Human Skill Transfer: Neural Networks as Learners and Teachers

Michael C. Nechyba and Yangsheng Xu

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Much work in recent years has focused on transferring human skill to robots by abstracting that skill into a machine-understandable, computational model. Such skill models, however, can be used not only for transferring human control strategy to robots, but also for helping less-skilled human operators improve their performance. We propose a two-step approach for transferring skill from human expert to human apprentice. An expert's relevant control strategies or skills are first abstracted into a sensory-based computational model. Afterwards, this trained computational model is used to generate on-line advice for less-skilled operators who need to improve their skill. This advice can take advantage of many different sensor modalities, thereby potentially improving both the quality and speed of learning for the apprentice. Furthermore, this approach allows for the efficient transfer of skill from a single expert to many apprentices, as well as from many experts to a single apprentice. In this paper, we first describe a flexible neural-network-based method for modeling human control strategy and provide motivation for its use. We then present a case study for teaching control strategy from one person to another in this two-step approach of transferring skill.

1. Introduction

Rapid advances in computer technology over the past decade have not been paralleled by corresponding rapid advances in robot capabilities or the development of "intelligent machines." This disparity is principally caused by the difficulty of formalizing intelligent human behavior and decision making processes into an *algorithmic* framework. Humans manage everyday tasks, such as visual processing, manipulation, and mobility, with relative ease; yet robots have not duplicated that performance adequately. Although humans are quite successful at executing these tasks, they are far less successful in formally describing them. In effect human skill remains locked away in the human mind where it is of little use in the development of robot functionality.

As a result, much work has focused on learning computational models of human skill so that human skill may be successfully transferred to robots and machines. In [13], a car is taught to drive autonomously, using a multilayer feedforward neural network to map camera images of the road ahead to a steering direction. In [2], a neural network learns the deburring task for a machining robot with training data from a human expert. In [12], we explore human-to-robot skill transfer for a simple, dynamically stable system. Hidden Markov Models (HMMs) have also been suggested as a possible means for modeling human performance [14] at higher levels of abstraction.

Models of human skill, however, can be used not only for transferring human control strategy to robots, but also for helping a less-skilled operator improve performance. Just as a father might guide his child's arm in learning to throw a football, an expert's skill model can be used to guide the actions of a learning apprentice. Thus, we propose a two-step approach to transferring skill from human expert to human apprentice. Sensory data is first collected from a human expert who is able to successfully perform a specified task; this data is then taken to train a neural network to model the expert's control strategy. Afterwards, the trained neural network, rather than the expert, can be used to give on-line advice to a less-skilled operator. Such advice can be given at each instant in time, and can take one of many forms, based on the error between the apprentice's control actions and those suggested by the expert's trained neural network.

Such an approach to transferring human skill offers several benefits over direct expert-apprentice interaction. By abstracting the expert's skill into a computational model, the model-generated advice can take advantages of many different sensor modalities, potentially improving both the speed and quality of learning for the apprentice. Also, a single expert can efficiently train many apprentices through the model of his/her skill. Conversely, a single apprentice can efficiently benefit from the advice of many experts at once (Figure 1).

This paper is divided into two sections. First, we describe a flexible, cascade neural network architecture for modeling dynamic human control strategy and provide motivation for its use over more conventional feedforward neural networks. Second, we present a detailed case study for training an apprentice from an expert indirectly by building a neural network-based computational model as an intermediary between expert and apprentice.

2. Neural network learning

In modeling human control strategy, we wish to approximate the functional mapping between sensory inputs and control action outputs. Function approximation, in general, is composed of two parts: (1) the selection of an appropriate functional form, and (2) the adjustment of free parameters in the functional model to optimize some criterion. For most neural networks used today, the learning process consists of (2) only, since a specific functional form is selected prior to learning; that is, the network architecture is usually fixed before learning begins.

