

UC Irvine

UC Irvine Electronic Theses and Dissertations

Title

Performance prediction of PEM fuel cell using artificial neural network machine learning

Permalink

<https://escholarship.org/uc/item/47t0j1pg>

Author

Tian, Pengjie

Publication Date

2020

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA,
IRVINE

Performance prediction of PEM fuel cell using artificial neural network machine learning

THESIS

Master of Science
in Mechanical and Aerospace engineering

by

Pengjie Tian

Thesis committee:

Professor Yun Wang, Chair

Professor Manuel Gamero-Castaño

Assistant Professor Penghui Cao

2020

TABLE OF CONTENTS

LIST OF FIGURES	iv
LIST OF TABLES	v
ACKNOWLEDGEMENTS	vi
ABSTRACT OF THE THESIS	vii
Chapter 1: Introduction	1
1.1 PEM fuel cell	1
1.2 PEM fuel cell performance	4
1.3 Machine learning	6
1.3.1 Introduction to machine learning	6
1.3.2 Artificial neural network	6
1.3.3 ANN study on PEM fuel cell	7
Chapter 2: Fuel cell performance and physics-based model	10
2.1 Effects of operating condition on fuel cell performance	10
2.1.1 Pressure	10
2.1.2 Temperature & relative humidity	11
2.1.3 Gas flow rate	11
2.2 Physics-based model	12
2.2.1 Electrical potential: Ethermo	12
2.2.2 Activation Loss: η_{act}	13
2.2.3 Ohmic loss	15
2.2.4 Concentration loss	16
2.3 Physics-based model prediction	18
Chapter 3: ANN method	23
3.1 Machine learning technique	23
3.1.1 The architecture of neural networks	24
3.1.2 A type of neuron: sigmoid neuron	25
3.1.3 Activation function	26
3.1.4 Cost function	28
3.1.5 Gradient descent	29
3.1.6 Back propagation algorithm	30
3.2 Data acquisition	34

3.3 Data standardization.....	35
3.4 Performance evaluation criterion	36
Chapter 4: Results and discussion.....	37
4.1 Effect of neuron number in hidden layer.....	37
4.2 Effect of training data size	39
4.3 Effect of activation function	42
4.4 Effect of random vs. ordered training data selection.....	45
4.5 Effect of selected training data set	46
Chapter 5: Conclusion.....	51
References.....	53
Appendix.....	57

LIST OF FIGURES

Figure 1 Typical fuel cell components [6].....	2
Figure 2 A schematic of a PEM fuel cell component thickness, and physical processes [7]	2
Figure 3 SEM images of catalyst layers: 20%Pt/C, I/C ratio of 0.5 (left) and 40%Pt/C, I/C ratio of 0.5 (center) and 0.33 (right). Each is measured at two different locations. [9]	3
Figure 4 A typical PEM fuel cell polarization curve. [14]	5
Figure 5 Relationship between j and η behaved as Butler-Volmer equation [12].....	14
Figure 6 3D fuel cell computational domain for database construction [19]	18
Figure 7 The 1-D prediction of liquid water profile across the anode and cathode GDLs, and comparison with the X-ray imaging data [41]	19
Figure 8 (a) Comparison with the high-resolution neutron imaging data, 2-D model prediction (with/without smoothing) using a uniform GDL property, and 2-D model prediction using a spatially varying GDL property (b) The varying GDL property in the present study. [41]	20
Figure 9 Polarization curve for different channel design [46].....	22
Figure 10 Flow chart for backpropagation and ANN algorithm.....	23
Figure 11 A typical architecture of a MLPs [48].....	24
Figure 12 Typical sigmoid neuron.....	25
Figure 13 (a) Plot of sigmoid function, (b) Plot of a step function [48]	26
Figure 14 (a) Simple linear function (b) Logistic sigmoid function (c) Hyperbolic tangent function (d) Rectified Linear Units (ReLu) [50].....	28
Figure 15 Schematic of a typical hidden layer.....	31
Figure 16 Architecture for our ANN	34
Figure 17 Predicted current density vs. actual current density & absolute percentage error [20 neuron configuration].....	38
Figure 18 General prediction of current density [training data number variation].....	39
Figure 19 Plot of current density vs. power output curve [training number variation]	40
Figure 20 Plot of polarization curve [training number variation].....	41
Figure 21 Linear fitting between predicted current density and actual current density [1000 training data]	41
Figure 22 General prediction of current density [activation function variation]	43
Figure 23 Plot of polarization curve [activation function variation]	44
Figure 24 Plot of current density vs. power output curve [activation function variation].....	44
Figure 25 Predicted current density vs. actual current density & absolute percentage error plot	45
Figure 26 Predicted current density vs. actual current density & absolute percentage error plot [randomly selected training data].....	46
Figure 27 Polarization curve plot for predicted current density and actual current density [with/without $RHc = 0.25$ data point]	47
Figure 28 Polarization curve plot with/without fake data point	49

LIST OF TABLES

Table 1 The governing equations of multi-physics fuel cell model.....	21
Table 2 Characteristics for common activation functions	27
Table 3 Table of sample data	35
Table 4 Specifications for input and output parameters	35
Table 5 Prediction accuracy using different number of neurons	38
Table 6 Table of prediction accuracy using different activation functions	43
Table 7 Linear fitting results for simulation trails with/without $RHc = 0.25$ data point.....	48
Table 8 Specifications for true & fake added into training data set.....	49
Table 9 Linear fitting results for simulation trails with fake/true data point	50

ACKNOWLEDGEMENTS

I would like to express the deepest appreciation to my committee chair, professor Yun Wang, who shows great genius in engineering teaching and research. He inspired me with this research topic: artificial neural network application on PEM fuel cell. He continually supports me with novel ideas and guidance, when I encountered troubles in my research. I also appreciate the time he spent discussing about the researching progress, correcting my mistakes in my thesis, giving the lesson about engineering professional ethics.

I would like to thank my other committee members, professor Manuel Gamero-Castaño, and professor Penghui Cao. They guided me producing a high standard professional engineering master thesis. In addition, their solid knowledge in engineering field helped me with my analysis of researching results.

I also appreciate my family's support throughout my thesis writing period. They encourage me when I fell into downfall. Without their financial support, I wouldn't be able to finish my master thesis in University of California, Irvine.

Finally, I would like to give credit to one of my friends, Dongheng Jing. He is a master student in Sibley School of Mechanical and Aerospace Engineering, Cornell University. He provided me with a lot of assistances in developing and diagnosing my artificial neural network code.

ABSTRACT OF THE THESIS

Performance prediction of PEM fuel cell using artificial neural network machine learning

by

Pengjie Tian

Master of Mechanical Engineering

University of California, Irvine, 2020

Professor Yun Wang, Chair

Proton exchange membrane (PEM) fuel cell is a promising candidate as a renewable energy source in the future. It is an electrochemical device that directly convert chemical energy in hydrogen fuel to electric energy with water as the only byproduct. In operation, multiple physical processes occur, including electrochemical reaction, heat and mass transfer, liquid water formation and vaporization, which govern fuel cell operation and performance. Various physical models have been developed, which couple the multi-physics with electrochemical reaction kinetics, to predict fuel cell performance for design and control purpose. Machine learning and data-driven approach have received a growing attention in recent years. This study investigates the application of artificial neural network (ANN) machine learning to predict PEM fuel cell performance. A novel four-layered backpropagation ANN is developed to achieve reasonable accurate prediction of fuel cell performance using the large amount ($> 1,500$) of performance data (I-V curves) obtained from a validated three-dimensional (3D) physical model for various operating conditions, including temperature, anode and cathode relative humidity. Various ANN parameters, including the number of neurons in each hidden layer, the sizes of training data, the activation functions, selection of training data set, are investigated to assess their impacts on prediction accuracy. Simulations show 20 neurons in each hidden layer fit best for our database.

The hyperbolic tangent function as the ANN's activation function in hidden layer performs the best in terms of prediction accuracy because of its high gradient value and smooth gradient profile. For data set selection, randomly selected training data show a good ANN prediction. For selected datasets, the ANN shows capabilities of predicting the I-V curves using incomplete input data information and filtering out noise signals or outliers in the input data set. These results can provide a valuable guidance of using the ANN to help PEM fuel cell experiment in which data for training are limited in most case.

Chapter 1: Introduction

1.1 PEM fuel cell

Seeking alternative automobile power source becomes an urgent topic in recent years as conventional internal combustion engine fossil fuel causing severe air pollution. Proton exchange membrane fuel cell (PEMFC) is an energy conversion device which convert chemical potential energy into electrical energy [1]. As alternative power source, PEMFC exhibits high efficiency, energy density, silent operation, reduction in pollution emission, low temperature operation, etc. [2] In recent years, PEMFC have been applied into many industrial fields, for example, fuel cell electric vehicles, forklift trucks, buses, small portable devices, etc. According to a fuel cell industry review report, a total of 39700 unit of PEMFC shipments is achieved in 2018, and 609.0 megawatts electricity was generated by PEMFC in 2018 [3]. Toyota launched its first fuel cell vehicle, Mirai, in 2017, and achieved total of 4000 shipment so far. Over 30 gas stations are being constructed and operated across U.S and over 60 in Germany. [4]

Main components for a typical PEMFC are membrane electrode assembly, gasket, bipolar plate (BPs) and current collective plates [1]. Figure 1 shows the components of a typical single PEM fuel cell. End plates provide robust mechanical support for the whole cell while uniformly distributing compressive pressure to all the components. What's more, the compression stress allows a tighter contact between each component of a PEMFC so that reduce the contact resistance. [5]

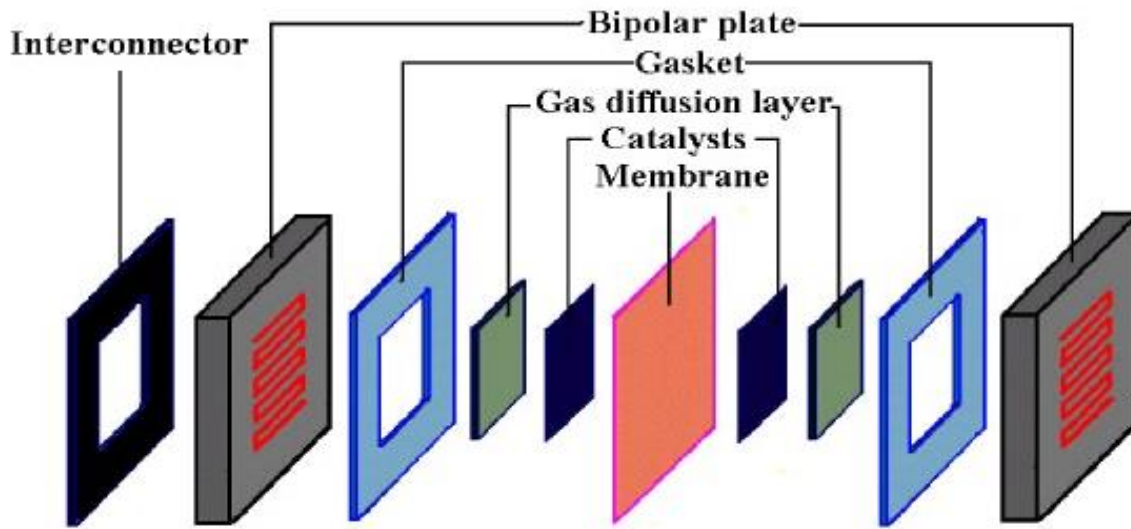


Figure 1 Typical fuel cell components [6]

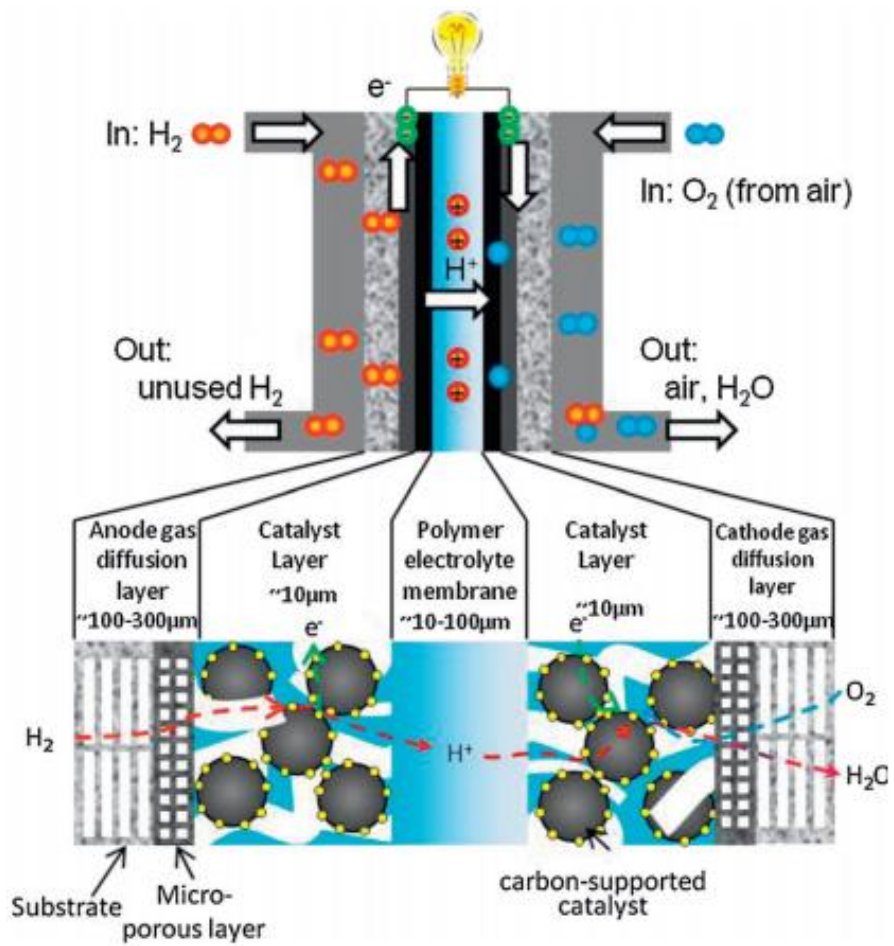


Figure 2 A schematic of a PEM fuel cell component thickness, and physical processes [7]

Flow channels usually grooved in the BPs that provide pass way for reactant gases, which help with the uniform distribution of reactants to the reactive area. Parallel, serpentine, pin-type flow channels are conventional channel design in early research. In addition, the high thermal conductive property of base material allows them to remove heat efficiently so that keep fuel cell under a proper operating temperature. Bipolar plates also play an important role at the fuel cell stack level. Plates can separate each single cell while remaining the electrical connection in series configuration. [8]

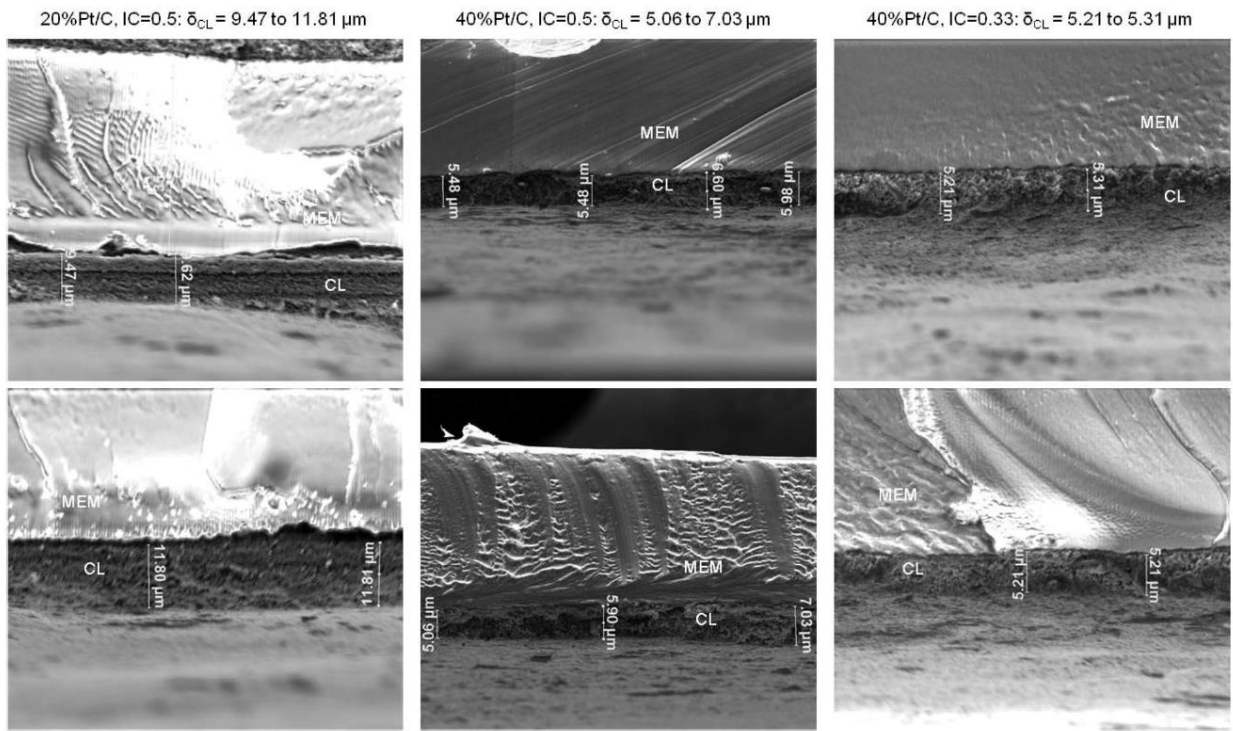
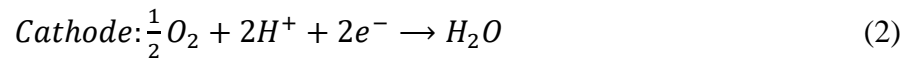


Figure 3 SEM images of catalyst layers: 20%Pt/C, I/C ratio of 0.5 (left) and 40%Pt/C, I/C ratio of 0.5 (center) and 0.33 (right). Each is measured at two different locations. [9]

The membrane electrode assembly (MEA) can be considered as the “heart” of a PEMFC, which consists of proton exchange membrane, catalyst layer (CL), gas diffusion layer (GDL). It is the location where all the electrochemical reactions take place. Proton exchange membrane separates the anode hydrogen and cathode oxygen and it only permit hydrogen ions to pass through. Currently, the most prominent membrane, Nafion, was developed based on

perfluorosulfonic acid. Polytetrafluoroethylene (PTFE) serves as its backbone structure to provide mechanical strength, while sulfonic acid groups offer charge transport sites [1]. Platinum nanoparticles are dispersed on the high-surface area carbon support to form conventional catalyst layer [10]. Commonly, the porous Pt/C catalyst particles mixed with electrolyte solution, known as ionomer, facilitates the transport of reactant gas, discharge the product of water and provide path for electrical conduction [11]. Figure 3 shows SEM images of catalyst layers with different Pt/C mass percentage loading, CL thickness, and ionomer to carbon black ratio. GDL distribute the reactant gas to the whole reactive area and provide robust mechanical support to protect catalyst layer [12].

