A Connectionist Implementation of Identical Elements

Paul Munro (pmunro@mail.sis.pitt.edu)
Jianghua Bao (jib2@pitt.edu)
Department of Information Science and Telecommunications
School of Information Sciences
Pittsburgh, PA 15260 USA

Abstract

In training two networks on tasks that are different on the surface, but similar, or even isomorphic, at a higher level of description, similarities between the network solutions are plausible if not expected. Such similarities tend to become evident when two networks with shared weights are trained on similar tasks. After training, the shared weights were used as part of a third network that was trained on a third task similar to the first two. This "head start" results in significantly shorter training times than a network that starts with random weights. Shared hidden unit response profiles were analyzed across networks trained on structurally analogous tasks to reveal parallel, but nonidentical features.

Background

Experimental evidence for transfer was described early in the last century by Thorndike & Woodworth (1901), who proposed the *identical elements theory* of transfer. The theory proposes that two different "mental functions" may share cognitive structures in their processing.

Two events may seem similar or identical at an abstract level of description or representation, yet very different on the surface. Analogical reasoning depends upon feature abstraction at a level that is sufficient to subsume concrete features of multiple situations. Using this higher-level representation, a mapping can be established between features in the concrete instances (Gentner, 1983). Such an analogical mapping is a mechanism, probably not unique, for transfer of knowledge from one domain to another.

To the extent that the structures underlying analogy can be represented as patterns, the connectionist framework is a natural approach (e.g., Holyoak & Thagard, 1988; Halford et al, 1993). A distributed approach to structural mapping was put forward by Hummel & Holyoak (1994). Several symbolic-connectionist models have addressed the processing of structural mapping in the context of language processing (e.g., Mitchell, 1993).

Some other connectionist models have addressed transfer phenomena without explicitly addressing analogy; that is, these models do not focus on the development of a structural mapping. Pratt et al. (1991) demonstrated that using weights from a network trained on one task to initialize a network to be trained on a similar task could improve learning performance. The Multi-task Learning (MTL) approach of Caruana (1997) demonstrated that a single network trained simultaneously on multiple tasks can learn faster and converge to a state of lower error than if it is trained on one of the tasks in isolation. This reinforcement among the tasks depends on whether the individual tasks can benefit from common intermediate features.

Identical Elements in a Network

The central idea introduced in this paper is that a network can learn high-level features common to different tasks, and use them in learning a novel task, assuming it has similar high-level attributes. This *Identical Elements Neural Network* (IENN) approach uses simultaneous training of multiple tasks, but differs from MTL in that the tasks do not share a common input. In principle, the tasks can have inputs that have very different coding, dimensionality, etc. Some of the weights learned from simultaneous training are then used to initialize a network that is trained on a novel task. These transferred weights remain fixed.

The IENN approach requires more than one hidden layer. Only the weights between hidden layers are shared by multiple tasks, thus allowing different numbers of input and output units for the individual tasks. This also enables a novel task to use the shared weights without interfering with performance on the original tasks.[1]

Network Architecture and Training

Feedforward networks were trained using backprop (cross-entropy error). The networks were strictly layered with two hidden layers. Two inputs activated all units in the first hidden layer (H_1), which subsequently activated all units in the second hidden layer (H_2), which activated the sole output unit. Each hidden layer had 16 units. Training was conducted in one of three modes: single network with random initial weights, single network with prespecified weights between the two hidden layers, and two networks with yoked (shared) weights between the hidden layers.

Mode 1: Single Network with random initial weights.

This was the control condition for each study. Network weights and biases were initialized to random values, and trained to a time/error criterion.

[1] An earlier version of this system appeared as a one page member's abstract at Cogsci '96 (Munro, 1996).

Mode 2: Single Network with fixed H_1-H_2 weights.

Here, weights between the two hidden layers and bias values of both hidden layers were initialized to values that were results of training on a different task. These weights and biases were not modified during training. Note that these weights were used to backpropagate errors to H_1 for modification of the *input*-H_1 weights.

Mode 3: Two networks with shared weights.

