Bouldin	0.3264	1.2155	0.2755	0.5631	3.5007	1.3308
	Silhouette	0.7586	0.4307	0.8059	0.6233	0.2247	0.3402
	Classification Error	0.0001	0.2398	0.0013	0.0302	0.3998	0.3173
$20,000$	Davies–Bouldin	0.2970	1.1418	1.1071	0.4795	1.3397	0.9942
	Silhouette	0.7804	0.4648	0.5120	0.6588	0.3314	0.4415
	Classification Error	0.0000	0.2685	0.1775	0.0306	0.4985	0.1320

			(α_3,β_3)			$(\alpha_4,\ \beta_4)$	
n	Metric	AE	PCA	Isomap	AE	PCA	Isomap
$1,000$	Davies–Bouldin	1.0309	1.1689	0.8682	0.8483	3.7612	0.7407
	Silhouette	0.3639	0.2467	0.3220	0.3698	0.1373	0.4514
	Classification Error	0.2460	0.4170	0.1830	0.1550	0.4850	0.7290
$5,000$	Davies–Bouldin	0.8852	1.2093	1.1879	0.5802	2.3971	0.7849
	Silhouette	0.5032	0.2524	0.3694	0.5308	0.2968	0.3552
	Classification Error	0.1758	0.4052	0.2028	0.0304	0.4666	0.7968
$10,000$	Davies–Bouldin	0.5631	3.5007	1.3308	0.5627	1.5323	0.6401
	Silhouette	0.6233	0.2247	0.3402	0.5231	0.3025	0.6247
	Classification Error	0.0302	0.3998	0.3173	0.0336	0.4777	0.8114
$20,000$	Davies–Bouldin	0.4795	1.3397	0.9942	0.5682	1.9291	1.4965
	Silhouette	0.6588	0.3314	0.4415	0.5110	0.2818	0.4551
	Classification Error	0.0306	0.4985	0.1320	0.0457	0.4543	0.7052

Table 3.1: Gaussian mixture model (GMM) clustering on embedding space.

3.6 Application

3.6.1 Data Description

The data includes about 240 million grocery transactions that involved $307,365$ households distributed across the US between 2015 and 2017. Purchases are collected directly from consumers via receipt images. This method makes the process much easier on the user end in addition to eliminating user input error. Additional incentives to report as often and as accurately as possible are provided through a rewards system. In addition, this source gathers household demographic information. This allows the panel makeup to mirror U.S. Census data. All household characteristics are weighted and balanced to provide an accurate sample of shopping behavior in the U.S.

Since this is a voluntary program and the participants can leave the panel without restrictions, households have various length of appearance. To remove the noise introduced by short-term, spontaneous participants, I use only a relatively stable subset of the households. I will refer to this sample as the "static" sample, which consists of households who have been in the panel for twelve or more consecutive months. The static sample accounts for 134,869 households.

3.6.2 Customer Segmentation Analysis

I will focus the analysis on household purchases of "snacks“ (Chips, Trail Mix, Candy, etc). I examine whether households can be grouped based on preferences for particular brands in this category. I examine household purchases among the top 200 snack brands in 2016. The data matrix is of the same form as in the simulations. In this case for any element y_{ij} household i either purchases a snack within a brand j or they do not. I do not consider the

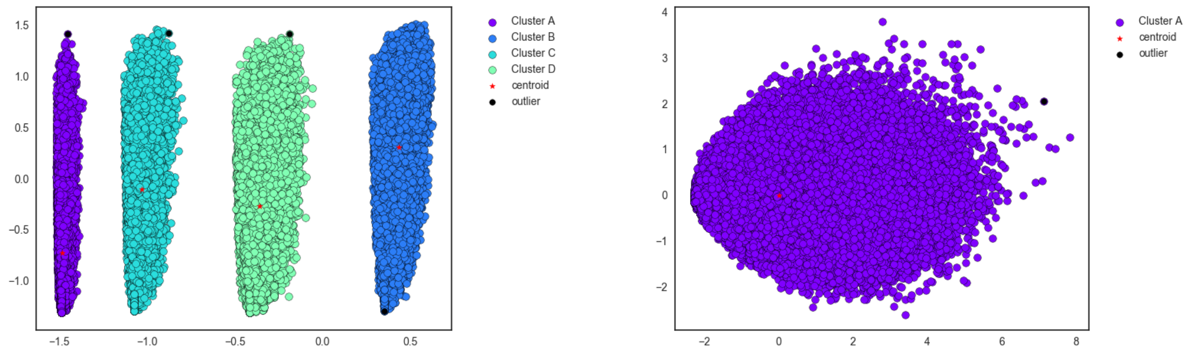


Figure 3.4: Embedding space of consumer types generated by the Autoencoder and PCA respectively.

number of products purchased, only whether a household purchased a brand at any point in 2016.

I maintain the network structure and specifications used during the simulations. The input and output layers have dimension J which is equal to the total number of brands.

I apply GMM to the embedding space for both PCA and the Deep Autoencoder, shown in figure 3.4. The number of clusters is based on visual evidence, which in this case is very clear. One can immediately see PCA has no ability to distinguish between household preferences in this category. However, the Autoencoder picks out four distinct clusters. Since this is an unsupervised learning task, the natural question to ask is whether these clusters have any interpretable meaning? Fortunately, the ability to reconstruct the embedding space gives us insight into the algorithm's process.

3.6.3 Cluster Analysis

Consider the stochastic shopping basket for each cluster centroid. From the reconstruction of these coordinates one can observe the underlying probability that households in each cluster purchase a particular brand. For visualization purposes, figure 3.5 shows the stochastic shopping basket of the centroid households for the top twenty brands.

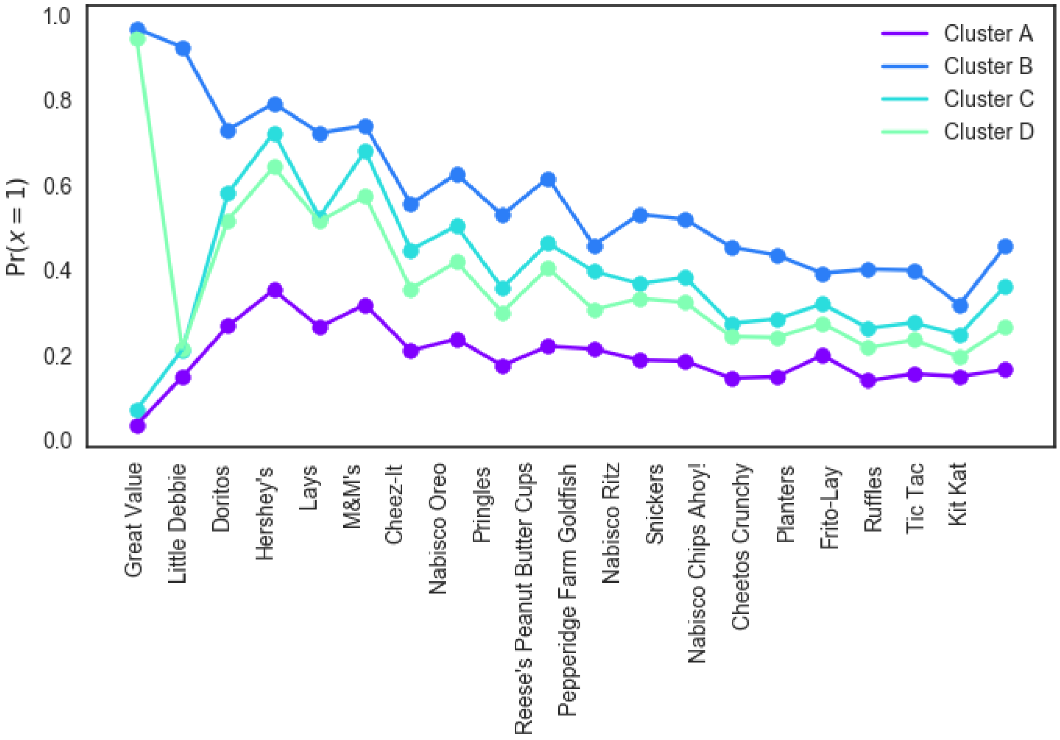


Figure 3.5: Reconstruction for cluster centroids corresponding to the top 20 brands.

One can see each cluster's preferences are well ordered in terms of purchasing probability for any particular brand. Households in Cluster B generally have a higher probability of purchasing these brands relative to the others. Interestingly, households in Cluster D have a very high probability of purchasing a 'Great Value' product, while falling below Cluster B and Cluster C for the remaining brands. This is largely indicative of where these households shop for snacks.

3.6.4 Expenditure and Counts

A further question of interest might be whether the deep Autoencoder picks up on information it does not directly observe. The algorithm is only trained on whether individuals

Multinomial Logistic Regression: Base is Cluster A

	Cluster B		Cluster C		Cluster D	
log(Total Purchases)	1.5292	(0.0185)	0.8849	(0.0204)	0.6974	(0.0155)
log(Total Expenditure)	0.3648	(0.0142)	0.0810	(0.0161)	0.1312	(0.0124)
Income ($< \$50K$)	-10.7875	(0.1026)	-5.9020	(0.1092)	-4.4888	(0.0784)
Income ($\$50K - \$100K$)	-11.4621	(0.1051)	-6.2528	(0.1117)	-4.7955	(0.0804)
Income ($> \$100K$)	-12.3825	(0.1090)	-6.6852	(0.1154)	-5.4014	(0.0837)
Household Size (1)	-0.2966	(0.3711)	-0.1182	(0.3776)	-0.3964	(0.3045)
Household Size ($2 - 4$)	-0.0519	(0.3700)	0.1564	(0.3760)	-0.3089	(0.3035)
Household Size (> 4)	0.0987	(0.3703)	0.3524	(0.3765)	-0.2770	(0.3039)
Asian	-0.5946	(0.3672)	-0.5808	(0.3715)	0.0316	(0.3007)
Black	1.0809	(0.3666)	0.8291	(0.3711)	0.5374	(0.3016)
Hispanic	0.3931	(0.3663)	0.1512	(0.3710)	0.5151	(0.3008)
White	0.6969	(0.3651)	0.2925	(0.3695)	0.5594	(0.2998)
Other Ethnicity	0.2905	(0.3677)	0.0517	(0.3728)	0.3886	(0.3020)
Unknown Demographics	-11.4291	(0.1015)	-6.2776	(0.1071)	-4.7964	(0.0765)

Notes: Coefficient and standard errors (in parentheses)

Table 3.2: Cluster assignment as a function of household characteristics and demographics.

purchased a snack brand during 2016, but perhaps clusters represent a household's expenditure on snacks, or the actual number of snacks purchased over the course of the year.

