International Conference on Neural Information Processing Systems: ICONIP'98, , pp. 691-694. Kitakyushu, Japan

A comparison of learning transfer in networks and humans

Steven Phillips
Email:stevep@etl.go.jp

Information Science Division, Electrotechnical Laboratory,
1-1-4 Umezono, Tsukuba, Ibaraki 305 Japan

ABSTRACT

Learning transfer is the improvement in performance on one task having learnt a related task. That the degree of transfer is significantly greater in humans than other primates and animals suggests it is a critical component of higher intelligence. One connectionist approach, *weight sharing*, represents common task knowledge as weighted connections shared by subnetworks dedicated to individual tasks. Although this technique permits transfer, recent analysis has shown that it does not support the same degree of transfer as humans. In this paper, several extensions are outlined, and their theoretical limits compared. The comparison points to a greater role for control mechanisms in connectionist cognitive models.
KEYWORDS: Cognition, generalization

1. Introduction

Learning transfer is the phenomenon whereby performance (as measured by the number of learning trials, for example) on one task is significantly better having learnt a structurally related task. Experiments on learning transfer typically involve a series of tasks with the same structure, but differing in stimulus materials. Subjects are tested on a task instance until some learning criterion is reached, at which point a new task is tested. If the number of learning trials to reach criterion on the subsequent task is significantly less than for the previous task, then the subject has demonstrated some degree of learning transfer.

Since the work of Harlow [4] it is known that the degree of transfer differentiates species (see [7] for a review). Rats, for instance, show almost no transfer even after hundreds of tasks, whereas humans show the greatest degree of transfer [7]. That the degree of transfer distinguishes humans from other primates and animals suggests the underlying mechanisms are critical components of intelligence.

One connectionist approach, *weight sharing*, represents common task knowledge as weighted connections shared by subnetworks dedicated to individual tasks (e.g., [5, 1]). In Hinton's simulations, a feedforward network was trained to make inferences about relationships in two isomorphic family trees. The network is required to learn the mapping from person and relation to person (e.g., John, Wife $\rightarrow$ Mary) for each family tree. The network consisted of two groups of input and output units (i.e., a group for each family) linked to common groups of hidden units. The weighted connections between hidden units code common knowledge across the two families. With this configuration the network was trained on 100 of the 104 possible person-relation pairs. In one simulation run, the network generalized to all four remaining test cases. In the other run, it generalized to three test cases.

Weight sharing appears as an elegant solution to the problem of learning transfer in networks. However, recent analysis [9, 8] showed that it was not sufficient to account for the degree of transfer in humans [2]. In this paper, I examine extensions to weight sharing and compare their theoretical limits to the psychological data.

2. Human subjects: Klein 4-group task

In a series of psychological experiments [2], human subjects exhibited rapid transfer of learning on a series of tasks sharing the structure. In one experiment, subjects are given a series of four task instances derived from the Klein-4 Group (Figure (a)), where a, b, c and d are states; and H and V are operations on states. For example, application of H to state a results in state b. When states are depicted as vertices of a square, H and V can be interpreted as horizontal and vertical transitions, respectively. Each task instance consisted for four unique three-letter strings and two unique shapes, corresponding to the states and operators in the klein group (e.g., Figure (b)). Subjects are presented with a string and a shape, and asked to predict the response string (e.g., **PEJ** and $\triangle$ predicts **BIP**).

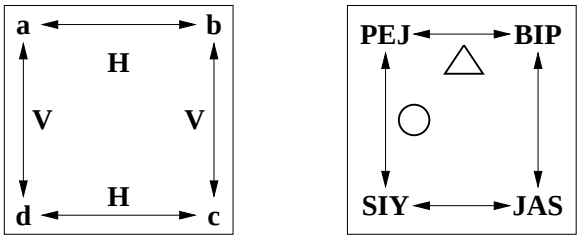

Figure 1: Klein 4-group (a) and task instance (b).