In this mode, two networks are trained on different tasks using the same values for the weights between the hidden units and the same bias values for units in both H_1 and H_2. The weight parameters for the two networks from the input to H_1 are independent, as are the weight and bias parameters for the output units. Patterns are not presented simultaneously to the two networks; i.e. the shared layer processes patterns from either task during a given training step, not from both simultaneously.

For each experiment, a network was trained on a target task (T) in three conditions:

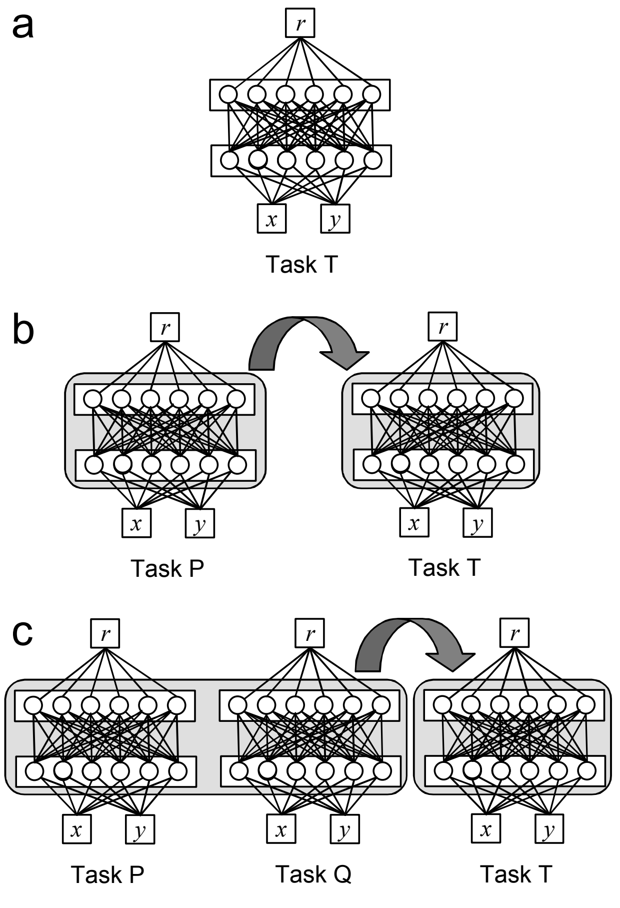

Figure 1. a: Control Condition (Mode 1). b: Pretraining with a single task (Modes 1,2) c: Pretraining with two tasks (Modes 3,2).

Control Condition (CC): Network weights and biases are randomly initialized (Mode 1), and trained on a task for comparison with the two experimental conditions below (Figure 1a).

Experimental Condition I (EC I): Pretraining with a single task. A network is trained with random initial weights (Mode 1) on a preparatory task (P). The H_1 - H_2 weights and biases are then used as initial parameters for a network trained on T in Mode 2 (Figure 1b).

Experimental Condition II (EC II): Pretraining with two simultaneous tasks. Two networks are trained with shared H_1 - H_2 weights and biases on two different tasks (Mode 3), which are then used to initialize a network to be trained on T in Mode 2 (Figure 1c).

The Tasks

The tasks were all classification of regions in a two dimensional space. The two inputs were x-y coordinate values in the unit square. The output unit was trained using the cross-entropy error function to classify a region; the target value was 1 for inputs inside the region and 0 for inputs outside. Based on the intuition that topological similarity is more abstract than geometrical similarity, the tasks were chosen to be topologically equivalent but geometrically different.

Study 1. Closed Regions.

All tasks here were simple convex closed regions (Figure 2).

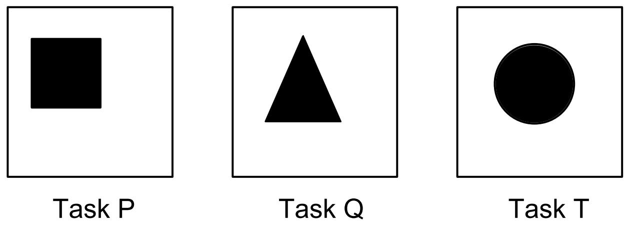

Figure 2.

Study 2. Annular Regions.

All tasks in this study were annular, having two similarly shaped regions, one completely inside the other. The target region was defined as points between the two boundaries (Figure 3).