To this end consider a multinomial logistic regression with the cluster assignment as the dependent variable. The probability any household i is assigned to cluster j is estimated as:

$$p_{ij} = \frac{\exp\{z_i'\beta_j\}}{\sum_{l \in m} \exp\{z_i'\beta_l\}}$$

where z_i is a vector of observables containing indicators for race, household size, and income in addition to the variables of interest, total expenditure on snacks and total number of items purchased in the snack category. The results are summarized in 3.2.

I conduct pairwise tests for combining alternatives via Wald and Likelihood-Ratio tests. In both cases I strongly reject the null that alternatives can be collapsed. This is reassuring, as it implies, the clusters are informative, conditional on this model specification. These results corroborate this estimator's ability to cluster individual preferences based only on observing purchasing decisions.

3.7 Conclusions and Future Work

In settings of high dimensionality, it is often convenient and necessary to examine underlying structures of the data. Dimensionality reduction is an obvious means of accomplishing this task. However, I have shown that linear methods like PCA and even more complex methods like Isomap fail when the underlying DGP is sufficiently entangled. In this setting autoencoders, particularly deep autoencoders can be leveraged to great success.

There are many avenues for future work with autoencoders. The most useful for both theoretical and empirical work would be a rigorous identification result for the embedding space. This is incredibly difficult due to the geometry of lower dimensional manifolds and nonlinear dimensionality reduction. In appendix C, I discuss a simplification to the autoencoder which may reduce the complexity of the identification problem in addition to resolving some of the difficulties with optimization.

Bibliography

Ahn, S. C., and A. R. Horenstein (2013): "Eigenvalue ratio test for the number of factors," *Econometrica*, 81(3), 1203–1227.

Andrews, D. W. (1991): "Asymptotic normality of series estimators for nonparametric and semiparametric regression models," *Econometrica: Journal of the Econometric Society*, pp. 307–345.

Angrist, J. D., and J.-S. Pischke (2008): *Mostly harmless econometrics: An empiricist's companion*. Princeton university press.

Arellano, M., and S. Bond (1991): "Some tests of specification for panel data: Monte Carlo evidence and an application to employment equations," *The review of economic studies*, 58(2), 277–297.

Arellano, M., J. Hahn, et al. (2007): "Understanding bias in nonlinear panel models: Some recent developments," *Econometric Society Monographs*, 43, 381.

Bai, J., and S. Ng (2002): "Determining the number of factors in approximate factor models," *Econometrica*, 70(1), 191–221.

Bajari, P., D. Nekipelov, S. P. Ryan, and M. Yang (2015): "Machine learning methods for demand estimation," *American Economic Review*, 105(5), 481–85.

Baldi, P., and K. Hornik (1989): "Neural networks and principal component analysis: Learning from examples without local minima," *Neural networks*, 2(1), 53–58.

Barron, A. R. (1993): "Universal approximation bounds for superpositions of a sigmoidal function," *IEEE Transactions on Information theory*, 39(3), 930–945.

Belloni, A., V. Chernozhukov, and L. Wang (2011): "Square-root lasso: pivotal recovery of sparse signals via conic programming," *Biometrika*, 98(4), 791–806.

Bester, C. A., and C. Hansen (2009): "A penalty function approach to bias reduction in nonlinear panel models with fixed effects," *Journal of Business & Economic Statistics*, 27(2), 131–148.

Bickel, P. J., Y. Ritov, A. B. Tsybakov, et al. (2009): "Simultaneous analysis of Lasso and Dantzig selector," *The Annals of Statistics*, 37(4), 1705–1732.

Blundell, R., and R. L. Matzkin (2010): "Conditions for the existence of control functions in nonseparable simultaneous equations models," Discussion paper, cemmap working paper.

Calonico, S., M. D. Cattaneo, and M. H. Farrell (2018): "On the effect of bias estimation on coverage accuracy in nonparametric inference," *Journal of the American Statistical Association*, pp. 1–13.

Carreira-Perpinán, M. A. (1997): "A review of dimension reduction techniques," *Department of Computer Science. University of Sheffield. Tech. Rep. CS-96-09*, 9, 1–69.

Carro, J. M. (2007): "Estimating dynamic panel data discrete choice models with fixed effects," *Journal of Econometrics*, 140(2), 503–528.

Chen, X. (2007): "Large sample sieve estimation of semi-nonparametric models," *Handbook of econometrics*, 6, 5549–5632.

Chen, X., and Z. Liao (2014): "Sieve M inference on irregular parameters," *Journal of Econometrics*, 182(1), 70–86.

Chen, X., Z. Liao, and Y. Sun (2014): "Sieve inference on possibly misspecified semi-nonparametric time series models," *Journal of Econometrics*, 178, 639–658.

Chen, X., J. Racine, and N. R. Swanson (2001): "Semiparametric ARX neural-network models with an application to forecasting inflation," *IEEE Transactions on neural networks*, 12(4), 674–683.

Chen, X., and X. Shen (1998): "Sieve extremum estimates for weakly dependent data," *Econometrica*, pp. 289–314.

Chen, X., and H. White (1999): "Improved rates and asymptotic normality for nonparametric neural network estimators," *IEEE Transactions on Information Theory*, 45(2), 682–691.

Chernozhukov, V., M. Goldman, V. Semenova, and M. Taddy (2017): "Orthogonal Machine Learning for Demand Estimation: High Dimensional Causal Inference in Dynamic Panels," *arXiv preprint arXiv:1712.09988*.

Chernozhukov, V., J. A. Hausman, and W. K. Newey (2019): "Demand analysis with many prices," Discussion paper, National Bureau of Economic Research.

Fan, J., F. Han, and H. Liu (2014): "Challenges of big data analysis," *National science review*, 1(2), 293–314.

Farrell, M. H., T. Liang, and S. Misra (2018): "Deep neural networks for estimation and inference: Application to causal effects and other semiparametric estimands," *arXiv preprint arXiv:1809.09953*.

Fernández-Val, I. (2009): "Fixed effects estimation of structural parameters and marginal effects in panel probit models," *Journal of Econometrics*, 150(1), 71–85.

Fernández-Val, I., and F. Vella (2011): "Bias corrections for two-step fixed effects panel data estimators," *Journal of Econometrics*, 163(2), 144–162.

Fernández-Val, I., and M. Weidner (2016): "Individual and time effects in nonlinear panel models with large N, T," *Journal of Econometrics*, 192(1), 291–312.

——— (2018): "Fixed effects estimation of large-t panel data models," *Annual Review of Economics*, 10, 109–138.

Friedman, J. H., and W. Stuetzle (1981): "Projection pursuit regression," *Journal of the American statistical Association*, 76(376), 817–823.

Gallant, A. R., and H. White (1988): "There exists a neural network that does not make avoidable mistakes," in *Proc. of the International Conference on Neural Networks, San Diego*.

Grenander, U. (1981): "Abstract inference," Discussion paper, Wiley Series, New York.

Hahn, J., and G. Kuersteiner (2002): "Asymptotically unbiased inference for a dynamic panel model with fixed effects when both n and T are large," *Econometrica*, 70(4), 1639–1657.

Hahn, J., Z. Liao, and G. Ridder (2018): "Nonparametric two-step sieve M estimation and inference," *Econometric Theory*, pp. 1–44.

Hahn, J., and W. Newey (2004): "Jackknife and analytical bias reduction for nonlinear panel models," *Econometrica*, 72(4), 1295–1319.

Harding, M., E. Leibtag, and M. F. Lovenheim (2012): "The heterogeneous geographic and socioeconomic incidence of cigarette taxes: evidence from Nielsen homescan data," *American Economic Journal: Economic Policy*, 4(4), 169–98.

Hausman, J. A., and W. K. Newey (2017): "Nonparametric welfare analysis," *Annual Review of Economics*, 9, 521–546.

Heckman, J. (1979): "Sample Selection Bias as a Specification Error," *Econometrica*, 47(1), 153161.

Heckman, J. J. (1976): "The common structure of statistical models of truncation, sample selection and limited dependent variables and a simple estimator for such models," in *Annals of economic and social measurement, volume 5, number 4*, pp. 475–492. NBER.

Hendel, I., and A. Nevo (2002): "Measuring the Implications of Sales and Consumer Stockpiling Behavior1," Discussion paper, Working Paper, Northwestern University.

Hinton, G. E. (2002): "Training products of experts by minimizing contrastive divergence," *Neural computation*, 14(8), 1771–1800.

Hinton, G. E., and R. R. Salakhutdinov (2006): "Reducing the dimensionality of data with neural networks," *science*, 313(5786), 504–507.

Hornik, K., M. Stinchcombe, and H. White (1989): “Multilayer feedforward networks are universal approximators,” *Neural networks*, 2(5), 359–366.

——— (1990): “Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks,” *Neural networks*, 3(5), 551–560.

Hotelling, H. (1933): “Analysis of a complex of statistical variables into principal components.,” *Journal of educational psychology*, 24(6), 417.

Hyvärinen, A., and E. Oja (2000): “Independent component analysis: algorithms and applications,” *Neural networks*, 13(4), 411–430.

Jolliffe, I. T. (2002): “Graphical representation of data using principal components,” *Principal component analysis*, pp. 78–110.

Kolmogorov, A., and V. Tikhomirov (1959): “epsilon-entropy and epsilon-capacity,” *Uspekhi Mat. Nauk*, 14, 3–86.

Lewbel, A., and K. Pendakur (2009): “Tricks with Hicks: The EASI demand system,” *American Economic Review*, 99(3), 827–63.

