Application of Knowledge-based Cascade-correlation to Vowel Recognition

François Rivest
School of Computer Science
McGill University
Montreal, QC Canada H3A 1B1
francois.rivest@mcgill.ca

Thomas R. Shultz
Department of Psychology and
School of Computer Science
McGill University
Montreal, QC Canada H3A 1B1
shultz@psych.mcgill.ca

Abstract - Neural network algorithms are usually limited in their ability to use prior knowledge automatically. A recent algorithm, knowledge-based cascade-correlation (KBCC), extends cascade-correlation by evaluating and recruiting previously learned networks in its architecture. In this paper, we describe KBCC and illustrate its performance on the problem of recognizing vowels.

I EXISTING KNOWLEDGE AND NEW LEARNING

Neural network algorithms rarely allow prior knowledge to be incorporated into their learning. Most start from scratch and those that do use prior knowledge require that knowledge to have a specific form, such as having the same architecture [1], being a symbolic domain theory [2], or being given as hints [3]. However, prior knowledge can often take the form of some existing classifier or function approximator and no algorithm is flexible enough to permit the integration of such a wide variety of knowledge.

It is clear that humans do not learn from scratch, but make extensive use of their knowledge in learning [4-6]. Use of prior knowledge in learning can ease and speed learning and lead to better generalization as well as interference effects. The current difficulty in using prior knowledge is arguably the major limitation in neural network modeling of human learning and cognition. In this paper, we describe and test a neural learning algorithm that implements a general mechanism of knowledge reuse.

Knowledge-based cascade-correlation (KBCC) is a fundamental extension of cascade-correlation (CC), a constructive learning algorithm that has been successfully used in many real applications [7] and in simulations of cognitive development [8-13]. CC builds its own network topology by adding new hidden units to a feedforward network in cascade fashion, i.e., new units receive inputs from each non-output unit already in the network [14]. Our KBCC extension recruits previously learned networks in addition to the untrained hidden units recruited by CC. These recruitable networks could potentially be any functional form knowledge, although being differentiable is a must. We refer to existing networks as *source* knowledge and to the current task to learn as a *target*. Previously learned source classifiers or approximators compete with each other and with standard hidden units to be recruited into the learning network.

In artificial bivariate dichotomous tasks, KBCC successfully recruited networks representing parts of a target task, equivalent-knowledge networks, and more complex networks embedding equivalent knowledge, with substantial learning speed ups [15]. KBCC was also shown to be superior to multi-task learning (MTL) in these respects [16].

II PREVIOUS WORK ON KNOWLEDGE AND LEARNING

KBCC is similar to recent neural network research on transfer [1], sequential learning through multi-task learning [17], and knowledge insertion [2,18]. But KBCC is more ambitious and principled because it stores and searches for knowledge within a generative network approach and has no real limitation in the structure of the recruited knowledge.

Pratt [1] studied the idea of transferring knowledge from a source neural network to a target network through copying the network structures and parameters (weights). She found that literally copying a network could sometimes slow down the training and reduce generalization performance compared to random networks. She therefore developed a technique to re-scale the weight vector feeding hidden units. If a hidden unit has good discrimination power, its weight vector is scaled up to reduce training effects, and conversely, if the discrimination hyperplane is bad, its weight vector is scaled down, or even randomized, in order to avoid copying bad effects. This technique is limited to discrete output networks where the target task requires a network at least as big as the source network and where input and output perfectly matches the source network.

Silver and Mercer [17] developed a transfer of knowledge technique based on Caruana's multi-task learning [19]. The basic idea derives from a proof that if a network has multiple related tasks to learn, it requires fewer examples to learn them, because the hidden layer can develop a more general representation. Silver and Mercer's idea is to re-learn the prior knowledge while learning the new task, in parallel, on the same network. The target network has an output for the target task, and extra outputs to represent each source network's outputs. Prior knowledge is used to generate the desired values for these extra outputs to learn. This can be simply done by processing the input patterns through the prior knowledge, thus permitting the prior knowledge to be any sort of function. This still has a major limitation in that

0-7803-7278-6/02/$10.00 ©2002 IEEE

target inputs must match source inputs, and the new network must be big enough to learn the prior knowledge. Moreover, relearning of prior knowledge is required, which does not seem very efficient.