We believe, however, that both (1) and (2) above have a place in the learning process. Thus, we look towards the flexible cascade learning architecture [4], which adjusts the structure of the neural network as part of learning, for modeling human control strategy. The cascade learning architecture combines the following two notions: (1) a cascade architecture, in which hidden units are automatically added one at a time to an initially minimal net-

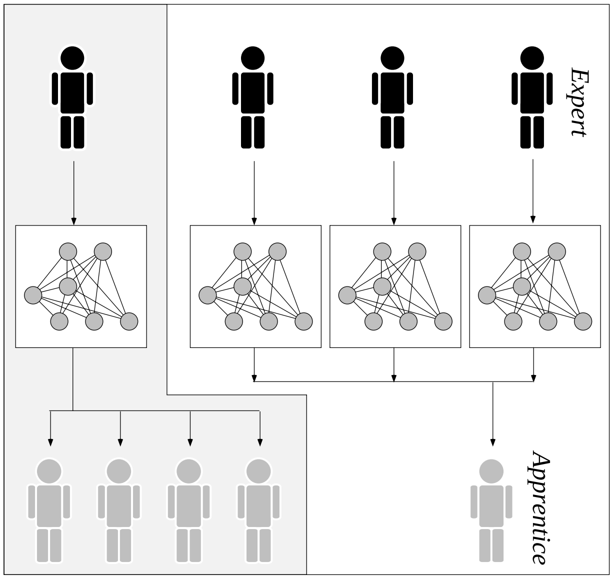

Fig. 1: Following the two-step approach in human-to-human skill transfer allows one expert to teach many apprentices (left) and many experts to contribute to the learning of a single apprentice (right).

work, and (2) the learning algorithm which creates and installs new hidden units as the learning requires in order to reduce the sum-squared difference between the scaled unit outputs and the residual error.

Network training proceeds as summarized below. Initially, there are no hidden units in the network, only direct input-to-output connections. These weights are trained first, thereby capturing any linear relationship between the inputs and outputs. With no further depreciable decrease in the error measure, a first hidden unit will be added to the network from a pool of *candidate* units. Using the quickprop algorithm [3], these candidate units are trained independently and in parallel with different random initial weights.

Again, after no more appreciable error reduction occurs, the best candidate unit is selected and installed in the network. Once installed, the hidden unit input weights are frozen, while the weights to the output units are retrained. By freezing the input weights for all previous hidden units, each training cycle is equivalent to training a three-layer feedforward neural network with a single hidden unit. This allows for much faster convergence of the weights during training than in a standard backpropagation network where many hidden unit weights are trained simultaneously. The process is repeated until the algorithm succeeds in reducing the sum-squared error sufficiently for the training set or the number of hidden units reaches a specified maximum number. Figure 2 below illustrates, for example, how a two-input, single-output network grows as two hidden units are added.

Thus, the cascade architecture relaxes *a priori* assumptions about the functional form of the model to be learned by dynamically adjusting the network size. These assumptions can be further relaxed by allowing new hidden units to have variable activation functions [11][12]. In the pool of candidate units, we can assign a different nonlinear activation function to each unit, rather than just the standard sigmoidal function. During candidate training, the algorithm will select for installment whichever candidate unit reduces the sum-squared error of the training data the most. Hence, the unit with the most appropriate activation function at that point during training is selected. It was shown in [11], that the theoretical properties for multilayer feedforward networks as universal function approximators hold for cascade networks with variable activation functions.

