A Cascade-Correlation Model of Balance Scale Phenomena

Thomas R. Shultz and William C. Schmidt

Department of Psychology
McGill University
1205 Penfield Avenue
Montreal, Quebec, Canada H3A 1B1
shultz@psych.mcgill.ca

Abstract

The Cascade-Correlation connectionist architecture was used to model human cognitive development on balance scale problems. The simulations were characterized by gradual expansion of the training patterns, training bias in favor of equal distance problems, and test problems balanced for torque difference. Both orderly rule stages and torque difference effects were obtained. Analyses of the development of network structure revealed progressive sensitivity to distance information. It was noted that information salience effects, such as that for torque difference, are particularly difficult to capture in symbolic level models.

Introduction

An emerging benchmark for detailed computational modeling in cognitive development is the balance scale. The clarity and replicability of balance scale phenomena with humans, coupled with the classical developmental appeal of its stage-like character, have led to both connectionist (McClelland 1988) and rule based (Newell 1990) models.

Psychological assessments present the child with a rigid balance beam in which differing numbers of weights are placed on pegs at various distances to the left or right of a fulcrum. The child's task is to determine which side of the scale will go down when supporting blocks are removed. A 5 position, 5 weight version of the balance scale is presented in Figure 1. Ordinarily, all of the weights on one side are placed on a single peg. *Balance* problems have equal numbers of weights placed at equal distances. In *weight* problems, the side with more weights goes down since the distances are equal. In *distance* problems, the side with greater distance goes down since the sides have equal weights. The *conflict* problems have greater weight on one side and greater distance on the other side. The side that goes down is the one with greater weight for *conflict-weight* problems, and the one with greater distance for *conflict-distance* problems. The scale balances in *conflict-balance* problems.

Siegler (1976, 1981) has indicated that children's performance on the balance scale progresses through 4 distinct rule based stages: (1) use weight alone to determine if the scale will balance, (2) emphasize weight, but consider distance (correctly) in the event that the weights to the left and right of the fulcrum are equal, (3) consider both weight and distance but get confused when one side has greater weight and the other has greater distance, (4) multiply distance by weight for each side and compare the products. Siegler has noted that each rule makes specific predictions about the kinds of problems that will be solved as illustrated by the predicted percentages correct in Figure 1.

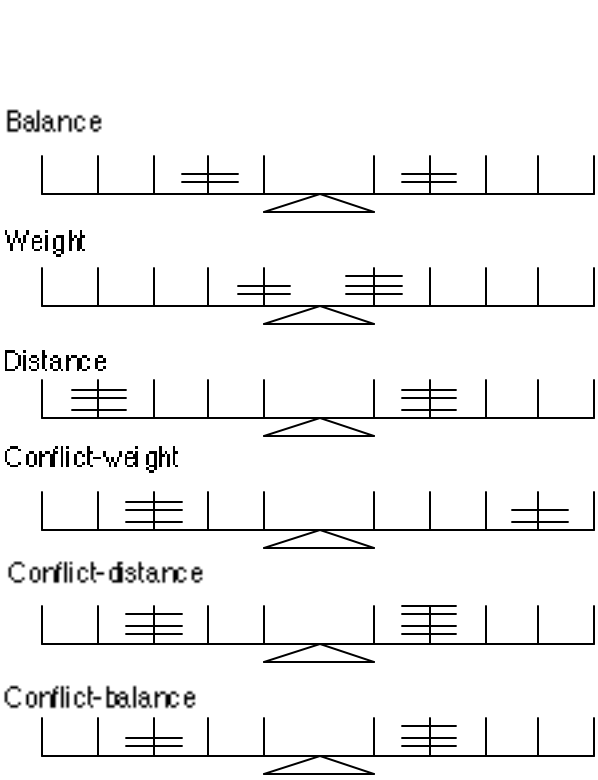
McClelland (1988) reported a pioneering simulation of balance scale stages using a connectionist network with a back-propagation learning rule. This model required a number of limiting assumptions: a strong bias in the training patterns favoring equal distance problems, a local binary representation of weight and distance information, and a forced segregation of weight vs. distance information in connections to the hidden units. Even with these assumptions satisfied, McClelland reported that there was a great deal of shifting back and forth between rules 3 and 4, with stage 4 never being clearly established.