With a slightly different goal in mind, Towell and Shavlik [2] invented an algorithm to transform rule-based knowledge into a neural network (KBANN). The idea was to refine that knowledge in neural network form and then to later extract improved rules. We believe that this technique could be used with KBCC, by taking rule-based knowledge and transforming it into differentiable functional form.

This kind of idea was also developed by Parekh and Honavar [18], who proposed to use KBANN in conjunction with constructive algorithms. KBANN was used to create a neural network that would serve as a basis for a constructive algorithm that could build on the source knowledge outputs and inputs. Again, this requires the same encoding for prior and new knowledge. Moreover, it does not allow composition of prior knowledge like most other approaches.

III DESCRIPTION OF KBCC

Because KBCC is a generalization of CC, it is quite similar to CC. As in CC, candidates are installed on top of the network, just below the output; hence new units receive inputs from every non-output unit already in the network. Unlike CC, KBCC is not limited to a pool of candidate units that are univariate single-valued functions. KBCC can recruit any multivariate vector-valued component. The connection scheme in KBCC as shown below is similar to the CC connection scheme, except that a hidden unit may have a matrix of weight connections (as opposed to a single vector) at their inputs and their outputs as shown in figure 1.

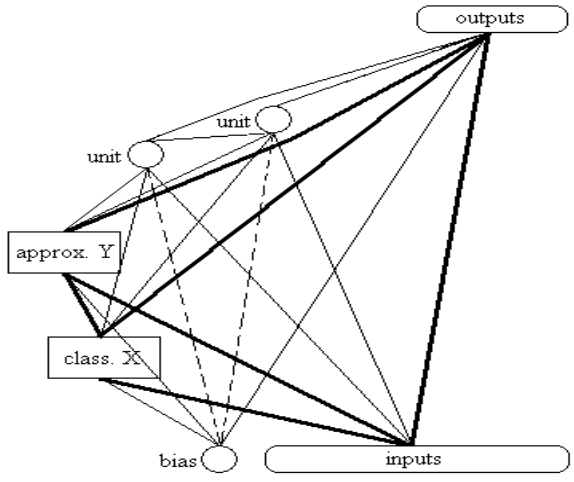

Figure 1: A KBCC network with four hidden units. The first one is an existing classifier, the second one is an existing approximator, and the last two are single sigmoid units. Dash lines represent single weights, while solid thin lines represent weight vectors, and solid thick lines weight matrices.

KBCC training is composed of two phases: In output phase, only the weights feeding the output units are trained. In input phase, only the weights feeding the candidate units (and networks) are trained.

The network begins in output phase with a set of output units fully connected to the inputs. These weights are trained to minimize the sum squared error:

$$F = \sum_{o} \sum_{p} \left(V_{o,p} - T_{o,p}\right)^2 \qquad (1)$$

Where $V_{o,p}$ is the network output o at pattern p and $T_{o,p}$ the corresponding target value. The training uses QuickProp[1] [14], a gradient based algorithm that employs the current and previous gradient to estimate the second order derivative of the objective function with respect to the weights to be trained. The output phase stops either when it successfully learns the task, or when the sum squared error stagnates or a maximum number of epochs is reached, in which case the algorithm goes into input phase.

The input phase begins by initializing a pool of candidate units and networks (or other functional knowledge) with random weights from every non-output unit of the target network to the candidate inputs. These weights are then trained using QuickProp to maximize the covariance between the candidate outputs and the target network residual error:

$$G_c = \frac{\sum_{o_c} \sum_{o} \left\| Cov\left(V_c, E\right) \right\|^F}{\sum_{o} \sum_{p} E_{o,p}^2} \qquad (2)$$

Where $E_{o,p}$ is the error at output unit o for pattern p, V_c is the candidate output patterns, E the network error patterns and $\|C\|^F$ the Frobenius norm of matrix $C=Cov(V_c,E)$ defined as:

$$\left\|C\right\|^F = \sum_{i,j} C_{i,j}^2 \qquad (3)$$

Again, whenever the best score $max\{G_c\}$ stagnates, or a maximum number of epochs is reached after a minimal score, the input phase stops and the best candidate is installed into the target network by adding connections from its outputs to the target network outputs using small random values and the sign of the covariance. The other candidates are discarded.