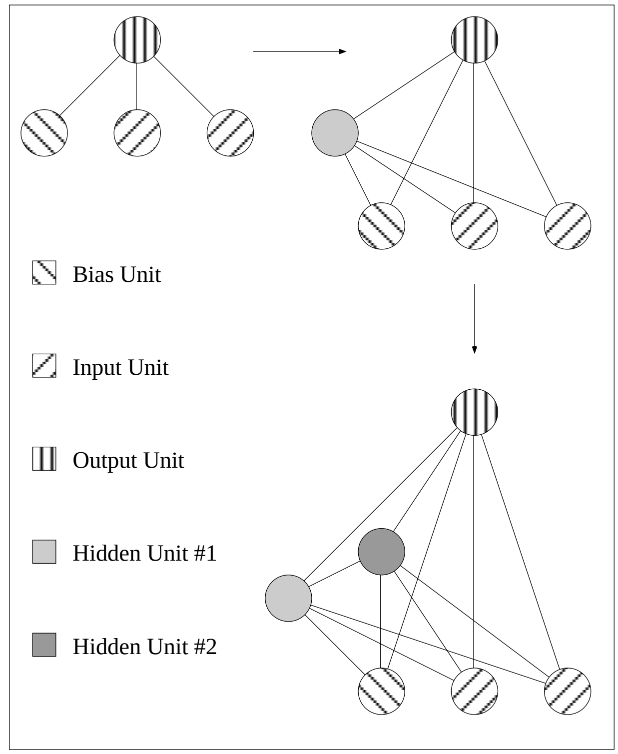

Fig. 2: The cascade two learning architecture adds hidden units one at a time. All connections are feedforward.

The performance of cascade networks with variable activation functions is significantly better in both approximation error as well as convergence rate, compared to cascade networks with only sigmoidal hidden units, or standard multilayer feedforward networks [11][12]. In addition, over repeated trials, it was found that more than 80 percent of all the variable activation functions chosen by the learning algorithm were of some sinusoidal type (i.e sine or cosine). Due to this algorithmic preference for sinusoidal activation functions, cascade networks with exclusively sinusoidal units perform almost as well as cascade networks with variable activation functions in both approximation error and learning speed.

For various case studies of approximating known functions, both variable cascade networks as well as sinusoidal cascade networks outperform sigmoidal cascade networks by a factor of 3 to 10 in approximation error, and learn approximately three times as quickly. These cascade nets also outperform multilayer feedforward networks of comparable size (i.e. equivalent number of free parameters) by a factor of 3 to 15 in approximation error and a factor of 3 to 7 in learning speed [11][12]. Therefore, we feel well motivated in preferring this learning architecture over others due to (1) its efficiency in learning speed, (2) its flexibility in functional form, and (3) its good function approximation properties.

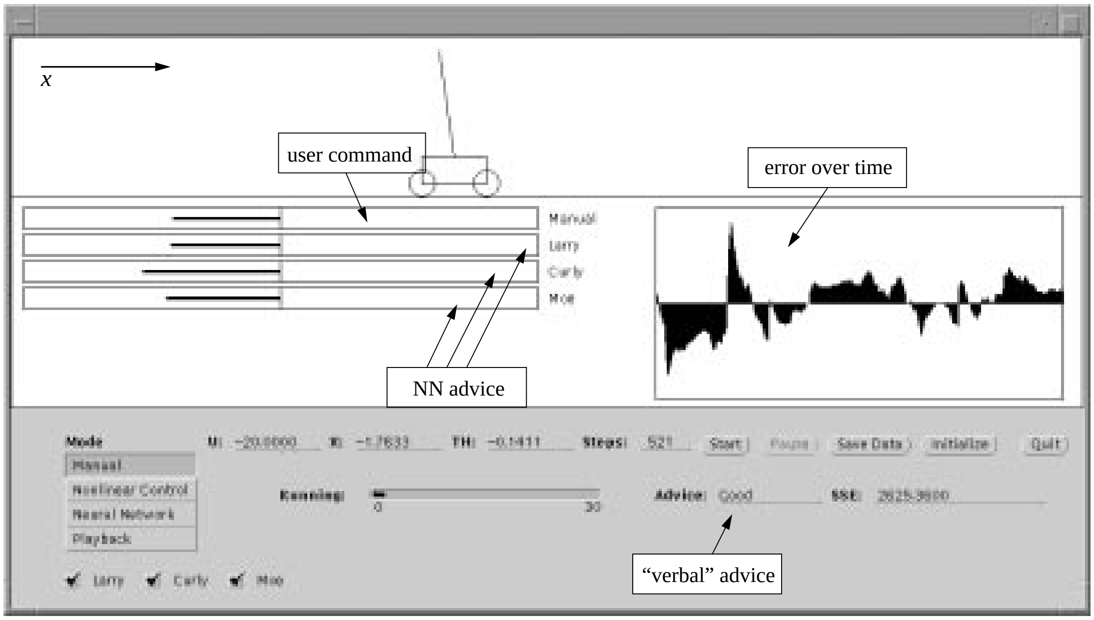

Fig. 3: This interface was used to (1) collect training data, and (2) for training a less skilled individual using a neural network.