Interestingly, the leading rule learning program, Soar, has also been applied to balance scale phenomena (Newell 1990). Soar acquired rules 1-3 but, like the back-propagation model, did not manage rule 4. It is unclear how dependent the Soar model was on getting balance scale problems in a certain order. It may well be that different problem orders yield different orders of rules.

Neither the back-propagation model nor the Soar model attempted to capture the other major balance scale phenomenon, the torque difference effect (Ferretti & Butterfield 1986). The torque on each side of the fulcrum is the product of weight x distance. Torque difference is the absolute difference between the torques on the two sides. The larger the torque difference, the easier the problem is for children to solve. This could be regarded as an effect of information salience. It is not explainable by Siegler's rules since any such rule should apply regardless of the torque difference. Nor is the torque difference effect explainable by the additive or multiplicative rules of information integration theory (Wilkening & Anderson 1982).

In this paper, we report our attempt to cover rule stages and the torque difference effect in the balance scale using a relatively new connectionist architecture called Cascade-Correlation (Fahlman & Lebiere 1990). Cascade-Correlation builds its own network topology by recruiting new hidden units as it learns to solve problems. Thus, it affords a more principled approach to network construction than is typical in connectionist research. It starts with a minimal topology containing only the input and output units defined by the programmer. In what is called the output phase, connections to the output units are trained until error can no longer be reduced. Then, in the

input phase, a pool of candidate hidden units receives trainable input from the input units and any existing hidden units. Outputs from the candidate hidden units are not yet connected to the output units in this phase. The purpose of the input phase is to identify the candidate unit whose



	Rules			
	1	2	3	4
Balance	100	100	100	100
Weight	100	100	100	100
Distance	0	100	100	100
Conflict-weight	100	100	33	100
Conflict-distance	0	0	33	100
Conflict-balance	0	0	33	100

Figure 1. Balance scale problems and predicted success.

activations correlate best with the output errors. This best candidate unit is then installed in the network, receiving input from all input units and any hidden units already in place and sending output to all of the output units. Once installed, the input side weights to the new hidden unit are frozen, and its output side weights are allowed to change with learning (output phase). Because Cascade-Correlation uses second order error minimization in computing weight changes and learns only one level at a time, it is typically 10-50 times faster at learning than back-propagation.

Although Cascade-Correlation has not yet been applied to cognitive developmental phenomena, it appears to afford a novel and natural interpretation of both qualitative and quantitative developmental changes. Qualitative changes occur through the recruitment of new hidden units; quantitative changes through the adjustment of network weights.

The Network and the Training

We report two experiments, one on rule diagnosis and the other on the torque difference effect. Both experiments used a 5 peg, 5 weight version of the balance scale, as did McClelland (1988). Our initial network had 4 input units, the obligatory bias unit (which is always on), and 2 output units. Of the 4 input units, one encoded left-side distance, a second encoded left-side weight, a third encoded right-side distance, and the fourth encoded right-side weight. The input coding of distance and weight information was done using integers from 1 to 5. There were 2 sigmoid output units which represented actual balance scale results in a distributed fashion. Left-side down was conveyed by excitation of the first output and inhibition of the second output; right-side down was conveyed by the reverse pattern; and balance was conveyed by neutral values on both outputs. Any recruited hidden units also used a sigmoid activation function.