Makovoz, Y. (1996): “Random approximants and neural networks,” *Journal of Approximation Theory*, 85(1), 98–109.

Marron, J. S., and M. P. Wand (1992): “Exact mean integrated squared error,” *The Annals of Statistics*, pp. 712–736.

Newey, W. K. (1994): “Series estimation of regression functionals,” *Econometric Theory*, 10(1), 1–28.

——— (1997): “Convergence rates and asymptotic normality for series estimators,” *Journal of econometrics*, 79(1), 147–168.

Newey, W. K., and J. L. Powell (2003): “Instrumental variable estimation of nonparametric models,” *Econometrica*, 71(5), 1565–1578.

Newey, W. K., J. L. Powell, and F. Vella (1999): “Nonparametric estimation of triangular simultaneous equations models,” *Econometrica*, 67(3), 565–603.

Neyman, J., and E. L. Scott (1948): “Consistent estimates based on partially consistent observations,” *Econometrica: Journal of the Econometric Society*, pp. 1–32.

Onatski, A. (2010): “Determining the number of factors from empirical distribution of eigenvalues,” *The Review of Economics and Statistics*, 92(4), 1004–1016.

——— (2015): “Asymptotic analysis of the squared estimation error in misspecified factor models,” *Journal of Econometrics*, 186(2), 388–406.

Pearson, K. (1901): “LIII. On lines and planes of closest fit to systems of points in space,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11), 559–572.

Politis, D. N., and H. White (2004): "Automatic block-length selection for the dependent bootstrap," *Econometric Reviews*, 23(1), 53–70.

Roweis, S. T., and L. K. Saul (2000): "Nonlinear dimensionality reduction by locally linear embedding," *science*, 290(5500), 2323–2326.

Safran, I., and O. Shamir (2017): "Depth-width tradeoffs in approximating natural functions with neural networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2979–2987. JMLR. org.

Sanderson, E., and F. Windmeijer (2016): "A weak instrument F-test in linear IV models with multiple endogenous variables," *Journal of Econometrics*, 190(2), 212–221.

Shen, X. (1997): "On methods of sieves and penalization," *The Annals of Statistics*, pp. 2555–2591.

Shen, X., and W. H. Wong (1994): "Convergence rate of sieve estimates," *The Annals of Statistics*, pp. 580–615.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov (2014): "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, 15(1), 1929–1958.

Stock, J. H., and M. Yogo (2002): "Testing for weak instruments in linear IV regression," Discussion paper, National Bureau of Economic Research.

Sun, Y., Y. Y. Zhang, and Q. Li (2015): "Nonparametric panel data regression models," *The Oxford Handbook of Panel Data*, pp. 285–324.

Telgarsky, M. (2016): "Benefits of depth in neural networks," *arXiv preprint arXiv:1602.04485.*

Tenenbaum, J. B., V. De Silva, and J. C. Langford (2000): "A global geometric framework for nonlinear dimensionality reduction," *science*, 290(5500), 2319–2323.

White, H. (1990): "Connectionist nonparametric regression: Multilayer feedforward networks can learn arbitrary mappings," *Neural networks*, 3(5), 535–549.

White, H., and J. Wooldridge (1991): "Some results on sieve estimation with dependent observations," *Nonparametric and Semiparametric Methods in Economics*, pp. 459–493.

Yarotsky, D. (2017): "Error bounds for approximations with deep ReLU networks," *Neural Networks*, 94, 103–114.

Appendix A

This appendix provides a supplement to chapter 1. I provide the complete entropy calculations used for the consistency and convergence rate results (section 1.5) below in section A.1. In addition I discuss an interpretation of ReLU networks in section A.2. The latter is an interesting aside and may be considered for future research.

A.1 Entropy Calculations

I suppress the dependence on i and define $z_j \equiv (1, s(\tilde{x}'\delta_j))'$ and $\zeta_j \equiv (1, s(\tilde{x}'d_j))'$.

Let $\eta > 0$ be given such that B_η, G_η, and D_η are η-nets for $\mathcal{B} = \{\beta : ||\beta|| \leq \Delta\} \subset \mathbb{R}^{K+1}$, $\Gamma = \{\gamma : ||\gamma|| \leq K\Delta\} \subset \mathbb{R}^{(d+1)K}$, and $\mathcal{D} = \{\delta : ||\delta|| \leq d\Delta\} \subset \mathbb{R}^{(p+1)d}$.

Further the feasible parameter space can be written as: $\mathcal{H}_\eta = \mathcal{B}_\eta \times \Gamma_\eta \times \mathcal{D}_\eta$. Let $f(x)$ be an arbitrary two-layer neural network with parameters (β, γ, δ). There exists corresponding parameters $\varrho = (b, c, d) \in \mathcal{H}_\eta$ that satisfy $||\beta - b|| \leq \eta$, $||\gamma - c|| \leq \eta$, and $||\delta - d|| \leq \eta$.

$$|f(x) - t(x)| = \left|\beta_0 + \sum_{j=1}^{K} \beta_j s(z_j'\gamma_j) - b_0 - \sum_{j=1}^{K} b_j s(\zeta_j' c_j)\right| \tag{A.1}$$

$$\leq \left|\beta_0 - b_0 + \sum_{j=1}^{K} (\beta_j - b_j) s(z_j'\gamma_j)\right| + \left|\sum_{j=1}^{K} b_j \left(s(z_j'\gamma_j) - s(\zeta_j' c_j)\right)\right| \tag{A.2}$$

The first term is bounded by $||\beta - b|| \leq \eta$ by the choice of b and using the fact that $\sup_{x \in \mathcal{X}} |s(t)| = 1$. For the second term:

$$\left|\sum_{j=1}^{K} b_j \left(s(z_j'\gamma_j) - s(\zeta_j' c_j)\right)\right| \leq \sum_{j=1}^{K} |b_j| \sum_{j=1}^{K} \left|s(z_j'\gamma_j) - s(\zeta_j' c_j)\right| \tag{A.3}$$

The magnitude of the b vector is restricted by Δ and by the Lipschitz condition on $s(\cdot)$:

$$\sum_{j=1}^{K} |b_j| \sum_{j=1}^{K} \left|s(z_j'\gamma_j) - s(\zeta_j' c_j)\right| \leq \Delta \sum_{j=1}^{K} \left|z_j'\gamma_j - \zeta_j' c_j\right| \tag{A.4}$$

This would be complete in the single layer case. However, since I have an additional layer, I plug-in the definition of z_j and ζ_j:

$$\Delta \sum_{j=1}^{K} |z'\gamma_j - \zeta' c_j| \leq \Delta \sum_{j=1}^{K} \left|\gamma_{0j} + \sum_{\ell=1}^{d} \gamma_{j\ell} s(\tilde{x}'\delta_\ell) - c_{0j} - \sum_{\ell=1}^{d} c_{j\ell} s(\tilde{x}' d_\ell)\right| \tag{A.5}$$

One will notice that to bound the remaining components the procedure is identical to above.

Examining each component of the sum individually:

$$\left|\gamma_{0\ell} + \sum_{\ell=1}^{d} \gamma_{j\ell} s(\tilde{x}'\delta_\ell) - c_{0\ell} - \sum_{\ell=1}^{d} c_{j\ell} s(\tilde{x}' d_\ell)\right| \tag{A.6}$$

$$\leq \sum_{\ell=0}^{d} |\gamma_{j\ell} - c_{j\ell}| + \Delta \sum_{\ell=1}^{d} |\tilde{x}'\delta_\ell - \tilde{x}' d_\ell| \tag{A.7}$$

$$\leq \sum_{\ell=0}^{d} |\gamma_{j\ell} - c_{j\ell}| + \Delta \sum_{\ell=1}^{d} \left[\left(\sum_{m=0}^{p} |x_m| \right) \sum_{m=0}^{p} |\delta_{m\ell} - d_{m\ell}| \right] \tag{A.8}$$

$$= \sum_{\ell=0}^{d} |\gamma_{j\ell} - c_{j\ell}| + + \Delta p \sum_{\ell=1}^{d} \left[\sum_{m=0}^{p} |\delta_{m\ell} - d_{m\ell}| \right] \tag{A.9}$$

Substituting back:

$$\Delta \sum_{j=1}^{K} \left(\sum_{\ell=0}^{d} |\gamma_{j\ell} - c_{j\ell}| + \Delta p \sum_{\ell=1}^{d} \left[\sum_{m=0}^{p} |\delta_{m\ell} - d_{m\ell}| \right] \right) \tag{A.10}$$

$$= \Delta \sum_{j=1}^{K} \sum_{\ell=0}^{d} |\gamma_{j\ell} - c_{j\ell}| + \Delta^2 p \sum_{j=1}^{K} \sum_{\ell=1}^{d} \left[\sum_{m=0}^{p} |\delta_{m\ell} - d_{m\ell}| \right] \tag{A.11}$$

$$\leq \Delta\eta + \Delta^2 p\eta \tag{A.12}$$

Since this holds for any $x \in \mathcal{X}$:

$$\sup_{x \in \mathcal{X}} |f(x) - t(x)| \leq \eta(1 + \Delta + \Delta^2 p) = \varepsilon$$

Let # denote the cardinality operator. Then: $\#T = (\#B)(\#G)(\#D)$. Each of which can be bounded by results in Kolmogorov and Tikhomirov (1959).

$$\#B \leq 2(2\Delta/\eta)^{K+1} \tag{A.13}$$

$$\#G \leq 2(2K\Delta/\eta)^{(d+1)K} \tag{A.14}$$

$$\#D \leq 2(2d\Delta/\eta)^{(p+1)d} \tag{A.15}$$

With $\eta = \epsilon / \left(1 + \Delta + \Delta^2 p\right))$ and let $\omega = 1 + K(d+2) + d(p+1)$

$$\log \#T \leq \log 8 + (K+1)\log\frac{2\Delta}{\eta} + K(d+1)\log\frac{2K\Delta}{\eta} + (p+1)d\log\frac{2d\Delta}{\eta} \tag{A.16}$$

$$= \log 8 + \omega\log\frac{2\Delta}{\eta} + K(d+1)\log K + (p+1)d\log d \tag{A.17}$$

$$\leq \omega\left[\log\frac{16}{\epsilon} + \log\left(\Delta(1+\Delta+p\Delta^2)\right) + \log dK\right] \tag{A.18}$$

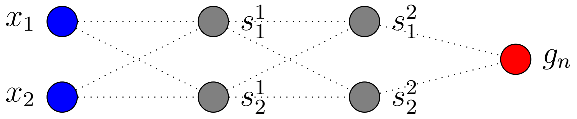
A.2 ReLU Networks

A natural question that may arise is how to generalize the results of chapter 1 to deeper architectures? In the case of smooth activation functions, it does not seem reasonable to continue to stack layers without explicitly restricting connections to reduce the entropy. In the approximation results from 1.4 I argue that an additional layer reduces the burden placed on each 'basis' (defined by ψ_k) by approximating only pieces of the underlying function rather than the whole. This could be argued to hold iteratively, each subsequent layer reducing the complexity necessary for the previous layer to estimate the underlying function. However, it is not immediately clear how one would show this rigorously. Interestingly, stacking layers in practice is not typically done with smooth activation functions, but rather the ReLU ($s(t) = \mathbf{1}\left\{t > 0\right\} t$) function mentioned in 1.3. This section presents a case for why these activation functions are useful, but also why they are fundamentally different from neural networks with smooth activation functions.

