Cooperating Artificial Neural and Knowledge-Based Systems in a Truck Fleet Brake-Balance Application

Michael Lawrence Smith

A proprietary air brake-balance analysis system for trucks gathers five sets of data relating air pressure, time, braking force, and temperature. Each test produces a complex, color graph plotted against axes chosen from pressure, time, and temperature. A human expert can make impressive diagnoses about the brake and air systems after studying these graphs. I describe five artificial neural networks that are trained to render a judgment about these graphs and a knowledge-based system that accepts these judgments and combines them with additional information to arrive at a precise problem identification and a procedure to solve the problem. The brake-balance system is innovative because it uses a rare approach to a real problem: cooperative problem solving and diagnostics between a knowledge-based system and a suite of neural networks. Success rates are 90 percent for the neural nets and 100 percent for the knowledge-based system. The annual savings is at least $100,000.

Truck Fleet Brake-Balance Analysis

Since the fuel crisis of the 1970s, the braking force required to stop and slow trucks has risen dramatically. Fuel economy considerations have eliminated two mechanisms formerly relied on to assist with the braking process: air resistance, caused by the boxy shape of older trucks, and the additional braking force provided by large, powerful engines. Trucks are now more slippery to the air, decreasing drag, and the engine size and horsepower have been sized down, reducing engine-braking effectiveness.

Thus, issues of maintenance not only become critical to brake system integrity but also assume a primary role in reducing operating costs. Reducing these expenses is especially critical to owners and operators of large truck fleets, for whom savings in maintenance costs can be leveraged many times over. The most crucial aspect of maintenance is brake balancing. A truck with an unbalanced braking system performs poorly in emergency situations. In addition, when braking systems don't do their share of the work in bringing a 40-ton vehicle to a halt, the components bearing the brunt of this effort wear out sooner. Sometimes, failed components create a chain reaction affecting many other components. Not only will these components fail ahead of schedule, but they can mask the true cause of the failure.

The principal task of brake balance is to assure coordinated and controlled braking performance between the tractor and the trailer. An unbalanced vehicle costs far more to operate and maintain because its components wear prematurely. Unsafe operation is also likely. Jackknifing and tire blowouts are frequent results of poorly balanced truck brake systems.

For example, a jackknife situation can occur if the trailer's brakes don't work as hard as the tractor's brakes, so the trailer actually moves faster than the tractor when the brakes are applied. The driver's first indication of brake imbalance might be the disconcerting sight of the trailer trying to pass him backwards!

Brake balancing is the art of adjusting the many components of the average truck air brake system, so work is evenly divided, and components operate in a synchronous manner. A partial list of the elements that are considered during brake balancing is air compressors, brake size, brake pad friction coefficients, the length of the air lines, the number and type of hose fittings, and relay valve pressure activation values (the crack pressures).

A truck is composed of a tractor (where the driver sits) and a trailer. The tractor has three axles: a steer axle and two drive axles. The trailer has two axles, known as the front and rear trailer axles. By "truck," I

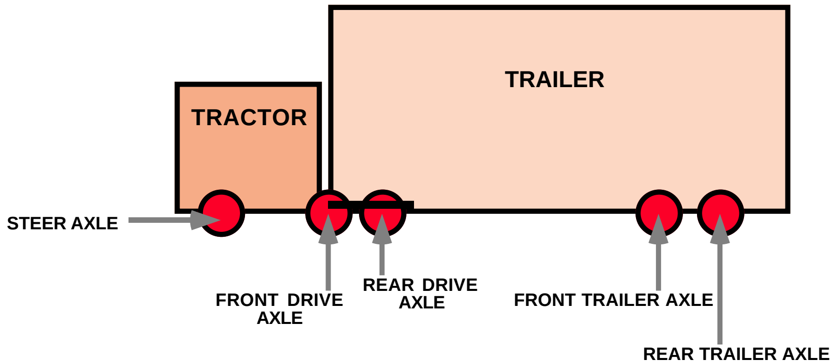

Figure 1. Schematic of a Typical Class 8 Vehicle.
Brake-balance diagnosis revolves around the components identified in the figure.

refer to class 8 vehicles, most commonly known as *18 wheelers*. Figure 1 shows a schematic of a typical class 8 vehicle.

The issue of brake balance comes down to ensuring that the tractor and trailer brake in a safe, coordinated manner. This action can be difficult to achieve because every independent driver owns his(her) own tractor but pulls somebody else's trailer, usually from a truck fleet. It's likely that the maintenance procedures, the frequency and quality of the maintenance performed, and the after-market parts will vary considerably between the driver's tractor and the fleet's trailer. This situation virtually assures brake system imbalance.

During the brake-balance analysis phase, a proprietary brake-balance analyzer is attached to various tractor-trailer combinations (trucks). Data are collected according to five different test procedures that simultaneously measure relevant parameters concerning the brake and air systems. The duration of a test varies from 2 seconds to 30 minutes depending on the test and environmental factors. The data are displayed and plotted as a graph, with axes representing time, pressure, or temperature. For most graphs, six lines are plotted in six colors, one for each axle and one for the pressure control line. The five tests are the snub pressure (shown in figure 2), full apply, full release, static pressure, and temperature.