By the fourth task instance, subjects predict six of the eight responses. Furthermore, in a related experiment, subjects were also able to predict the start state, given the operator and final state; and the operator given the start and final states. In both cases, subjects predicted six of eight responses.[1]

3. Feedforward networks with shared weights

Before detailing and expanding upon an analysis of transfer initiated in [9], some justification is given for the choice of local input/output representations (i.e., single unit with activation 1, and the rest 0) for task elements used here, and in [5]. Generalization performance is sensitive to the

[1] Since the assignment of strings and shapes to states and operators is arbitrary and unique across task instances, it is not possible to predict the first two responses (see [2]).

choice of input/output representations. In an extreme case, if all elements are represented by a single real number, then arbitrarily many generalizations are possible after training on only two examples in a linear system with one variable. Although there is some similarity between elements in the Klein 4-group task (e.g., strings may share common characters), this similarity is not the basis for generalization, since the assignment of task elements is arbitrary across task instances. The use of local representations acts as a control measure to eliminate any possibility of generalization on the basis of surface (input pattern) similarity.

3.1. Summary of previous results

In this and subsequent sections, the network consists of four input and four output units representing strings for each task instance, and two input units for shapes. The inputs are connected to a common layer of hidden units, which are connected to a second common layer of hidden units, which are connected to the outputs.

Since generalization is maximized by reducing the number of free parameters, while still being able to represent a solution, the strategy was to test a network with the minimum number of hidden units. With only three hidden units in the first layer and two in the second, the network could learn one task instance but could not transfer that learning to a subsequent task [9]. Lack of transfer could have been due to a poor set of learning parameters. However, a plot of the error surface around a global minimum showed that even when only two of the weights where free the error surface for the training set was not constrained to coincide with the testing set. In others words, there were many solutions (minima) in the training space that were not solutions to the test space, and so generalization was highly unlikely. Thus it was concluded that weight sharing along could not demonstrate any degree of transfer across the isomorphic task instances [8].

4. Other techniques for improving transfer

As well as connectivity, activation and error functions also determine the shape of the error surface. Given the possibly infinite variations one can make in terms of these architectural components it is not feasible to canvas all possibilities. However, we can identify specific properties and examine their capacity to support learning transfer.

4.1. Reduced representations

The effective weight space of the network can be restricted by enforcing fewer activation states for its units (e.g., binary, rather than real valued). In the extreme case, the smallest number of identifiable states for the second hidden layer is 4 (i.e., one state for each possible response). Fewer states means that at least one state must be mapped to two different responses. Reduced representations can be implemented with a signal binary valued unit for each internal state and 0/1 weights connected to the output; or, by one real valued unit and weights with non-monotonic activation functions at the output units. In the second case, since it is not possible to partition 4 points on a line using single threshold units, double threshold units are required. Gaussian functions, for example, have two thresholds permitting each point to be separated from every other point.

In either case, how many patterns are required to learn a new task? The lower bound is four (i.e., one pattern for each possible response). Each output unit must be trained to discriminate between two types of points: the pattern it represents; and all other patterns. With fewer patterns, at least one of the output units will be trained on only one type of point (i.e., the patterns represented by the other output units). In this case, the training set provides no information about discrimination for that output unit. Therefore, the upper bound on generalization is four patterns.

4.2. Bidirectional connections

One limiting property of the feedforward network is the unidirectional nature of its flow of information. Learning the mapping between task specific input/output representations and common structural elements proceeds in one direction only: from stimulus to internal representation and then from internal representation to response. These two directions are independent. The consequence is that learning to map a task element (stimulus) to a suitable internal representation (i.e., alignment) does not permit the related generalization of mapping the aligned internal representation to the same task element, but treated as a response. Even with the most optimized internal representational space (four states) four is an upper bound on generalization.

An obvious extension to the feedforward network is to introduce bidirectional weights and linear units so that the interpretation and its inverse are learnt concurrently. Following previous analysis, this architecture assumes only four possible states at the hidden layer encoding strings, and two possible states for shapes. Since the mapping is bidirectional there is one unit per state and binary valued bidirectional weighted connections between each pair of units.