3. Neural network teaching

We focus on a simulated inverted pendulum system for studying the transfer of human control strategy from an experienced, expert user, to a less-experienced, less-skilled individual. Below, we describe the experimental setup, the training of the neural network from the expert, as well as the learning from the neural network by the apprentice.

3.1 Experimental Setup

For these experiments, a user is shown an inverted pendulum-cart system on a computer screen (Figure 3), and is able to control the horizontal force to be applied to the cart via the horizontal mouse position. The horizontal force u that may be applied is limited to $-64\ \mathrm{N} \le u \le 64\ \mathrm{N}$, where the dynamics of the system are given in [12]. The angle θ is defined as the deviation from the vertical position; in Figure 3, for example, the angle of the pendulum is $-28.9°$. We simulate the system using the Euler approximation at a frequency of 100Hz, while the task for the user is to keep the pendulum from falling over.

While the expert is controlling the inverted pendulum, the “NN advice,” “verbal advice,” and “error over time” portions of the display in Figure 3 are not shown. That is, the expert is given no advice on how to stably control the system. By “expert,” we refer to those people who keep the inverted pendulum from falling for longer than 30 seconds. The control of the system is difficult enough so that some three fourth of the people (about 20) who attempted to control the system failed, even after repeated attempts.

Data from successful runs are then taken to train cascade networks to model the expert control strategy. Afterwards, these trained networks are used to give on-line advice to less-skilled operators at each time step. The user can choose one of a number of trained expert cascade networks for guidance. In Figure 3, that guidance appears in several formats. The “NN advice” portion of the display graphically illustrates the control that each of the expert cascade networks for Larry, Curly and Moe would execute at that instant in time. The “verbal advice” indicates a suggested control correction verbally based on one of the expert cascade networks. Finally, the “error over time” portion of the display provides a time history of the error between the user’s control actions and those suggested by one of the expert cascade networks.

3.2 Human Control Strategy Modeling

In all, five different people succeeded in controlling the pendulum simulation for longer than 3000 time steps, or 30 seconds in a single trial: Larry, Curly, Moe, Groucho, and Harpo. From each person’s training run, 1500 data points were randomly selected to train cascade networks, while another 1500 data points were used for cross validation. The networks to be trained are provided 11 inputs, namely, the past 10 values of θ [9][10], as well as the velocity of the cart $\dot{x}$,

$$\{\theta(k-9), \theta(k-8), \ldots, \theta(k-1), \theta(k), \dot{x}\} \qquad \text{(Eq. 1)}$$

As output, the networks are to generate the horizontal force to be applied at the next time step, $u(k+1)$. Learning was stopped after the error in the cross validation set no longer decreased.

Over many trials, all trained cascade networks formed stable trajectories for initial values in at least some region of the complete state space. That is, the cascade networks are able to successfully abstract the essential features of each person’s control strategy. Furthermore, Table 1 illustrates good generalizing properties of the resulting cascade networks. The first two data columns list the range of θ values in the training data presented to each cascade network. The third and fourth data columns compare these ranges to the minimum and maximum initial θ values for which the best resulting cascade networks converge to a stable trajectory. These values indicate that the cascade networks appear to generalize very well to states outside the range of the training data.

Below, we show how learning proceeds as hidden units are added to the cascade network. Figure 4, for example, shows part of the training run for Groucho.

Table 1: Generalization of Control Strategies

	Range of θ in training data		*Range of initial θ for stable trajectories in best cascade networks*	
	θ_{min}	θ_{max}	θ_{min}	θ_{max}
Larry	–24.4°	36.2°	–76.8°	76.7°
Curly	–24.9°	30.2°	–78.5°	77.6°
Moe	–24.5°	31.2°	–76.4°	76.0°
Groucho	–31.1°	19.4°	–69.8°	67.7°
Harpo	–21.9°	21.2°	–84.1°	83.7°

Figures 5(a)-(c) show the pendulum trajectory (in phase space) generated by Groucho's cascade network for $\theta_{init} = 0.2$ rad. Figure 5(a) corresponds to the cascade network at one hidden unit; Figure 5(b) corresponds to six hidden units; and Figure 5(c) corresponds to 11 hidden units.