According to Figure 2 the electrolyte prevents direct reaction between fuels, hydrogen and oxygen (air), while it keeps the ability allowing proton (hydrogen ion: H^+) to travel through the proton exchange membrane. Hydrogen is introduced and ionized at the anode side, while one of the products, hydrogen ions, travel through the membrane and reach to cathode side of the PEMFC. Hydrogen ions reduce oxygen to form water at the cathode side. [13]



1.2 PEM fuel cell performance

The performance of a PEM fuel cell is usually indicated by polarization (I-V) curve, which displays the voltage output for a given current density. Mainly three distinct regions consist a polarization curve, respectively, low current density region, moderate current density region, and high current density region. Figure 4 shows a typical I-V curve. At the initiation stage of a PEM fuel cell (low current region), the sharp voltage drop is mainly caused by the

activation loss. At moderate current density region, the fuel cell voltage exhibits a linearly decreasing behavior, which is resulted by the domination of ohmic loss. While, the voltage depart from a linear drop because of the more significant concentration loss occurs in high current density region. These three major losses will be discussed in detail in latter chapter.

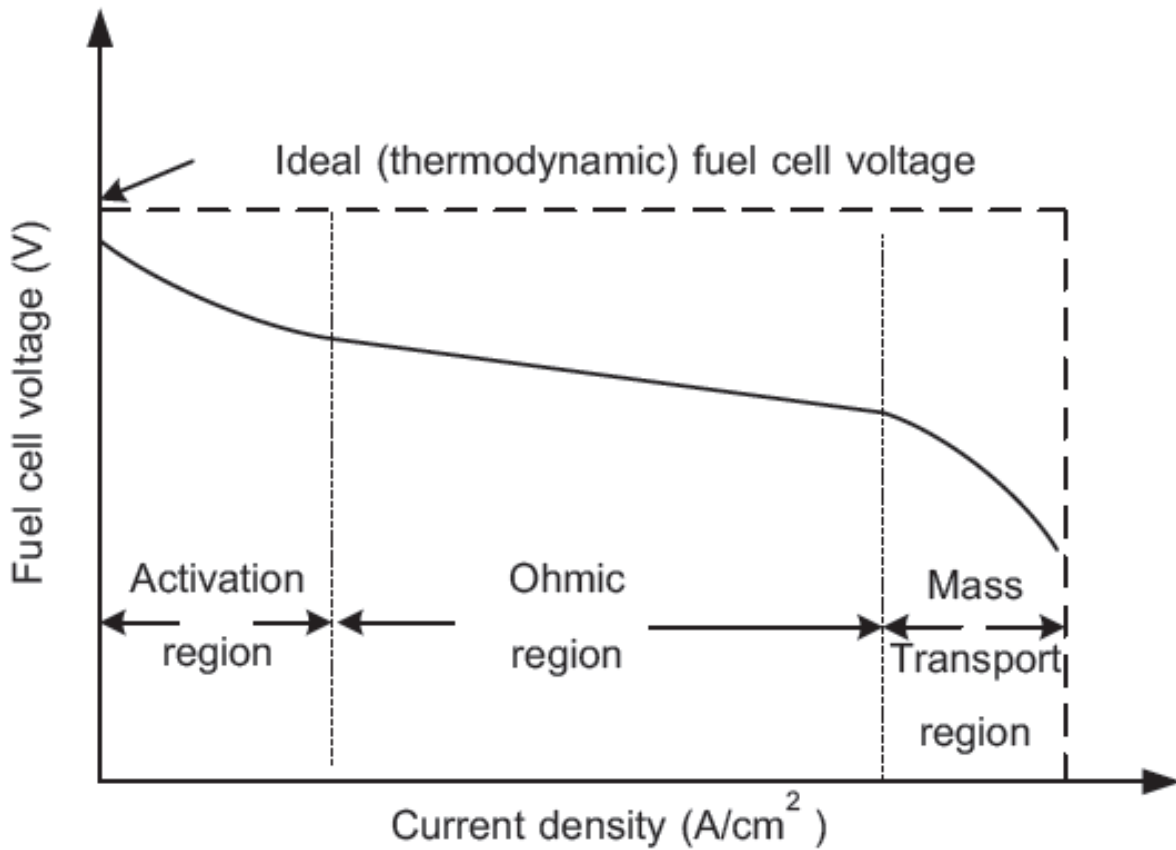


Figure 4 A typical PEM fuel cell polarization curve. [14]

The performance of a PEMFC can be affected by various factors. Operating condition including operating temperature, reactant inlet/out relative humidity, pressurization condition, reactant flow rates, can influence the voltage output value. In addition, materials used in a PEMFC such as the catalyst layer components (Pt/C wt%, I/C wt%), the type of Nafion, can affect the performance as well. In this study, we focus on the factor of operating condition.

1.3 Machine learning

1.3.1 Introduction to machine learning

Recently, applying machine learning technique on the study of PEM fuel cell performance has gained a lot attention. It provides a way for systems automatically learn and improve from the previous experience, large amount of data [15]. In other words, Machine learning algorithm allows computer to build a mathematical model automatically based on the training data. Machine learning algorithm has been implemented in many practical areas, including, web searching, spam filters, stock trading, etc. [16] Machine learning can be classified into three main types, supervised learning, unsupervised learning, reinforcement learning. Supervised learning is most popular type in PEM fuel cell research field. For a given input signal, if the training data contains the explicit correct output data, then it is in the supervised learning [17].

1.3.2 Artificial neural network

Many researchers developed both analytical and numerical models for prediction of PEMFC performance in recent years. Wang et al developed a two-phase flow analytical model based on two-phase Darcy's law and the M^2 formalism. Analytical solutions of liquid water saturation and species concentration are obtained [18]. Zhang et al built a 3D multi-phase model of PEM fuel cell and validated the model using PEM fuel cell experimental data in terms of I-V curves and spatial variation of current density distribution from two different resources. His work showed a decreasing reaction rate from inlet to outlet for 100% RH operation and the peak local current shifts downstream for dry operation [19]. A three-dimensional agglomerate model for cathode catalyst model was built by Das et al. their numerical model successfully assesses the

activation polarization in which calculated the cathode activation overpotentials, membrane and solid phase current densities, investigated the oxygen transport process in cathode catalyst layer. [20] Those previous studies contributed significant progress in investigating PEMFC fuel cell performance under different operating conditions, exploring mechanism in thermal management, species transport, two-phase flow, etc. During a PEMFC operation, a large number of parameters need to be considered, because too many physicochemical reactions entangle together, which make it especially hard to evaluate the performance. What's more, complicated numerical models trying to simulate all the inner reactions that require a lot computational power to solve those coupled differential equations. Further, two-phase phenomena, droplet dynamics and cold start are complex in PEM fuel cells [21] [22] [23] [24] [25] [26]. Though many models claim to consider the two-phase flow, it is challenging to fully consider their impacts due to two-phase instability, random surface heterogeneity and dynamics.

Artificial neural network is a typical machine learning algorithm that simulates the biological systems composed of “neurons” [27]. Instead of taking every physicochemical parameter into account, the artificial neural network focus on several key parameters (weight values) to seek for estimate solutions for the model demonstrating performance. ANN shows ability to solve for problem with incomplete data set, incomplete information provided, and complex phenomena situations [28].

1.3.3 ANN study on PEM fuel cell

Chavez-Ramirez et al, modeled a 5(kW) PEMFC stack with 7 inputs and 2 outputs, namely voltage output and cathode outlet temperature. Based on backpropagation (BP) learning algorithm, their ANN model successfully predicted voltage and cathode outlet temperature, with error percentage of 9.4%, and 5.6% respectively [29]. A two-layered backpropagation neural

network was built to predict a commercial PEMFC system's (1.2 kW NexaTM) output voltage and current by Saengrungs et al. The model learning speed and accuracy were evaluated by how many epochs is needed to reach designed error goal of 0.001. Their results show the two-layered BP neural network can smoothen the prediction of fuel cell voltage and current output, while the BP network with only one hidden can provide a good prediction as well. [30] A feedforward backpropagation neural network created by Bhagavatula et al perform a good prediction for the cell output voltage, and cell temperatures, humidification temperatures, H_2/Air flow rates and current density were used as inputs for their ANN. The simulation results indicated model improvements can be observed with the increasing number of neurons in each hidden layer. In addition, increasing the input parameters can significantly reduce the number of neurons required in each hidden layer, and improve the ANN model. They used trained ANN successfully predicted the performance of a fuel cell at higher temperature. [31] Lee et al trained their ANN model with data obtained from a 300 cm² single cell in H_2/Air operation using Nafion 115 and Nafion 1135 membrane electrolytes. The architecture of their ANN is a two-layered feed forward backpropagation neural network with hyperbolic tanh function as activation and 30 neurons in each layer, while 10 independent input variables are given to train this ANN. The simulated results are compared with experimental data, which exhibit a good agreement with error percentage less than 1%. [32] Seyhan produced a parametric study on the PEMFC with different wave amplitude, different flow rates and different cell operating temperature, and used those parameters as inputs for their ANN to predict the performance of a PEMFC with wavy serpentine and conventional serpentine flow channel. The linear regression model for their experimental data and predicted data shows a great agreement, with regression value R for both training and testing data equal 0.99975 and 0.99966 respectively [33]. For regression simulation,

Wang et al suggested that the ANN and SVM are most common algorithm for data-driven models, which perform better in small database in a range of 100 to 1000 data points [34].

Chapter 2: Fuel cell performance and physics-based model

PEM fuel cell performance is measured by the I-V curve or output voltages as a function of operating current density, which is determined by the transportation and electrochemical processes in operation. Physics-based models have been developed to describe the processes and predict fuel cell performance.

2.1 Effects of operating condition on fuel cell performance

2.1.1 Pressure

According to Nernst equation, a PEMFC operated under a pressurized condition, a significant performance enhancement can be observed. For a PEMFC under a normal operating condition: the overall reaction: $H_2 + \frac{1}{2}O_2 \rightarrow H_2O$. Therefore, apply the overall reaction into Nernst equation:

$$E = E_0 - \frac{RT}{2F} \ln \left(\frac{1}{P_{H_2} * P_{O_2}^{\frac{1}{2}}} \right) \quad (4)$$

where E_0 is standard-state reversible voltage, R is ideal gas constant, T is operating temperature in (Kelvins), F is Faraday's constant, P is partial pressure for hydrogen and oxygen. It is obvious that pressurizing the fuel cell which increase the reactant gas partial pressure can increase the reversible voltage. [12] Lu et al developed a 3D CFD model using ANSYS FLUENT for a single serpentine channel PEMFC. They discovered the apply back pressure can increase the relative humidity at the cathode channel result in an increasing of membrane water content. [35]

2.1.2 Temperature & relative humidity

The relative humidity of reactant gas is depended on the fuel cell operating temperature.

[36]

$$RH(\%) = \frac{P_{H_2O}}{P_{H_2O}^0} * 100\% \quad (5)$$

Where $P_{H_2O}^0$ is a function of temperature, which can be expressed as [36],

$$P_{H_2O}^0 = 6.02724 * 10^{-3} + 4.38484 * 10^{-4}(T - 273.15) + 1.39844 * 10^{-5}(T - 273.15)^2 + 2.71166 * 10^{-7}(T - 273.15)^3 + 2.57731 * 10^{-9}(T - 273.15)^4 + 2.82254 * 10^{-11}(T - 273.15)^5 \quad (6)$$

According to Nernst equation, the increasing in temperature will reduce the reversible voltage. In addition, increasing in temperature can accelerate electrochemical reaction rate, bring down the activation energy needed for initializing fuel cell. Kim et al performed testing on a fuel cell stack of 10 cm^2 active area and 10 cells to investigate the effects of humidity and temperature on performance of a PEMFC stack. As a result, the cathode reactant humidity has more influence on the overall performance. Increasing in cathode humidity can strongly increase the performance of PEMFC. At the stack level, they claimed the humidity has more influence than that of temperature. [37] On the contrary, introducing excessive water (high humidity) content may cause non-uniform temperature distribution on membrane, water flooding in cathode channel, thermal management issue, etc [18].

2.1.3 Gas flow rate

Providing enough reactant gas is necessary to keep fuel cell operating under expected performance. Guveligoglu et al built an analytical model based on a new algorithm, which

explores the cell performance under various hydrogen/air flow rates and humidification level. The results showed the high hydrogen flow rates and low air flow rate at high humidification level can increase the overall performance for a PEMFC. High water concentration in membrane result in a high membrane conductivity, while high hydrogen flow rate can ensure enough reactant supply. [38]

2.2 Physics-based model

The real voltage output for a PEMFC can be obtained by firstly starting from the thermodynamic voltage, then subtracting three main voltage losses, respectively, activation loss, ohmic loss, and concentration loss.

$$V_{real} = E_{thermo} - \eta_{act} - \eta_{ohmic} - \eta_{conc} \quad (7)$$

2.2.1 Electrical potential: E_{thermo}

Gibbs free energy characterizes the work potential for a typical reaction. The Gibbs free energy is defined as:

$$G = H - TS \text{ or } \Delta g = \Delta h - T\Delta s \text{ (differential form and holding a constant temperature)} \quad (8)$$

The maximum electrical work that can be extract from a constant-temperature and constant-pressure process is given by:

$$W_{elec} = -\Delta g \quad (9)$$

Based on the equation above, the voltage (electrical potential) can be expressed by the electrical work that required to move a charge Q in a electrical potential field E,

$$W_{elec} = EQ = E * (nF) \quad (10)$$

Combine with equation 2.2 and 2.3, we can get

$$\Delta g = -nFE \text{ or } E = -\frac{\Delta g}{nF} \quad (11)$$

Thus, for a typical PEMFC reaction,



The reversible voltage generated under standard condition:

$$E^0 = -\frac{\Delta g}{nF} = 1.23 \text{ (V)} \quad (13)$$

Where, Δg is standard-state free-energy change for the reaction, E^0 is standard-state reversible voltage.

However, in most cases, the PEMFCs commonly are operated under Non-standard-state conditions. In those cases, the Nernst equation is applied to predict the thermodynamic reversible cell voltage, in which the effects of temperature and operating pressure (species concentration) are considered,

$$E = E^0 - \frac{RT}{nF} \ln \left(\frac{1}{P_{H_2} * P_{O_2}^{\frac{1}{2}}} \right) \quad (14)$$

2.2.2 Activation Loss: η_{act}

At low current density region, activation of reaction dominates at this stage. The activation loss (voltage loss) is labeled as the voltage loss due to overcome the activation barrier associated with the electrochemical reaction. [12] The equation, known as Butler-Volmer equation, is used to describe the relationship between current density and overpotential,

$$j = j_0 \left(e^{\frac{\alpha n F \eta}{RT}} - e^{-\frac{(1-\alpha) n F \eta}{RT}} \right) \quad (15)$$

Where, j_0 is exchange current density, α is transfer coefficient, η is activation potential.

Basically, it states the current density grows in an exponential behavior with activation overpotential. If the electrochemical reaction with a higher current density, the fuel cell must experience a higher overpotential.

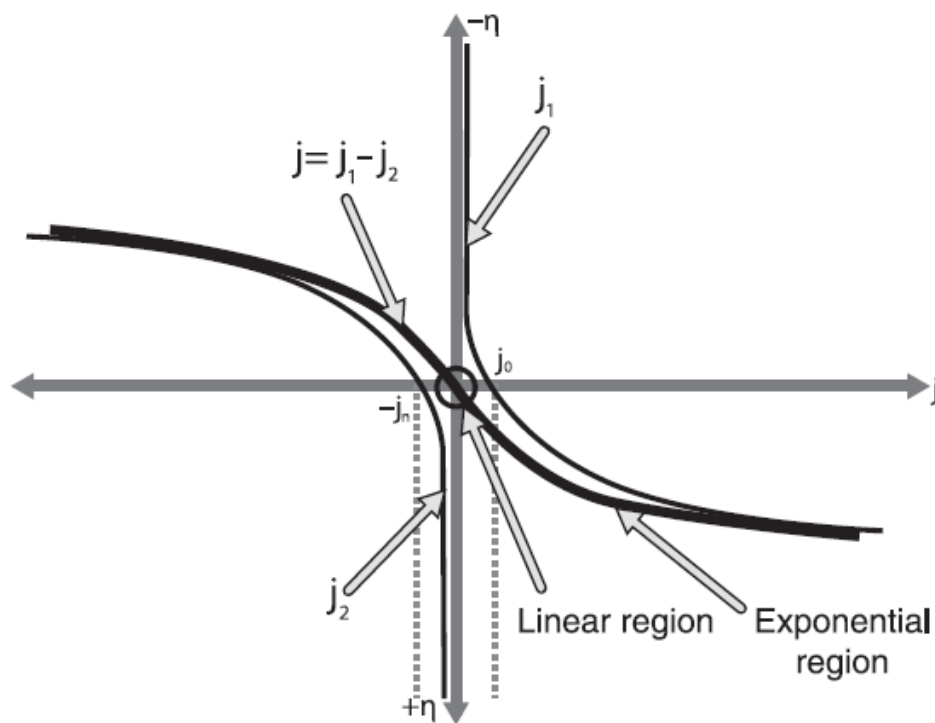


Figure 5 Relationship between j and η behaved as Butler-Volmer equation [12]

Figure 5 shows plot of activation overpotential with current density at low current density level. With the simplification of Butler-Volmer equation, two distinct regions can be observed on this figure. At extremely low current density region ($\eta_{act} < 15mV$), the equation can be simplified into,

$$j = j_0 * \frac{nF\eta_{act}}{RT} \quad (16)$$

A linear relationship between current density and activation overpotential can be observed in this region. At relatively high current density region ($\eta_{act} > 50 - 100 \text{ mV}$), the second exponential term can be neglect from the original Butler-Volmer equation. Then it becomes,

$$j = j_0 e^{\frac{\alpha n F \eta_{act}}{RT}} \quad (17)$$

Expressed in terms of η_{act} ,

$$\eta_{act} = -\frac{RT}{\alpha n F} \ln(j_0) + \frac{RT}{\alpha n F} \ln(j) \text{ or } \eta_{act} = a + b * \log(j) \text{ (Tafel equation)} \quad (18)$$

For fuel cells, when under operation, large current density with high power output is desired. Therefore, $\eta_{act} = -\frac{RT}{\alpha n F} \ln(j_0) + \frac{RT}{\alpha n F} \ln(j)$ is considered in computing the real fuel cell voltage output.