A more detailed description of the KBCC algorithm with all the default parameter values can be found in [20].

[1] Although training is not limited to QuickProp.

IV DEMONSTRATION OF KBCC: PETERSON-BARNEY VOWEL RECOGNITION

We created six transfer scenarios with the Peterson-Barney vowel recognition problem from the CMU AI repository.[2] The data set can be split into three subsets based on the speaker type: male, female or child. One scenario was originally used by Pratt [1] and involved training networks on the female data and then using them as sources to train target networks on male data. The other scenarios are similar and complete all permutations of the three subsets.

The data set contains the two middle formants of the speech sound made by 76 speakers saying 10 different vowels twice. The speakers were 33 males, 28 females, and 15 children, all speaking English. The inputs were scaled down by 1000 resulting in input values in the range [0.0, 1.5] and [0.0, 4.0] for the first and second formants, respectively. The outputs were encoded on 10 units (one per vowel) with a value of 1.0 for the correct vowel and 0.0 for the others. A network was considered to properly classify a pattern if the output with highest activation corresponded to the target vowel.

The scenarios are constructed using the following scheme. Starting with the three subsets (male, female, child), one subset is used to train the source networks, and a different subset for training the target networks. This scheme generates six scenarios. In order to compare KBCC with CC without knowledge, we added three more scenarios where we trained CC nets on one of the subsets without any prior knowledge.

A. Experimental setup

First, for each subset, we generated 10-fold cross-validation train/test set pairs. We trained 10 CC networks (for up to 15 hidden units) for each train/test pair for a total of 100 CC networks per subset. Those represent the three no-knowledge baseline scenarios.

We found that a good CC source network (similar to Pratt's sources) has about 10 hidden units. Since each subset is used as source in two scenarios, we have trained 200 CC networks on each of the three subsets for 10 recruitments each.

For each scenario, we have 100 CC networks per source data set. Given a scenario, for each of the 10 train/test pairs of the target data set, we trained 10 KBCC networks (for up to 15 hidden units/networks). For each of the 100 resulting KBCC networks we used a different source network.

During the training of the target networks, we evaluated the network on the data subset of their source to measure retention and on the third data subset, the one that wasn't used in their training nor in the training of their sources. For example, in the scenario where the source networks are trained on female data and the target networks are trained on male data, the third subset is the child data.

B. Early learning comparison

In one of our scenarios, similar to Pratt, we used the female data (560 patterns) to train the source networks. We first found that good sources had about 10 hidden units, so we trained 100 CC sources with 10 hidden units each. We then tested these female-trained sources on the male data and obtained 52%±3% accuracy. This is quite close to Pratt's result using static back propagation networks.

Then we generated 10-fold cross-validation train/test sets. For each of the 10 train/test set pairs we trained 10 CC networks and 10 KBCC networks (each KBCC network used a different source) for up to 15 recruitments. We computed the train and test percentage correct at every epoch and analyzed the resulting learning curves.

Before the first recruitment, the linear solution scored around 71% correct on the train set and 69% on the test set. Before the second recruitment (which happened sooner in KBCC than in CC), KBCC reached 86% and 84% on the train and test sets while CC had only reached 80% and 77% on these same sets. Moreover the peak generalization of the KBCC-averaged test curves (85.5%) is reached at epoch 438 while that peak on the CC-averaged test curves (85.8%)[3] is reached at epoch 1699. Finally, we looked at the average number of epochs for each network curve to reach its peak generalization. We computed a paired sample t-test using the average for each fold. KBCC was significantly faster, taking an average of 827 epochs to reach its peak generalization while CC took 1279 epochs, with $t(9) = 4.418$ and $p < 0.005$. Both average peak values were 89% correct. Results are plotted in figure 2, also showing that the effect of the first recruitment is even stronger with child sources.