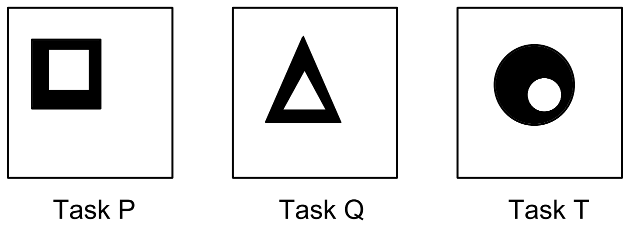

Figure 3.

Results

The test set for the tasks was the entire input space ($[0,1]^2$) sampled with a resolution of 0.01. Errors were averaged over all 10000 points. In order to analyze the details of network computation, response profiles were plotted for the output unit on each task as well as the units at both H_1 and H_2. These correspond to receptive field plots from studies of neurons in visual cortex.

Study 1. Closed Regions.

Error curves were generated for the duration of the simulation. Figure 4 shows error curves for Task T in the 3 conditions: control (a), using H_1 and H_2 weights from a network trained on task P alone (b) and using H_1 and H_2 weights from shared training on Tasks P and Q together (c). Note that all three curves converge to approximately the same low error value, but that transfer accelerates the error reduction.

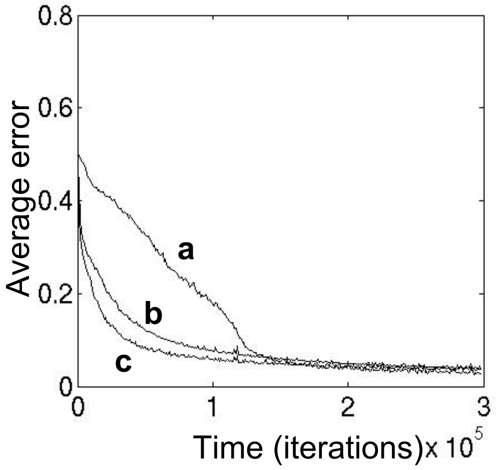

Figure 4. Error curves for the closed region. Each curve is an average over 5 trials. **a.** Task T alone (control) **b.** Task T using weights transferred from Task P network (EC I). **c.** Task T using weights transferred from Task P/Q network (EC II).

Response profiles generated for the output unit at 10000 iterations and 300000 iterations illustrate the correspondence between average error and the shape of the boundary predicted by the network (Figure 5).

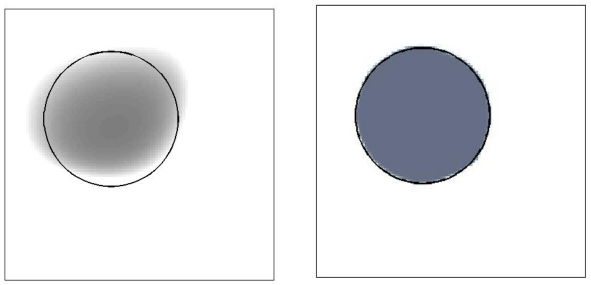

Figure 5. Output unit response profiles at 10000 iterations (left) and 300000 iterations (right).

Response profiles of hidden units at both H_1 and H_2 were also examined. Sample response profiles from networks trained on tasks P and Q with shared weights are shown in Figures 6 and 7. Note that the left and middle profiles in the two figures are from the same H_1 and H_2 units in the shared weight matrix. The response profiles are different because the *input*-H_1 weights are trained independently. But for both tasks, the boundary in the H_1 profile approximately corresponds to a significant portion of the H_2 boundary. Also, note that the shading is reversed; the 1 (white side) is opposite in the H_1 and H_2 profiles. This reversal occurs for both figures since they share the same *negative* weight value between the hidden layers.

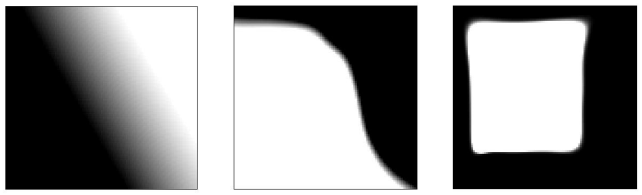

Figure 6. Response profiles from a network trained on the square task (P). Left: an H_1 unit. Middle: an H_2 unit. Right: the output unit.