Consider the following illustration. Let $x_i \in \mathcal{X} = [0,1]^2$ and $y \in [0,1]$. I claim that for any fixed value of z_i one can embed a deep ReLU network into a local linear model.

Suppose without loss of generality the network has two hidden layers as in equation 1.4 where s_0 and s_1 are ReLU functions denoted $\mathbf{1}_+$ for simplicity. Then consider any pair $x_{i1} = x_1, x_{i2} = x_2$ with outcome $y_i = y$. The network can be visually represented as:

Consider the output of each hidden node:



$$s_1^1 = \mathbf{1}_+ \left\{x_1\gamma_{11} + x_2\gamma_{21} + \gamma_{01}\right\}$$

$$s_2^1 = \mathbf{1}_+ \left\{x_1\gamma_{12} + x_2\gamma_{22} + \gamma_{02}\right\}$$

$$s_1^2 = \mathbf{1}_+ \left\{s_1^0\omega_{11} + s_2^0\omega_{21} + \omega_{01}\right\}$$

$$s_2^2 = \mathbf{1}_+ \left\{s_1^0\omega_{12} + s_2^0\omega_{22} + \omega_{02}\right\}$$

Then given model parameters γ and ω the network estimates $y = c_0 + c_1x_1 + c_2x_2$. In this simple example there are $(2^2-1)^2 = 9$ possible (non-degenerate) submodels. However, it is important to note that when z is fixed only one of these models is realized. For example if z, γ, ω are such that all $s_j^k > 0$ for $j, k \in \{1, 2\}$

$$\begin{aligned} y &= \beta_1 \left[\left(x_1\gamma_{11} + x_2\gamma_{21} + \gamma_{01}\right)\omega_{11} + \left(x_1\gamma_{12} + x_2\gamma_{22} + \gamma_{02}\right)\omega_{21} + \omega_{01}\right] \\ &\quad + \beta_2 \left[\left(x_1\gamma_{11} + x_2\gamma_{21} + \gamma_{01}\right)\omega_{12} + \left(x_1\gamma_{12} + x_2\gamma_{22} + \gamma_{02}\right)\omega_{22} + \omega_{02}\right] \\ c_0 &= \beta_1\left(\omega_{01} + \gamma_{02}\omega_{21} + \gamma_{01}\omega_{11}\right) + \beta_2\left(\omega_{02} + \gamma_{02}\omega_{22} + \gamma_{01}\omega_{12}\right) \\ c_1 &= \beta_1\left(\omega_{11}\gamma_{11} + \omega_{21}\gamma_{12}\right) + \beta_2\left(\omega_{12}\gamma_{11} + \omega_{22}\gamma_{12}\right) \\ c_2 &= \beta_1\left(\omega_{11}\gamma_{21} + \omega_{21}\gamma_{22}\right) + \beta_2\left(\omega_{12}\gamma_{21} + \omega_{22}\gamma_{22}\right) \end{aligned}$$

Therefore, deep neural networks with ReLU activation functions are local linear models with data driven partitions. This is in sharp contrast to neural networks with smooth activation functions which construct approximations to the underlying function using all of the data. Given the empirical success of linear models and even those of local linear models it is unsurprising that ReLU networks have had such great success in practice. However, this does give us some insight as to why copious amounts of data is vital to the success of these networks. The estimate for each partition will depend on how many observations lie in that subset.

A potential area for further research is to prepend a ReLU network to the classical (smooth single layer) neural network. This would result in a local neural network where all the theoretical results for classical networks would follow conditional on the subset selection defined in the ReLU portion.

A.3 Inference Supplemental

This section presents the conditions for asymptotic normality to hold for the evaluation functional of extended or classical neural network estimator. In section 1.6 I take these as given and leave verification to future work.

AA.3.1 Suppose $g_{0,n} = \arg\min_{g \in G_n} ||g - g_0||$. The approximation error of the sieve functional satisfies:

$$\frac{\left|\frac{\partial h(g_0)}{\partial g}[g_{0,n} - g_0]\right|}{||v_n^\star||_\ell} = o(n^{-1/2}) \tag{A.19}$$

and $||v_n^\star|| \, / \, ||v_n^\star|| = O(1)$.

AA.3.2 Let $\epsilon_n = o_p(n^{-1/2})$. Then the following stochastic equicontinuity conditions hold:

$$\sup_{g \in \mathcal{G}_n} \mu_n \left\{\ell(g \pm \epsilon_n u_n^\star, z_i) - \ell(g, z_i) - \Delta(g_0, z_i)[\pm \epsilon_n u_n^\star]\right\} = O_p(\epsilon_n^2) \tag{A.20}$$

$$\sup_{g \in \mathcal{G}_n} \left| \mathbb{E}\left[\ell(g, z_i) - \ell(g \pm \epsilon_n u_n^\star, z_i)\right] - \frac{||g \pm \epsilon_n u_n^\star - g_0||^2 - ||g - g_0||^2}{2} \right| = O_p(\epsilon_n^2)$$

(A.21)

AA.3.3 The following central limit theorem holds:

$$\sqrt{n}\mu_n[\Delta(g_0, z_i)[u_n^\star]] \overset{d}{\rightarrow} N(0,1) \tag{A.22}$$

Appendix B

This appendix provides a supplement to chapter 2. In section B.1 I provide additional tables describing the various samples discussed in 2.7.1. I examine a map between total expenditure observed in the data and the reported income levels from the consumer panel data in section B.2. The issue of missing at random is further discussed in section B.3 and first stage relevance is examine in section B.4.

B.1 Descriptive Statistics

The following tables compare the full Nielsen sample for each panel block with the sample of users who purchase cigarettes and the 'active' user group. The samples for the latter two are similar across income, education, and race, but tend to differ across household composition. In general, the 'active' sample tends to have a higher proportion with no children under 18 and tends towards older (50+) heads of household.

Panel Years	Household Income	Full Sample	Cigarette Sample	Cigarette Sub-Sample
07-10	<$10,000	2.77	4.78	4.84
	$10,000-$24,999	13.15	18.63	18.63
	$25,000-$49,999	31.93	38.55	41.37
	$50,000-$99,999	38.87	31.77	30.00
	$100,000+	13.27	6.27	5.16
08-11	<$10,000	2.72	4.57	3.78
	$10,000-$24,999	12.75	19.35	18.67
	$25,000-$49,999	31.15	37.41	39.98
	$50,000-$99,999	39.12	31.76	31.27
	$100,000+	14.26	6.91	6.30
09-12	<$10,000	2.81	4.83	4.45
	$10,000-$24,999	12.84	19.46	17.57
	$25,000-$49,999	30.50	36.13	38.03
	$50,000-$99,999	39.17	32.13	32.73
	$100,000+	14.68	7.45	7.22
10-13	<$10,000	2.97	5.24	4.51
	$10,000-$24,999	13.15	20.15	18.76
	$25,000-$49,999	30.62	35.76	35.99
	$50,000-$99,999	39.00	31.63	33.85
	$100,000+	14.25	7.21	6.89
11-14	<$10,000	3.17	5.77	4.21
	$10,000-$24,999	13.59	21.70	19.18
	$25,000-$49,999	30.68	36.00	37.03
	$50,000-$99,999	38.26	29.78	32.71
	$100,000+	14.30	6.74	6.87
12-15	<$10,000	3.21	5.75	4.31
	$10,000-$24,999	13.40	21.48	19.28
	$25,000-$49,999	30.65	36.52	37.49
	$50,000-$99,999	38.06	29.40	32.22
	$100,000+	14.69	6.85	6.71
13-16	<$10,000	3.11	5.57	4.52
	$10,000-$24,999	12.70	21.01	19.10
	$25,000-$49,999	30.12	36.38	40.45
	$50,000-$99,999	38.54	29.80	28.14
	$100,000+	15.53	7.23	7.79
14-17	<$10,000	2.93	5.36	3.86
	$10,000-$24,999	12.05	20.92	20.00
	$25,000-$49,999	29.41	36.17	36.97
	$50,000-$99,999	39.00	29.51	31.59
	$100,000+	16.61	8.04	7.59

Table B.1: Sample proportions for reported income brackets.