[2] http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/speech/database/pb/pb.tgz

[3] These two peaks are not significantly different in percent correct.

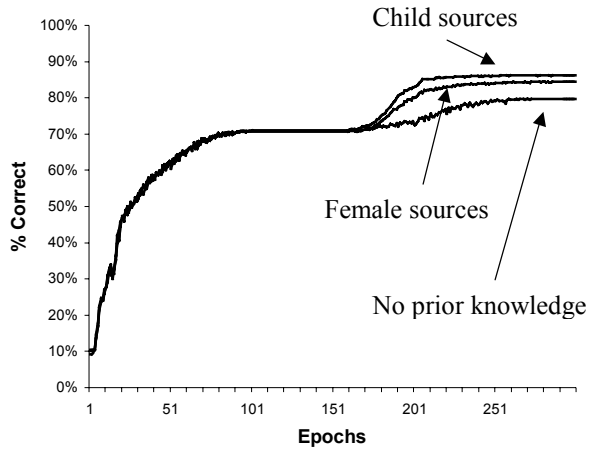

Figure 2: Averaged learning curves for the KBCC networks learning the male data. The curve reaching the lowest point is averaged over networks without prior knowledge. The one in the middle is averaged over networks using source networks trained on female data and the highest one represents networks using sources trained on the child dataset.

Similar results were obtained in the other scenarios. Each row in tables 1 and 2 represents one target situation. For each of these, the proportion correct after the first recruitment is given for every source condition. In all cases, the existence of prior knowledge shows a clear advantage in proportion correct after the first recruitment.

TABLE 1
TRAIN PROPORTION CORRECT AFTER THE FIRST RECRUITMENT

Target	Source			
	None	Male	Female	Child
Male	80%	N/A	86%	84%
Female	81%	86%	N/A	87%
Child	76%	81%	84%	N/A

TABLE 2
TEST PROPORTION CORRECT AFTER THE FIRST RECRUITMENT

Target	Source			
	None	Male	Female	Child
Male	78%	N/A	84%	82%
Female	78%	83%	N/A	83%
Child	71%	76%	80%	N/A

C. Learning time comparison

To evaluate the learning time, we simulated early-stopping after training. Since during training we recorded train and test set proportion correct, we could reconstruct for every target network the number of epochs it took to reach its highest generalization peak before over training. Given a target subset, we compared the learning speed of the three conditions (for example, given male data as target subset, the three conditions are without knowledge, using knowledge of female data and using knowledge of chid data).

Figures 4- 6 show the results grouped by target task. For each of those figures, we ran an ANOVA and looked at the Scheffe post hoc test. For all three targets, the prior knowledge conditions were significantly faster than the no knowledge condition at the .05 level.

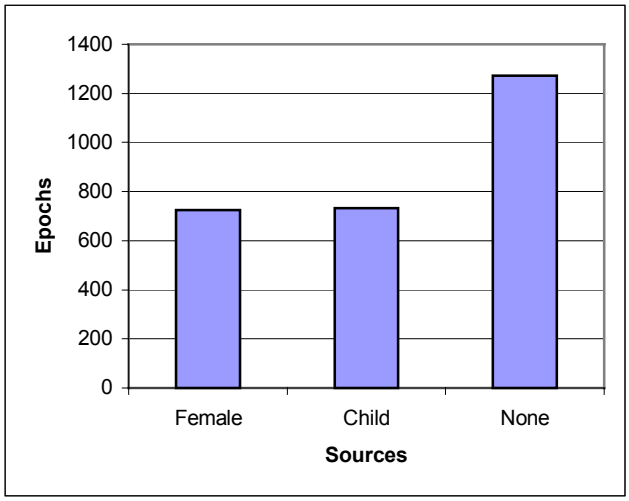

Figure4: Mean number of epochs to learn male data in three different source conditions.

