The Utility of Knowledge Transfer with Noisy Training Sets

Steven Gutstein, Olac Fuentes and **Eric Freudenthal**
Department of Computer Science
University of Texas at El Paso
El Paso, Texas, 79968, U.S.A.

Abstract

Knowledge transfer has traditionally concerned itself with the transfer of relevant features. Yet, in this paper, we will highlight the importance of transferring knowledge of which features are irrelevant.

When attempting to acquire a new concept from sensory data, a learner is exposed to significant volumes of extraneous data. In order to use knowledge transfer for quickly acquiring new concepts, within a given class (e.g. learning a new character from the set of characters, a new face from the set of faces, a new vehicle from the set of vehicles etc.), a learner must know which features are ignorable or repeatedly be forced to relearn them.

We have previously demonstrated knowledge transfer in deep convolutional neural nets (DCNN's) (Gutstein, Fuentes, & Freudenthal 2007). In this paper, we give experimental results that demonstrate the increased importance of knowledge transfer when learning new concepts from noisy data.

Additionally, we exploit the layered nature of deep convolutional neural nets (DCNN's) to discover more efficient and targeted methods of transfer. We observe that most of the transfer occurs within the 3.2% of weights that are closest to the input image.

Introduction

Sensory data often contains as much, if not more, information that is irrelevant to a given task as is relevant. Natural images of objects don't depict them in a vacuum. Not only do these objects have a set of characteristic invariants, but they are also invariant with respect to all the irrelevant data in the image.

Abu-Mostafa (Abu-Mostafa 1994) pointed out that by adroitly choosing training samples, one could ensure that desired invariants, would be learned. Admittedly, he was referring to active properties of the class in question (i.e. invariance with respect to rotation, scaling etc.) and not to irrelevant information contained within the training data. However, the concept is still applicable.

In order to demonstrate the importance of transferring knowledge of what information is irrelevant and to better localize where this transference takes place, we have performed experiments using a deep convolutional neural net. This architecture is very robust to random noise once it has been trained. However, it will still have trouble when trained on images which have not had the noise removed. This issue should become more pronounced for images of natural scenes, which have a great variety of backgrounds that need to be ignored when a neural net is trained to recognize a particular object or class of objects.

The results shown here will both demonstrate the increased utility of knowledge transfer when training on data with extraneous information, and that the 3.2% of weight parameters associated with the lower 3 levels of the net are responsible for nearly all the knowledge transfer.

Background & Related Work

Knowledge transfer has traditionally focused on transfer of relevant features - the use of previously learned features from one task to help in learning another. For instance when explaining the benefits of knowledge transfer, Thrun (Thrun 1996) describes how learning to drive a car is aided by prior knowledge of basic motor skills, traffic patterns, language etc. We feel it is equally important to remember to ignore irrelevant features, such as the currently selected radio station. When learning to drive a manual instead of an automatic, qualities that retain their irrelevance should be ignored. Any effort spent relearning this is effort wasted.

Most identification or classification tasks involve a flood of data, much of which is unrelated to the task being learned. Again, it should be stressed that the irrelevance that is being described here is not that of acceptable variations, such as facial expressions, which are acceptable variations with respect to facial recognition, or handwriting styles, which are acceptable variations with respect to character recognition. Instead, it is the irrelevance of data not associated with the presumed focus of attention.

The challenge of identifying objects in cluttered backgrounds is usually approached by sliding a window across an image and applying a binary classifier for the desired object to such a window. However these windows will still contain data not associated with the focus of attention.

Viola and Jones (Viola & Jones 2001) have developed a method for object detection, which concentrates on finding regions of an image which absolutely cannot contain a given object and then discarding them. Their method draws its speed and efficacy by concentrating on quickly and correctly

identifying regions to ignore.

Knowledge Transfer

Some of the earliest successful work done in knowledge transfer dates back to Pratt's work in discriminability-based transfer with neural nets (Pratt 1993). Previous work had unsuccessfully attempted to use the weights learned by a net in solving one problem as the initial conditions for solving a new problem. Pratt's success hinged upon identifying which weights of her neural net were irrelevant to the new problem, resetting these weights and allowing them to retrain, while preserving the network weights that were still salient.