The expert who performs the tests and offers consistently thorough and correct analyses of the data must be highly trained and experienced. Unfortunately, the expertise is difficult to teach. Thus, AI technologies were brought to bear to accomplish two objectives: (1) to automate the process as much as possible, so less experienced individuals can perform the tests, and (2) to render timely diagnoses of the col-

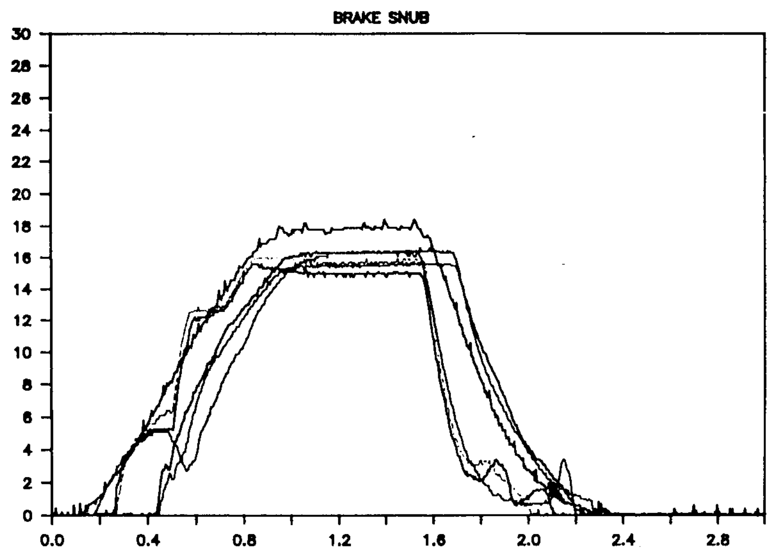

Figure 2. A Typical Graph for Analysis by Eaton's Brake-Balance System.

lected data in the field. In effect, the goal of the project was to remove the expert from the task of performing brake-balance analysis of truck fleets at customer sites by cloning the expert and spreading around his(her) expertise.

Use of AI Technology

Two AI technologies are used by the brake-balance analyzer system: knowledge-based systems and artificial neural systems (Anderson and Rosenfeld 1988; Arbib 1989; McCulloch 1988).

A knowledge-based system was written using Lisp and IntelliCorp's KEE software product. It supports two knowledge bases. One knowledge base performs a preprocessing function, and the other is a diagnostic system. The knowledge-based system also consists of a model of a truck, reasoned over by a number of rule classes and metarules (rules about rules) in an object-oriented environment using multiple inheritance. The knowledge bases and truck model are partially controlled by, and communicate through, a blackboard architecture (Barr, Cohen, and Feigenbaum 1981–1982; Erman and Lesser 1975; Erman et al. 1980; Hayes-Roth 1985; Arbib 1989). Where possible, methods and demons replace rules to speed execution times. (*Methods* are compiled, inheritable, and, possibly, rulelike procedures that are stored in frame slots. *Demons* are autonomous, self-actuating methods.) Figure 3

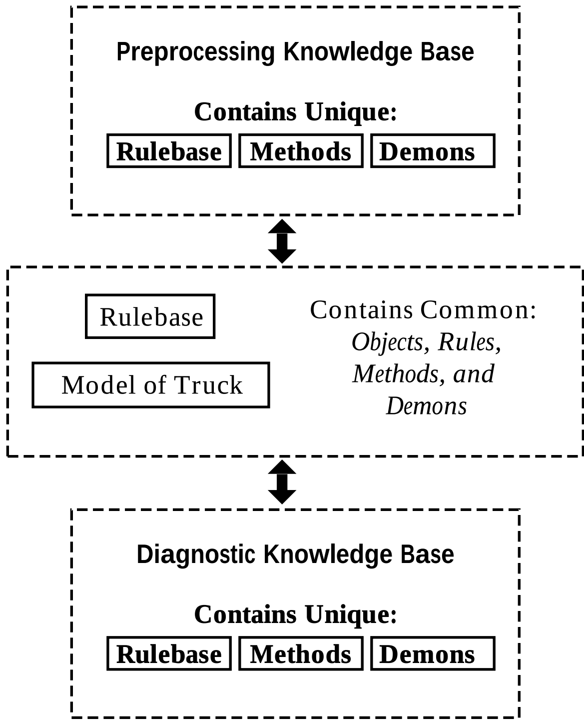

Figure 3. A Schematic Diagram of the Basic Structure of the Knowledge-Based System Portion of the Brake-Balance Analyzer.

shows the basic structure of the knowledge-based system.

About 40 rules and 120 methods are used. The rules are reasoned over using both forward- and backward-chaining processes. Inheritance is bi-directional wherever appropriate; that is, values are also inheritable from instance (child) to subclass (parent) to class object. Figure 4 illustrates this concept for the simple case where a slot representing crack pressure is inherited by subclasses and their respective instances (child objects). Subsequent measurements by the brake-balance analyzer provide each instance with its unique pressure value. To reason about, say, the trailer's crack-pressure value, we average the values of its instances (children) and assign this value to the trailer's crack-pressure slot. (We must, of course, inhibit the inheritance of this value back to its children, or we defeat the purpose of bi-directional inheritance and create dangerous looping behavior).