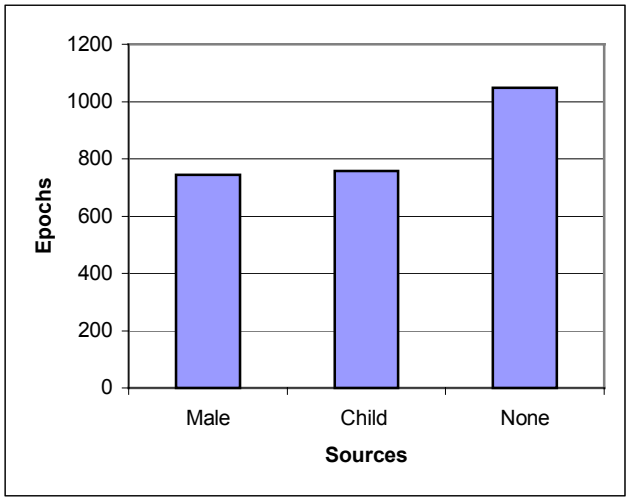

Figure 5: Mean number of epochs to learn female data in three different source conditions.

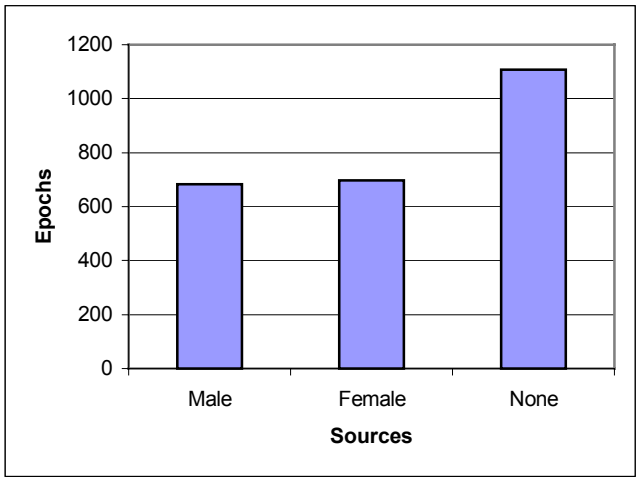

Figure 6: Mean number of epochs to learn child data in three different source conditions.

D. Learning quality

We also compared the best train and test percent correct reached by our networks. Results are presented in Table 3. We did an ANOVA to compare the three source conditions under each target task separately. None of the three ANOVAs yielded significance at the .05 level on the Scheffe test. Hence, the quality of the final solution did not seem to be affected by prior knowledge.

TABLE 3
TEST PROPORTION CORRECT AT HIGEST GENERALIZATION

Target	Source			
	None	Male	Female	Child
Male	89%	N/A	88%	88%
Female	89%	89%	N/A	89%
Child	84%	85%	85%	N/A

E. Retention and 3rd set generalization

We compared the retention and third set generalization of the source knowledge conditions for each target task. Even though in few cases, the prior knowledge condition had a slightly significant advantage, in others it had a slight disadvantage. In most cases there was little difference.

V DISCUSSION

These results show that KBCC is able to adapt and use its prior, related knowledge in the learning of a large and realistic new problem. Moreover, the availability of relevant knowledge significantly shortens KBCC learning time, without any loss of accuracy. Effective use of prior knowledge in new learning is the sort of quality one would like from both engineering and cognitive modeling viewpoints.

In contrast to previous methods for using knowledge in learning, KBCC has almost no restrictions on the format of prior knowledge. First, because prior knowledge is recruited into the network topology instead of being relearned, there is basically no limit to the internal complexity of the sources. Second, KBCC automatically searches for the best way to connect recruited sources in its architecture, removing any necessity for source inputs and outputs to perfectly match those of the target task.

Moreover, KBCC can use one or multiple sources to build a compositional solution. Because every candidate receives input from every previously recruited module, KBCC can combine them in a compositional way, for example, processing input data first through some classifier and then through some approximator (as shown in figure 1).

KBCC also seamlessly integrates learning by analogy with learning by induction. It learns by induction whenever it recruits single hidden units and by analogy when it recruits a previously trained source network. Finally, KBCC is consistent with the CC algorithm that has been successful in solving many real problems and in simulating many aspects of cognitive development.

Application of KBCC to other real problems such as DNA junction-splicing is currently being studied in our laboratory. Another area under study is the effect of prior knowledge on KBCC in impoverished training environments.