This was an early demonstration not only of knowledge transfer, but also the importance of task similarity. The more similar a new task was to previously mastered tasks, the more already trained nodes could be used in learning the new task. In fact without similarity between tasks, there is no point in transferring knowledge. One early method of grouping tasks by similarity is the Task-Clustering method, introduced by Thrun and O'Sullivan (Thrun & O'Sullivan 1996).

Multi-Task Learning, a technique very different than Pratt's, was developed by Caruana (Caruana 1997). The basic approach taken was to improve both generalization and learning speed by learning many related tasks simultaneously. This depended upon the various tasks being sufficiently related so they would constructively reinforce learning each other. An earlier paper describes several mechanisms for this reinforcement (Caruana 1995).

Although these two techniques both rely upon the creation of an internal representation of raw data to enhance a net's ability to learn multiple tasks, they represent two fundamentally different methods of knowledge transfer - representational and functional (Silver & Mercer 1996). In representational transfer, as Pratt used, one finds the subsets of an existing internal representation which are useful for the new task being learned. In functional transfer, as Caruana used, one trains in such a way as to require learning a single internal representation, which is suitable for several tasks. This technique highlights the importance of developing an internal representation for the different classes which is both concise and consistent.

Baxter has stated (Baxter 2000) that one way to view the problem of knowledge transfer is as the problem of learning an appropriate inductive bias. When attempting to learn a 'family of concept learning tasks' (Thrun 1996), which is a potentially infinite set of mutually distinct binary classification problems (e.g. distinguishing among characters, faces, vehicles etc.) the real challenge is to learn a set of features that, with an appropriate distance metric, will enable one to differentiate among classes. (Baxter 2000). Again, we want to stress that learning this metric involves not only learning the relevant features, but also learning to filter out irrelevant features.

Baxter used this approach to train a neural net to recognize Japanese kanji taken from a dataset from the CEDAR group at the State University of New York at Buffalo. His net consisted of 4 layers. The first layer was the input layer. The second layer was trained to provide the common internal representation for all classes. The last two layers were grouped by task. Nodes in the third layer acted like a classical hidden layer, taking raw input and providing enough capacity for the given subnet to learn to recognize any individual kanji character. Each node in the final layer would act as boolean classifier for a particular character, relying solely upon nodes in the third layer for input. He trained this net first to recognize 400 kanji. Then, he used the common internal representation to do 1 Nearest Neighbor classification on 2,618 different kanji. He achieved a classification error of 7.5% (Baxter & Bartlett 1998).

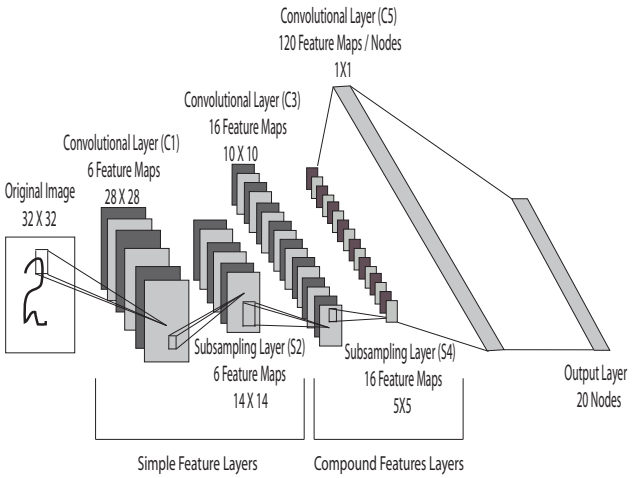

Figure 1: Architecture of our net, which is a slightly modified version of LeNet5. It should be noted that the feature maps in the C5 & Output layers are 1 node $\times$ 1 node. So, they could with equal accuracy be considered as traditional nodes in a non-weight sharing feed-forward neural net.

A more recent example of learning an appropriate representation and metric to distinguish among the elements of a set of classes was made by Chopra et al. (Chopra, Hadsell, & LeCun 2005). In this work, a siamese net (Bromley *et al.* 1993) was trained on a relatively small number of faces to recognize whether a given pair of faces were from the same person. This technique was used to correctly label pairs of faces, which came from people not seen during training, as being same or different. However, the images used contained very few features that were not relevant to the faces.

Like Chopra et al., we use deep convolutional neural nets. Our net employs a slight variation on the LeNet 5 architecture first employed by LeCun et al. (LeCun *et al.* 1999) as is shown in Figure 1.