Suppose the network has been trained on two patterns for the second task: (A,H) $\rightarrow$ B, and (D,V) $\rightarrow$ A. Now, for the sake of identifying bounds on generalization, suppose each task element in the first two patterns has been aligned to its corresponding structural element. That is, the five task elements (A,B,D,H,V) represented at the input/output units will have been mapped to their corresponding internal representations. However, the sixth element C does not appear in the first two patterns, so weights from its corresponding input/output unit will not have been trained. There are four possible states that element C can be mapped to, only one of which is correct. Consequently, when given the third pattern: (C,H) or (C,V), there is no guarantee the network will make the correct prediction. The network requires training on the third pattern to identify the correct alignment for element C. Hence the upper bound on generalization is five, which is still less than human performance.

4.3. Non-local weight modification

The limitation of the previous techniques is that the mapping of each task element was considered independent of the other task elements. Human subjects can make use of the *one-to-one correspondence* principle to improve generalization. That is, each element of a task instance maps to one and only one unique structural element. For example, if we know from the first two stimulus-response trials that elements A, B and D map to particular structural elements, then we can infer that element C maps to the last remaining unaligned structural element, even though C does not appear in the first two trials. Use of the one-to-one correspondence principle permits the correct prediction after two training examples.

This principle can be implemented by modifying the weight updating function. A purely local weight update function does not implement one-to-one correspondence, since the weights linking the unaligned task and structure elements are not updated during training on the first two patterns. Consequently, a third training pattern containing the unaligned element C is required to align all elements. Yet, one-to-one correspondence can be enforced using a non-local weight update function[2] (e.g., [6]).

Suppose a group of input/output units (I_a, I_b, I_c, I_d) for representing task stimuli, and a group of hidden units (H_a, H_b, H_c, H_d) representing corresponding structural elements. Following along the lines of Hummel and Holyoak's model, a procedure for ensuring one-to-one correspondence is outlined in the following example:

- Suppose input pattern A is presented at the input/output units. The unit I_a becomes active and propagates its activation to the hidden units. Mutual inhibition between hidden units ensures only one unit (say, H_a) is most active.
- Weights linking co-active units have their weights increased. The outgoing weights from a unit are normalized. Normalization increases the weight between the two most co-active units (I_a-H_a) at the expense of the other outgoing weights. It ensures that unit I_a only activates the hidden unit H_a (i.e., task element A is aligned to one and only one structural element). However, at this stage unit H_a could be activated by other input/output units.
- To constrain hidden units from being activated by other input/output units, incoming weights to a common unit are also normalized. As with outgoing weights, normalization increases one weight at the expense of others. In this case, it has the effect of ensuring that a structural element corresponds to one and only one task element.
- If weights are bounded above by one, and normalized to sum to zero [6], then weights between aligned pairs (e.g., I_a-H_a) will approach one and non-aligned pairs (i.e., I_i-H_a and I_a-H_j, where $i, j \neq a$) will be negative.

Suppose the first two trials consisted of stimulus-response patterns: (A,H) $\rightarrow$ B; and (D,V) $\rightarrow$ A. By the end of the second trial, all three elements have been aligned to their corresponding structural elements. For the sake of exposition, suppose elements are aligned by setting weights I_a-H_a, I_b-H_b and I_d-H_d to one. By virtue of normalization, all other weights into or out of these six units will have small negative values. This leaves the single I_c-H_c connection. Although, this weight was not directly trained by the presence of the C element (since it does not appear in the first two trials), it is indirectly trained by its absence. All other connections to the unit I_c will have small negative weights, since they are connected to non-aligned units (i.e., hidden units already aligned to other input/output units). Similarly, all other connections to the unit H_c will have small negative weights. Assuming a small positive starting weight for connection I_c-H_c, then normalization will force this value to become strongly positive. On the third trial, when element C is presented to the network it will already be aligned to the structural element represented at H_c, resulting in correct prediction of the response element. After training on two patterns, non-local weight modification permits generalization to the other six patterns.