2.2.3 Ohmic loss

Charge transport process involves in PEMFC operation. Electrons and ions are two main species been transported in the PEMFC reaction. In actual situation, because ions are more difficult to be transported than that of electrons, therefore, the resistance of electron charge transport usually dominates the ohmic resistance.

Equation for ohmic loss:

$$R = \frac{L}{A\sigma} \quad (19)$$

Where L is conductor length, A is cross-section area, σ is conductive resistivity. [39]

$$\eta_{ohmic} = i * R_{ohmic} = i * (R_{elec} + R_{ion} + R_{contact}) \quad (20)$$

When research group investigate this issue, a resistance term named area-specific resistance (ASR) is commonly introduced to normalize the effect of ohmic resistance with area. A simplified expression for ohmic loss can be written as,

$$\eta_{ohmic} = j * (ASR_{ohmic}) \quad (21)$$

Where ASR is area-specific resistance, j is operating current density.

According to ohmic loss equation, it can be observed a linearly relationship between ohmic loss and current density. In order to increase performance of PEMFC, many researchers investigated possible ways to bring down area-specific resistance in many aspects, such as reducing electrolyte thickness, reducing contact resistance by applying various compression stress.

2.2.4 Concentration loss

Species concentration influence performance of a PEMFC in two ways. Firstly, according to Nernst equation, the reactant and product concentration on the active area affect the reversible voltage output. Secondly, species concentration can affect the reaction rate as well. [12]

For the Nernst thermodynamic reversible voltage output, it is determined by the species concentration at reactive area not the bulk concentration at inlets. The reactant concentration depletion at catalyst layer many cause a voltage loss, $\eta_{conc,Nernst}$.

$$\eta_{conc,Nernst} = \frac{RT}{nF} \ln \left(\frac{j_L}{j_L - j} \right) \quad (22)$$

Where, j_L is the limiting current characterize the condition when reactant concentration at catalyst layer is depleted to zero, $c_{R,catalyst} = 0$. From equation (22), it can be observed, as

current density approaches the limiting current density, the concentration loss will increase significantly.

The electrochemical reaction rate is affected by species concentration as well. Starting from Butler-Volmer (BV) equation, as we concentrated at high current density region, the second part of BV equation can be dropped out, which becomes to,

$$j = j_0^0 \left(\frac{c_R}{c_R^0} e^{\frac{\alpha n F \eta}{RT}} \right) \quad (23)$$

Here, the ratio between catalyst layer reactant concentration and bulk reactant concentration: $\frac{c_R}{c_R^0}$ need to be considered. In terms of overpotential,

$$\eta_{conc} = \frac{RT}{\alpha n F} \ln \left(\frac{j c_R^0}{j_0^0 c_R} \right) \quad (24)$$

Substitute the term $\frac{c_R^0}{c_R} = \frac{j_L}{j_L - j}$. Eventually, we can get,

$$\eta_{conc, BV} = \frac{RT}{\alpha n F} \ln \left(\frac{j_L}{j_L - j} \right) \quad (25)$$

Combining two part of concentration loss together,

$$\eta_{conc} = \left(\frac{RT}{nF} \right) \left(1 + \frac{1}{\alpha} \right) \ln \left(\frac{j_L}{j_L - j} \right) \quad (26)$$

The real fuel cell voltage output can be obtained by putting all the parts previous discussed together. A typical PEMFC I-V curve as shown in Figure 4 A typical PEM fuel cell polarization curve. , at the initiation stage (low current density region), the curve shows a nonlinear behavior similar as BV equation. While, at medium current density region, the ohmic loss (loss due to charge transport), contributes most overpotential to the overall voltage decreasing, and a linear relationship between current density and voltage output can be seen. The

concentration loss dominant at high current density region, mainly, because of the depletion of reactant concentration at catalyst layer.

For more details about the physics-based model, it can be found in the Ref [12].

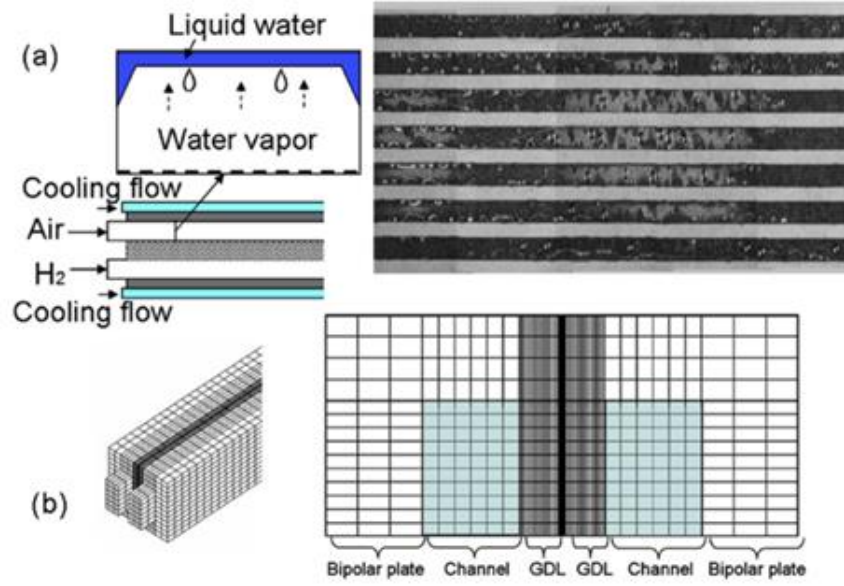


Figure 6 3D fuel cell computational domain for database construction [19]

2.3 Physics-based model prediction

To construct a database for ANN, a 3D multi-physics model of PEM fuel cells was adopted. The 3D fuel cell computational domain is shown in Figure 6, which contains the BP channel, GDL, catalyst layer and membrane. The governing equations are given in Table 1 and details regarding the catalyst layer physics can be found in ref [40]. The model was also validated in detail and thus is reliable to generate the database for ANN. For example, Wang et al investigated the through-plane water profiles with spatially varying properties of gas diffusion layers and their multi-dimensional two-phase model of PEM fuel cell was validated by their

experimental data [41]. Figure 7 shows the model predicted 1-D liquid water distribution profile which is very close to the experimental data probed by X-ray imaging.

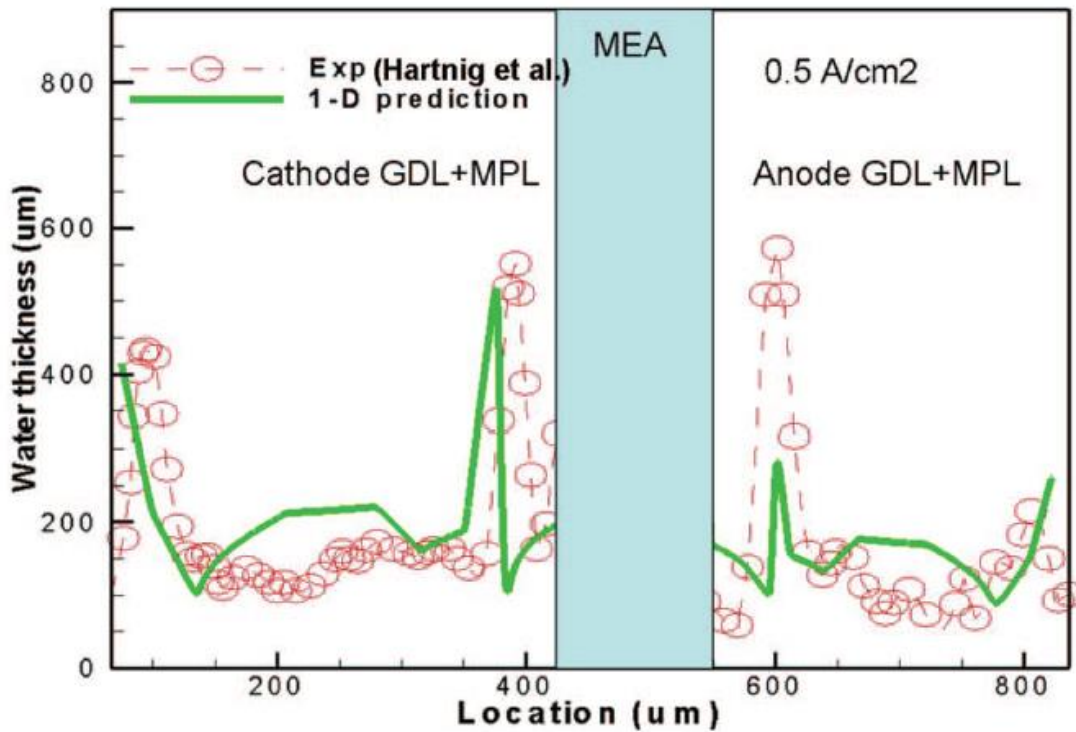


Figure 7 The 1-D prediction of liquid water profile across the anode and cathode GDLs, and comparison with the X-ray imaging data [41]

Figure 8 shows their results of 2-D model prediction based on a uniform GDL property in comparison with neutron imaging data. It can be observed that the model prediction agrees the experimental data well. In addition, the prediction results of a smoothed model by applying Gaussian smoothing technique and a model considering spatially varying GDL property are plotted in Figure 8 as well.

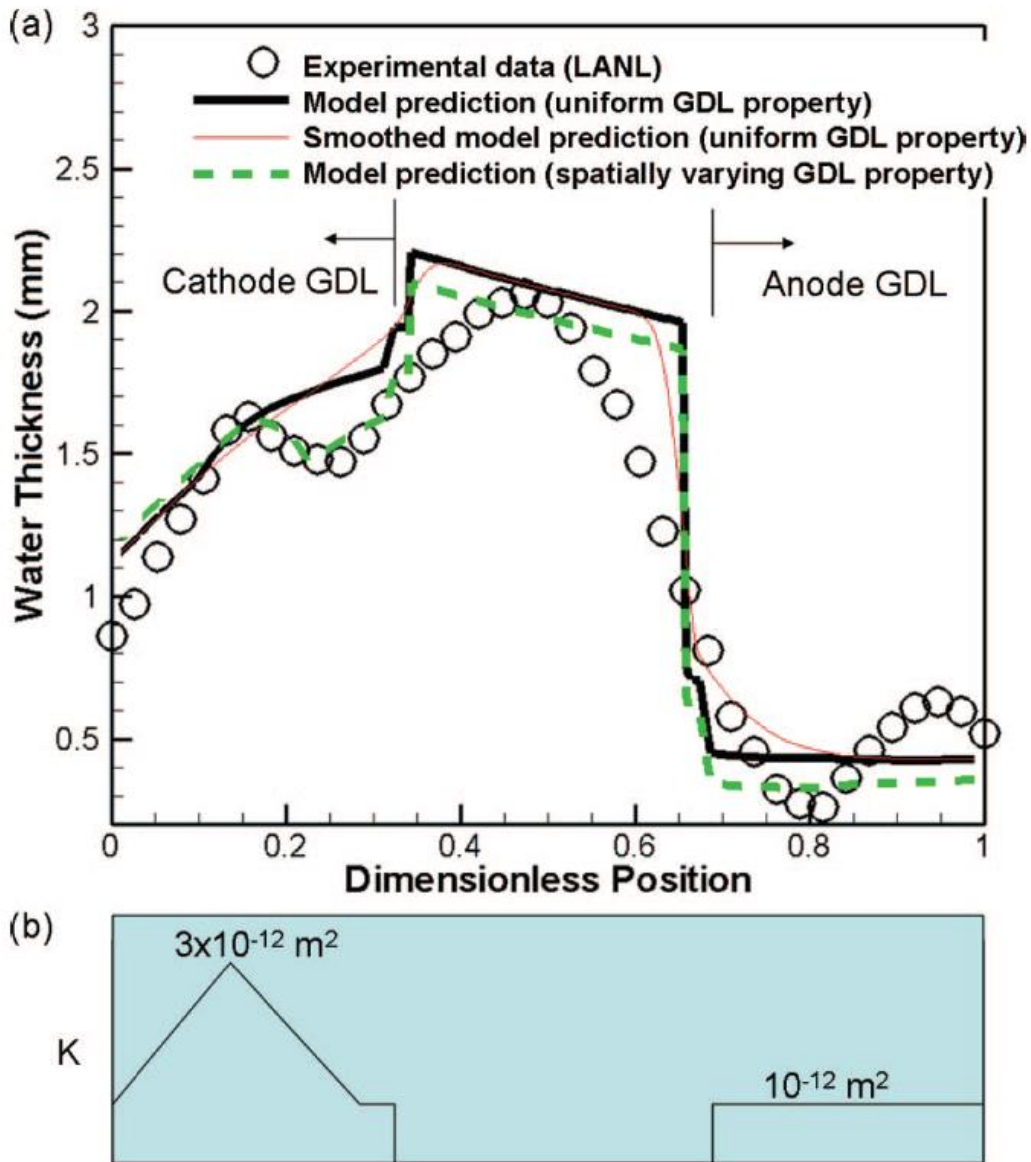


Figure 8 (a) Comparison with the high-resolution neutron imaging data, 2-D model prediction (with/without smoothing) using a uniform GDL property, and 2-D model prediction using a spatially varying GDL property (b) The varying GDL property in the present study. [41]

Table 1 The governing equations of multi-physics fuel cell model

Continuity equation:	$\varepsilon \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = S_m$	(27)
Momentum conservation:	$\frac{1}{\varepsilon} \left[\frac{\partial \vec{u}}{\partial t} + \frac{1}{\varepsilon} \nabla \cdot (\vec{u} \vec{u}) \right] = -\nabla \left(\frac{p}{\rho} \right) + \nabla \cdot \tau + S_u$	(28)
Energy conservation:	$\frac{\partial \bar{\rho} \bar{c}_p T}{\partial t} + \nabla \cdot (\gamma_T \rho c_p \vec{u} T) = \nabla \cdot (k^{eff} \nabla T) + S_T$	(29)
Species conservation (H_2O, H_2, O_2)	$\varepsilon^{eff} \frac{\partial C^k}{\partial t} + \nabla \cdot (\gamma_c \vec{u})$ $= \nabla \cdot (D^{k,eff} \nabla C^k) - \nabla \cdot \left[\left(\frac{m_{f_l}^{fk}}{M^k} - \frac{C_g^k}{\rho_g} \right) \vec{j}_l \right] + S_k$	(30)
Charge conservation (electrons):	$0 = \nabla \cdot (\sigma^{eff} \nabla \Phi_s) - S_\varphi$	(31)
Charge conservation (protons)	$0 = \nabla \cdot (K^{eff} \nabla \Phi_e) + S_\varphi$	(32)

Note: Detail regarding the equations and parameters can be found in ref [42]

In addition, other physics-based fuel cell models have been developed in the literature [43] [44] [45]. Afshari et al studied the performance of a PEM fuel cell with three different cathode channel design, namely, two parallel channel design (model A), locally restricted flow channels (model B), and metal foam (model C). They applied most basic mathematical models of several conservation laws, including mass, momentum, species, electrical charge and energy [46]. The numerical results obtained by their mathematical model were validated by the experimental data provided from Ticianelli et al [47]. Figure 9 is their polarization curve for different channel design, which is plotted based on their numerical results. It clearly shows the effects of reactant

flow rate, caused by different channel design, on the PEM fuel cell performance. The metal foam design shows best performance especially in high current density region [46].

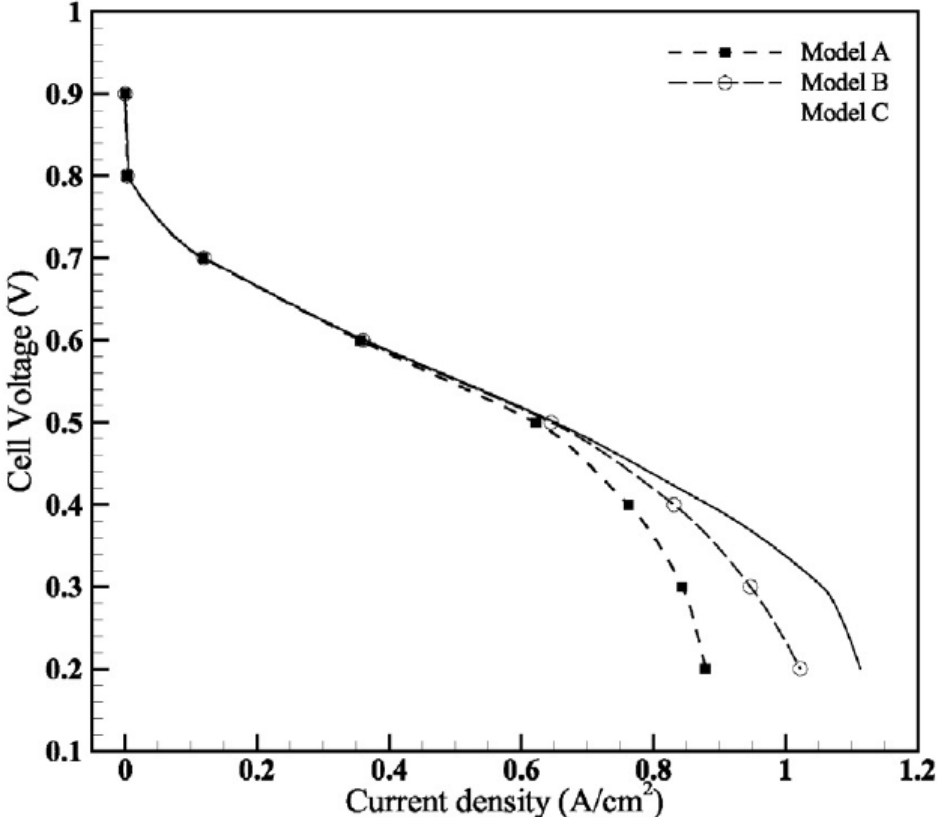


Figure 9 Polarization curve for different channel design [46]

Chapter 3: ANN method

3.1 Machine learning technique

The artificial neural network shows great ability to handle problems with incomplete data & information, and it is also able to solve complex multiple phenomena involved problems in an intuitive way. In this paper, a standard feed forward backpropagation neural network was developed to capture the PEMFC I-V curve behavior.