Deep Convolutional Neural Nets

This brings us to the second stream of research from which we draw - the use of deep neural nets, in particular, convolutional ones. Although these nets are usually referred to as just Convolutional Neural Nets, we refer to them as Deep Convolutional Neural Nets, in order to equally emphasize

the use of a deep architecture as well as the properties that make them convolutional. Both of these properties are important for us.

Deep convolutional neural nets (DCNN's) are designed to limit the hypothesis space available to be searched. Specifically, they limit themselves to solutions that display (LeCun *et al.* 1999):

1. Shift Invariance
2. Moderate Insensitivity to Rotations
3. Moderate Insensitivity to Geometric Distortions

Almost all image recognition problems require these properties. Because the net is architecturally restricted to hypotheses with these properties, there is no need for exhaustive training with hints, or extensive pre-processing of images.

Additionally, a DCNN, as described in (LeCun *et al.* 1999) is a deep architecture. It uses both convolutional layers, which learn to detect individual features in a position invariant way, and sub-sampling layers, which provide modest insensitivity to rotations and geometric distortions. This architecture forces the early layers to act as feature extractors. Each layer of the net learns an internal representation consisting of higher level features than the layer beneath it. Learning of simpler features is concentrated at the lower levels which act as pre-processed input to higher levels, which learn higher level features. This should give successive internal representations with greater levels of specificity for a given task or set of tasks. For the sake of knowledge transfer, it also means that there will be multiple inductive biases from which to choose. This permits various degrees of knowledge transfer between different sets of tasks.

The most popular type of feed-forward neural net has three layers - an input layer, a hidden layer and an output layer. This architecture has achieved popularity because of its simplicity and its ability to search a large hypothesis space. However, it is not terribly efficient. One tends to need far more nodes to represent a given function with a shallow architecture than with a deep architecture (Bengio & LeCun 2007; Utgoff & Stracuzzi 2002; Allender 1996). This suggests that deep nets would be superior to shallow ones, because they need fewer parameters. However, training deep nets accurately seems to require specialized architectures or training techniques to avoid falling into erroneous local minima (Bengio & LeCun 2007; Sima 1994).

Fortunately though, there are techniques that avoid this pitfall. Some of these techniques, such as DCNN's and Neural Abstraction Pyramid nets (Behnke 2003), rely upon their architectural restrictions to avoid the spurious minima which plague other deep nets. Others rely upon specialized training techniques, such as Cascade Correlation (Fahlman & Lebiere 1990), Knowledge Based Cascade Correlation (Schultz & Rivest 2000) and Greedy Layerwise Training (Hinton, Osindero, & Teh 2006; Hinton & Salakhutdinov 2006).

Experiments

Our experiments involved training our net with the NIST Special Database 19, which contains 62 classes of handwritten characters corresponding to '0'-'9', 'A'-'Z' and 'a'-'z'. We first trained the net on one subset of characters from the NIST database, using about 375 samples of each character. Each character was assigned a 20 bit random target vector, which the net was supposed to reproduce when presented with an image of the corresponding character.

This net was trained for 150 epochs. We then took the net at the epoch that had the best performance on a smaller validation set with about 20 samples of each character to be the source of any knowledge we transferred. This will be referred to as the 'source net'.

In our previous work (Gutstein, Fuentes, & Freudenthal 2007), our objective was merely to demonstrate the existence of knowledge transfer with DCNN's and how well-suited they are for this technique. In these experiments our goal is to demonstrate how the importance of knowledge transfer increases as the variations in the data being learned increases. This will hold even, or perhaps especially, when the variations have nothing to do with the concepts being learned.

The images of hand-written characters that were used came from the NIST Special Dataset 19, which consists of binary images. Our first set of experiments was noise free. These images are referred to as having 0% noise. The second set involved randomly changing 10% of the background pixels of each image from black to white. These images are referred to as having 10% noise. The third set of experiments involved randomly changing 20% of the background pixels from black to white. These images are referred to as having 20% noise.

A set of experiments was performed for each degree of noise. The same set of 20 character classes was used to create the nets from which we were attempting to transfer knowledge to aid in the learning of a set of 20 different character classes. This different set consisted of the same set of character classes for each experiment.