The initial training pattems were 100 randomly selected without replacement from the 625 possible 5 peg, 5 weight problems, subject to a 0.9 bias in favor of equal distance problems (balance and weight problems, as illustrated in Figure 1). This bias ensured that the probability of drawing an equal distance problem during construction of the training patterns was 0.9. On each epoch in the output phase, another training pattern was randomly selected with replacement, also subject to the 0.9 equal distance bias, and added to the training patterns. We call this Expansion training of the 1+ type. The training set gradually expanded, with 1 new pattern added each output phase epoch. Expansion training conforms to our assumptions that the child's environment changes gradually and that these changes are marked by exposure

to more aspects of the environment. The constant bias for equal distance problems reflects the assumption that children have lots of experience lifting differing numbers of objects but relatively little experience placing objects at different distances from a fulcrum.

Pilot experiments had established that rule stages 1 and 2 could not be obtained without a strong bias for equal distance problems; the network went directly to stages 3 and 4. Other pilot experiments indicated that learning was extremely difficult when 100 training patterns were randomly selected each epoch.

We used default parameter values for Cascade-Correlation (Fahlman & Lebiere 1990), with two exceptions. We lowered the input and output Epsilons (learning rates) by 1/2 in order to reduce the bounce in errors from epoch to epoch. We have noticed such error bounce often in using integer-coded input. Also, we used a score-threshold of 0.25. Normally, Cascade-Correlation continues training until all the activations of output units are within score-threshold of their targets in the training patterns. The default score-threshold of 0.4 is appropriate for sigmoid units on a threshold and margin criterion. But because our output units were also coding neutral (balance) patterns, we lowered score-threshold to 0.25 in order to achieve non-overlapping scoring ranges. An output activation had to be equal to or greater than absolute 0.25 in order to count as anything but balance.

Each experiment involved 16 runs. Each run was terminated at 300 epochs because pilot testing had established that most runs were well within stage 4 by that time. With Expansion training, complete mastery of the training patterns is quite difficult to achieve until most of the training patterns have been seen.

Experiment 1: Rule Diagnosis

Each of the 16 runs used distinct, randomly selected training and test patterns. The 24 test patterns in this experiment were balanced for both problem type and torque difference, such that there were 4 patterns from each of the 6 problem types in Figure 1, 1 pattern representing each of 4 levels of torque difference: 1, 2-5, 6-9, and 10-20. On each epoch during the output phase, the network was tested with the 24 test patterns. A test problem whose output activations were both within score-threshold of their correct targets was coded as correct; any other test problems were coded as incorrect.

The patterns of correct and incorrect problems were used to diagnose rule use. A diagnosis of rule 4 required 20 or more test problems correct; rule 2 required 13 or more correct on balance, weight, distance, and conflict-weight problems and less than 3 correct on conflict-distance and conflict-balance problems; rule 3 required 10 or more correct balance, weight, and distance problems and less than 10 correct on conflict problems; rule 1 required 10 or more correct balance, weight, and conflict-weight problems and less than 3 correct distance, conflict-distance, and conflict-balance problems. Scoring priority for these rules, in decreasing order, was 4, 2, 3, and 1. Rule 2 was given a higher priority than rule 3, because rule 2 produces fewer errors on conflict-weight problems, as shown in Figure 1.

Figure 2 shows a plot of the rule diagnosed at each output epoch for a representative computer subject. It shows a typically orderly progression through Siegler's rules. The *H* on the bottom of the plot signifies where a hidden unit was added to the network. Tabulation of rule diagnosis results revealed that 11 of the 16 subjects showed the predicted 1 2 3 4 ordering. Two other subjects showed rules 1 2 3; 1 showed rules 1 2 4; 1 showed rules 1 2 4 with regression to 3 and 2; and 1 showed rules 1 2. It is quite likely that all subjects would have reached rule 4 with continued training. The overlap between the diagnoses of adjacent rules near transition points reflected the tentative nature of each transition.

Of the 16 computer subjects, 9 recruited 1 hidden unit, 6 recruited 2 hidden units, and 1 recruited 3 hidden units. Of these 24 hidden units, 13 were associated with a very quick progression from one rule to the next: 5 moved up to rule 4, 7 to rule 3, and 1 to rule 2.