Theoretically, the neural

network using gradient descent algorithm to minimize the cost function, and a cost function is an assessment of the accuracy between predicted value and actual value. As shown in Figure 10, the training & validation data are fed into neural network. In the first iteration, the output can error

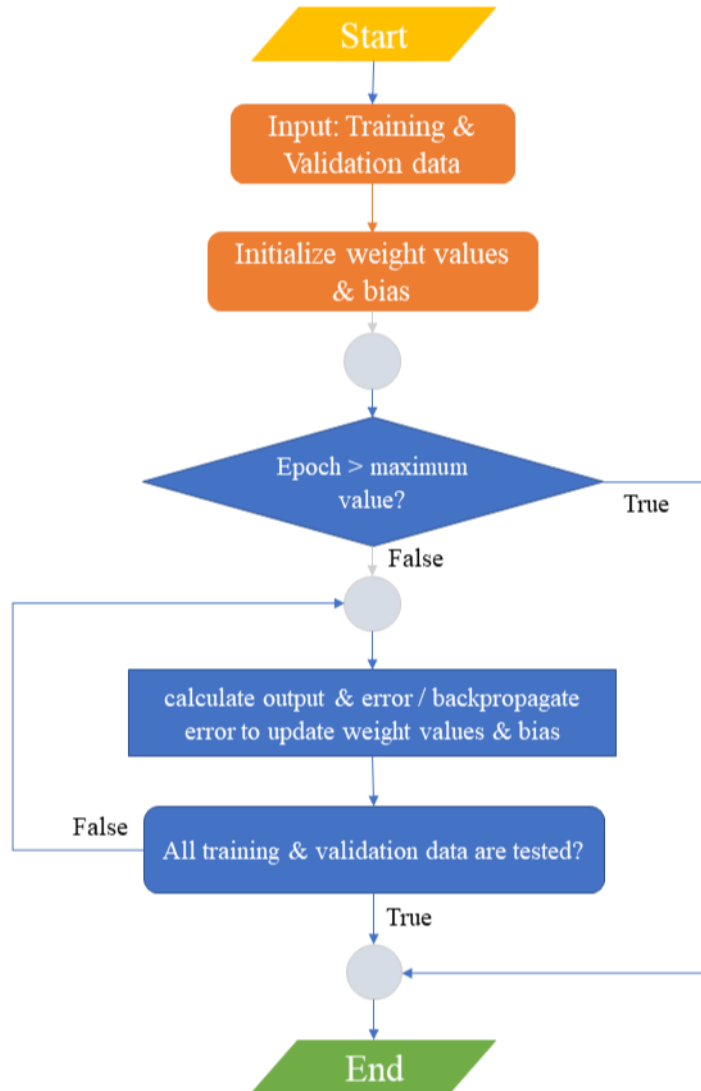


Figure 10 Flow chart for backpropagation and ANN algorithm

can be calculated based on the beginning weight and bias values. After all the validation data have been checked, the ANN starts the following iteration. The same processes are performed until the epoch reaches the designated maximum value. Via gradient descent algorithm, the weights and bias values are continuously updated to find the global minimum of the error value.

3.1.1 The architecture of neural networks

Figure 11 shows a typical multilayered neural network configuration. The leftmost layer is called input layer, and the neurons in this layer are known as input neurons. The rightmost layer is named output layer where the prediction values are output from this layer. Last, the layers located in between those two are called hidden layers. Specifically, Figure 11 indicates this typical neural network has two hidden layers. Commonly, those neural networks with multiple layers are named as multilayer perceptrons (MLPs), even if the neurons in the hidden layers are sigmoid neurons (not perceptrons). [48]

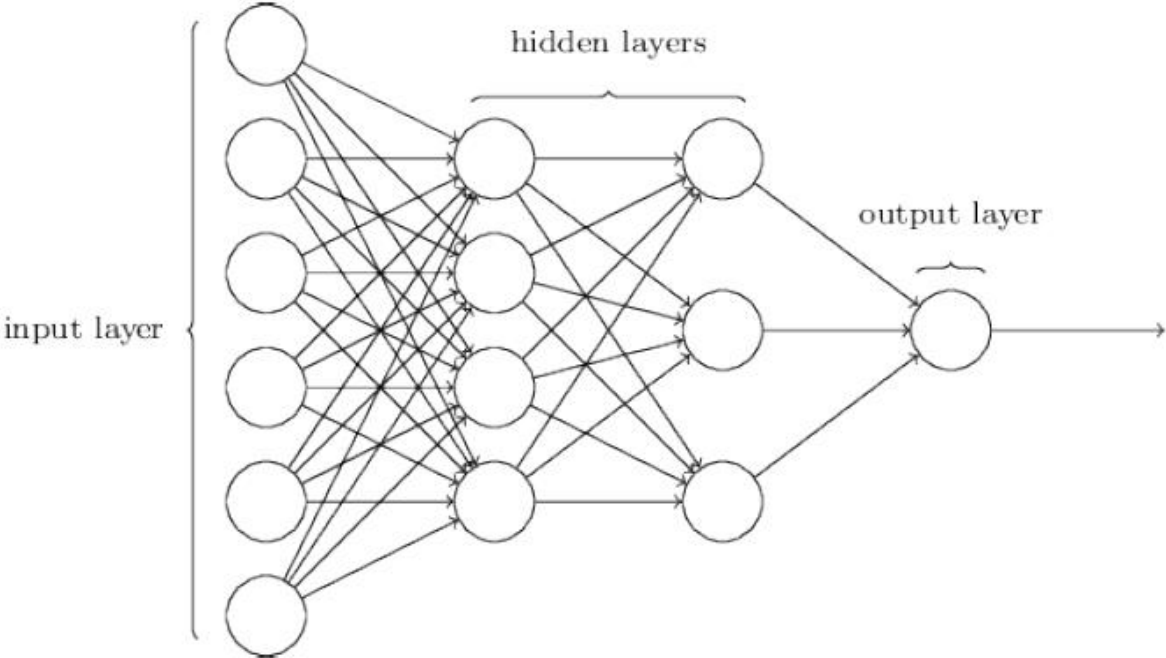


Figure 11 A typical architecture of a MLPs [48]

3.1.2 A type of neuron: sigmoid neuron

The ANN model predicts PEMFC current density which requires neurons inside hidden layer have ability that contain continuous values. Therefore, the concept of sigmoid neuron needs to be introduced in this regression model, because a simple perceptron can only contain value either 1 or 0. A sigmoid neuron have inputs $x_1, x_2, x_3 \dots, x_n$, and an overall bias b . The output for a sigmoid neuron:

$$\text{Output} = \sigma(\sum_j^n w_j x_j + b) = \sigma(z) = \frac{1}{1 + \exp(-\sum_j^n x_j - b)} \quad (33)$$

Where $\sigma(z)$ is sigmoid function,

$$\sigma = \frac{1}{1 + e^{-z}} \quad (34)$$

According to the equation (34), as $z \equiv w * x + b$ approaches positive infinite, the $\sigma(z) \approx 1$. When z is a large negative value, then we get the output $\sigma(z) \approx -1$. Sigmoid function is a smoothed version of a step function which results in the sigmoid neuron obtaining the ability to hold values between 0 and 1. [48]

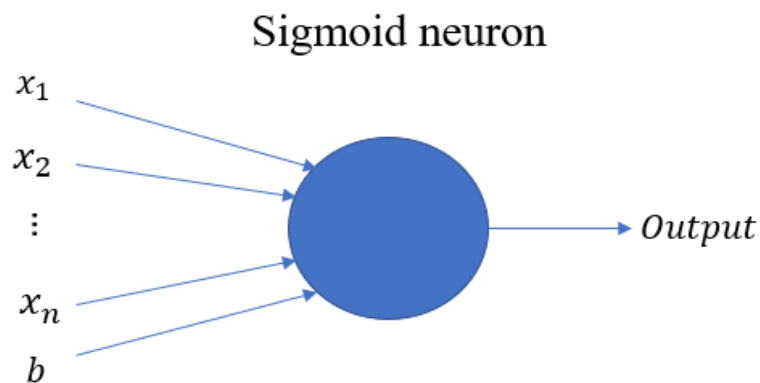


Figure 12 Typical sigmoid neuron

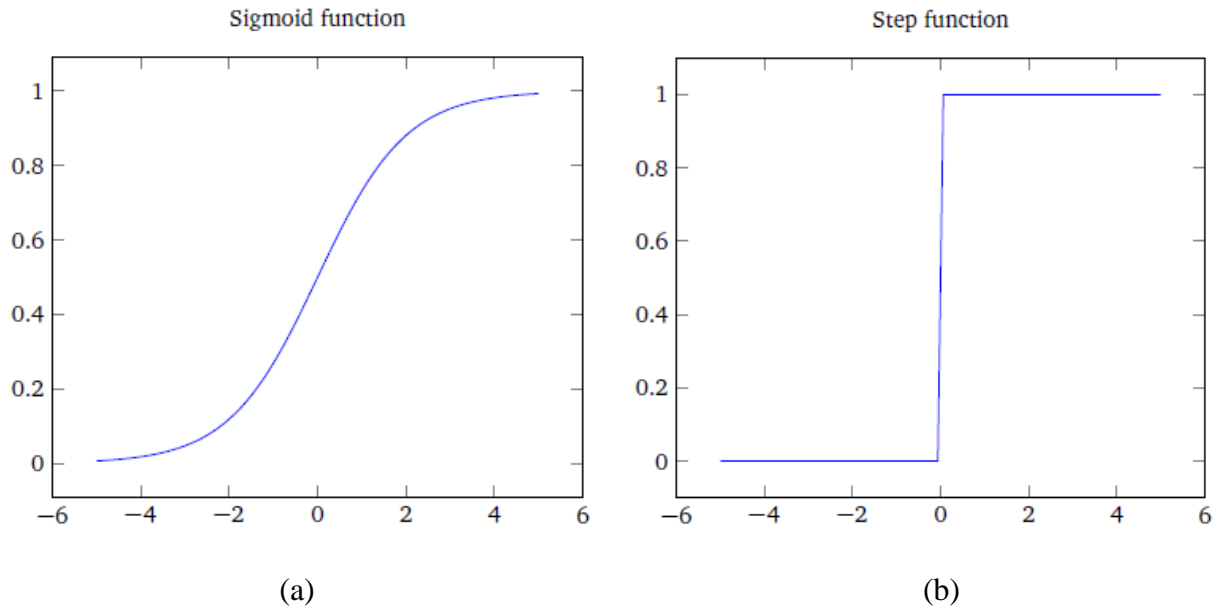


Figure 13 (a) Plot of sigmoid function, (b) Plot of a step function [48]

Sigmoid neuron is just an example of various kinds of neurons, and the name sigmoid neuron is just because this kind of neurons select sigmoid as activation function. However, to build and optimize the neural network, the selection of activation functions is one of important factors.

3.1.3 Activation function

The selection of activation function is a crucial step when building a ANN. They determine the output type, training efficiency, and converging speed of an ANN. Activation functions are attached to each neuron in ANN, they determine whether to activate the related neuron or not by judging the neuron input's relevance with the model prediction. [49]

There are several types of activation functions, and they can be classified into linear and nonlinear types. For linear activation function, the simple linear activation function is often used at the output layer to get a real output value, which serves as a scaling function in the ANN. However, it cannot be used as activation functions for hidden layers when applying

backpropagation algorithm, because this function has constant gradient. The classical nonlinear

Table 2 Characteristics for common activation functions

Activation function	Mathematical expression	Advantages	Disadvantages	Figure #
Linear	$f(x) = x$	<ul style="list-style-type: none"> Used as scaling factor in output layer 	<ul style="list-style-type: none"> Missing nonlinearity All layers collapse into one layer 	<i>a</i>
Sigmoid	$f(x) = \frac{1}{1 + e^{-x}}$	<ul style="list-style-type: none"> Smooth gradient Output value between 0 to 1 	<ul style="list-style-type: none"> Vanishing gradient: for very high/low x values, the gradient almost equal to 0. 	<i>b</i>
tanh	$f(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$	<ul style="list-style-type: none"> Smooth gradient Output both positive and negative values Larger gradient that that of sigmoid function 	<ul style="list-style-type: none"> Vanishing gradient problem 	<i>c</i>
<u>ReLU</u>	$f(x) = \max \{x, 0\}$	<ul style="list-style-type: none"> Computational efficiency Gradient function is nonlinear 	<ul style="list-style-type: none"> Dying ReLU problem: the input approach zero or negative, the gradient becomes zero 	<i>d</i>

activation function that applied in early development of ANN are sign, sigmoid and hyperbolic tanh function. The sigmoid function can hold values located in (0,1), which gives its ability to solve prediction problems with outputs can be interpreted as probabilities. Hyperbolic tanh function has the similar properties as that of sigmoid functions, however, tanh function can re-scale the output value from -1 to 1. Therefore, tanh function is preferred when the outputs contain both positive and negative values. [50] In recent years, Rectified Linear Unit (ReLU) function has gained a lot attention, because of its computational efficiency and ease in training neural networks. Table 2 lists the most commonly used activation in ANN applications.

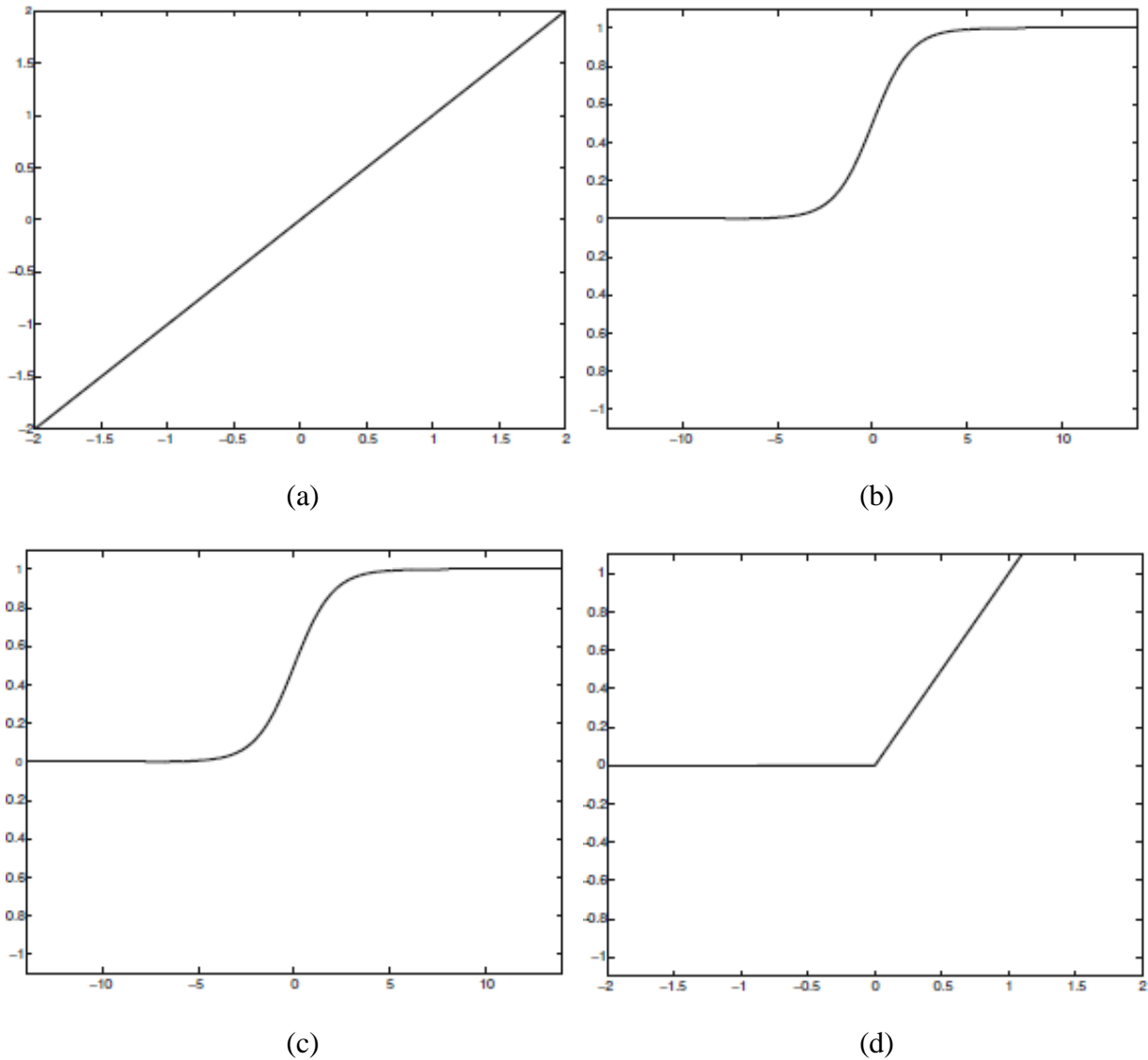


Figure 14 (a) Simple linear function (b) Logistic sigmoid function (c) Hyperbolic tangent function (d) Rectified Linear Units (ReLU) [50]

3.1.4 Cost function

A cost function is an assessment that measures how well the predicted result fit for its corresponding original result. When the value of cost function approaches to zero, the model successfully found proper weight and bias values which provide a prediction output very close to its actual value. A typical cost function is called mean squared error (MSE) function:

$$C(w, b) = \frac{1}{2n} \sum \|y(x) - a^L(x)\|^2 \quad (35)$$

Where w and b denotes the weight values and bias namely, n is the total number of training data, a is the predicted output from $L - th$ layer (output layer) of an ANN. The primary goal for training a neural network is to minimize the cost as a function of weight and bias values. Therefore, the best weight and bias values can be found out to describe the problem intuitively.