Each experimental set involved attempting to learn the new set of 20 character classes, given 1, 5, 10, 20 or 40 samples/class. Attempts were made to transfer knowledge by copying weights from the bottom n layers of the source net over to the new net, where $0 \leq n \leq 5$. Transferred weights were kept fixed and not allowed to change during training. To find the best choice for n, we ran a series of experiments beginning with $n = 5$ and culminating with $n = 0$. This last scenario, of course, corresponds to the absence of any knowledge transfer. Were we to have tried allowing $n = 6$, that would correspond to transferring all the weights from the source net and not allowing any training.

Additionally, we ran each individual experiment for a given noise-level, number of samples/class and number of retained layers 5 times each. Each run involved learning a different set of training instances. For each run we let the net train for 150 epochs and then took the net which had the best performance on a validation set, consisting of 50 samples/class (1,000 images) and would measure its performance on a testing set which also consisted of 50 sam-

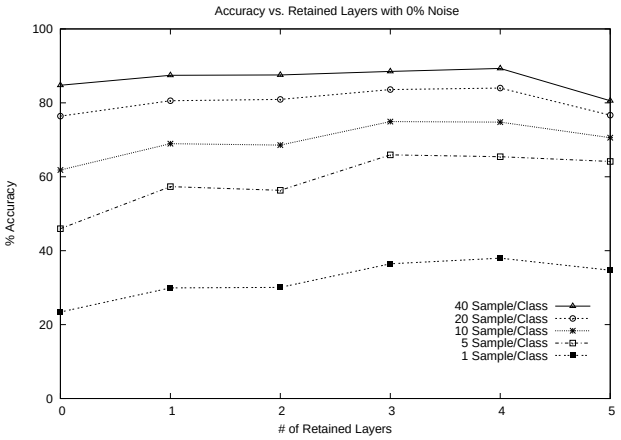

Figure 2: Comparison of learning curves showing accuracy vs. number of retained levels for various numbers of samples per class in the 0% Noise training set. Curves show, from top to bottom, results for 40, 20, 10, 5 and 1 sample per class. Each point represents the average of 5 trials on a testing set with 1,000 character samples.

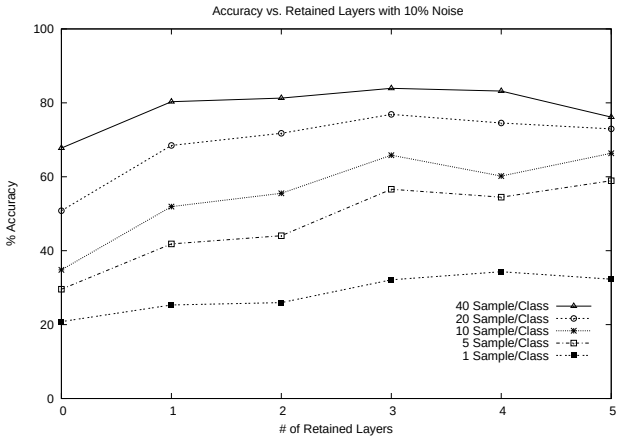

Figure 3: Comparison of learning curves showing accuracy vs. number of retained levels for various numbers of samples per class in the 10% Noise training set. Curves show, from top to bottom, results for 40, 20, 10, 5 and 1 sample per class. Each point represents the average of 5 trials on a testing set with 1,000 character samples.

ples/class. All results, unless otherwise notated, are for the average accuracies achieved on the testing sets over 5 runs.

We observed, as expected, a uniform degradation in performance with an increase of noise. This may even be seen in the source nets, which achieved validation set accuracies of 93.48%, 88.47% and 85.21% respectively for 0% noise, 10% noise and 20% noise experiments.

Results and Analysis

The results for the 3 sets of experiments are shown in Figures 2-4.

We can see in these curves, that the most significant increases in accuracy come from retaining a convolutional layer. This may be seen by the relatively abrupt increases in accuracy that occur between retaining 0 levels (no transfer) and 1 level (C1), and between retaining 2 levels (C1 & S2) and retaining 3 levels (C1, S2 & C3). This makes sense, because the C (convolutional) layers are where features are detected. The sub-sampling layers just provide some minor blurring effects to aid robustness. The occasional deleterious effect of having a sub-sampling layer as the final retained layer, rather than the *prior* convolutional layer is slightly puzzling.