To better understand developing network structure, we drew Hinton diagrams in the middle of each rule stage. Each such diagram shows the size and sign of incoming weights at a particular epoch. Size of weight is indicated by the size of the square; sign of the weight is indicated by the color of the square, with white indicating positive and black negative. Hinton diagrams for a representative subject are presented in Figure 3. The first epoch number in each diagram excludes input phases; the second epoch number (in parentheses) includes input phases.

During rule 1, which uses only weight information, the output units were highly sensitive to weight information. The right-side down output received a positive signal from the right-side weight input, whereas the left-side down output received a positive signal from the left-side weight input. During rule 2, which continues to use weight but begins to use distance when the weights on each side are equal, the network's outputs became more sensitive to distance information. The differential sensitivity to sides was retained, and the new hidden unit was particularly sensitive to weight information. During rule 3, which is characterized by the use of both weight and distance information but confusion when these are in conflict, the outputs became about as sensitive to distance as to weight. And in rule 4, which signifies nearly correct performance, a new hidden unit emerged that was particularly sensitive to distance information. The two hidden units, one representing mainly weight and the other mainly distance, sent opposite signals to the outputs.

More generally, we found that, of the 21 hidden units with Hinton relevance, 8 were especially sensitive to side information, 11 were mainly sensitive to side x distance or side x weight, and 2 were mainly sensitive to the bias unit or to older hidden units.

Experiment 2: Torque Difference Effect

This experiment employed exactly the same techniques as Experiment 1, except that the principal interest was in recording errors for 4 torque difference levels: 1, 2-5, 6-9,

and 10-20. For each run, 4 sets of test patterns were randomly selected with 4 problems of each of the 6 problem types in Figure 1. Each set of test patterns contained only problems representing 1 of the 4 torque difference levels.

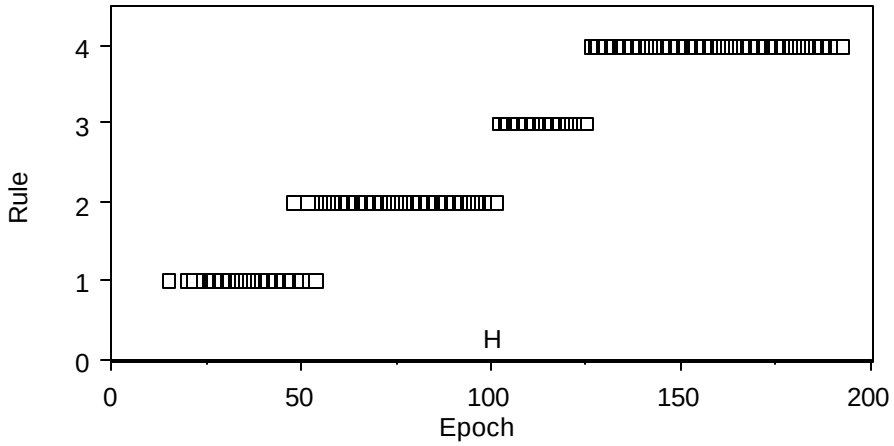


Figure 2. Rule diagnosis for 1 subject.