3.1.5 Gradient descent

The algorithm named gradient descent is usually applied to find the minimum of cost function. When adjusting weight values w and bias b , a small change of cost function can be expressed as

$$\Delta C(w, b) \approx \frac{\partial C}{\partial w} \Delta w + \frac{\partial C}{\partial b} \Delta b \quad (36)$$

We can denote the gradient vector: ∇C of cost function, and define Δv

$$\nabla C = \left(\frac{\partial C}{\partial w}, \frac{\partial C}{\partial b} \right), \Delta v \equiv (\Delta w, \Delta b) \quad (37)$$

Then equation (36) can be written as:

$$\Delta C(w, b) \approx \nabla C * \Delta v^T \quad (38)$$

In order to guarantee cost function $C(w, b)$ will always decrease, we need to keep $\Delta C(w, b)$ always as a negative value. Then, we can particularly choose:

$$\Delta v = -\eta \nabla C \quad (39)$$

Eventually, the change in cost function: $\Delta C(w, b)$ becomes,

$$\Delta C = -\eta (\nabla C)^2 < 0, \text{ where } \eta \text{ (learning rate) is a positive value} \quad (40)$$

This illustrates that the value of cost function always decreases, with the changing of weight and bias values. Via this process, we can find the best weights and bias values that describes the model by iterating this process repeatedly. In each iteration, the change of weight and bias values can be written as [48],

$$(w', b') = (w, b) - \eta \nabla C \quad (41)$$

3.1.6 Back propagation algorithm

The back propagation algorithm provide a tool that tracks the error δ_j^l for every layer and then builds connection between error and the gradient of the cost function with respect to weight values and biases, respectively, $\frac{\partial C}{\partial w_j^l}$ and $\frac{\partial C}{\partial b_j^l}$. The error for $j - th$ neuron in $l - th$ layer is defined as [48]:

$$\delta_j^l \equiv \frac{\partial C}{\partial z_j^l} \quad (42)$$

Starting from notations, as shown in Figure 15, the activation a_j^l is influenced by all the weight values and biases in previous layer, $(l - 1) - th$ layer. Thus, the activation a_j^l of $j - th$ neuron in $l - th$ layer can be expressed as [48]:

$$a_j^l = \sigma * (\sum_k w_{jk}^l * a_k^{l-1} - b_j^l) = \sigma * (z_j^l) \quad (43)$$

For simplicity, the equation (43) can be written in a matrix form:

$$a^l = \sigma * (w^l a^{l-1} - b^l) = \sigma * (z^l) \quad (44)$$

Where, the w^l is a $[J * K]$ weight matrix which contains all the weight values for $l - th$ layer, a^{l-1} is a $[K * 1]$ activation matrix for $l - th$ layer, and b^l is a $[J * 1]$ bias matrix.

From simplified equation (44), it can be observed that the $l - th$ layer activation a^l is the activation at $(l - 1) - th$ layer a^{l-1} times the weight matrix for $l - th$ layer, then minus the bias matrix for $l - th$ layer, b^l .

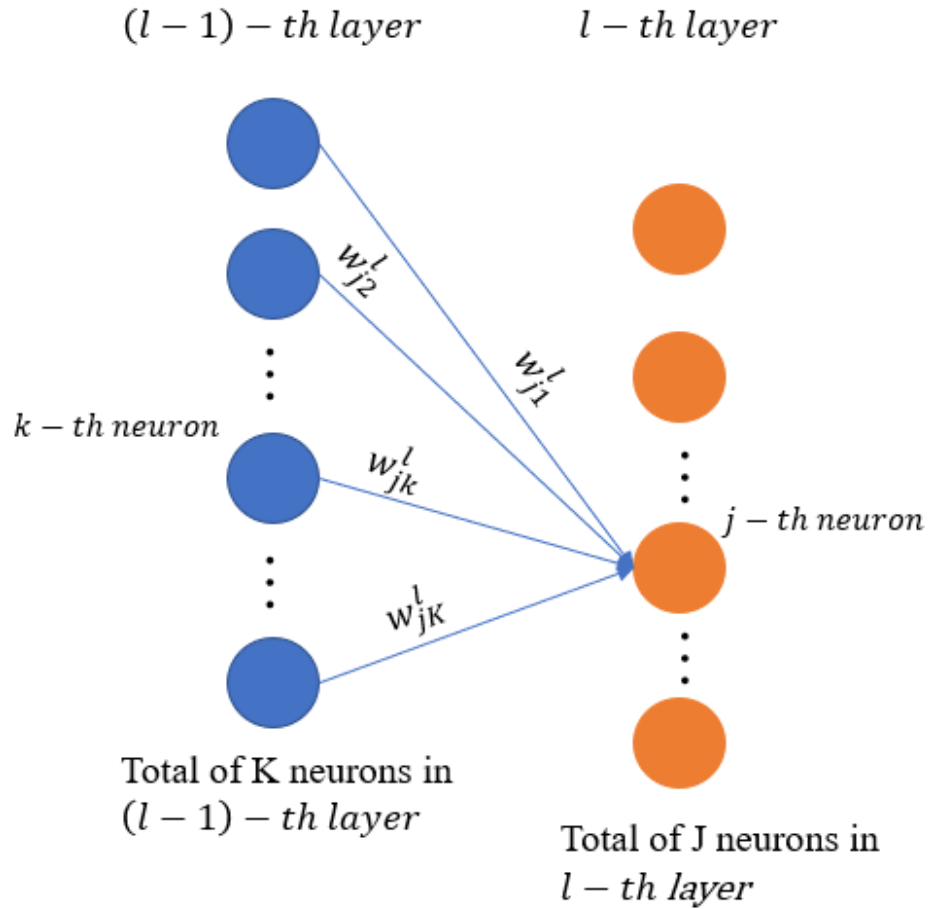


Figure 15 Schematic of a typical hidden layer

As stated previously, the back propagation monitoring the error in each layer. To illustrate the progress that the error is propagated backward from output layer to input layer, there are several remarkable equations need to be understood.

The error in $l - th$ layer can be expressed as an equation of error in $(l + 1) - th$ layer.

$$\delta^l = [w^{l+1}]^T \circ \delta^{l+1} * \sigma'(z^l) \quad (45)$$

Here, the symbol \circ is the Hadamard product between two matrices. From the equation, the error in $(l + 1) - th$ layer δ^{l+1} can be propagated backward to find out the error in $l - th$ layer, δ^l .

Applying chain rule on equation (42) and equation (44), the error in the output layer $L - th$ layer can be expressed as:

$$\delta_j^L = \frac{\partial C}{\partial z_j^L} = \frac{\partial C}{\partial a^L} * \sigma'(z^L) \quad (46)$$

In matrix form,

$$\delta^L = \nabla C \circ \sigma'(z^L) \quad (47)$$

Generally, starting from output layer, we can compute error for any layer by combining equation (46) and equation (47). In addition, the neural network needs to adjust the weight values and biases to find the best combination of weight values and biases, and the error should be able to send messages to neural network to determine whether the adjustment of weights and biases is necessary or not. In terms of error δ^l , the rate in change of cost function with respect to weights and biases can be expressed as following,

$$\frac{\partial C}{\partial w^l} = a^{l-1} * \delta^l \quad (48)$$

$$\frac{\partial C}{\partial b^l} = \delta^l \quad (49)$$

Several insights can be observed from those equations,

1. The term $\sigma'(z^l)$ equation (47) implies when the gradient of activation function is small, which results in a small value of δ^l . Consequently, the neuron learning rate determined by term: $\frac{\partial C}{\partial w^l}$ will be small. In other words, the neuron is at saturated state.
2. The equation (48) shows that a small activation input lead to a small gradient in cost with respect to weight values.

This simplified ANN mathematical model with some symbol correction was based on a basic model. Details regarding with ANN mathematical model can be found in Ref [48].

A four-layered feed forward backpropagation artificial neural network was developed with python programming tool. The build-in neural network algorithms in TensorFlow 1.1.5 open source machine learning library were used in our ANN. Figure 16 shows the architecture of our feed forward backpropagation neural network. It consists of one input layer with four input parameters, respectively, operating temperature (T), anode and cathode relative humidity (RH_a/RH_c), output voltage (V). Two hidden layers were built in our ANN, and various layer configurations such as number of neurons and activation functions, are applied to explore the best ANN parameter for prediction our PEMFC performance. Current density is the only output, which is combined with experimental voltage data to plot the predicted I-V curve.

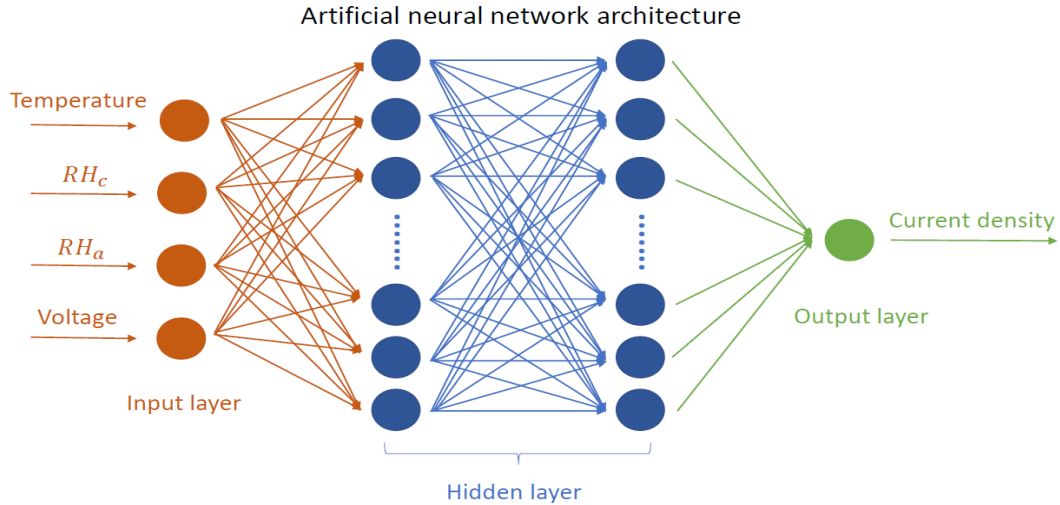


Figure 16 Architecture for our ANN

3.2 Data acquisition

The data imported as training, validation and testing data were from the study of liquid water distribution by wang et al [41]. As stated in previous chapter, the experimental data were validated by their numerical model, therefore the experimental data were imported as training validation and testing data in our ANN. Total of 1460 data were selected as training, validation and testing data among total number of 1615 data provided, because several experimental tests were conducted under wrong operating conditions which resulted in extremely high current density outcome in those experimental runs. Table 3 lists sample data points for each operating condition. Experiments were performed under combinations of those operating conditions. Sample data obtained under specific operating conditions are shown in Table 4.

Table 3 Table of sample data

Sample data	Input: operating conditions				Output
	Temperature	RH_a	RH_c	Voltage	Current density
Data1	353	0.75	0.75	0.9	217
Data2	333	0.75	0.25	0.34	15400
Data3	293	0.99	0.99	0.74	94.6

Note: Because of the unit of current density, the value of current density at low voltage region is very large.

Table 4 Specifications for input and output parameters

Parameter name	symbol	Unit	Data points
Temperature	T	K	293,313,333,353
Relative humidity (anode)	RH_a		0.25,0.5,0.75,0.99
Relative humidity (cathode)	RH_c		0.25,0.5,0.75,0.99
Voltage	V	V	0.3-0.9
Current density	j	A/m^2	

3.3 Data standardization

As it can be seen from figure of sigmoid function, it can only operate input data from (0,1). Similar for the hyperbolic tangent function, which can handle data between -1 and 1. Therefore, data normalization is necessary at the beginning. In addition, as the input value x is close to 1 or 0, the gradient of sigmoid activation function will be very close to 0. The vanishing of gradient can cause the adjustment of weights and biases very slow. In other words, the neuron

will learn slowly. To avoid vanishing gradient issue, all the inputs value including temperature, relative humidity, voltage, were normalized to (0.1,0.9).

Secondly, for output data, as noted previously, the current density value could be very large at low voltage density region. Thus, an extra step $\log_{10}(\text{current density})$ was applied firstly. After that, a normalization is performed on those data. Eventually, all the input and output data were reverted to their original scale.

3.4 Performance evaluation criterion

It is very important to propose a method to evaluate the prediction results for our ANN. Since current density has very large value, because of its unit ($\frac{A}{m^2}$), the resulted root mean square error (RMSE) values are always very high, with its value reach up to 1000 (A/m^2) in some simulation trails. Therefore, we mainly evaluate our model accuracy with two statistical criteria, mean absolute percentage error (MAPE) and coefficient of determination (CoD or R^2),

$$MAPE = \frac{1}{N} \sum_i^N \frac{|Y_{pred,i} - Y_{actual,i}|}{|Y_{actual,i}|} \quad (50)$$

$$R^2 = 1 - \frac{\sum_i^N (Y_{pred} - Y_{actual})^2}{\sum_i^N (Y_{pred} - Y_{avg})^2} \quad (51)$$

Where, Y_{pred} is predicted current density, Y_{actual} is corresponding experimental current density, and Y_{avg} is the average experimental current density.

Small MAPE value indicates the ANN model prediction accuracy is very high. Coefficient of determination is a measurement for the fitting between prediction value and actual experimental value. With $R^2 \approx 1$, the prediction values fit the experimental value very well.

Chapter 4: Results and discussion

4.1 Effect of neuron number in hidden layer

Determining number of neurons in each hidden layer is a very important part when constructing the architecture of ANN. Using too few neurons in each hidden layer may cause the issue named underfitting where neuron cannot effectively capture all the information provided by training data set. On the contrary, too many neurons in hidden layer will add complexity on the ANN, which can lead to overfitting. Basically, in the case of overfitting, not all the neurons in hidden layer are trained because of the excessiveness of information processing ability. [51] Bhoopal et al developed a back-propagation ANN with 9 neurons in each hidden layer, and their model performed a successfully predicted their fuel cell stack voltage and current. [52]

Figure 17 shows a sample prediction results for ANN with the number of neuron equal to 20. Obviously, our ANN prediction results show a great agreement with our experimental data. Specifically, total of 4 ANN configurations were tested, namely, 5 neurons, 10 neurons, 20 neurons, and 30 neurons. According to the results shown in Table 5, it yields that minor difference can be observed when changing the number of neurons in hidden layer, with the lowest MAPE equals to 5.52% using 20 neuron hidden layer configuration. However, when using 30 neurons in each hidden layer the prediction results are even worse than that of other trials, which implies the total of 1180 data fed into this ANN is not enough. What's more, excessive neurons in ANN will cause a longer time for training our ANN. Therefore, we set neuron number to 20 in our latter simulation.

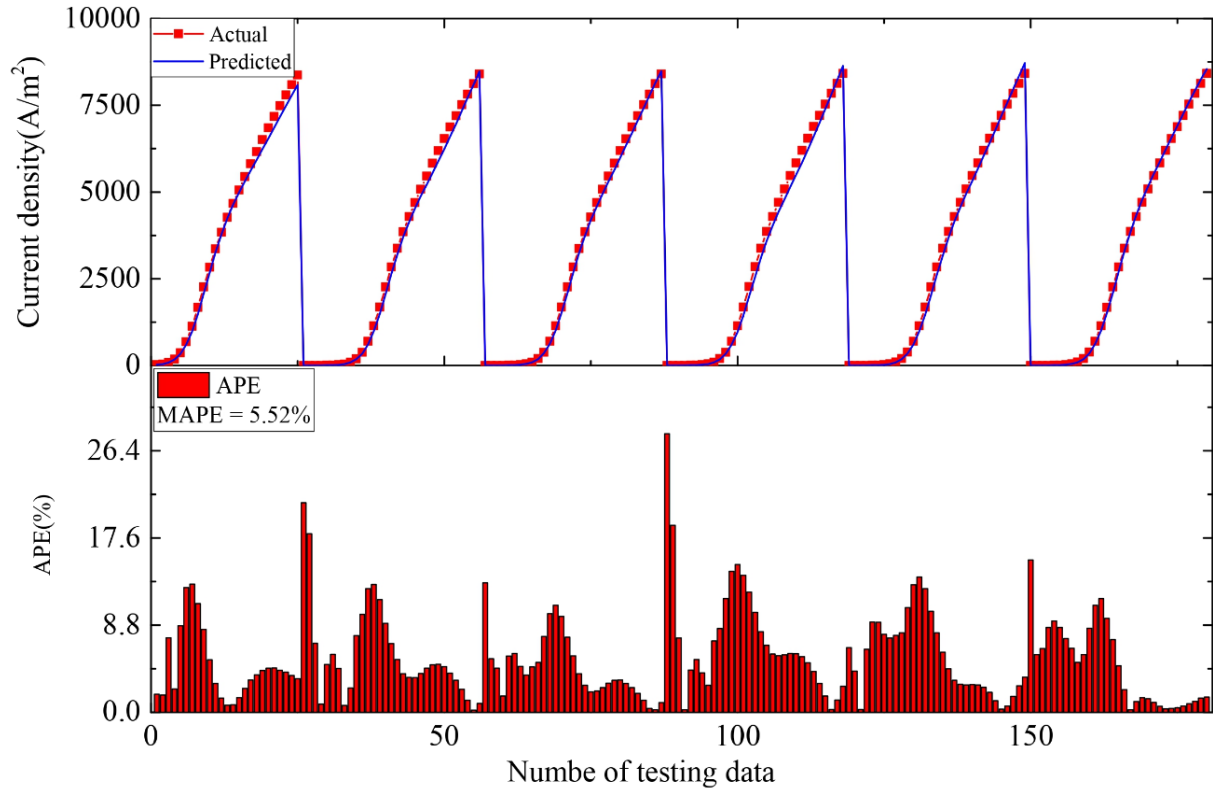


Figure 17 Predicted current density vs. actual current density & absolute percentage error [20 neuron configuration]

Table 5 Prediction accuracy using different number of neurons

Neuron number in hidden layer	MAPE (%)	APE_{max} (%)
5	6.88	37.4
10	5.65	39
20	5.52	28.1
30	7.47	17.7