[2] In the sense that weight change is also a function of other weights in the same layer (e.g., normalization).

4.4. Omnidirectionality

In the Klein 4-group tasks, subjects also showed the ability to predict the missing operator (shape) given the initial and final states, and the initial state given the operator and final states. Bidirectional links between task and structure elements are not sufficient to make this generalization, since the mapping between structure elements is still unidirectional (i.e., from initial state and operator to final state). In fact, the first test from initial and final states to operator is meaningless to a feedforward network without input units to represent the initial state and output units to represent the operator.

In general, omnidirectionality between related elements is supported by tensor networks [10, 3]. For full implementation details the reader is referred to these articles. Here, only an outline of a tensor-based architecture is provided.

Based on the work of [10], [3] showed how tensor networks can support relations as the sum of the tensor outer products of vectors representing each tuple element. A rank n tensor (T^n) is constructed by taking the outer product of vectors representing each tuple element of an n-ary relational instance. For example, the ternary relational instance (b, v, c) is represented by the rank 3 tensor $T^3 = \vec{b} \otimes \vec{v} \otimes \vec{c}$. The inner product is used to extract tuple elements, as in $\vec{b} \otimes \vec{v} \odot T^3 = \vec{c}$.

For example, $R_k(I, O, F) = \{(b, v, c), \ldots\}$ corresponds to the tensor $T_k = \vec{b} \otimes \vec{v} \otimes \vec{c} + \ldots$, under the assumption that element vectors are mutually orthonormal. Also, $T_i = B\vec{I}P \otimes \vec{b} + J\vec{A}S \otimes \vec{c} + \vec{\bigcirc} \otimes \vec{v} \ldots$ represents the interpretations of strings and shapes to structural elements, deduced from the first two patterns. The response to the third stimulus pair is predicted as follows:

1. $B\vec{I}P \odot \vec{T_i} \rightarrow \vec{b}$, $\vec{\bigcirc} \odot \vec{T_i} \rightarrow \vec{v}$;
2. $\vec{b} \otimes \vec{v} \odot \vec{T_k} \rightarrow \vec{c}$; and
3. $\vec{T_i} \odot \vec{c} \rightarrow J\vec{A}S$.

A tensor based network captures the same degree of generalization as subjects at the fourth task by embodying some of the properties of relational systems. Those properties are omnidirectional access to relational elements, and representation of the relational structure via groups of units dedicated to particular relational arguments (roles). Given the same two first trial tuples, the corresponding tensor operations supporting omnidirectional prediction are:

1. $B\vec{I}P \odot \vec{T_i} \rightarrow \vec{b}$, $\vec{T_i} \odot J\vec{A}S \rightarrow \vec{c}$;
2. $\vec{b} \otimes \vec{T_k} \otimes \vec{c} \rightarrow \vec{v}$; and
3. $\vec{T_i} \odot \vec{v} \rightarrow \vec{\bigcirc}$.

Tensors are one mechanism for implementing omnidirectional access to knowledge elements. With respect to the Klein 4-group task, after the first two trials they permit prediction for the remaining six final state trials, plus six operator trials, plus six initial state trials.

5. Summary: Five techniques for transfer

Weight sharing, in itself, does not result in any transfer of learning. Essentially, with real-valued activations and weights, there are too many possible internal states. The number of internal states at the second hidden layer can be reduced to a minimum of four. In this case, learning transfer is increased to four patterns on the subsequent task. However, four is the absolute upper limit with a purely feedforward network. Further increase in learning transfer must make use of the bidirectional relationship between task and structure elements. The use of bidirectional connections between input/output units and hidden units permits five patterns to be correctly predicted. However, bidirectional connections alone do not enforce the one-to-one correspondence principle between task and structure elements. This principle can be enforced by a non-local weight update function, permitting the maximum transfer in one direction (i.e., from initial state and operator to final state) of six patterns. Yet, each task is a relation (not just a function), permitting elements to be predicted in other directions (e.g., predict the operator that results in a transition between two given states). Omnidirectional access to task elements can be implemented by omnidirectional connections between hidden units, which encode common structural information. This architectural component permits six final states to be predicted plus six operators plus six initial states (6^{++}), matching the capability of humans. These results are summarized in Table 1.