B. Acknowledgments

This work was supported by a grant from the Centre de Recherche Informatiqe de Montréal to the first author and from the Fonds de la Formation de Chercheurs et l'Aide à la Recherche to each author. We are grateful for comments on this work from Doina Precup.

C. References

[1] L. Y. Pratt, "Discriminality-based transfer between neural networks," *Advances in neural information processing systems 5*, pp. 204-211. San Mateo, CA: Morgan Kaufmann, 1993.
[2] G. G. Towell and J. W. Shavlik, "Knowledge-based artificial neural networks," *Artificial Intelligence* **70**:119-165, 1994.
[3] Y. S. Abu-Mostafa, "A method for learning from hints," *Advances in Neural Information Processing Systems 5*, pp. 73-80. San Mateo, CA: Morgan Kaufmann, 1993.
[4] E. Heit, "Models of the effects of prior knowledge on category learning," *Journal of Experimental Psychology: Learning, Memory, and Cognition* **20**:1264-1282, 1994.
[5] M. J. Pazzani, "Influence of prior knowledge on concept acquisition: Experimental and computational results," *Journal of Experimental Psychology: Learning, Memory, and Cognition* **17**:416-432, 1991.
[6] E. J. Wisniewski, "Prior knowledge and functionally relevant features in concept learning," *Journal of Experimental Psychology: Learning, Memory, and Cognition* **21**:449-468, 1995.
[7] J. Yang and V. Honavar, "Experiments with the cascade-correlation algorithm," *Microcomputers Applications* **17**:40-46, 1998.
[8] D. Buckingham and T. R. Shultz, "A connectionist model of the development of velocity, time, and distance concepts," *Proceedings of*

the Sixteenth Annual Conference of the Cognitive Science Society, pp. 72-77. Hillsdale, NJ: Erlbaum, 1994.
[9] D. Mareschal and T. R. Shultz, "Development of children's seriation: A connectionist approach," *Connection Science* **11**: 149-186.
[10] T. R. Shultz, "A computational analysis of conservation," *Developmental Science* **1**:103-126, 1998.
[11] T. R. Shultz, D. Buckingham and Y. Oshima-Takane, "A connectionist model of the learning of personal pronouns in English," *Computational learning theory and natural learning systems, Vol. 2: Intersection between theory and experiment*, pp. 347-362. Cambridge, MA: MIT Press, 1994.
[12] T. R. Shultz, D. Mareschal and W. C. Schmidt, "Modeling cognitive development on balance scale phenomena," *Machine Learning* **16**:57-86, 1994.
[13] S. Sirois and T. R. Shultz, "Neural network modeling of developmental effects in discrimination shifts," *Journal of Experimental Child Psychology* **71**:235-274, 1998.
[14] S. E. Fahlman and C. Lebiere, "The cascade-correlation learning architecture," *Advances in neural information processing systems 2*, pp. 524-532. Los Altos, CA: Morgan Kaufmann, 1990.
[15] T. R. Shultz and F. Rivest, "Knowledge-based cascade-correlation," *Proceedings of the International Joint Conference on Neural Networks*, Vol. V, pp. 641-646. Los Alamitos, CA: IEEE Computer Society Press, 2000.
[16] T. R. Shultz and F. Rivest, "Using knowledge to speed learning: A comparison of knowledge-based cascade-correlation and multi-task learning," *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 871-878. San Francisco: Morgan Kaufmann, 2000.
[17] D. L. Silver and R. E. Mercer, "The task rehearsal method of sequential learning," Technical Report #517, Department of Computer Science, University of Western Ontario, 1998.
[18] R. Parekh and V. Honavar, "Constructive theory refinement in knowledge based neural networks," *Proceedings of the International Joint Conference on Neural Networks*. Anchorage, Alaska, 1998.
[19] R. Caruana, "Multitask learning: A knowledge-based source of inductive bias," *Proceedings of the Tenth International Machine Learning Conference*, pp. 41-48. San Mateo, CA: Morgan Kaufmann, 1992.
[20] T. R. Shultz and F. Rivest, "Knowledge-based cascade-correlation: Using knowledge to speed learning," *Connection Science* **13**:1-30, 2001.