The drop in accuracy found between retaining 4 levels (C1 - S4) and retaining 5 levels (C1 - C5) is surprising, because nominally, C5 is a convolutional layer. We would expect it to have learned new relevant features. However, as can be seen in Table 1, this layer contains about 92% of the free parameters of our net. So, when we retain this layer and all those prior to it, our net has less than 5% of its original capacity. Furthermore, when we transfer this layer in its entirety, all of our learning must take place within a single layer. This forces our hypothesis space to be even more severely restricted and further hampers learning. Yet, especially with noisy data, enough knowledge is usually transferred to enable this net to learn a new set of characters using fewer than 5% of its free parameters with greater accuracy than if it had trained from scratch. This improvement in accuracy can be as high as about 20%. However, the major benefits of transfer are contained in the 3.2% of the net's free parameters at the bottom 3 layers of the net (C1-C3).

Since we trained on data containing both relevant features and irrelevant ones, it is not immediately clear how to distinguish between the benefit of providing prior knowledge of features to use and prior knowledge of features to ignore. In Figures 5 & 6, we examine how accuracy vs. retained layers varies with noise for a given number of samples per class. In Table 2, we find the benefit obtained if we transferred the optimal number of levels. This tended to be C1-C3, but was sometimes C1-S4.

Although these figures and table show that knowledge transfer has increased utility in the presence of irrelevant information, they do not do so in an unambiguous manner. This is likely due to various competing effects.

Firstly, with enough samples per class, knowledge transfer will lose relevance, since more of the necessary information will exist within the training set. Yet, if there are too few samples per class, our approach to knowledge transfer will still not be as effective. Although it is biased to discriminate based upon relevant features, training is still necessary. This may be seen in Table 2 by observing that the peak improvements we obtain for each set of noisy images initially increases with the number of samples per class, reaches a maximum and then decreases.

Secondly, it is inherently more difficult to learn from noisy data. We observed this in the performance of our source nets, which achieved validation set accuracies of 93.48%, 88.47% and 85.21% respectively for the 0%, 10% and 20% noise experiments. So, we would expect better transfer with less noisy data sets. Better transfer means

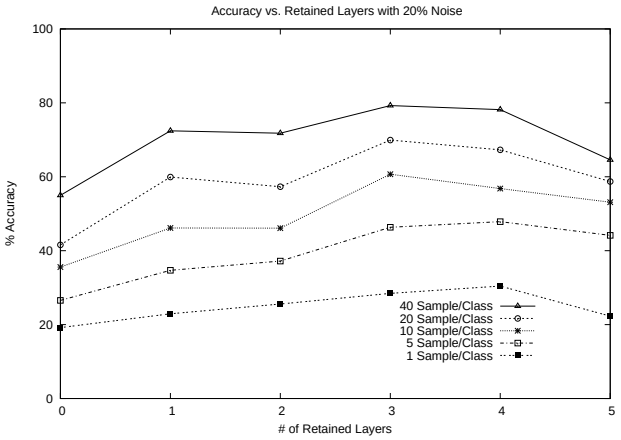

Figure 4: Comparison of learning curves showing accuracy vs. number of retained levels for various numbers of samples per class in the 20% Noise training set. Curves show, from top to bottom, results for 40, 20, 10, 5 and 1 sample per class. Each point represents the average of 5 trials on a testing set with 1,000 character samples.

needing fewer samples per class to find the correct relevant features upon which to focus. It also means that since the benefits of transfer are realized more quickly, the maximum improvements due to transfer should also be achieved more quickly. Once the transferred features have been successfully transferred, additional samples are superfluous. This is also reflected in Table 2.

Yet, bearing these issues in mind, Table 2 still shows that the more irrelevant data a learner is forced to filter, the more it benefits from using prior knowledge to ignore the useless data.

Conclusions & Future Work

Our results show that when learning a new task in the presence of irrelevant data, knowledge transfer techniques acquire added utility by biasing the learner away from a large, distracting region of hypothesis space. Additionally, in the case of a DCNN learning to recognize some fairly simple images, we can see that most of the transferrable knowledge resides in the lower levels, which contain relatively few free parameters. This should make it practical to record them for future use when one finds a task that seems sufficiently similar to one that has already been mastered.