With only one hidden unit, the cascade network model is forced to be nearly linear. The resulting pendulum trajectory is therefore smooth and convergent, and can bear little resemblance to the training trajectory (Figure 4). By the time six hidden units have been added to the network, the trajectory pattern is beginning to show initial indications of nonlinear behavior. At eleven hidden units, the resulting pendulum trajectory forms a nonconvergent, stable attractor, with significantly more intricate and nonlinear behavior. Some of the finer details of Groucho's control strategy now begin to emerge.

Thus, the cascade network learning process allows one to choose from a number of resulting models of the human's control strategy, ranging from simple to complex. One may be interested in both a coarse approximation of the control behavior, as well as more detailed models which expose long-term nonlinear patterns of behavior. By examining the magnitude of the weights in the simple cascade network with one hidden unit, for example, one can observe that the important inputs for the neural network in the sequence,

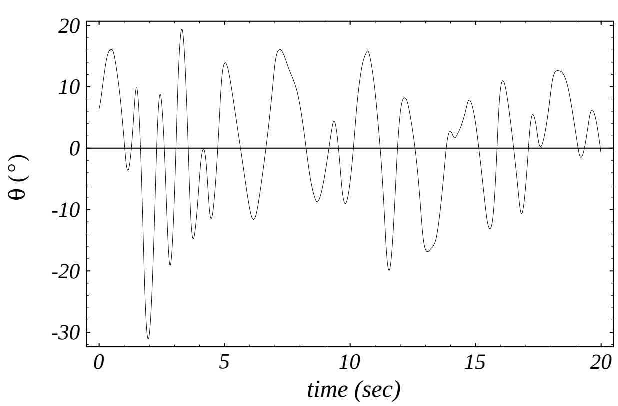

Fig. 4: Training data for cascade network from Groucho.

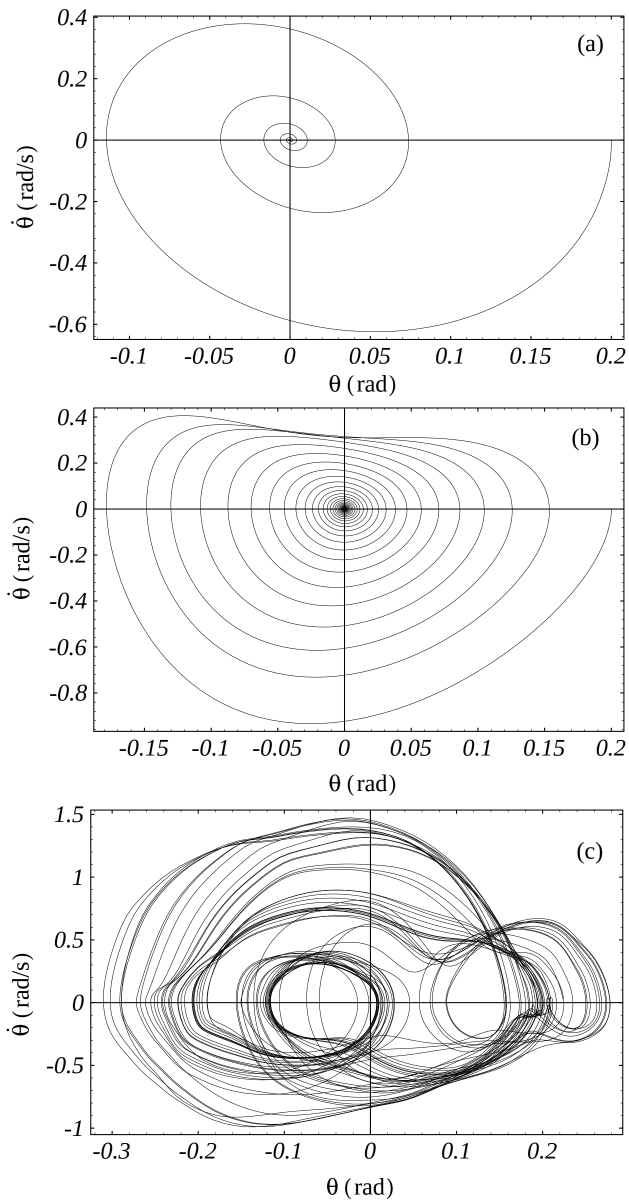

Fig. 5: Pendulum trajectory in phase space for Groucho's network with (a) one, (b) six, and (c) eleven hidden units.