4.2 Effect of training data size

As stated previously, selecting a proper size of data set to fit for the complexity of ANN is a very important step in ANN construction. Because of the limitation in our available experimental data, thus we need to find an optimum number of training data fed into our ANN which does not cause either underfitting or overfitting. Three different size of data were fed to our ANN, namely, 500, 700, and 1000 training & validation data. Figure 18 shows a general result for all three prediction trails. Specifically, Figure 19 and Figure 20 show prediction results for selected data points from one I-V curve, and it can be observed that for our ANN model complexity, the 1000 training & validation data size can provide more information to training our ANN, while 500 and 700 data set cannot produce accurate results. Figure 21 is a linear fitting between

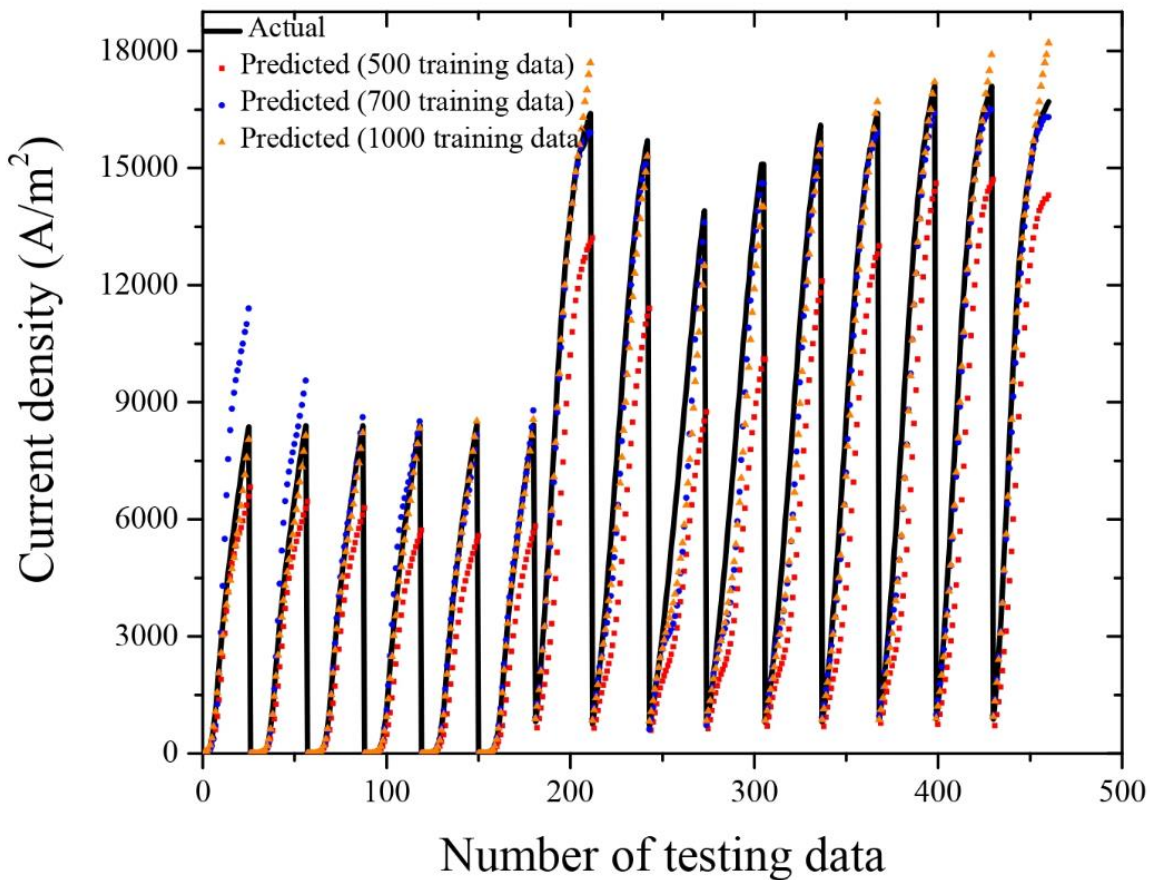


Figure 18 General prediction of current density [training data number variation]

predicted and actual experimental results for 1000 training & validation data set. The relationship between predicted and actual current density is: $predicted\ value = 0.966 * actual\ value - 368.82$, and the slope is very close to 1 shows the similarity between predicted and actual experimental data is high. However, the relatively “large” interception value is caused by the very large value of current density, which can reach up to around $18000\ (\frac{A}{m^2})$. The high coefficient of determination ($CoD = 0.976$) demonstrates a very high portion of prediction current data can be predicted by our experimental data by using this linear fitting model.

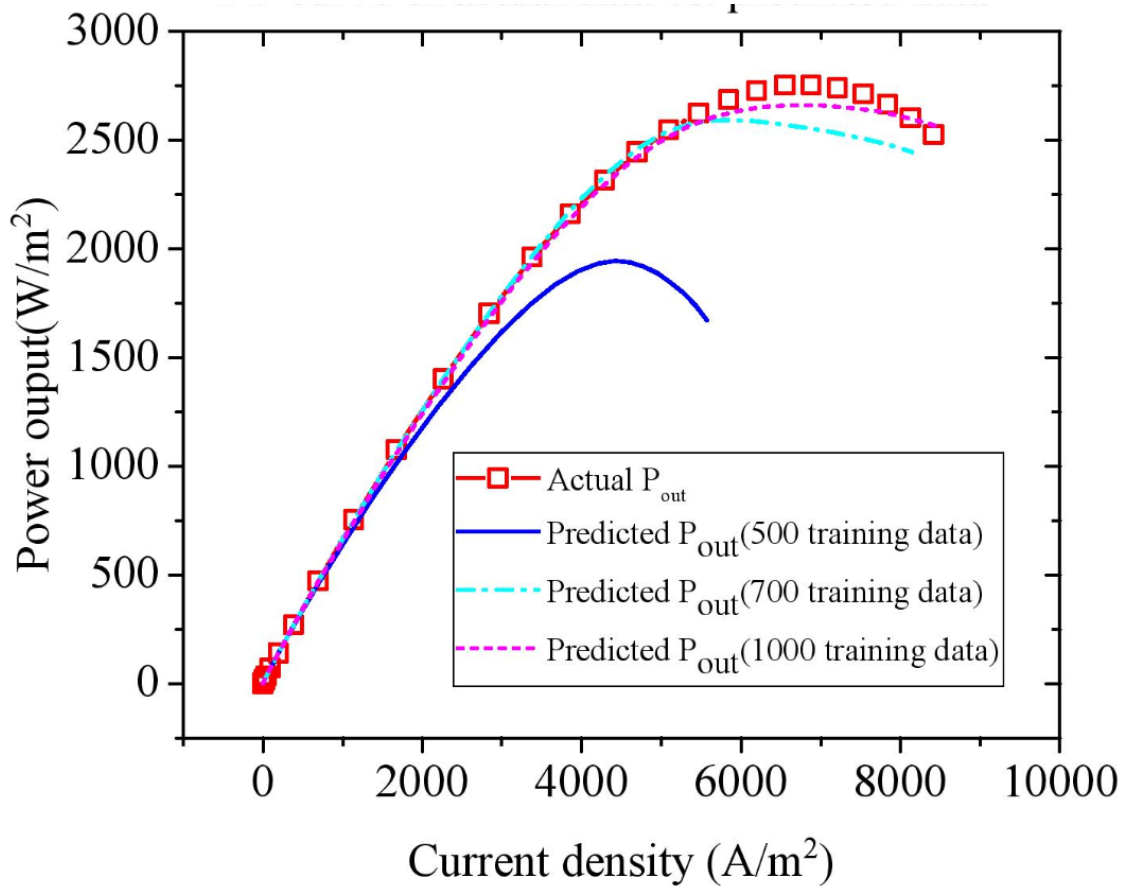


Figure 19 Plot of current density vs. power output curve [training number variation]

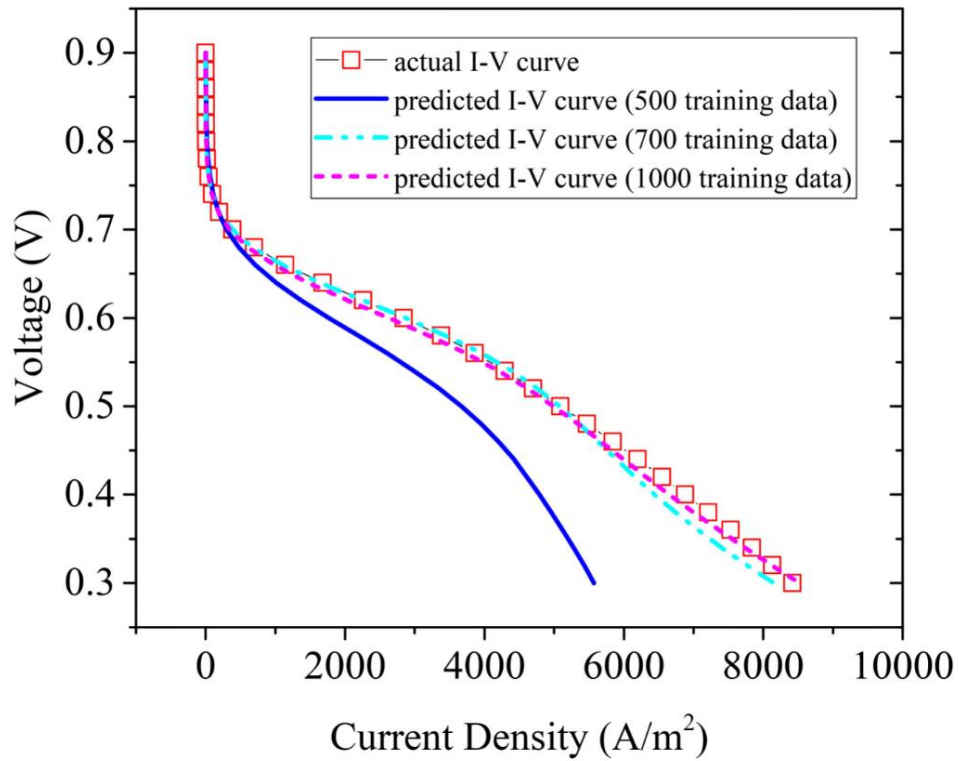


Figure 20 Plot of polarization curve [training number variation]

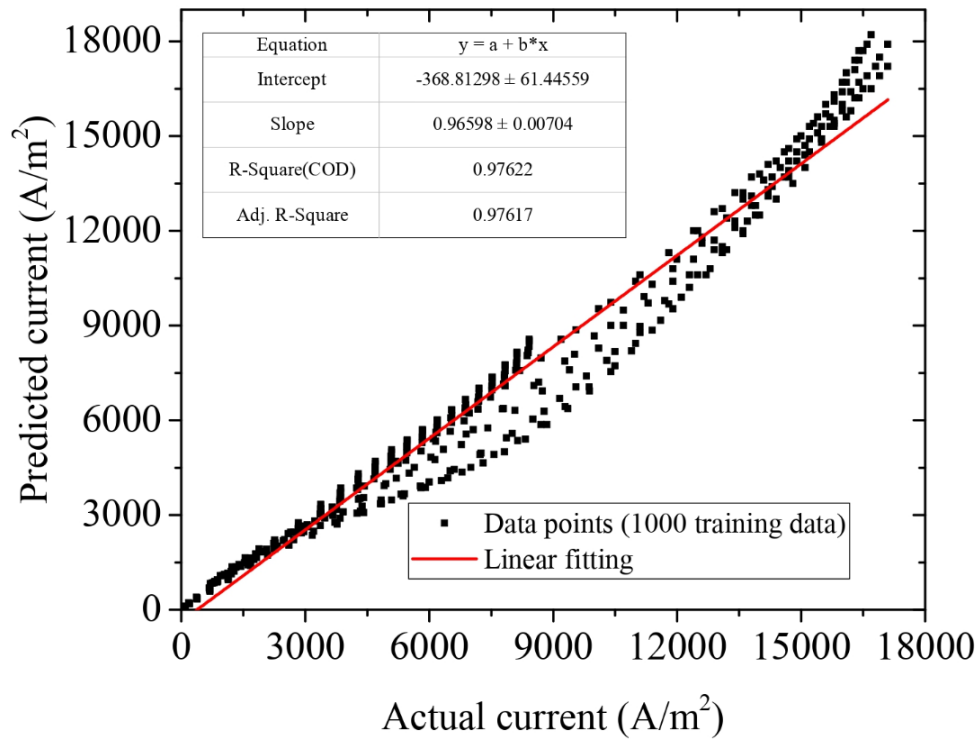


Figure 21 Linear fitting between predicted current density and actual current density [1000 training data]

4.3 Effect of activation function

Selecting a proper activation function for ANN can enhance model's ability to solve various types of problem while remain computational efficiency and converging speed. This section discusses using different activation function and their training results. The functions we used are logistic sigmoid, hyperbolic tangent, rectified linear units and simple linear function. From the general prediction result in Figure 22, it is obvious that simple linear activation function doesn't have the ability to capture the highly nonlinear behavior of PEMFC performance curve. A I-V and I-P curve selected from whole testing data set were plotted in Figure 23 and Figure 24 as well. Clearly, hyperbolic tangent ($\tanh(x)$) function as activation function in our model exhibits the most accurate prediction, with mean absolute percentage error of 6.15%. $ReLU(x)$ function is popular in recent ANN research because of its ease in computation, fast in training speed, and non-saturate property. However, it is not the best choice for our model, which is mainly because $ReLU(x)$ function has dying neuron problem when input value x is less than or equal to 0. What's more, the $ReLU$ basically is a linear function with a constant gradient when $x > 0$, and it may not be able to capture the highly nonlinear behavior of PEMFC performance. The relation between hyperbolic tangent and logistic sigmoid,

$$\tanh(x) = 2\text{sigmoid}(2x) - 1 \quad (52)$$

The $\tanh(x)$ function shows greater gradient in $x \in (0,1)$ than that of sigmoid function.

Therefore, the ANN learning speed when using $\tanh(x)$ function as activation function can be increased in a limited number of training epoch.

Table 6 Table of prediction accuracy using different activation functions

Type of activation function	Mean absolute percentage error (%)
Simple linear	1110
Rectified Linear Units (ReLU)	23.2
Logistic sigmoid	17.6
Hyperbolic tangent (Tanh)	6.15

Note: MAPE is calculated by all predicted data, not data from that particular I-V curve

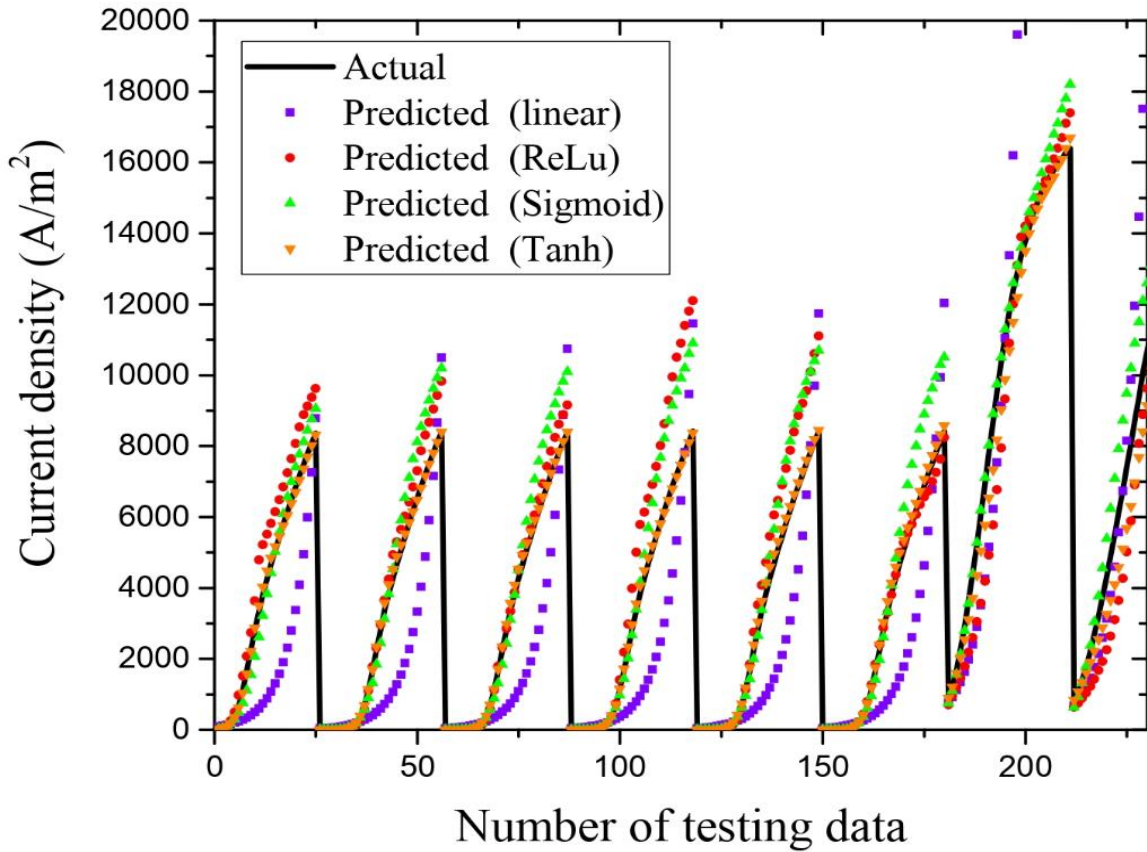


Figure 22 General prediction of current density [activation function variation]