We are planning experiments to more closely approximate Baxter & Bartlett's approach and use a much greater number of classes in our training sets. Their results strongly suggest that ultimately knowledge transfer will achieve accuracy comparable to the best achievable for full capacity nets with large training sets.

Additionally, we plan to investigate techniques to identify which parts of a specific layer should be transferred. This would enable us to transfer knowledge from layers in a more controlled manner, especially from the C5 layer. An initial set of experiments that attempted to use saliency as described by LeCun, Denker et al. (LeCun *et al.* 1990) to

Layer	*Free Parameters*	*%of Total*
C1	156	0.30%
S2	12	0.02%
C3	1,516	2.90%
S4	32	0.06%
C5	48,120	92.09%
Output Layer	2,420	4.63%
Total	52,256	100.00%

Table 1: Number of free paramters at each layer of our net

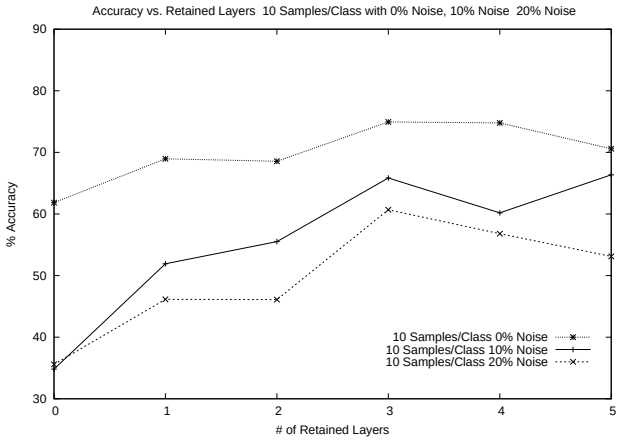

Figure 5: Comparison of curves showing accuracy vs. number of retained levels for 10 samples/class for no noise, 10% noise and 20% noise Each point represents the average of 5 trials on a testing set with 1,000 character samples.

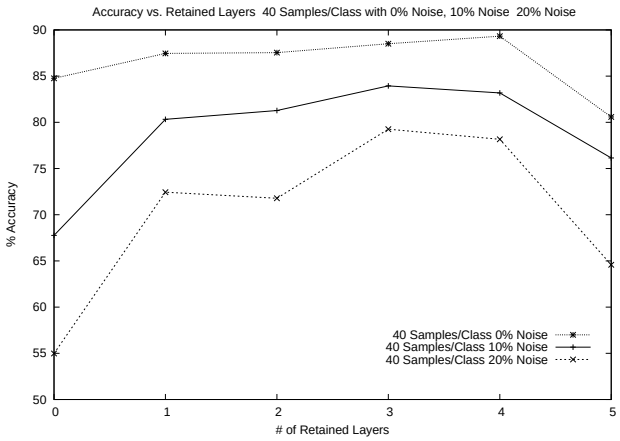

Figure 6: Comparison of curves showing accuracy vs. number of retained levels for 40 samples/class for no noise, 10% noise and 20% noise Each point represents the average of 5 trials on a testing set with 1,000 character samples.

Samples per Class	Maximum Improvement Obtained		
	0% Noise	10% Noise	20% Noise
1	14.58%	13.50%	11.24%
5	19.98%	29.36%	21.32%
10	13.10%	31.54%	25.10%
20	7.60%	26.08%	28.34%
40	4.56%	16.18%	24.28%

Table 2: Maximum improvement in accuracy obtained from our knowledge transfer technique for various combinations of noise and samples per class

determine which nodes to transfer, were unsuccessful. However, we are not yet convinced that this is a fruitless path.

We will also try to let the topology of our net change to adapt to the new task using a technique like Knowledge Based Cascade Correlation (Schultz & Rivest 2000) or NEAT (Stanley & Miikkulainen 2002) in order to determine which feature maps should be transferred and which should be retrained.

Finally, we will investigate techniques to select an optimal set of classes for knowledge transfer.

References

Abu-Mostafa, Y. 1994. Learning from hints. *Journal of Complexity* 10(1):165–178.

Allender, E. 1996. Circuit complexity before the dawn of the new millennium. In *FSTTCS*, 1–18.