Table 1: Architectural components and their impact on learning transfer as measured by the number of correctly predicted patterns on the subsequent task.

Architectural component	Transfer (Max. 6^{++})
Weight sharing	0
Reduced representations	4
Bidirectional i/o connections	5
Non-local weight modification	6
Omnidirectional connections	6^{++}
Humans	6^{++}

6. Discussion: Nature of knowledge representation

Much of connectionist research has focussed on the issue of representation in the sense of data values (e.g., local versus distributed vectors). The first two techniques are examples. They result in reduced descriptions (vector representations) of task knowledge so as to increase generalization. The control process remains largely unchanged (i.e., forward propagation of activation, and backward propagation of error signal).

The other three components are essentially process oriented in the sense that they concern the flow of information, rather than its value. With bidirectional connections, for example, activation flows both from input/output to hidden units and from hidden to input/output units. One may remark that bidirectional connections play the same role of reduced descriptions as shared weights in that two previously separate weight matrices are compressed to one. But the difference is that bidirectional units have a temporal (serial) component. It makes no sense for the hidden units to activate the input/output units until they themselves have been activated by the input/output units. Tensor networks also assume additional connectivity to place inputs onto the appropriate axis of the tensor. The STAR model of analogical reasoning [3] and the LISA model of analogical reasoning and schema induction [6] are two examples of networks with more complex control mechanisms.

7. Conclusion

Control affords a particular form of generalization. From the five techniques sketched here, a pattern begins to emerge. Increasing generalization comes by overloading network resources (e.g., units, weights dedicated to more than one related task). But reuse of resources requires co-ordination (control).

In the past, connectionists have looked for more powerful learning algorithms while keeping the basic feedforward/feedback control structure the same. The work here suggests a greater need for more complex control mechanisms, which is evident in more recent models. One can speculate that the advance of humans over other animals is the control, rather than the distribution of information.

References

[1] D J Chalmers. Syntactic transformations on distributed representations. *Connection Science*, 2:53–62, 1990.

[2] G S Halford, J D Bain, M T Maybery, and G Andrews. Induction of relational schemas: Common processes in reasoning and complex learning. *Cognitive Psychology*, in press.

[3] G S Halford, W H Wilson, J Guo, R W Gayler, J Wiles, and J E M Stewart. Connectionist implications for processing capacity limitations in analogies. In K J Holyoak and J Barnden, editors, *Advances in Connectionist and Neural Network theory*, chapter 7. Ablex, Norwood, NJ, 1994.

[4] H F Harlow. The formation of learning sets. *Psychological Review*, 42:51–65, 1949.

[5] G E Hinton. Mapping part-whole hierarchies in connectionist networks. *Artificial Intelligence*, 46(1-2):47–76, November 1990.

[6] J E Hummel and K J Holyoak. Distributed representations of structure: A theory of analogical access and mapping. *Psychological Review*, 104(3):427–466, 1997.

[7] T S Kendler. *Levels of cognitive development.* Lawrence Erlbaum Associates, Mahwah, NJ, 1995.

[8] S Phillips. Limits of generalization: An error surface view. In *Proceedings of the 1997 Annual Conference of Japanese Neural Network Society: JNNS'97*, pages 188–189, 1997.

[9] S Phillips and G S Halford. Systematicity: Psychological evidence with connectionist implications. In M G Shafto and P Langley, editors, *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society*, pages 614–619, 1997.

[10] P Smolensky. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, 46:159–216, 1990.