$$\{\theta(k-9), \theta(k-8), \ldots, \theta(k-1), \theta(k)\} \quad \text{(Eq. 2)}$$

are in fact,

$$\{\theta(k-9), \theta(k-4), \theta(k)\} \quad \text{(Eq. 3)}$$

and, moreover, together, they form a first-order derivative. On the other hand, the more complex eleven-hidden-unit network exposes a long-term control strategy which more closely resembles that of the original training data, and therefore gives a truer model of the human control strategy. Each network of increasing complexity imparts different important information about the underlying human model.

3.3 Human Control Strategy Transfer

We are now interested in using one of the models of an expert's human control strategy in order to teach a less skilled individual to successfully control the inverted pendulum system. Two principal causes for failure in the stable control of the system are (1) overreaction, (i.e. the “control gains” are too high), and (2) force limits (in the interface) which do not allow recovery from error. Thus, in order to successfully control the system, a human operator must deal effectively with these two problems.

One of the individuals (Zipo) who had repeatedly failed in controlling the simulation for the reasons cited above was asked to train himself by getting constant feedback advice in the form of a visual display (as explained in section 3.1 above) during the simulation runs. This advice would be generated by one of the previously trained cascade networks.

First, Zipo experimented with different cascade networks from which to train. Eventually, he settled on a cascade network with two hidden units from Curly. This model was chosen because the “recommended” control forces were (1) relatively small and (2) varied relatively smoothly compared to other networks. The model was also tested independently and converged from a wide range of extreme initial states.

Zipo then spent one hour learning from the cascade network model by observing the “recommended” control force as he was controlling the cart on the simulation display (see Figure 3). The length of each consecutive successful trial increased steadily, where, near the end of learning, successful trials of up to 220 seconds were achieved.

At the conclusion of training with the visual aids, the “advice” portions of the interface were then turned off, and Zipo was left to control the system on his own once again. Without help, Zipo now demonstrated the ability to control the simulation virtually indefinitely, with only patience and eye fatigue being limiting factors. Thus, a person who had previously failed in repeated trials to control the simulation successfully, had now managed to learn a stable control strategy with the advice of an expert-trained neural network. Figures 6(a)-(b) below shows one portion of a post-training successful run. From t=262 sec to t=269 sec, it is clear that Zipo has, at least to some extent, learned how to deal with extreme conditions (i.e a large θ value) without overreacting and within the confines of limited force.

Below, we analyze how closely Zipo did adopt the control strategy of Curly. First, we take one of the long post-training successful runs (of about 285 seconds), and compare, at each time step, the “recommended” control actions for the fully trained cascade networks of Larry, Curly, Moe, Groucho, and Harpo. Table 2 below summarizes those results. The control actions of Zipo do approximate, most closely, those of Curly, as the mean RMS error is the lowest for Curly’s cascade networks.