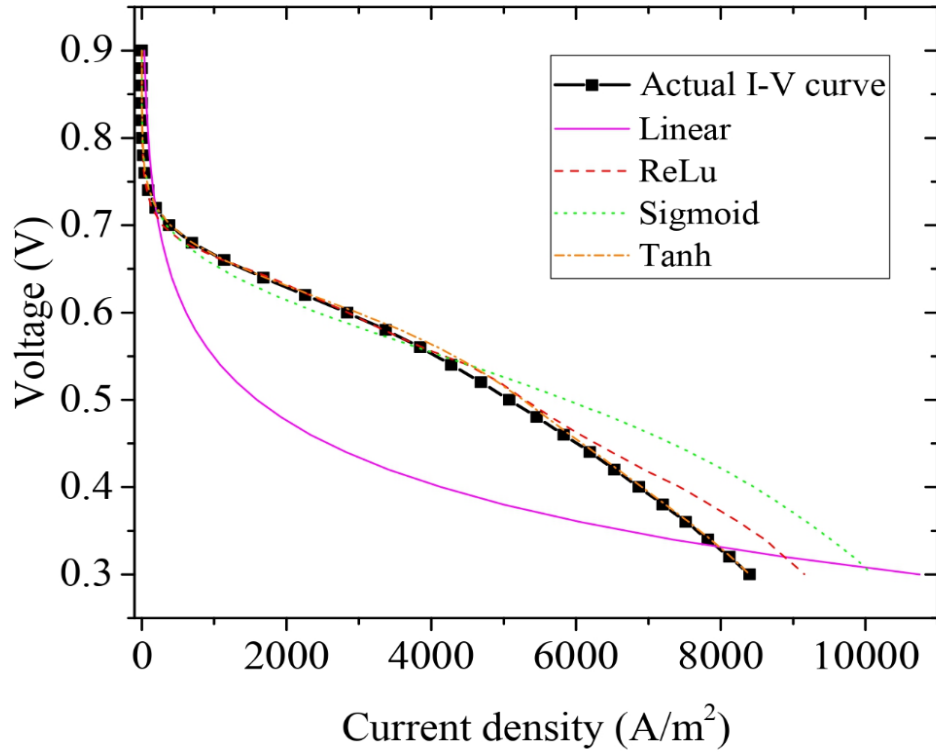


Figure 23 Plot of polarization curve [activation function variation]

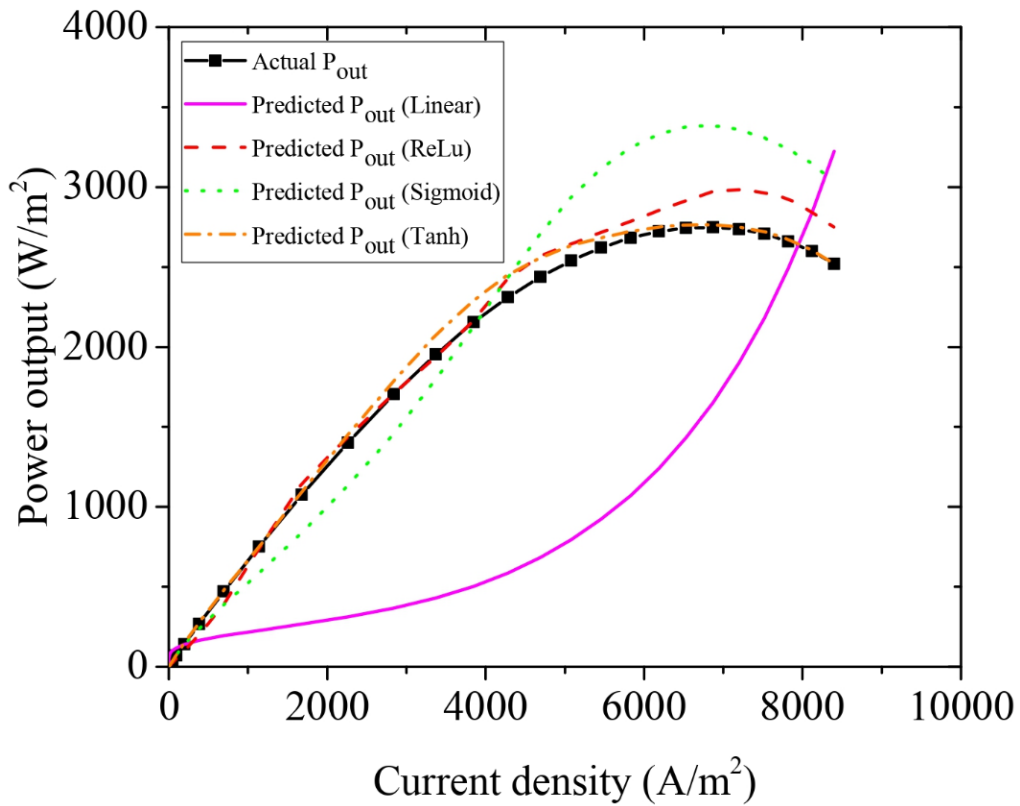


Figure 24 Plot of current density vs. power output curve [activation function variation]

4.4 Effect of random vs. ordered training data selection

The results for two different simulations are compared in this section. 1000 training data and 180 testing data were selected in two different manners, respectively, randomly selection and ordered selection. Prediction result for randomly selected data trial shows a higher accuracy than that of ordered selected data. According to Figure 25 and Figure 26, even the maximum absolute percentage error (APE_{max}) for two trails are very similar, with value of $APE_{max,ordered} = 15.4\%$ and $APE_{max,random} = 14.9\%$, the overall performance for random data trail is better than that of ordered data trail. Because the shuffle of raw data set before selection can help with a completed coverage of input information, while the training data selected in order may cause the missing of some input information. In addition, from Figure 25, another significant discovery can be observed that in small current density region, the prediction accuracy is not as high as it in large current density region.

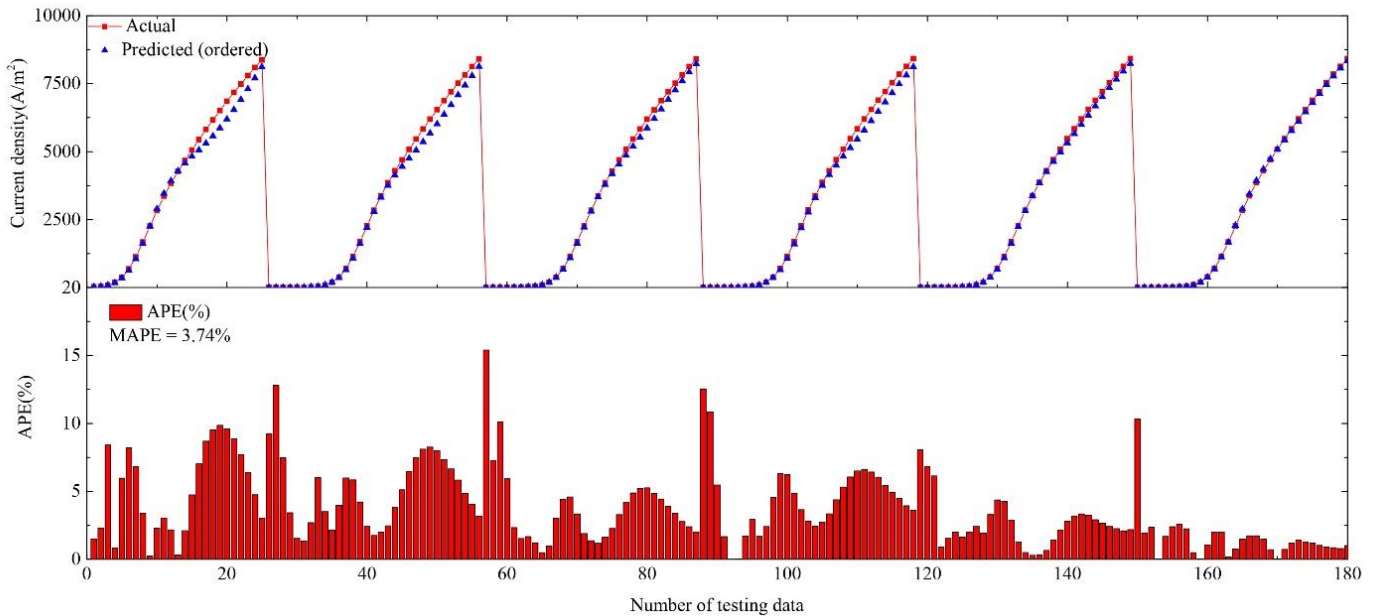


Figure 25 Predicted current density vs. actual current density & absolute percentage error plot

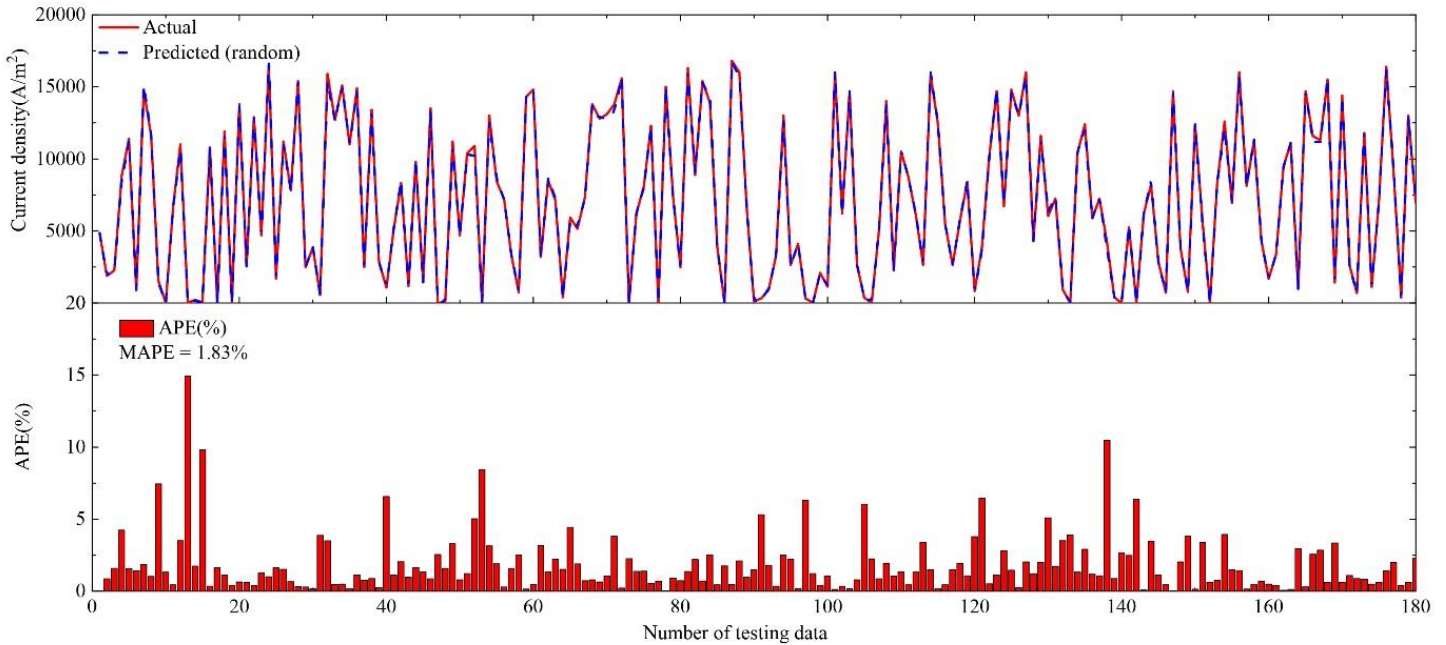


Figure 26 Predicted current density vs. actual current density & absolute percentage error plot [randomly selected training data]

4.5 Effect of selected training data set

Olabi et al successfully used ANN into Taguchi analysis with lack of experimental available data. [53] To investigate the ANN's ability to provide accurate prediction with incomplete input data signal, and filter out noisy signal, we intentionally excluded all data with $RH_c = 0.25$ out of training and validation data set. We picked one I-V curve (total of 30 data points) from $RH_c = 0.25$ group to test whether our ANN can perform a prediction with those void experimental data. Furthermore, to compare our results, we picked one additional data from the testing I-V curve and put into the training and validation data set. Then, we repeated this process with three individual data from low, middle and high current density region. Figure 27 shows the ANN can provide an accurate prediction without the information from $RH_c = 0.25$

data set. Table 7 shows the prediction accuracy and the improvement by adding one data point from $RH_c = 0.25$ from small, middle and large current density region. The very high coefficient of determination (R^2) means the predicted value can be expressed by our linear fitting model very well.

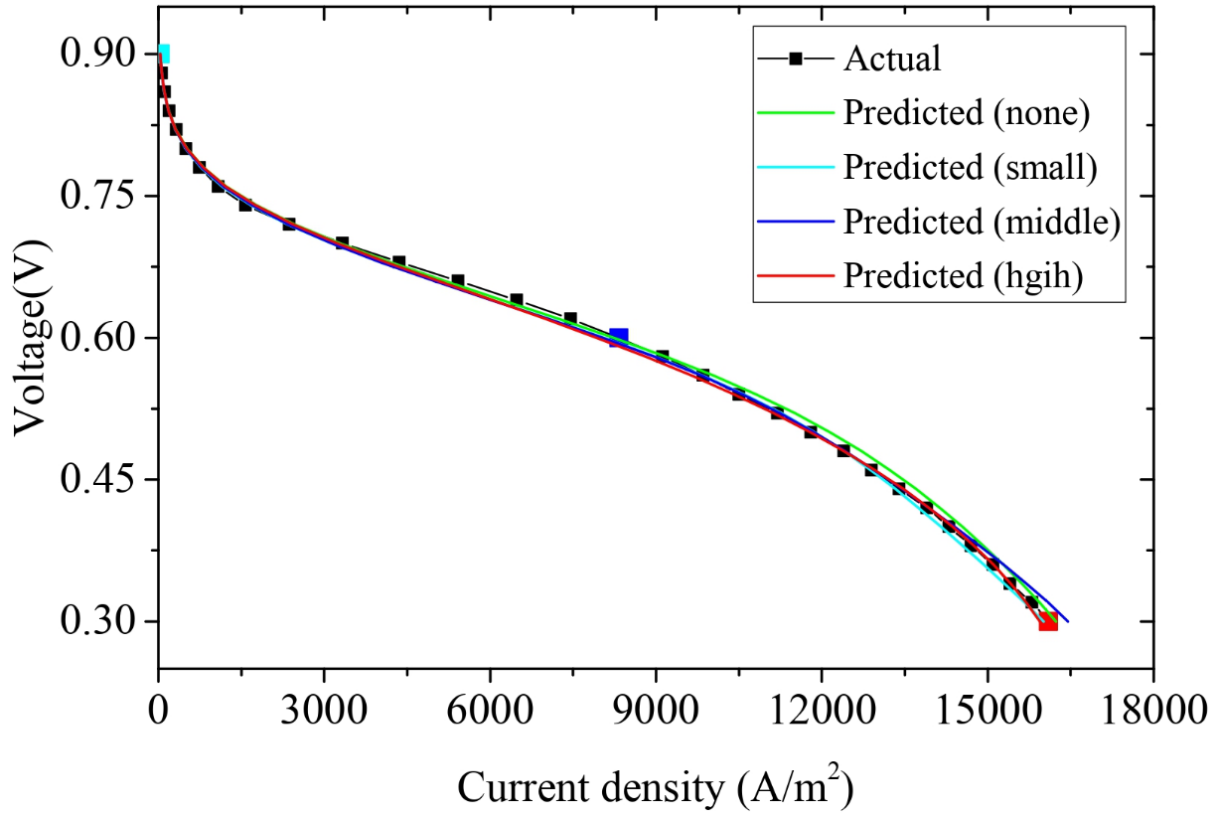


Figure 27 Polarization curve plot for predicted current density and actual current density [with/without $RH_c = 0.25$ data point]

Table 7 Linear fitting results for simulation trails with/without $RH_c = 0.25$ data point

With/without $RH_c = 0.25$ data set	MAPE (%)	R^2	Linear fitting equation
None	3.42	0.99939	$y = 1.016x - 27.92$
Small	3.05	0.99936	$y = 0.996x - 63.16$
Middle	3.08	0.99902	$y = 1.013x - 117.8$
Large	3.18	0.99917	$y = 0.998x - 55.46$

Moreover, we investigated the ANN's ability to filter out the noisy signal in the training data set. One fake data point was created on purpose. For comparison, a correct data point is selected at the same current density. Table 8 shows specifications for those two data points. Each of them was added into 500 training & validation data set in two different trails, and the results shows that the fake data point can reduce the model prediction accuracy. However, even fed with noisy data, the model still came out with a solid prediction on the testing I-V curve from $RH_c = 0.25$ data group, with its MAPE = 4.48%.

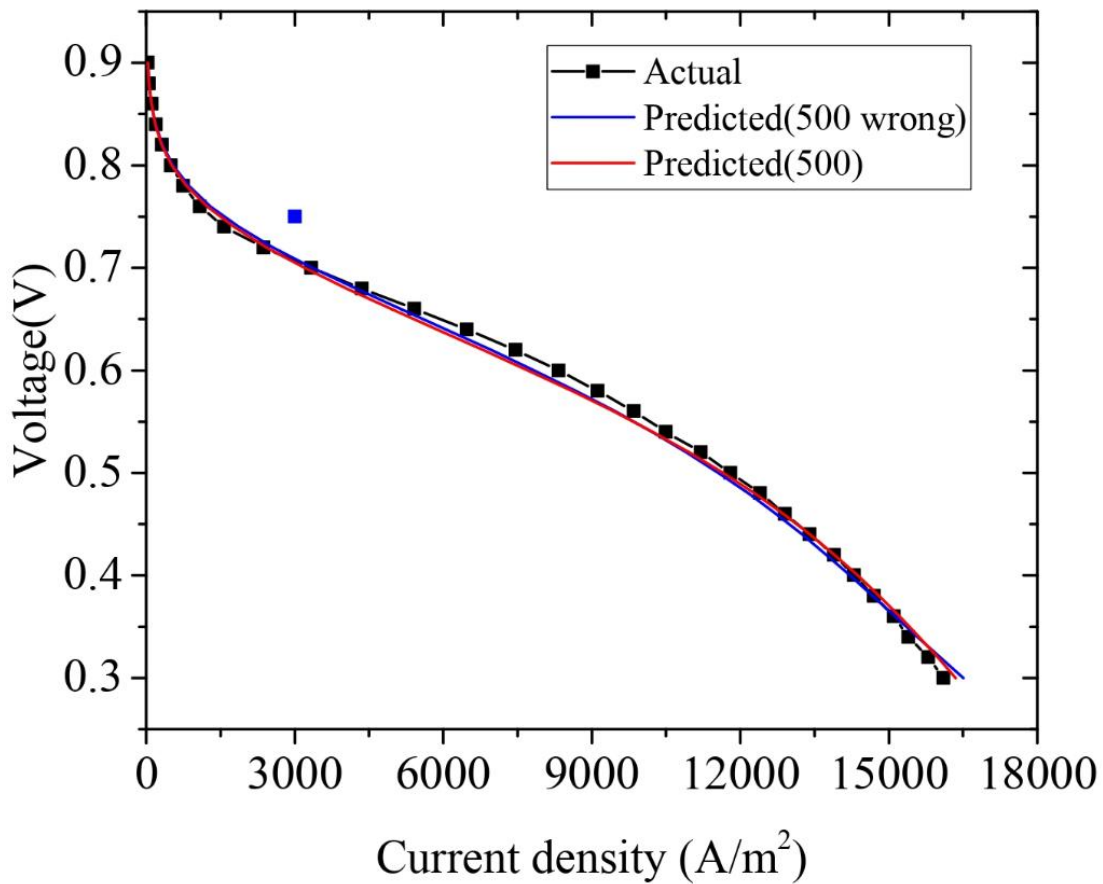


Figure 28 Polarization curve plot with/without fake data point

Table 8 Specifications for true & fake added into training data set

Data type	Specifications				
	Temperature	RH_c	RH_a	Voltage	Current density
	(K)			(V)	($\frac{A}{m^2}$)
True data	333	0.25	0.75	0.72	3000
Fake data	333	0.25	0.75	0.75	3000

Note: True data is measured from the testing I-V curve, the fake data is created by adding some output voltage.