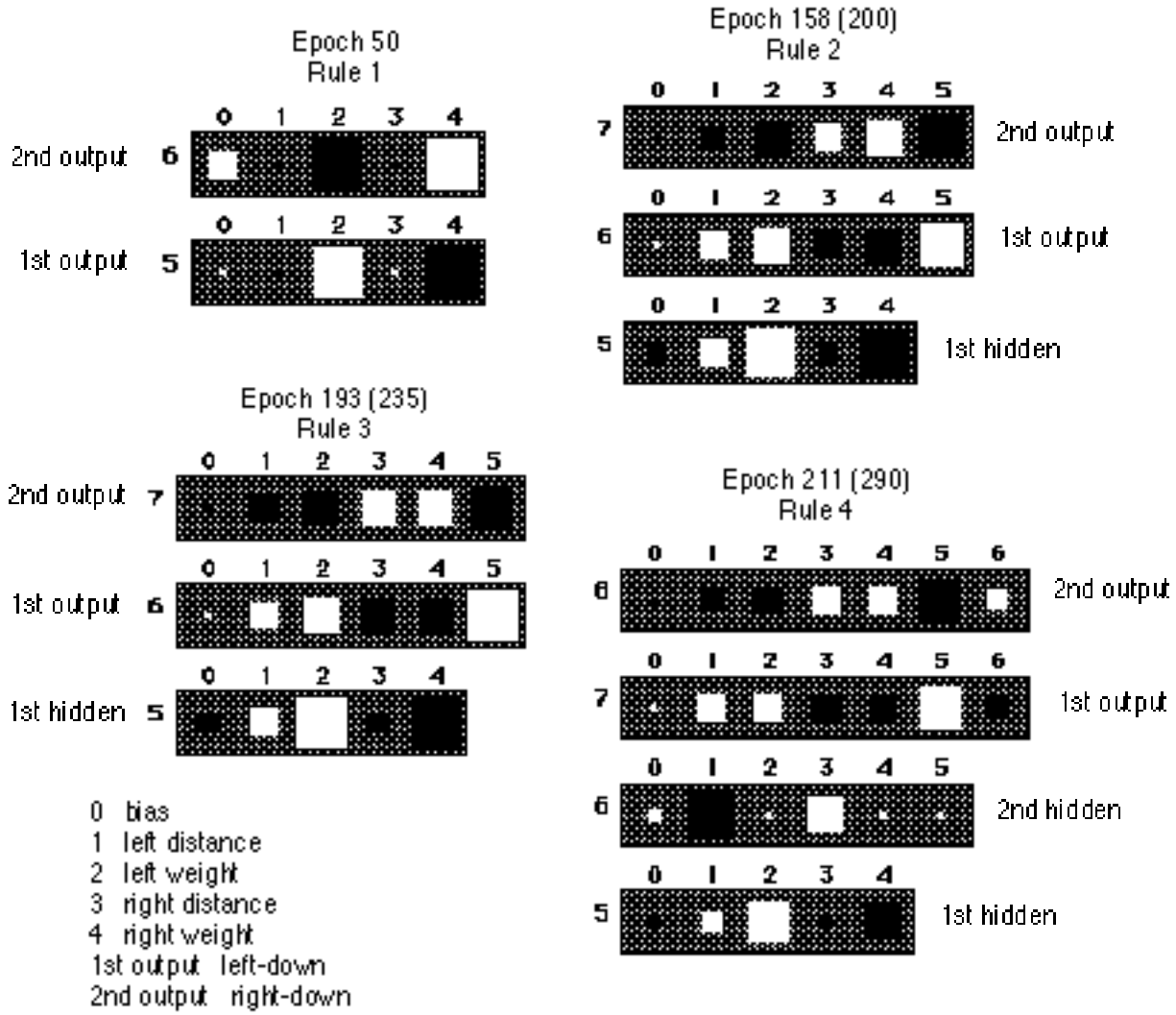


Figure 3. Hinton diagrams of incoming weights for 1 subject.

Errors are plotted over epochs for a representative subject in Figure 4. As expected, this subject showed faster and deeper error reduction with increasing torque difference.

An ANOVA of these error signals midway (epoch 75) and late (last epoch) in learning was performed for all subjects, yielding only a main effect for torque difference level, $F(3, 42) = 48.57$, $p < .001$, with a strong negative linear trend, $F(1, 42) = 140.45$, $p < .001$. The mean errors at these two epochs for the 4 torque difference levels are presented in Figure 5. The larger the torque difference, the smaller the error.

Discussion

In these simulations, Cascade-Correlation networks learned to perform on balance scale problems as if they were following rules, including clear performance at the level of stage 4. Further, these rules emerged in the psychologically correct order, almost without exception. Some developmental regressions and stage-skipping were observed, just as in human subjects. Unlike previous models, these nets also captured the torque difference effect. The present models are, of course, highly simplified compared to the environment and computational resources of children.