Panel Years	Head of Household Race	Full Sample	Cigarette Sample	Cigarette Sub-Sample
07-10	Asian	2.51	0.83	0.53
	Black	9.11	8.26	10.42
	Other	4.65	4.71	3.68
	White	83.73	86.19	85.37
08-11	Asian	2.65	0.91	0.80
	Black	9.32	8.74	10.54
	Other	4.43	4.56	3.89
	White	83.60	85.79	84.77
09-12	Asian	2.80	0.99	1.08
	Black	9.47	9.19	12.15
	Other	4.51	4.60	3.49
	White	83.22	85.22	83.27
10-13	Asian	2.91	0.98	0.83
	Black	9.85	9.77	11.52
	Other	4.65	4.82	3.68
	White	82.59	84.43	83.97
11-14	Asian	3.00	1.10	1.11
	Black	10.15	9.91	11.09
	Other	4.64	4.69	3.22
	White	82.21	84.30	84.59
12-15	Asian	3.16	1.27	0.96
	Black	10.56	10.39	10.90
	Other	4.93	4.66	3.47
	White	81.35	83.68	84.67
13-16	Asian	3.27	1.31	0.75
	Black	10.63	9.97	11.43
	Other	5.05	4.65	3.02
	White	81.05	84.07	84.80
14-17	Asian	3.50	1.34	0.97
	Black	10.58	9.68	11.31
	Other	5.15	4.82	3.72
	White	80.76	84.16	84.00

Table B.2: Sample proportions for reported race indicators.

Panel Years	Age of Children	Full Sample	Cigarette Sample	Cigarette Sub-Sample
07-10	13-17 only	7.99	8.32	5.26
	6-12 and 13-17	4.32	3.37	1.79
	6-12 only	6.63	6.06	3.37
	No Children under 18	70.86	75.15	86.11
	Under 6 and 13-17	0.69	0.82	0.63
	Under 6 and 6-12	3.75	2.16	1.05
	Under 6 and 6-12 and 13-17	0.82	0.78	0.11
	Under 6 only	4.94	3.35	1.68
08-11	13-17 only	7.79	7.95	5.15
	6-12 and 13-17	4.14	3.09	1.60
	6-12 only	6.40	5.77	3.21
	No Children under 18	72.31	76.80	86.03
	Under 6 and 13-17	0.62	0.68	0.57
	Under 6 and 6-12	3.52	2.09	1.03
	Under 6 and 6-12 and 13-17	0.78	0.71	0.23
	Under 6 only	4.45	2.92	2.18
09-12	13-17 only	7.53	7.17	4.69
	6-12 and 13-17	4.17	2.92	1.20
	6-12 only	6.21	5.32	3.25
	No Children under 18	72.87	78.96	87.73
	Under 6 and 13-17	0.57	0.54	0.48
	Under 6 and 6-12	3.62	1.88	1.08
	Under 6 and 6-12 and 13-17	0.77	0.56	0.12
	Under 6 only	4.26	2.65	1.44
10-13	13-17 only	7.42	6.95	4.63
	6-12 and 13-17	4.13	2.83	1.43
	6-12 only	6.33	5.20	3.80
	No Children under 18	73.34	79.73	88.24
	Under 6 and 13-17	0.57	0.51	0.36
	Under 6 and 6-12	3.49	1.68	0.48
	Under 6 and 6-12 and 13-17	0.75	0.53	0.00
	Under 6 only	3.97	2.57	1.07

Table B.3: Sample proportions for reported presence and ages of children in the household.

Panel Years	Age of Children	Full Sample	Cigarette Sample	Cigarette Sub-Sample
11-14	13-17 only	7.37	6.40	5.10
	6-12 and 13-17	4.01	2.45	1.88
	6-12 only	6.32	4.99	3.33
	No Children under 18	73.44	81.01	88.14
	Under 6 and 13-17	0.59	0.53	0.22
	Under 6 and 6-12	3.56	1.71	0.11
	Under 6 and 6-12 and 13-17	0.75	0.47	0.00
	Under 6 only	3.97	2.45	1.22
12-15	13-17 only	7.37	6.60	4.91
	6-12 and 13-17	4.02	2.49	1.20
	6-12 only	6.44	5.25	2.63
	No Children under 18	72.81	80.10	89.82
	Under 6 and 13-17	0.60	0.58	0.12
	Under 6 and 6-12	3.68	1.87	0.24
	Under 6 and 6-12 and 13-17	0.82	0.50	0.24
	Under 6 only	4.27	2.61	0.84
13-16	13-17 only	7.34	6.64	4.90
	6-12 and 13-17	4.12	2.83	1.38
	6-12 only	6.58	5.36	2.14
	No Children under 18	71.57	79.53	89.95
	Under 6 and 13-17	0.65	0.56	0.13
	Under 6 and 6-12	4.05	1.92	0.75
	Under 6 and 6-12 and 13-17	0.88	0.42	0.13
	Under 6 only	4.82	2.74	0.63
14-17	13-17 only	7.40	6.85	3.45
	6-12 and 13-17	4.24	3.16	2.07
	6-12 only	6.64	5.23	2.62
	No Children under 18	70.38	78.25	89.66
	Under 6 and 13-17	0.68	0.69	0.41
	Under 6 and 6-12	4.35	2.23	0.41
	Under 6 and 6-12 and 13-17	0.95	0.50	0.00
	Under 6 only	5.36	3.09	1.38

Table B.4: Sample proportions for reported presence and ages of children in the household.

Panel Years	Max Household Age	Full Sample	Cigarette Sample	Cigarette Sub-Sample
07-10	Under 25	0.29	0.17	0.00
	25-34	7.28	4.47	0.63
	35-49	32.59	33.10	24.21
	50-64	40.20	49.01	58.00
	65+	19.64	13.26	17.16
08-11	Under 25	0.23	0.13	0.00
	25-34	6.56	3.84	0.69
	35-49	30.60	30.18	20.96
	50-64	41.70	51.33	59.68
	65+	20.91	14.51	18.67
09-12	Under 25	0.27	0.15	0.24
	25-34	6.63	3.78	0.72
	35-49	29.46	28.08	19.01
	50-64	41.96	52.47	59.69
	65+	21.67	15.54	20.34
10-13	Under 25	0.30	0.10	0.12
	25-34	6.55	3.27	0.48
	35-49	28.26	25.80	16.63
	50-64	42.67	54.59	61.28
	65+	22.23	16.24	21.50
11-14	Under 25	0.36	0.11	0.00
	25-34	6.94	3.50	0.67
	35-49	27.51	23.99	15.74
	50-64	42.64	54.60	58.98
	65+	22.55	17.80	24.61
12-15	Under 25	0.44	0.13	0.12
	25-34	7.77	3.94	0.60
	35-49	27.24	23.19	14.13
	50-64	41.51	53.66	58.08
	65+	23.04	19.08	27.07
13-16	Under 25	0.52	0.23	0.13
	25-34	9.10	4.70	0.50
	35-49	27.61	22.83	12.31
	50-64	39.75	52.21	59.05
	65+	23.02	20.03	28.02
14-17	Under 25	0.58	0.28	0.00
	25-34	10.04	5.12	0.41
	35-49	28.13	23.32	10.48
	50-64	38.40	50.84	59.31
	65+	22.85	20.45	29.79

Table B.5: Sample proportions for reported ages pertaining to the head of household.

Panel Years	Max Education Attained	Full Sample	Cigarette Sample	Cigarette Sub-Sample
07-10	Grade School	0.16	0.17	0.11
	Graduated College	34.56	25.60	23.68
	Graduated High School	17.08	25.30	27.79
	Post College Grad	16.18	5.57	6.74
	Some College	30.84	41.01	39.05
	Some High School	1.18	2.35	2.63
08-11	Grade School	0.18	0.21	0.23
	Graduated College	35.12	26.13	26.69
	Graduated High School	16.66	24.75	24.40
	Post College Grad	16.62	5.93	7.10
	Some College	30.34	40.64	39.40
	Some High School	1.09	2.35	2.18
09-12	Grade School	0.16	0.17	0.00
	Graduated College	35.68	26.91	27.32
	Graduated High School	16.45	23.70	23.71
	Post College Grad	16.77	6.15	7.34
	Some College	29.86	40.72	39.23
	Some High School	1.07	2.35	2.41
10-13	Grade School	0.16	0.20	0.00
	Graduated College	36.16	26.81	29.45
	Graduated High School	16.22	23.98	23.28
	Post College Grad	16.73	6.16	6.53
	Some College	29.68	40.61	38.84
	Some High School	1.05	2.23	1.90
11-14	Grade School	0.17	0.24	0.00
	Graduated College	36.23	26.86	27.05
	Graduated High School	16.00	23.91	22.73
	Post College Grad	16.85	6.10	7.10
	Some College	29.71	40.44	40.69
	Some High School	1.04	2.45	2.44

Table B.6: Sample proportions for reported maximum education obtained in a household.

Panel Years	Max Education Attained	Full Sample	Cigarette Sample	Cigarette Sub-Sample
12-15	Grade School	0.16	0.18	0.00
	Graduated College	36.54	26.56	27.66
	Graduated High School	15.60	23.82	24.55
	Post College Grad	16.92	5.88	6.71
	Some College	29.69	40.98	38.08
	Some High School	1.09	2.57	2.99
13-16	Grade School	0.17	0.24	0.25
	Graduated College	37.31	27.48	27.01
	Graduated High School	15.07	23.41	23.62
	Post College Grad	17.52	6.18	6.53
	Some College	28.95	40.44	40.45
	Some High School	0.98	2.25	2.14
14-17	Grade School	0.13	0.20	0.14
	Graduated College	36.01	25.94	24.00
	Graduated High School	16.25	25.92	28.41
	Post College Grad	18.24	6.29	6.62
	Some College	28.39	39.37	38.62
	Some High School	0.97	2.27	2.21

Table B.7: Sample proportions for reported maximum education obtained in a household.

B.2 Income and Total Expenditure

This section discusses the mapping between total expenditure $y_{it} = \sum_j p_{ijt}q_{ijt}$ and income. This is a useful digression as one may want to know how elasticities vary across income levels. However, the results from section 2.7 apply only to total expenditure. To tease out how these are related I examine a regression of total expenditure on the reported income brackets in the consumer panel, i.e.:

$$y_{it} = \alpha + \gamma_t + z_i'\beta + \varepsilon_{it} \tag{B.1}$$

where z_i is the binary vector indicating whether household i belongs to each income bracket. The income brackets are the same as in table B.1 with the exclusion of the lowest bracket ($< \$10,000$) which is left out and absorbed into α. I include time effects to capture seasonal or other unobserved shocks. The results are summarized in table B.8.