Table 2: Comparison of Control Strategies

	# networks tested	*Mean. RMS error*
Larry	10	4.76
Curly	10	2.96
Moe	10	6.95
Groucho	10	3.23
Harpo	10	4.44

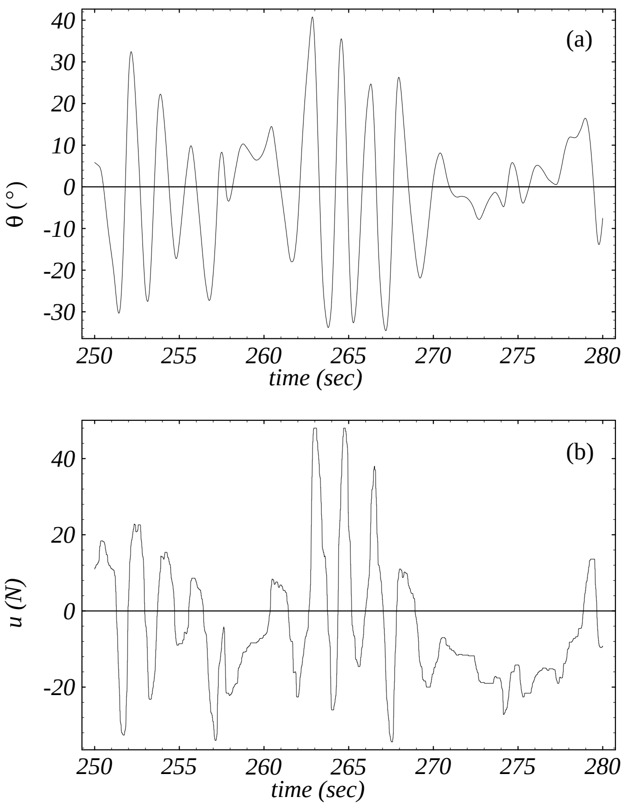

Fig. 6: (a) Segment of pendulum trajectory for learned control strategy (b) Corresponding commanded control force shows ability to deal with overreaction and limited force.

Second, we train a cascade network on one of Zipo’s post-training successful runs, and compare, at each time step, the control action of Zipo’s network versus the control actions of the other networks for trajectories with various initial conditions. Predictably, Zipo’s and Curly’s networks compare the closest, as is summarized in Table 3 below.

4. Discussion

The results in the previous two sections give some empirical evidence that (1) cascade networks are capable of abstracting human control strategy to various level of detail, and (2) that these networks can function as teaching tools for less skilled individuals. Although, the case study presented in this paper is limited, it does suggest that the two-step paradigm for human-to-human transfer of skill is a valid one, and that it may offer benefits beyond those of direct expert-to-apprentice instruction. A number of interesting areas for future research arise.

First, we note that even for this relatively simple simulation of an inverted pendulum, the expert-based control strategy is preferred by the apprentice for learning over other, more conventional mathematical models, such as a nonlinear partitioned control law. Although a mathematically derived control strategy may be more optimal in some sense, it fails as a teacher because (1) it relies on precision of which neither the human nor the human-computer interface is capable, and (2) a human’s reaction times are necessarily slower than the internal simulation speed. Thus, even when good mathematical control laws exist for a given problem, human-based models are still required for human-to-human transfer of skill.

Table 3: Comparison of Control Strategies

	# cascade networks tested	*Mean RMS error*	*95% confidence interval* [a]
Larry	10	3.90	[1.60, 2.01]
Curly	10	2.10	---
Moe	10	4.93	[2.57, 3.09]
Groucho	10	2.25	[0.03, 0.29]
Harpo	10	3.29	[0.98, 1.41]

a. Confidence interval for difference in mean RMS error, based on apprentice's *t*-test, between Curly's and the other networks.

Second, a number of interesting issues arise in both the collection of sensory data from humans, as well as the best sensor modalities to exploit in providing feedback to a apprentice. For the case study presented in this paper, the sensory data from the human was easy to collect, since we needed only to measure the position of the mouse. In more complicated examples, however, where part or all of the human body may be involved in a control strategy, we might be required to monitor the motion of key points (i.e. joints, perhaps) on the entire body. Additionally, these sensors must be as nonintrusive as possible, lest by observing control strategy, we actually alter it.

The type of feedback advice provided to a apprentice is also a critical issue. For our example, the feedback was provided visually and/or verbally. A apprentice may be tempted, however, to begin to rely too much on the feedback signal, and to focus too little on the task at hand. In our case, Zipo was continually reminded to pay as much or more attention to the consequences of his control actions, as to the advice by Curly's cascade network. Thus, initially a apprentice can rely heavily on the feedback advice, and as time goes on, begin to rely less and less on that advice.