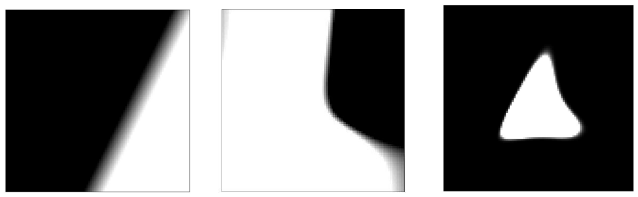

Figure 7. Response profiles from a network trained on the triangle task (Q). Left: an H_1 unit. Middle: an H_2 unit. Right: the output unit. The H_1 unit and H_2 unit are in the same position as those in Figure 6 with respect to the shared weights.

Study 2. Annular Regions.

The three error traces in Figure 8 correspond to those in Figure 4. Note that these traces to not seem to converge to a common value. In the control condition the curve seems to keep decreasing (a), while the curves for the experimental conditions appear to converge to distinct values.

Response profiles of the output unit for tasks P, Q, and T show a progression in task complexity during the course for learning that resembles that seen as one progresses through layers of the network (Figure 9). The shapes of the response profiles in the early/intermediate stages (left column) do not all show the same degree of complexity. The square annulus seems to be at a more advanced stage than the triangle and the circle. Of course, this can be attributed to the differences in the tasks, but the differences are also due to the random weight initialization.

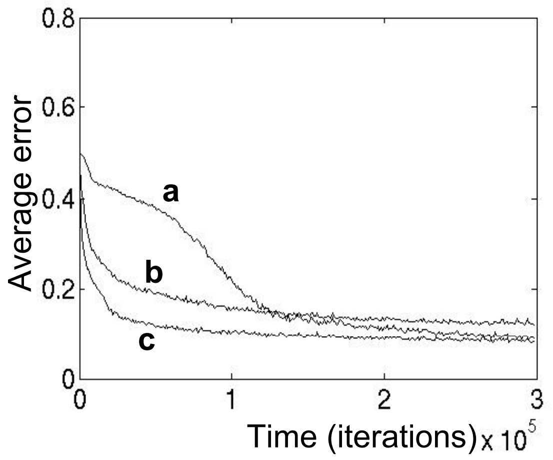

Figure 8. Error curves for the annulus. Each curve is an average over 5 trials. **a.** Task *T* alone (control) **b.** Task *T* using weights transferred from Task *P* network (EC I). **c.** Task *T* using weights transferred from Task *P/Q* network (EC II).

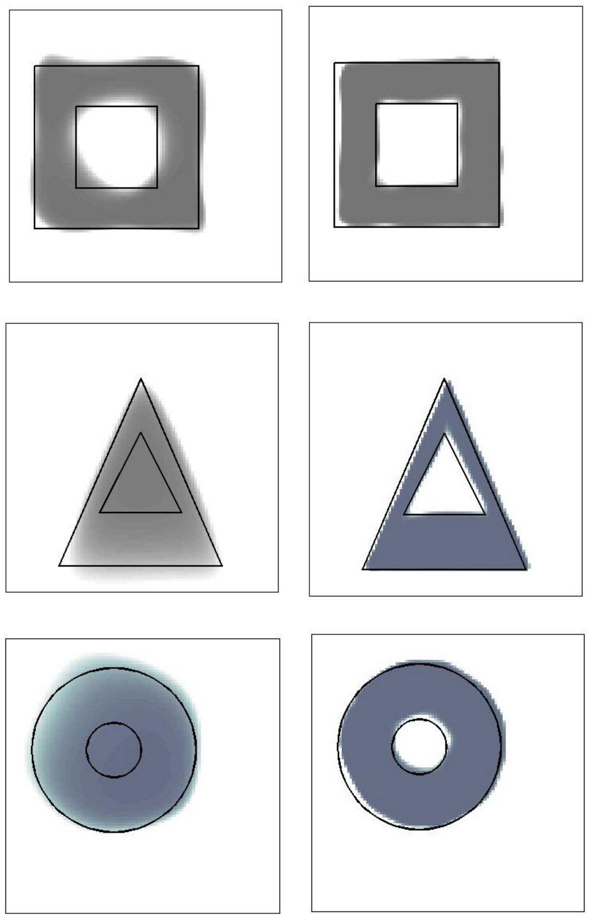

Figure 9. Intermediate response profiles for the three annular patterns (left) and final response profiles (right). The sharp curves indicate the boundaries used to generate the training data.