Coefficients	07-10	08-11	09-12	10-13	11-14	12-15	13-16	14-17
$\hat{\alpha}$	8.0380	8.0681	8.1463	8.1585	8.2177	8.2975	8.2772	8.3579
	(0.0343)	(0.0349)	(0.0362)	(0.0374)	(0.0377)	(0.0394)	(0.0402)	(0.0414)
$\$10,000 - \$24,999$	0.2319	0.2052	0.1866	0.1849	0.1480	0.1194	0.1052	0.0585
	(0.0356)	(0.0362)	(0.0372)	(0.0385)	(0.0389)	(0.0405)	(0.0416)	(0.0429)
$\$25,000 - \$49,999$	0.5372	0.4948	0.4574	0.4526	0.4225	0.4038	0.4041	0.3631
	(0.0349)	(0.0355)	(0.0367)	(0.0379)	(0.0384)	(0.0399)	(0.0408)	(0.0420)
$\$50,000 - \$99,999$	0.7566	0.7064	0.6588	0.6416	0.6161	0.5992	0.5992	0.5541
	(0.0350)	(0.0355)	(0.0367)	(0.0379)	(0.0384)	(0.0399)	(0.0410)	(0.0422)
$\$100,000+$	0.8610	0.8036	0.7572	0.7092	0.6585	0.6398	0.6428	0.6132
	(0.0395)	(0.0401)	(0.0413)	(0.0428)	(0.0434)	(0.0447)	(0.0456)	(0.0469)

Table B.8: Regression output from B.1. Standard errors in parentheses.

Total expenditure does monotonically increase with income brackets and these brackets are typically informative. Most of the indicators are significant at the 5% level except for the lower income bracket $\$10,000 - \$24,999$ in the final two panel blocks. If one takes these results seriously it would then be possible to estimate elasticities conditional on income bracket using the same approach as in table 2.9, but with cutoffs determined by table B.8.

B.3 Missing Quantities

As discussed in section 2.7.1, I assume that missing purchases of cigarettes for any given month are missing at random (MAR). While this assumption is untestable in general, one can attempt to falsify it by examining the conditional distributions of the missing outcomes as well as the outcomes themselves. In the former case the unconditional distribution is summarized in table B.9. I report the frequency of missing cigarette purchases in each sub-panel for the 'active' cigarette user sample. Across all panels most households miss reporting one or fewer months, but still a non-negligible (slightly less than ten percent) amount have greater than four missing months. I check for potential correlation between missing values and past purchases to determine if the source of missingness is related to stockpiling. One might imagine a situation where a household buys many cigarettes every other month. However, these kinds of patterns are not supported by the data as can be seen in B.10. A large majority of the patterns are unique. There are no consistent recurring patterns observed in the data in any of the panel blocks.

Another approach to check for stockpiling is to estimate the probability of observing a missing observation $\Pr(q_{it} = 0)$ as a function of past quantities in levels or logs. Under stockpiling behavior, one would expect a positive coefficient on last period's quantities. To allow households specific fixed effects, I use additional lags as instruments for lagged quantities. The model is run as a linear probability model where the second and first stages are respectively:

$$\Pr(q_{it} = 0) = \alpha_i + \rho q_{i,t-1} + \varepsilon_{it} \tag{B.2}$$

$$q_{i,t-1} = \gamma_i + \delta q_{i,t-1-m} + \eta_{it} \tag{B.3}$$

for some $m \in \{2, 3, \ldots, T-2\}$. It is possible to utilize additional moment conditions to identify this model as in Arellano and Bond (1991) However, I focus on the case with only one instrument and set m equal to four as this is consistent with the lag choices used in the main specification.

It is clear from table B.11 that lagged quantities do have significant correlation with the missing values, but in the opposite direction that would be explained by stockpiling. This effect remains when including contemporaneous expenditure, which is insignificant when controlling for household fixed effects and using the instruments as shown in tables B.12 and

Missing periods	07-10	08-11	09-12	10-13	11-14	12-15	13-16	14-17
0	0.4200	0.4307	0.4308	0.4477	0.4401	0.4539	0.4749	0.4869
1	0.2442	0.2440	0.2298	0.2257	0.2361	0.2275	0.2098	0.2193
2	0.1305	0.1203	0.1288	0.1306	0.1308	0.1114	0.1193	0.1172
3	0.0800	0.0710	0.0782	0.0689	0.0698	0.0790	0.0678	0.0579
4	0.0474	0.0527	0.0542	0.0499	0.0299	0.0551	0.0528	0.0414
>4	0.0779	0.0813	0.0782	0.0772	0.0931	0.0731	0.0754	0.0772

Table B.9: Number of missing time periods for each household as a percentage of the total.

num. households	07-10	08-11	09-12	10-13	11-14	12-15	13-16	14-17
1	307	272	276	266	272	267	256	213
2	13	17	14	17	14	10	11	9
3	8	11	9	9	9	5	9	15
4	8	3	9	8	10	13	8	5
5	5	6	6	7	9	9	3	7
6	5	6	5	1	7	3	3	4
7	7	6	2	3	3	2	3	0
8	2	1	1	3	0	2	1	1
9	1	0	0	0	1	1	1	1
10	2	0	0	2	1	0	1	0
11	0	0	0	0	1	0	0	0
12	0	0	2	0	0	0	0	0
13	1	0	0	0	0	0	0	0
14	0	1	0	0	0	0	0	0
16	0	1	0	0	0	0	0	0
total	359	324	324	316	327	312	296	255

Table B.10: The number of households belonging to a distinct pattern of missingness.

B.13. It may be the case that it becomes 'worth it' in some sense to report when quantities are larger. However, the effect is small, and more investigation is needed to tease out the exact mechanism.

The fixed effects absorb all the household specific characteristics in the previous regressions. However, it may be useful to determine if any of these observed attributes play a role in the missing purchases. I check this using another linear probability model:

$$\Pr(q_{it} = 0) = \alpha + x_i'\beta + \gamma_t + \varepsilon_{it} \tag{B.4}$$

where x_i are household specific characteristics and γ_t are time fixed effects. Table B.14 reports the joint significance tests for each demographic variable with p-values in parentheses.

None of the demographic variables is jointly significant in all panel blocks, but all are significant in at least one block at the 5% level. I would argue that these tests are inconclusive in the direction for or against MAR. It suggests there is some non-stochastic behavior in the

Panel Block		OLS	IV	OLS	IV
07-10	$\hat{\rho}$	-0.0195	-0.0166	-0.0241	-0.0253
		(0.0009)	(0.0013)	(0.0014)	(0.0068)
	Hausman Test		10.3338		0.0363
			(0.0013)		(0.8490)
08-11	$\hat{\rho}$	-0.0168	-0.0157	-0.0218	-0.0345
		(0.0009)	(0.0013)	(0.0015)	(0.0064)
	Hausman Test		1.4257		4.1612
			(0.2325)		(0.0414)
09-12	$\hat{\rho}$	-0.0167	-0.0156	-0.0207	-0.0362
		(0.0009)	(0.0013)	(0.0015)	(0.0073)
	Hausman Test		1.4502		4.7092
			(0.2285)		(0.0300)
10-13	$\hat{\rho}$	-0.0124	-0.0113	-0.0154	-0.0183
		(0.0009)	(0.0013)	(0.0015)	(0.0077)
	Hausman Test		1.8820		0.1535
			(0.1701)		(0.6952)
11-14	$\hat{\rho}$	-0.0151	-0.0134	-0.0166	-0.0221
		(0.0009)	(0.0012)	(0.0015)	(0.0069)
	Hausman Test		4.5848		0.6650
			0.0323		0.4148
12-15	$\hat{\rho}$	-0.0135	-0.0128	-0.0142	-0.0272
		(0.0009)	(0.0012)	(0.0015)	(0.0069)
	Hausman Test		0.7754		3.7060
			0.3785		0.0542
13-16	$\hat{\rho}$	-0.0122	-0.0099	-0.0152	-0.0159
		(0.0010)	(0.0013)	(0.0016)	(0.0070)
	Hausman Test		7.5257		0.0097
			0.0061		0.9215
14-17	$\hat{\rho}$	-0.0141	-0.0146	-0.0147	-0.0296
		(0.0009)	(0.0013)	(0.0016)	(0.0070)
	Hausman Test		0.2581		4.8667
			0.6114		0.0274
	Fixed Effects	False		True	

Table B.11: Results from estimating equation B.2. Pooled OLS and fixed effects models are reported with and without IV along with corresponding Hausman tests. Standard errors in parentheses for coefficient estimates and p-values for Hausman tests.

patterns of missingness, but the exact mechanism remains unclear.

Panel Block		OLS	IV	OLS	IV
07-10	$\hat{\rho}$	-0.0162	-0.0181	-0.0189	-0.0375
		(0.0008)	(0.0013)	(0.0013)	(0.0080)
	$\hat{\beta}_{exp}$	-0.0101	0.0004	-0.0286	0.0173
		(0.0007)	(0.0010)	(0.0012)	(0.0135)
	Hausman Test		209.6263		13.0831
			(0.0000)		(0.0014)
08-11	$\hat{\rho}$	-0.0137	-0.0156	-0.0169	-0.0387
		(0.0009)	(0.0012)	(0.0013)	(0.0073)
	$\hat{\beta}_{exp}$	-0.0086	0.0022	-0.0270	0.0306
		(0.0007)	(0.0010)	(0.0012)	(0.0126)
	Hausman Test		243.0150		24.0499
			(0.0000)		(0.0000)
09-12	$\hat{\rho}$	-0.0134	-0.0158	-0.0148	-0.0320
		(0.0009)	(0.0013)	(0.0014)	(0.0087)
	$\hat{\beta}_{exp}$	-0.0070	0.0029	-0.0242	0.0193
		(0.0007)	(0.0010)	(0.0012)	(0.0112)
	Hausman Test		191.7854		15.5963
			(0.0000)		(0.0004)
10-13	$\hat{\rho}$	-0.0096	-0.0130	-0.0098	-0.0326
		(0.0008)	(0.0012)	(0.0014)	(0.0101)
	$\hat{\beta}_{exp}$	-0.0073	0.0042	-0.0281	0.0197
		(0.0007)	(0.0010)	(0.0012)	(0.0101)
	Hausman Test		287.8669		22.7686
			(0.0000)		(0.0000)
	Fixed Effects	False		True	

Table B.12: Results from estimating equation B.2 adding the additional control total expenditure. Pooled OLS and fixed effects models are reported with and without IV along with corresponding Hausman tests. Standard errors in parentheses for coefficient estimates and p-values for Hausman tests.