Table 9 Linear fitting results for simulation trails with fake/true data point

Type	MAPE (%)	R^2	Linear fitting equation
With fake data	4.48	0.9984	$y = 0.9942x - 37.46$
With true data	4.18	0.9981	$y = 1.003x - 123.3$

Chapter 5: Conclusion

In this study, a backpropagation ANN model was developed in python programming tool to predict PEM fuel cell performance or I-V curves. A database of > 1500 fuel cell performance were generated from a 3D physical model to cover the fuel cell operation in a wide range of temperatures, anode relative humidity, cathode relative humidity, etc. The effects of number of neurons in each hidden layer, the activation function applied on each hidden layer, the number of training & validation data set, and the selective data set were investigated. Various prediction results show that a backpropagation ANN model can predict the behavior of a typical PEMFC's performance curve (I-V curve) with a reasonable accuracy. As a result, ANN can be an alternative method to find the relationship between the inputs and outputs when physical models are not present. The ANN simulations were carried out using Python platform. The main results are listed as below,

1. The number of neurons applied on each hidden can affect the complexity of our ANN model. To avoid either underfitting or overfitting issues, a proper complexity of our ANN model needs to be found to fit with the training data provided. With current size of available data set, it is found that using 20 neurons in each hidden layer yielded the most accurate result with MAPE equals to 5.52%

2. In terms of training & validation data size, feeding 1000 data into ANN resulted the high accuracy prediction. In addition, 500 and 700 training data set fed into ANN can provide reasonable results.

3. Using hyperbolic tangent as activation function in each hidden layer and simple linear function as activation function on output layer can capture the behavior of PEMFC performance curve best, with its prediction MAPE equals to 6.15%.

4. Randomly selected training & validation data set can cover more signal information over the entire range of input parameters, which cause a better prediction than that of ordered selected training & validation data set.

5. ANN successfully predicted the $RH_c = 0.25$ I-V curve when the training data set is lack of $RH_c = 0.25$ information. In other words, ANN can provide performance prediction even with incomplete data information, which can significantly reduce the number of required experiments. In addition, ANN is also able to filter out the noise training data. Our ANN model achieved a prediction with MAEP equals to 4.48% when noise data is present.

References

- [1] Wang Y, Chen K S, Mishler J, et al. A review of polymer electrolyte membrane fuel cells: Technology, applications, and needs on fundamental research[J]. *Applied energy*, 2011, 88(4): 981-1007.
- [2] "Connecticut Hydrogen-Fuel Cell Coalition," [Online]. Available: <http://chfcc.org/resources/hydrogen-fuel-cell-benefits/>. [Accessed 14 4 2020].
- [3] David Hart, Franz L., Stuart J, Jonathan Lewis, "The Fuel Cell Industry Review 2019," E4tech, 2019.
- [4] Wang Y, Diaz D F R, Chen K S, et al. Materials, technological status, and fundamentals of PEM fuel cells—a review[J]. *Materials Today*, 2020, 32: 178-203.
- [5] Asghari S, Shahsamandi M H, Khorasani M R A. Design and manufacturing of end plates of a 5 kW PEM fuel cell[J]. *International journal of hydrogen energy*, 2010, 35(17): 9291-9297.
- [6] L. N. Cleemann, "Technical University of Denmark," 2 February 2019. [Online]. Available: <http://www.ele.energy.dtu.dk/Research/Fuel-cells>. [Accessed 23 April 2020].
- [7] Mishler J, Wang Y, Mukundan R, et al. Probing the water content in polymer electrolyte fuel cells using neutron radiography[J]. *Electrochimica acta*, 2012, 75: 1-10.
- [8] Hermann A, Chaudhuri T, Spagnol P. Bipolar plates for PEM fuel cells: A review[J]. *International journal of hydrogen Energy*, 2005, 30(12): 1297-1302.
- [9] Mishler J, Wang Y, Lujan R, et al. An Experimental Study of Polymer Electrolyte Fuel Cell Operation at Sub-Freezing Temperatures[J] *ournal of The Electronical Society*, 2013 160 (6) 514-521.
- [10] Office of Energy efficiency & renewable energy, "Parts of a Fuel Cell," [Online]. Available: <https://www.energy.gov/eere/fuelcells/parts-fuel-cell>. [Accessed 14 4 2020].
- [11] Holdcroft S. Fuel cell catalyst layers: a polymer science perspective[J]. *Chemistry of materials*, 2014, 26(1): 381-393.
- [12] Ryan O'Hayre, Suk-Won Cha, Whitney Colella, Fritz B. Prinz, *Fuel Cell Fundamentals 3rd Edition*, Wiley, 2016.
- [13] Baroutaji A, Carton J G, Sajjia M, et al. Materials in PEM fuel cells[J]. 2015
- [14] Chanpeng W, Khunatorn Y. The effect of the input load current changed to a 1.2 kW PEMFC performance[J]. *Energy Procedia*, 2011, 9: 316-325.
- [15] "Expert system," [Online]. Available: <https://expertsystem.com/machine-learning-definition/>. [Accessed 6 May 2020].
- [16] Domingos P. A few useful things to know about machine learning[J]. *Communications of the ACM*, 2012, 55(10): 78-87.
- [17] Abu-Mostafa Y S, Magdon-Ismail M, Lin H T. *Learning from data*[M]. New York, NY, USA: AMLBook, 2012.

- [18] Wang Y, Basu S, Wang C Y. Modeling two-phase flow in PEM fuel cell channels[J]. *Journal of Power Sources*, 2008, 179(2): 603-617.
- [19] Zhang G, Wu J, Wang Y, et al. Investigation of current density spatial distribution in PEM fuel cells using a comprehensively validated multi-phase non-isothermal model[J]. *International Journal of Heat and Mass Transfer*, 2020, 150: 119294.
- [20] Das P K, Li X, Liu Z S. A three-dimensional agglomerate model for the cathode catalyst layer of PEM fuel cells[J]. *Journal of Power Sources*, 2008, 179(1): 186-199.
- [21] Benner J, Mortazavi M, Santamaria A D. Numerical Simulation of Droplet Emergence and Growth from Gas Diffusion Layers (GDLs) in Proton Exchange Membrane (PEM) Fuel Cell Flow Channels[C]//ASME 2018 International Mechanical Engineering Congress and Exposition. American Society of Mechanical Engineers Digital Collection, 2018.
- [22] Carrere P, Prat M. Impact of non-uniform wettability in the condensation and condensation-liquid water intrusion regimes in the cathode gas diffusion layer of proton exchange membrane fuel cell[J]. *International Journal of Thermal Sciences*, 2019, 145: 106045
- [23] Cho J, Oh H, Park J, et al. Effect of the micro porous layer design on the dynamic performance of a proton exchange membrane fuel cell[J]. *International journal of hydrogen energy*, 2014, 39(1): 459-468.
- [24] Randive P, Dalal A, Mukherjee P P. Probing the influence of superhydrophobicity and mixed wettability on droplet displacement behavior[J]. *Microfluidics and nanofluidics*, 2014, 17(4): 657-674.
- [25] Liu X, Peng F, Lou G, et al. Liquid water transport characteristics of porous diffusion media in polymer electrolyte membrane fuel cells: A review[J]. *Journal of Power Sources*, 2015, 299: 85-96.
- [26] Qin C Z, Hassanizadeh S M. A new approach to modelling water flooding in a polymer electrolyte fuel cell[J]. *international journal of hydrogen energy*, 2015, 40(8): 3348-3358.
- [27] S. sharma, "Data Science Central," 8 August 2017. [Online]. Available: <https://www.datasciencecentral.com/profiles/blogs/artificial-neural-network-ann-in-machine-learning>. [Accessed 8 5 2020].
- [28] Kalogirou S A. Artificial neural networks in renewable energy systems applications: a review[J]. *Renewable and sustainable energy reviews*, 2001, 5(4): 373-401.
- [29] Chávez-Ramírez A U, Muñoz-Guerrero R, Durón-Torres S M, et al. High power fuel cell simulator based on artificial neural network[J]. *International Journal of Hydrogen Energy*, 2010, 35(21): 12125-12133.
- [30] Saengrung A, Abtahi A, Zilouchian A. Neural network model for a commercial PEM fuel cell system[J]. *Journal of Power Sources*, 2007, 172(2): 749-759.
- [31] Bhagavatula Y S, Bhagavatula M T, Dhathathreyan K S. Application of artificial neural network in performance prediction of PEM fuel cell[J]. *International journal of energy research*, 2012, 36(13): 1215-1225.

- [32] Lee W Y, Park G G, Yang T H, et al. Empirical modeling of polymer electrolyte membrane fuel cell performance using artificial neural networks[J]. *International Journal of Hydrogen Energy*, 2004, 29(9): 961-966.
- [33] Seyhan M, Akansu Y E, Murat M, et al. Performance prediction of PEM fuel cell with wavy serpentine flow channel by using artificial neural network[J]. *International Journal of Hydrogen Energy*, 2017, 42(40): 25619-25629.
- [34] Wang Y, Seo B, Wang B, et al. Fundamentals, Materials, and Machine Learning of Polymer Electrolyte Membrane Fuel Cell Technology[J]. *Energy and AI*, 2020, In press.
- [35] Lu J B, Wei G H, Zhu F J, et al. Pressure Effect on the PEMFC Performance[J]. *Fuel Cells*, 2019, 19(3): 211-220.
- [36] Zhang J, Tang Y, Song C, et al. PEM fuel cell relative humidity (RH) and its effect on performance at high temperatures[J]. *Electrochimica Acta*, 2008, 53(16): 5315-5321.
- [37] Kim S, Hong I. Effects of humidity and temperature on a proton exchange membrane fuel cell (PEMFC) stack[J]. *Journal of Industrial and Engineering Chemistry*, 2008, 14(3): 357-364.
- [38] Guvelioglu G H, Stenger H G. Flow rate and humidification effects on a PEM fuel cell performance and operation[J]. *Journal of Power Sources*, 2007, 163(2): 882-891.
- [39] *Distributed generation systems: design, operation and grid integration*[M]. Butterworth-Heinemann, 2017.
- [40] Wang Y, Feng X. Analysis of the Reaction Rates in the Cathode Electrode of Polymer Electrolyte Fuel Cells II. Dual-Layer Electrodes[J]. *Journal of The Electrochemical Society*, 2009, 156(3): B403-B409.
- [41] Wang Y, Chen K S. Effect of spatially-varying GDL properties and land compression on water distribution in PEM fuel cells[J]. *Journal of The Electrochemical Society*, 2011, 158(11): B1292-B1299
- [42] Wang Y, Chen K S. *PEM fuel cells: thermal and water management fundamentals*[M]. Momentum Press, 2013
- [43] Wu H W. A review of recent development: Transport and performance modeling of PEM fuel cells[J]. *Applied Energy*, 2016, 165: 81-106.
- [44] Demuren A, Edwards R L. *Modeling Proton Exchange Membrane Fuel Cells—A Review*[M]//50 Years of CFD in Engineering Sciences. Springer, Singapore, 2020: 513-547..
- [45] Fadzillah D M, Rosli M I, Talib M Z M, et al. Review on microstructure modelling of a gas diffusion layer for proton exchange membrane fuel cells[J]. *Renewable and sustainable energy reviews*, 2017, 77: 1001-1009.
- [46] Afshari E, Mosharaf-Dehkordi M, Rajabian H. An investigation of the PEM fuel cells performance with partially restricted cathode flow channels and metal foam as a flow distributor[J]. *Energy*, 2017, 118: 705-715.

- [47] Ticianelli E A, Derouin C R, Redondo A, et al. Methods to advance technology of proton exchange membrane fuel cells[J]. Journal of The Electrochemical Society, 1988, 135(9): 2209-2214.
- [48] Nielsen M. Neural Networks and Deep Learning[J]
- [49] "missinglink.ai," [Online]. Available: <https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/>. [Accessed 20 4 2020].
- [50] Aggarwal C C. Neural networks and deep learning[J]. Springer, 2018, 10: 978-3.
- [51] J. Heaton, "Heaton Research," 01 06 2017. [Online]. Available: <https://www.heatonresearch.com/2017/06/01/hidden-layers.html>. [Accessed 23 4 2020].
- [52] Bhoopal N, Madhav G V, Pathapati P R, et al. Modeling of polymer electrolyte membrane fuel cell using artificial neural networks[J]. International Journal of Recent Trends in Engineering, 2009, 2(7): 75.
- [53] Olabi A G, Casalino G, Benyounis K Y, et al. An ANN and Taguchi algorithms integrated approach to the optimization of CO2 laser welding[J]. Advances in Engineering Software, 2006, 37(10): 643-648.

Appendix

1. code for ANN construction

```
import numpy as np
import csv

fc_input = []
with open("FC_2_Data_Input.csv") as csvfile:
    reader = csv.reader(csvfile, quoting=csv.QUOTE_NONNUMERIC) # change contents to floats
    for row in reader: # each row is a list
        fc_input.append(row)
fc_input = np.asarray(fc_input)

fc_output = []
with open("FC_2_Data_Output.csv") as csvfile:
    reader = csv.reader(csvfile, quoting=csv.QUOTE_NONNUMERIC) # change contents to floats
    for row in reader: # each row is a list
        fc_output.append(row)
fc_output = np.asarray(fc_output)

fc_test_input = []
with open("FC_Test_Input.csv") as csvfile:
    reader = csv.reader(csvfile, quoting=csv.QUOTE_NONNUMERIC) # change contents to floats
    for row in reader: # each row is a list
        fc_test_input.append(row)
fc_test_input = np.asarray(fc_test_input)
fc_test_input = fc_test_input[:180,:]
```

```

X = fc_input
Y = fc_output
Y_true = Y
Z = fc_test_input

X_a = 0.1
X_b = 0.9
X_range = np.max(X,axis=0)-np.min(X,axis=0)
X_min = np.min(X,axis=0)
X = ((X_b-X_a)*(X - X_min) / X_range)+X_a

Y = np.log10(Y)
Y_a = 0.1
Y_b = 0.9
Y_range = np.max(Y,axis=0)-np.min(Y,axis=0)
y_log_true_min = np.min(Y,axis=0)
# Y = Y-np.min(Y,axis=0)/Y_range
Y = ((Y_b-Y_a)*(Y - np.min(Y,axis=0)) / Y_range)+Y_a

Z = ((X_b-X_a)*(Z - X_min) / X_range)+X_a

# Above portion is for loading data only

import tensorflow as tf
import config
import matplotlib.pyplot as plt

class train:

```

```

def __init__(self, X, Y):
    print("")
    print("Training Neural Networks. This might take a while...")
    print("")
    self.X = X
    self.Y = Y
    self.data = np.concatenate((X,Y),axis=1)
    self.n, self.xdim = self.X.shape
    self.ydim = self.Y.shape[1]
    self.x_train, self.y_train, self.x_valid, self.y_valid = self.splitData()
    self.model = self.trainXY()

def splitData(self, train_ratio=0.7, valid_ratio=0.3):
    k1 = np.int(np.floor(self.n * train_ratio))
    k2 = np.int(np.floor(self.n * (train_ratio+valid_ratio)))
    if k2>self.n:
        k2 = self.n
    d = self.data
    np.random.shuffle(d) # select data randomly
    xd = self.xdim
    x_train = d[:k1,:xd]
    y_train = d[:k1,xd:]
    x_valid = d[k1:k2,:xd]
    y_valid = d[k1:k2,xd:]
    return x_train, y_train, x_valid, y_valid

def trainXY(self): # train with tensorflow
    model = tf.keras.models.Sequential([
        tf.keras.layers.Dense(config.n_neurons, input_shape=(self.xdim,), activation=config.activation),

```

```

        tf.keras.layers.Dense(config.n_neurons,
input_shape=(config.n_neurons,),activation=config.activation),
        tf.keras.layers.Dense(self.ydim, input_shape=(config.n_neurons,), activation='linear']])
model.compile(optimizer=config.optimizer,
              loss=config.loss,
              metrics=config.metric)
history_1 = model.fit(self.x_train, self.y_train, epochs = config.epochs, verbose=1)
model.evaluate(self.x_valid, self.y_valid, verbose=2)
weights = model.get_weights()
np.savetxt('weight.csv', weights, fmt='%s', delimiter=',')
# print(weights)

#plot
plt.plot(history_1.history['loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epochs')
plt.legend(['train', 'test'], loc='upper left')

plt.show()
return model

# train
model = train(X,Y)

# predict
y_pred = model.model(Z)
y_p_1=tf.keras.backend.eval(y_pred)
y_p_2 = (y_p_1-0.1)/0.8
y_p = np.power(10,y_p_2*Y_range+y_log_true_min)

```

2. Code for parameters initialization

```
# configs for tf
activation = 'tanh' # choose from 'tanh' and 'sigmoid'
n_neurons = 30 # number of neurons in hidden layer
epochs = 5000
loss = 'mean_squared_error'
metric = []
optimizer = 'adam'
```