The Cascade-Correlation networks covered these psychological phenomena without at least some of the restrictive assumptions of McClelland's (1988) back-propagation networks. We didn't need to encode weight and distance inputs in local binary form, or implant segregated hidden units for weight vs. distance information, or indeed implant any hidden units at all. We did, however, follow McClelland's lead in strongly biasing the training patterns in favor of equal distance problems. Such input bias may not be the only way to obtain human-like stages in connectionist models of the balance scale, but it's effectiveness in producing stages may encourage researchers to examine biases in the child's environment.

Like other early connectionist attempts to model phenomena in cognitive development (Chauvin 1989; McClelland 1988; Plunkett & Marchman 1989), the present simulations suggest that the connectionist approach deserves serious consideration as a means of studying transition mechanisms for higher level reasoning. Connectionist networks appear capable of reproducing classic developmental phenomena such as rules and stages, as well as more subtle effects such as information salience that explicit symbolic rule systems have particular difficulty with.

An explicit symbolic rule-based model trying to capture the torque difference effect would presumably find itself in the paradoxical position of having to compute torque differences well before stage 4. It might require rules of the form *if torque difference is greater than x then apply rule i*, where x is some integer between 1 and 20 that decreases with age, and i is the current stage. Such a model would apparently have to compute and use torque differences to mimic the torque difference effect well before it could compute and use torques to solve balance scale problems. This would possibly fit the psychological data, but would be extraordinarily awkward.

Acknowledgements

This research was supported by a grant from the Natural Sciences and Engineering Research Council of Canada. We are grateful to Scott Fahlman for providing the code for the Cascade-Correlation algorithm and for suggestions on running the simulations, and to Chris Schunn for providing the code for automatic drawing of Hinton diagrams.

References

Chauvin, Y. 1989. Toward a connectionist model of symbolic emergence. In Proceedings of the Eleventh Annual Conference of the Cognitive Science Society, 580-587. Hillsdale, NJ: Lawrence Erlbaum.

Fahlman, S. E., & Lebiere, C. 1990. The Cascade-Correlation learning architecture. Technical Report, CMU-CS-90-100, School of Computer Science, Carnegie-Mellon University.

Ferretti, R. P., & Butterfield, E. C. 1986. Are children's rule assessment classifications invariant across instances of problem types? *Child Development* 57:1419-1428.

McClelland, J. L. 1988. Parallel distributed processing: Implications for cognition and development. Technical Report, AIP-47, Department of Psychology, Carnegie-Mellon University.

Newell, A. 1990. *Unified theories of cognition*. Cambridge, MA: Harvard University Press.

Plunkett, K., & Marchman, V. 1989. Pattern association in a back propagation network: Implications for child language acquisition. Technical Report, #8902, Center for Research in Language, University of California, San Diego.

Siegler, R. S. 1976. Three aspects of cognitive development. *Cognitive Psychology* 8:481-520.

Siegler, R. S. 1981. Developmental sequences between and within concepts. *Monographs of the Society for Research in Child Development* 46:Whole No. 189.

Wilkening, F., & Anderson, N. H. 1982. Comparison of two rule-assessment methodologies for studying cognitive development and knowledge structure. *Psychological Bulletin* 92:215-237.