Discussion

The weights in feed-forward networks can encode task information that can be utilized by related tasks. The results demonstrate that weights shared between networks trained on similar tasks encode the common aspects of the tasks. Thus, the shared weights come to encode the common abstract features when simultaneously trained on task pairs that are similar at a high level of description and dissimilar on the surface.

Examination of the response profiles from certain units in both hidden layers shows a progression in complexity similar to the receptive fields of simple and complex cells in visual cortex, first described by Hubel & Wiesel (1962). Units in the H_1 layer can only form linear boundaries. Complexity at the output layer is determined by the task and constrained by the network structure. Examination of H_2 response profiles seems to indicate that they compute functions of complexity that is intermediate between H_1 and H_2.

Unlike observations in visual cortex, however, the response properties here are task dependent, since the input-hidden weights are independently trained for the different tasks. Response profiles from a given H_2 unit can be composed from H_1 response profiles for a given task (as in Figures 6 and 7). Since the same H_1 - H_2 weights and biases are used to compute a different task, the response profile for the same H_2 unit should be composed from the same H_1 units, even though the response profiles are very different.

The error traces from Study 2 (Figure 4) for the three experimental conditions shows that transfer from a single task speeds learning, but that the control condition eventually catches up. Transfer from a set of weights that were shared by tasks learning two tasks that were not identical but had features in common with each other as well as the target task, clearly increased learning speed even more. This supports our hypothesis that shared weights between networks trained on multiple tasks tend to encode features shared by the tasks. Thus, this may be a mechanism for abstraction of high level features and transfer to other tasks.

The error traces for Study 2 (Figure 8) also show a progressive speedup from the control condition to the single task pretraining to the double task pretraining. However, there are also some important differences. Rather than converging to the same error, these curves converge to different values. This difference is explained by the increase in complexity relative to Study 1. When all network parameters are adjustable, the network can find a state of minimum error. However for this more complex task, the adjustable parameters are not sufficient to reduce the error as much as in the control case. So while pretraining with a single task (Figure 8, trace b) gives faster learning, the final state has a higher average error. With a larger number of hidden units, the system would be expected to reach an error state as low as the control condition. Presumably for that system, there would exist another task of even higher complexity that would "break" the system for fixed weights between the two hidden layers.

Pretraining on simultaneous tasks (EC II) shows even faster learning than does pretraining on a single task (Figure 8, traces b and c). It also gives a lower error in the final state, however not as low as the control condition. This can be explained as follows. As in EC I, the information in the pretrained weights is initially beneficial, but eventually they limit learning since they are not modifiable. The better asymptotic performance of EC II is consistent with the notion that the shared weights encode more abstract features that are more relevant to the target task.

Of course, one should tread very lightly when making inferences about neural function based on a connectionist model. With this caveat in mind, let us consider what insights can be extracted from IENN with respect to how the brain implements analogical reasoning. Examination of the simulations suggests that an individual neuron may play a role in different computations, and that its response properties may change drastically from one computation to another. However, if the computations have similarities at some level of description, they can take advantage of the same neural circuits.

Furthermore, we make the following conjecture. If two neural circuits are learning different tasks, overlapping components of those circuits will tend to encode common features of the tasks.

This idea does not explain how one of the input streams to the shared weights and synapses does not generate a response in the wrong output stream. For example if this model is taken literally, an input to the channel trained on Task *T* could elicit a response from the output units trained on Task *Q*. On the one hand, this can probably be handled by a mechanism (neural, of course) for maintaining the context of the input. Perhaps some direct connections from the input to the output would be sufficient. On the other hand, some crossover is potentially a very powerful mechanism for establishing corresponding elements of analogical structures. Ideally, there is some crossover that will enable reasoning about one situation to inform reasoning about another situation, but the crossover is controlled somehow so that context is preserved.