Panel Block		OLS	IV	OLS	IV
11-14	$\hat{\rho}$	-0.0125	-0.0171	-0.0123	-0.0539
		(0.0008)	(0.0012)	(0.0014)	(0.0094)
	$\hat{\beta}_{exp}$	-0.0076	0.0045	-0.0287	0.0468
		(0.0007)	(0.0010)	(0.0012)	(0.0125)
	Hausman Test		319.2842		40.0402
			(0.0000)		(0.0000)
12-15	$\hat{\rho}$	-0.0113	-0.0131	-0.0102	-0.0149
		(0.0009)	(0.0012)	(0.0014)	(0.0089)
	$\hat{\beta}_{exp}$	-0.0073	0.0026	-0.0265	0.0218
		(0.0007)	(0.0010)	(0.0013)	(0.0122)
	Hausman Test		211.4071		18.1614
			(0.0000)		(0.0001)
13-16	$\hat{\rho}$	-0.0103	-0.0125	-0.0113	-0.0174
		(0.0009)	(0.0013)	(0.0015)	(0.0085)
	$\hat{\beta}_{exp}$	-0.0069	0.0040	-0.0272	0.0164
		(0.0008)	(0.0011)	(0.0012)	(0.0110)
	Hausman Test		219.1828		17.0904
			(0.0000)		(0.0002)
14-17	$\hat{\rho}$	-0.0114	-0.0141	-0.0112	-0.0159
		(0.0009)	(0.0013)	(0.0015)	(0.0083)
	$\hat{\beta}_{exp}$	-0.0078	0.0022	-0.0266	0.0092
		(0.0008)	(0.0011)	(0.0013)	(0.0115)
	Hausman Test		165.0210		10.6476
			(0.0000)		(0.0049)
	Fixed Effects	False		True	

Table B.13: Results from estimating equation B.2 adding the additional control total expenditure. Pooled OLS and fixed effects models are reported with and without IV along with corresponding Hausman tests. Standard errors in parentheses for coefficient estimates and p-values for Hausman tests.

Panel Years	07-10	08-11	09-12	10-13	11-14	12-15	13-16	14-17
Head of HH	1.2817	0.6077	0.8856	1.2201	10.6716	6.9937	3.5671	3.2077
Race	(0.7335)	(0.8947)	(0.8289)	(0.7482)	(0.0136)	(0.0721)	(0.3122)	(0.3607)
HH Income	23.0539	37.8186	44.9954	13.6896	16.4379	17.4067	17.8177	16.5554
	(0.1885)	(0.0041)	(0.0004)	(0.5492)	(0.3536)	(0.2951)	(0.2724)	(0.3461)
HH Composition	10.1446	9.5890	12.3486	11.5413	7.9888	8.6485	9.7176	5.3914
	(0.1187)	(0.1431)	(0.0546)	(0.0730)	(0.2389)	(0.1943)	(0.1371)	(0.4947)
Age of Children	23.0661	7.0022	9.9737	8.3762	24.9862	14.2617	5.6444	18.9357
	(0.0017)	(0.4287)	(0.1258)	(0.2118)	(0.0003)	(0.0268)	(0.5818)	(0.0043)
Male Head	9.1844	23.0395	5.1902	30.4200	6.3522	8.0613	9.0735	11.2868
HH Age	(0.2397)	(0.0017)	(0.5197)	(0.0001)	(0.3849)	(0.1529)	(0.1695)	(0.1266)
Female Head	20.9249	25.0065	21.0374	21.7656	12.5053	4.9862	10.0039	11.2077
HH Age	(0.0073)	(0.0016)	(0.0070)	(0.0097)	(0.1300)	(0.7591)	(0.1884)	(0.1902)
Male Head	9.0480	19.4187	5.9902	2.6564	7.2260	8.2144	18.5083	5.6416
HH Education	(0.1072)	(0.0016)	(0.3072)	(0.7528)	(0.2044)	(0.1448)	(0.0024)	(0.3427)
Female Head	2.9533	3.6780	1.1681	2.6606	1.8271	4.7591	9.0413	23.2008
HH Education	(0.7072)	(0.5966)	(0.9479)	(0.7521)	(0.8725)	(0.4460)	(0.0601)	(0.0003)

Table B.14: Joint tests of significance for household demographics on missing indicators from the model in B.4. p-values in parentheses.

B.4 Control Function: First Stage

In this section I report the first stage results for the control functions discussed in section 2.7.3. The first stage for total expenditure and prices are respectively:

$$y_{it} = \alpha_i + \rho y_{i,t-4} + X'_{it}\beta + \varepsilon_{it} \tag{B.5}$$

$$p_{it} = \gamma_i + \rho p_{i,t-4} + Z'_{it}\delta + \nu_{it} \tag{B.6}$$

While the weak instrument literature is quite extensive in the single endogenous variable case, it is somewhat sparse when there are multiple endogenous regressors. I utilize the test developed in Sanderson and Windmeijer (2016) which is a rigorous adjustment to a test proposed in Angrist and Pischke (2008). Table B.15 reports the results. All test statistics are well above the relevant critical values found in Stock and Yogo (2002).

Panel Years	07-10	08-11	09-12	10-13	11-14	12-15	13-16	14-17
Expenditure	62.6548	43.4053	56.3685	51.2558	46.7733	59.4386	67.6067	53.5080
Prices	1069.00	899.479	359.319	48.6041	23.4240	52.0349	50.4284	97.7365

Table B.15: First stage conditional F statistics as a test for instrument relevance.

Appendix C

This appendix focuses on a modification of the autoencoder in chapter 3 and its application to nonlinear factor models. The motivation for the sliced autoencoder is to simplify the structure of the autoencoder while leveraging new results in approximately sparse models to automatically learn nonlinear factor representations. In chapter 3 I demonstrated the power of autoencoders as tools for dimensionality reduction. However, these models are difficult to work with from a theoretical perspective, outside of the single linear layer, which spans the same space as principle components analysis (PCA) Baldi and Hornik (1989). In addition, one is typically forced to utilize a sophisticated procedure, the restricted boltzmann machine (RBM), to initialize the parameters as shown in the previous chapter. In contrast, the sliced autoencoder combines ideas from the panel neural network, found in the second chapter, with autoencoders to generate a common factor structure that has cross-sectional specific slopes.

This approach simplifies the structure of the autoencoder by removing most of the reconstruction, which is forced to be a linear function. However, the approximation power of the estimator is maintained through the embedding. The embedding is itself extended from the previous chapter by allowing for a panel structure.

C.1 Model

I consider n iid realizations from the random vector $\{y_{it}\}_{t=1}^{T}$. Each cross-sectional unit is sampled from:

$$y_i = g_0(F_t)\gamma_i + e_i, \qquad \mathbb{E}[e_i|x_i] = 0, \qquad \mathbb{E}[e_i^2|x_i] = \sigma^2(x_i) \tag{C.1}$$

where F_t, γ_i, and $g_0(\cdot)$ are unobserved.

AC.1.1 For each $t \in \{1, 2, \ldots, T\}$, the random vectors $\{z_{it}\}_{i=1}^{n} = \{y_{it}, x_{it}'\}_{i=1}^{n}$ are independent. In addition, $y_{it} \in \mathcal{Y} \subset \mathbb{R}$ and $x_{it} \in \mathcal{X} \subset \mathbb{R}^p$ where $\mathcal{X}$ and $\mathcal{Y}$ are compactly supported.

AC.1.2 For each $i \in \{1, 2, \ldots, n\}$ the vectors $\{z_{it}\}_{t=1}^{T}$ are stationary ϕ-mixing sequences with $\phi(k) = \phi_0\zeta^k$, $\zeta \in (0, 1)$, and $k > 0$ where:

$$\phi(k) \equiv \sup_{t\in\mathbb{N}} \sup_{\Pr(G)>0, G\in\{z\}_{-\infty}^{t}, H\in\{z\}_{t+k}^{\infty}} |\Pr(H|G) - \Pr(H)| \tag{C.2}$$

AC.1.3 The unknown index $g_0 \in \mathcal{W}_2^q(\mathcal{X})$ where $\mathcal{W}$ is a Sobelev space with q weak derivatives and has a Fourier representation:

$$g_0(x_{it}) = \int \exp(i\delta' x_{it}) d\sigma_g(\delta)$$

where σ_g is a complex measure on $\mathbb{R}^p$ satisfying:

$$\int \max\{|\delta|, 1\}^{q+1} d\,|\sigma_g|\,(\delta) < \infty$$

AC.1.4 Let $m = nT$. The parameters of the sieve space $\mathcal{G}_m$ satisfy the following bounds:

$$||\delta||_1 \leq \Delta_m, \quad \sum_{j=1}^{d_m} ||\delta_j||_1 \leq d_m \Delta_m$$

where $\Delta_m, d_m \to \infty$ slowly with m

Here I utilize the same assumptions from **A2.3.3** which largely follow from White (1990) and Chen and White (1999).

C.2 Estimation

I consider a penalized sieve least squares framework where the objective function is:

$$\mathbb{Q}_n(y_{it}, \delta, \gamma; \lambda) = \sum_{i=1}^{n} \sum_{t=1}^{T} \ell_{it} + \sum_{j=1}^{K} \lambda_j |\psi_j^{-1} \delta_j|_1 \tag{C.3}$$

$$\ell_{it} = \left(y_{it} - \sum_{j=1}^{K} s(y'_{(-i),t} \delta_j) \gamma_{ij} \right)^2 \tag{C.4}$$

There are two key elements to consider in this estimation framework. The first is the presence of $y_{(-i),t}$ as regressors. These are the contemporaneous outcomes for each cross-sectional unit other than i. The inclusion of these as regressors is justified by the independence assumption **AC.1.1** along with the existence of the common component $g_0(F_t)$.