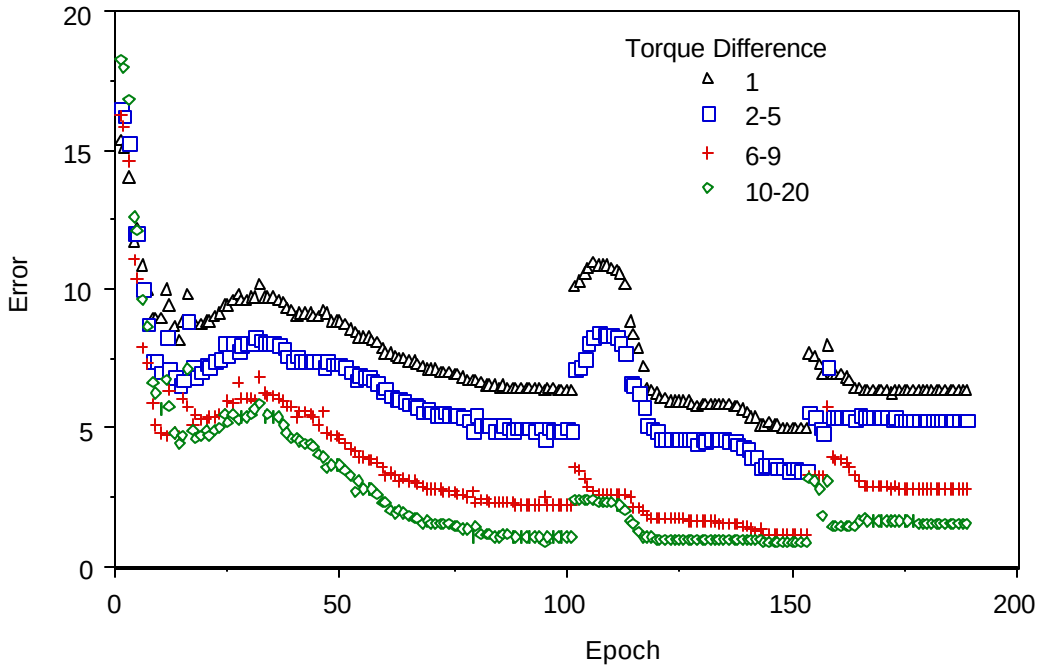


Figure 4. Errors on test problems at 4 torque difference levels for 1 subject.

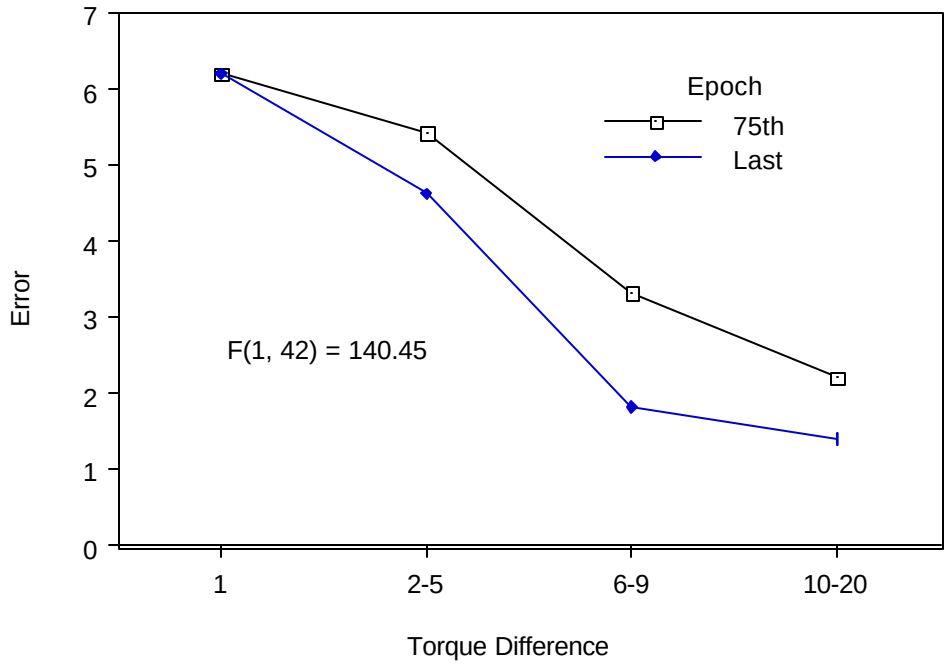


Figure 5. Mean errors on test problems at 4 torque difference levels.