A very similar network architecture was constructed by Dienes et al. (1999) using a modification of Elman's (1990) simple recurrent network, or SRN. Here, an SRN with two hidden layers was trained to learn an artificial grammar (as in Cleeremans et al., 1989). This research demonstrated the usefulness of weights from a previous task. In their experiments, the two tasks were structurally itentical. They demonstrate an increase in performance by the second network over the first. Interestingly, comparison of the modifiable layers from the two networks shows correlated but non-identical weight matrices.

Conclusions

Consider the following framework for neural implementation of analogical reasoning:

- Different tasks are learned in cortical circuits or pathways that may partially overlap. Some of the neurons in overlapping circuits come to compute features that support both tasks (high-level abstraction of concepts).
- These same features may have general utility as computational elements in a larger class of tasks (transfer to novel tasks).
- Thus, the influences that determine a neuron's computational properties can be described as not only "bottom-up" and "top-down", but also "cross-task".

Like the MTL work of Caruana (1997), the IENN model demonstrates that units serving multiple tasks can develop response properties that serve both tasks from simultaneous top-down influences. In the brain, it is expected that the overwhelming majority of units (if not all) are involved in multiple tasks.

The ability to recognize and apply high-level analogical correspondences across situations seems much more highly developed in humans than in other species. If the IENN account is a model of the neurobiological underpinnings of analogical reasoning, then we should look to the prefrontal cortex as a candidate brain structure, as others have suggested (e.g. Damasio, 1989).

The IENN approach bears extension in several directions. Anticipated work includes working with transfer in simple recurrent networks (Elman, 1990), models of transfer in motor learning and spatial cognition, and implementation in a more biologically realistic system. We intend to go well beyond the work of Dienes and Altman (1999), exploring the interaction of tasks that are structurally *similar* but *non-identical*.

Acknowledgments

The authors wish to thank colleagues at the University of Pittsburgh School of Information Sciences for valuable feedback. The input of the anonymous reviewers is also acknowledged, especially for directing us to some relevant papers in the literature.

References

Caruana, R. (1997) Multitask learning. *Machine Learning, 28*:41-75.

Cleeremans, A., D. Servan-Schreiber, and J.L. McClelland. (1989) Finite state automata and simple recurrent networks. *Neural Computation*, 1:372--381.

Damasio, A. (1989) Time-locked multiregional retroactivation: A systems level proposal for the neural substrates of recall and recognition. *Cognition 33*:25-62.

Dienes, Z., Altman G. T. M., and Gao, S.-J. (1999) Mapping across domains without feedback: A neural network model of transfer of implicit knowledge, *Cognitive Science 12*:53-82.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science, 14*:179--211.

Gentner, D., (1983) Structure-mapping: A theoretical framework for analogy, *Cognitive Science 7*:155-170.

Halford, G., Wilson, W., Guo, J., Gayler, R., Wiles, J., Stewart, J. (1994). Connectionist implications for processing capacity limitations in analogies. In: K. Holyoak & J. Barnden (eds.) *Advances in connectionist and neural computation theory, vol. 2, Analogical Connections,* pp. 363--415. Norwood, NJ: Ablex.

Holyoak, K. & Thagard, P. (1989) Analogical mapping by constrant satisfaction. *Cognitive Science 13,* 295-355.

Hubel, D. H., & Wiesel, T. (1962). Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex. *J. Physiol. 160*, 106-154.

Hummel, J. & Holyoak, K. (1997) Distributed representations of structure: A theory of analogical access and mapping. *Psychological Review*, *104*, 427- 466.

Mitchell, M. (1993) *Analogy-making as Perception: A computer model*. Cambridge, MA: MIT Press.

Munro, P. (1996) Shared network resources and shared task properties. In: *Proceedings of the Eighteenth Annual Conference of the Cognitive Science Society.* Mahwah NJ: Erlbaum

Pratt, L. Y., Mostow, J., and Kamm, C. A. (1991) Direct transfer of learned information among neural networks. In: *Proceedings of the Ninth National Conference on Artificial Intelligence* (*AAAI-91*) Anaheim CA

Thorndike, E. L.., & Woodworth, R. S. (1901). The influence of improvement in one mental function upon the efficiency of other functions. *Psychological Review, 8*, 247-261.