The feedback advice should, in general, be structured so as to make it as difficult as possible to rely too heavily on the feedback. For the case study presented in this paper, another possibility for sensor feedback might have been to play a sound whose frequency would be proportional to the error between the apprentice's control action and the cascade network's advice. Different sensors, the eyes for controlling the simulation and the ears for receiving feedback on performance, rather than just the eyes, would therefore have been utilized by the apprentice in learning. Clearly, more work needs to be done in exploring which sensor modalities lead to the best and fastest learning.

Finally, human-to-human transfer of control strategies or skill is, by far, a more difficult proposition than human-to-robot transfer of skill. Robots make better apprentices in the sense that they require no special interfaces, generally have the ability to measure their state with greater precision than humans, and can follow commands, without pause, at a faster rate than humans. Consequently, progress in transferring skill from one human to another human will lead to additional progress in transferring skill from humans to robots.

5. Conclusion

Modeling expert human control strategies for challenging dynamic tasks in order to train less skilled individuals is an important application of intelligent control and system modeling research. It relieves the burden on the expert, as well as allows many more apprentices to benefit indirectly from the skilled advice of a single expert. In addition, the control strategy model may afford the opportunity to provide a more diverse array of sensory feedback to the apprentice. This could both accelerate learning as well as improve the quality of the learned skill.

Acknowledgments

We would especially like to thank Ofer Barkai, Randy Casciola, Andrew Gove, John Murphy, Mark Ollis for their help and support in carrying out this work.

References

[1] Antsaklis, P. J., Guest Editor, Special Issue on Neural Networks in Control Systems, *IEEE Control System Magazine*, Vol. 12, No. 2, pp. 8-57, 1992.

[2] Asada, H., and Liu, S., "Transfer of Human Skills to Neural Net Robot Controllers," *Proc. IEEE Int. Conf. on Robotics and Automation*, Vol. 3, pp. 2442-2447, 1991.

[3] Fahlman, S. E., "An Empirical Study of Learning Speed in Back-Propagation Networks," Technical Report, CMU-CS-TR-88-162, Carnegie Mellon University, 1988.

[4] Fahlman, S.E., and Boyan, J.A., "The Cascade Two Learning Architecture," Technical Report, CMU-CS-TR-94-100, Carnegie Mellon University, 1994.

[5] Guez, A., Selinsky, J., "A Trainable Neuromorphic Controller," *Jour. of Robotic Sys.*, Vol 5., No. 4, pp. 363-388, 1988.

[6] Hunt, K. J., *et. al.*, "Neural Networks for Control Systems - A Survey," *Automatica*, Vol. 28, No. 6, pp. 1083-1112, 1992.

[7] Lee, C. C., "Fuzzy Logic in Control Systems: Fuzzy Logic Controller," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 20, No. 2, pp. 404-435, 1990.

[8] Miller, W. T., Sutton, R. S., and Werbos, P. I., Editors, "Neural Networks For Control," MIT Press, 1990.

[9] Narendra, K. S., "Adaptive Control of Dynamical Systems Using Neural Networks," *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*, White D. A., and Sofge D. A., ed., pp. 141-184, 1992.

[10] Narendra, K. S., Parthasarathy, K., "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Trans. on Neural Networks*, Vol. 1, No. 1, pp. 4-27, 1990.

[11] Nechyba, M., and Xu, Y., "Neural Network Approach to Control System Identification with Variable Activation Functions," P*roc. IEEE Int. Symp. Intelligent Control*, pp. 358-363, 1994.

[12] Nechyba, M., and Xu, Y., "Towards Human Control Strategy Learning: Neural Network Approach with Variable Activation Functions," Technical Report, CMU-RI-TR-95-09, Carnegie Mellon University, 1995.

[13] Pomerleau, D. A., "Neural Network Perception for Mobile Robot Guidance," Ph.D. Thesis, School of Computer Science, Carnegie Mellon University, 1992.

[14] Yang, J., Xu, Y., and Chen, C., "Hidden-Markov Model-Based Learning Controller," *Proc. IEEE Int. Symp. Intelligent Control*, pp. 39-44, 1994.