Baxter, J., and Bartlett, P. 1998. The canonical distortion measure in feature space and 1-NN classification. In Jordan, M. I.; Kearns, M. J.; and Solla, S. A., eds., *Advances in Neural Information Processing Systems*, volume 10, 245–251.

Baxter, J. 2000. A model of inductive bias learning. *Journal of Artificial Intelligence Research* 12:149–198.

Behnke, S. 2003. *Hierarchical Neural Nets for Image Interpretation*, volume 2766 of Lecture Notes in Computer Science. Springer-Verlag.

Bengio, Y., and LeCun, Y. 2007. Scaling learning algorithms towards AI. In Bottou, L.; Chapelle, O.; DeCoste, D.; and Weston, J., eds., *Large-Scale Kernel Machines*. MIT Press.

Bromley, J.; Guyon, I.; LeCun, Y.; Sackinger, E.; and Shah, R. 1993. Signature verification using a siamese time delay neural network. In Cowan, J., and Tesauro, G., eds., *Advances in Neural Information Processing Systems*, volume 6. Morgan Kaufmann.

Caruana, R. 1995. Learning many related tasks at the same time with backpropagation. In Tesauro, G.; Touretzky, D.; and Leen, T., eds., *Advances in Neural Information Processing Systems*, volume 7, 657–664. The MIT Press.

Caruana, R. 1997. Multitask learning. *Machine Learning* 28(1):41–75.

Chopra, S.; Hadsell, R.; and LeCun, Y. 2005. Learning a similarity metric discriminatively, with application to face verification. In *Proc. of Computer Vision and Pattern Recognition Conference*. IEEE Press.

Fahlman, S. E., and Lebiere, C. 1990. The cascade-correlation learning architecture. In Touretzky, D. S., ed., *Advances in Neural Information Processing Systems*, volume 2, 524–532. Denver 1989: Morgan Kaufmann, San Mateo.

Gutstein, S.; Fuentes, O.; and Freudenthal, E. 2007. Knowledge transfer in deep convolutional neural nets. In *Proceedings of the FLAIRS-07 Conference*, 104–109.

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

Hinton, G.; Osindero, S.; and Teh, Y. 2006. A fast learning algorithm for deep belief networks. *Neural Computation* 18(7):1527–1554.

LeCun, Y.; Denker, J.; Solla, S.; Howard, R.; and Jackel, L. 1990. Optimal brain damage. In Touretzky, D., ed., *Advances in Neural Information Processing Systems 2 (NIPS*89)*. Denver, CO: Morgan Kaufman.

LeCun, Y.; Haffner, P.; Bottou, L.; and Bengio, Y. 1999. Object recognition with gradient-based learning. In Forsyth, D., ed., *Shape, Contour and Grouping in Computer Vision*, 319–346. Springer.

Pratt, L. Y. 1993. Discriminability-based transfer between neural networks. In Hanson, S. J.; Cowan, J. D.; and Giles, C. L., eds., *Advances in Neural Information Processing Systems*, volume 5, 204–211. Morgan Kaufmann, San Mateo, CA.

Schultz, T., and Rivest, F. 2000. Knowledge-based cascade corellation. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks 2000*, volume 5, V641–V646.

Silver, D., and Mercer, R. 1996. The parallel transfer of task knowledge using dynamic learning rates based on a measure of relatedness. *Connection Science Special Issue: Transfer in Inductive Systems* 8(2):277–294.

Sima, J. 1994. Loading deep networks is hard. *NEURCOMP: Neural Computation* 6.

Stanley, K., and Miikkulainen, R. 2002. Efficient reinforcement learning through evolving neural network topologies. In *GECCO '02: Proceedings of the Genetic and Evolutionary Computation Conference*, 569–577. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Thrun, S., and O'Sullivan, J. 1996. Discovering structure in multiple learning tasks: The TC algorithm. In *International Conference on Machine Learning*, 489–497.

Thrun, S. 1996. Is learning the n-th thing any easier than learning the first? In Touretzky, D. S.; Mozer, M. C.; and Hasselmo, M. E., eds., *Advances in Neural Information Processing Systems*, volume 8, 640–646.

Utgoff, P., and Stracuzzi, D. 2002. Many-layered learning. *Neural Computation* 14(10):2497–2529.

Viola, P., and Jones, M. 2001. Rapid object detection using a boosted cascade of simple features. In *CVPR 1*, 511–518.