The construction here differs from the autoencoder found in the previous by excluding the y_i as a regressor and restricting the reconstruction to be a linear function. The second key element are the penalty terms λ_j and ψ_j^{-1}. Each element λ_j governs the magnitude of $|\delta_j|_1$ while ψ_j^{-1} rescales the individual components.

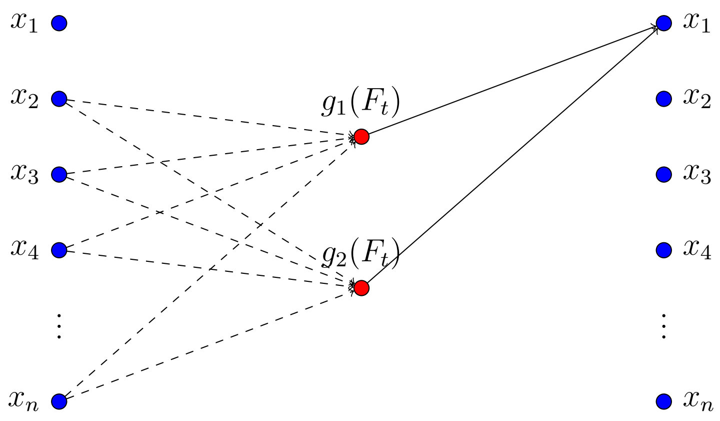


Figure C.1: The sliced autoencoder depicted for $i = 1$. The diagonal lines represent potential connections in the network which are fixed across $i \in [1, 2, \ldots, n]$. The solid lines are specific to i and represent γ.

C.2.1 Penalty Choice

One of the primary difficulties with deep learning and neural network estimation is the inclusion of hyperparameters. These models are difficult to optimize without additional complications, typically utilizing only first order methods and requiring many iterations. If one adds grid searches for hyperparameters the cost of estimation becomes prohibitively large. To alleviate the computational burden of such searches I consider a 'rigorous' result for the choice of λ_j and ψ_j based on ideas from Bickel, Ritov, Tsybakov, et al. (2009) and Belloni, Chernozhukov, and Wang (2011). The idea is to choose the penalty λ_j to be larger than the noise in estimation:

$$\lambda_j/n \geq c \max_k \frac{1}{n} \sum_{i=1}^{n} s_n(\delta_j) \tag{C.5}$$

for some $c > 1$ where $s_n(\delta_j)$ is the score function with respect to the index parameters δ_j. This result has an intuitive interpretation in the pure noise setting, i.e., $y_{it} = e_{it}$. In this case λ must be large enough to drive all the coefficients to zero. Interestingly in a pure noise setting, the score function for the neural network parameters δ, $s_n(\delta_j)$ is:

$$\lim_{\delta_j \to 0} s_n(\delta_j) = \gamma_{ij} x_i' \left(e_{it} + r_{it}\right) \tag{C.6}$$

where r_{it} is the approximation error of the estimator. This score is almost identical to the score one would work with in the Lasso model. The reason for this simplification is that the derivative of the activation function we choose $s(\cdot)$ with respect to the parameters δ approaches 1 as $\delta_j \to 0$. The primary complication here then is the presence of γ_{ij} and the approximation error r_{it}.

Automatic Penalty Choice

```
Set the desired tolerance ($tol$) for convergence.
Let $\theta_j^{(k)} = (\delta_j^{(k)}, \gamma_{ij}^{(k)})$ be the estimates at iteration $k$ for each $j \in \{1, 2, \ldots, K\}$.
while $\epsilon > tol$ do
    Given current values of $\delta_j^{(k)}$ estimate the score $s_{itj}^{(k)}$ and scale parameters $\psi_{itj}^{(k)}$
    Set $\lambda_j = 1.05 \max\left\{\frac{1}{nT}\sum_{i,t} s_{it,j}^{(k)}/\psi_{it,j}^{(k)}\right\}$
    Given an update rule $m(\cdot)$ the new value is: $\theta^{(k+1)} = \theta^{(k)} + m(\Delta_\ell^{(k)})$
    Threshold the new values of $\delta_j(k)$ using: $\delta_j = \delta_j^{(k+1)}\mathbf{1}\left\{|\delta_j^{(k+1)}|\psi_j > \lambda_j\right\}$
end while
```

C.3 Monte Carlo

In this section I use a simulation setting that is consistent with much of the literature including Bai and Ng (2002), Onatski (2010), Ahn and Horenstein (2013) and Onatski (2015). I deviate by adding an additional nonlinear component to the model and add sparsity to the factor loadings λ_i. The data follows the following process:

$$y_{it} = C_{it} + \sqrt{\theta} e_{it} \tag{C.7}$$

where C_{it} is independent from the stochastic error e_{it} and θ governs the inverse signal-to-noise ratio.

$$C_{it} = g(F_{t1}) + \sum_{j=1}^{r} \lambda_{ij} F_{tj}/\sqrt{r} \tag{C.8}$$

$$g(x_i) = \sin(3\pi F_{t1}/2)(1 + 18x^2[\text{sgn}(F_{t1}) + 1])^{-1} \tag{C.9}$$

$$e_{it} = \rho e_{i,t-1} + \nu_{it} + \sum_{j \neq 0, j=-J}^{J} \beta \nu_{i-j,t} \tag{C.10}$$

The common component is given a factor structure while the error term e_{it} is autoregressive of order one. In addition one can allow for cross-sectional correlation through the coefficient β and J, which determines the breadth of the correlation. Following Onatski (2015) let $\nu_{it} \overset{iid}{\sim} (0, \sigma_\nu^2)$ where $\sigma_\nu^2 = (1 - \rho^2)/(1 + 2J\beta^2)$ for $J = \min\{n/20, 10\}$.

C.3.1 Model A

In the first simulation I set $\beta = 0$ and $\rho = 0.5$. In this case all values of y_{-i} are valid regressors to identify $g_0(F_t)$. The true number of factors $r = 3$ and the distribution of e_{it}, ν_{it} are normal.

	PCA					Sliced Autoencoder				
n	r	RMSE	abs. Bias	rel. Bias	ise	k	RMSE	abs. Bias	rel. Bias	ISE
100	3	0.4766	0.1216	0.0027	0.2651	10	0.3587	0.0613	0.0067	0.3176
	4	0.2950	0.0157	0.0007	0.2860	11	0.3589	0.0618	0.0065	0.3194
	5	0.3508	0.0187	0.0008	0.3451	12	0.3578	0.0625	0.0074	0.3202
	6	0.3955	0.0207	0.0009	0.3910	13	0.3579	0.0621	0.0072	0.3212
	7	0.4335	0.0222	0.0011	0.4296	14	0.3575	0.0624	0.0057	0.3217
	8	0.4666	0.0237	0.0010	0.4632	15	0.3588	0.0637	0.0068	0.3229
150	3	0.4725	0.1379	0.0030	0.2492	10	0.3346	0.0560	0.0030	0.2947
	4	0.2695	0.0151	0.0004	0.2617	11	0.3311	0.0551	0.0030	0.2945
	5	0.3216	0.0176	0.0005	0.3164	12	0.3293	0.0553	0.0027	0.2941
	6	0.3635	0.0194	0.0005	0.3594	13	0.3295	0.0551	0.0030	0.2959
	7	0.3992	0.0210	0.0005	0.3956	14	0.3293	0.0547	0.0028	0.2975
	8	0.4305	0.0227	0.0006	0.4273	15	0.3292	0.0552	0.0029	0.2979

Table C.1: PCA vs Sliced autoencoder for $t = 200$.

C.3.2 Model B

In the second simulation I set $\beta = 0.2$ and $\rho = 0.5$. In this case values many values of y_{-i} are endogenous and would ideally be removed from the valid instrument set. This is a violation of **A1.3.1** and is useful to determine how much this matters in practice.

n	r	RMSE	abs. Bias	rel. Bias	ise	k	RMSE	abs. Bias	rel. Bias	ISE
100	3	0.5321	0.1218	0.0048	0.3550	10	0.4536	0.0598	0.0037	0.4276
	4	0.3877	0.0368	0.0017	0.3784	11	0.4584	0.0577	0.0037	0.4349
	5	0.4760	0.0373	0.0014	0.4696	12	0.4700	0.0569	0.0033	0.4492
	6	0.5408	0.0371	0.0015	0.5352	13	0.4778	0.0550	0.0036	0.4591
	7	0.5923	0.0373	0.0015	0.5873	14	0.4896	0.0543	0.0031	0.4726
	8	0.6347	0.0365	0.0015	0.6301	15	0.5006	0.0531	0.0032	0.4858
150	3	0.5452	0.1393	0.0032	0.3697	10	0.4609	0.0527	0.0018	0.4395
	4	0.3928	0.0387	0.0009	0.3840	11	0.4652	0.0521	0.0021	0.4454
	5	0.4856	0.0379	0.0011	0.4794	12	0.4717	0.0515	0.0019	0.4535
	6	0.5527	0.0377	0.0010	0.5473	13	0.4800	0.0497	0.0019	0.4636
	7	0.6059	0.0364	0.0010	0.6010	14	0.4891	0.0494	0.0016	0.4741
	8	0.6496	0.0357	0.0010	0.6450	15	0.5023	0.0487	0.0020	0.4890

Table C.2: PCA vs Sliced autoencoder for $t = 200$.

C.4 Conclusion

The sliced autoencoder has great potential as an extension to linear factor models. In simulations it is shown that the performance, measured in RMSE, is largely invariant to k and outperforms PCA across most specifications. However, the PCA estimator is not uniformly dominated in either simulation design as $r = 4$ is the best performing model. The optimal selection informed by the score criterion and continuous updating is also shown to be incredibly effective for practical implementation of this algorithm.