In this example, slot values for the front- and rear-axle instances of

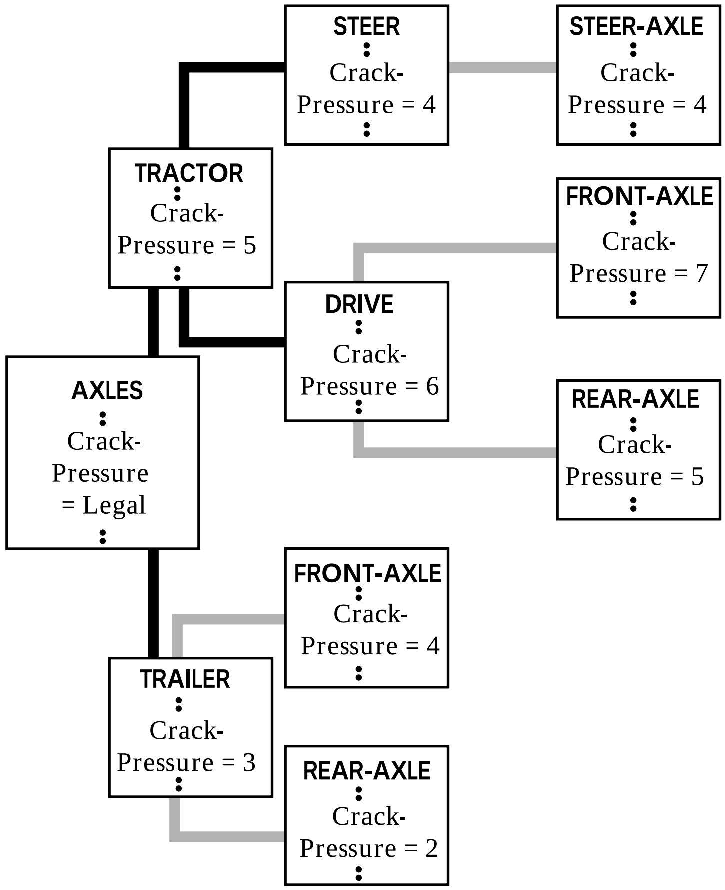

Figure 4. A Sample of the Inheritance Schemes Used in the Brake-Balance System Is Illustrated by the Class Axles.

the subclass DRIVE are averaged to give DRIVE a crack-pressure value of 6. More specifically, the slot CRACK-PRESSURE of the subclass DRIVE of the subclass TRACTOR of the class AXLES is given a value of 6. This process continues until eventually the class AXLES is itself given a value. In this case, this value is no longer numeric because a Boolean value such as LEGAL or ILLEGAL best suits our purposes. An advantage of this bi-directional inheritance scheme is that the variable type of the object's slot value can dynamically be changed.

The task of rendering detailed diagnoses based on an analysis of the graphs is difficult. Field-service personnel were unable to learn the task within a reasonable time frame. I believe knowledge-based system technology cannot efficiently make judgments about the graphic data. Therefore, an effective system for doing so using neural networks was developed.

The neural networks were built with the assistance of scientists at HNC. The development platform consisted of an HNC Anza Plus board in a PC-AT installed at Eaton Corporation and a similarly equipped Sun workstation at HNC. Time was too short to allow for much experimentation, so a simple back-propagation scheme with feed forward was used to meet the delivery date (Rummelhart, Hinton, and Williams 1986a, 1986b). The number of input nodes, hidden layers and hidden units varies between networks. It is not surprising that graphs turned out to be difficult to learn. What is surprising is that particular classes of graphs, thought to be easy ones, were, in fact, the hardest to learn.

Special brake-balance analysis equipment and software are used to acquire most of the raw data. This equipment and the neural net software run on a PC-AT class machine. The knowledge-based system requires an 80386-based machine. New avenues for deployment have opened and are discussed in Results and Future Work.

The Preprocessing Knowledge Base

The first knowledge base performs a preliminary checking function. While installing the sensors and microcontrollers used by the brake-balance computer, the brake-balance technician scrapes mud and rust from the components to identify their manufacturers and record sizes, adjustments, and wear and tear. This information is entered into the preprocessing knowledge base by the technician. The knowledge base examines the data provided by the human operator from this visual inspection of the vehicle systems along with the results produced by the brake-balance analyzer. The preprocessor ensures that each component of the truck (the tractor and trailer) is itself balanced. Testing will not proceed until incompatibilities are resolved because an attempt to balance the performance of the tractor and trailer is wasted if either of them is unbalanced.

Neural Networks

In the second phase of testing, the proprietary brake-balance analyzer is used to test various tractor-trailer combinations. It simultaneously measures characteristics of various brake components for each axle.

A suite of five neural networks is used to render expert analysis of the graphs. Each network classifies one particular type of graph as good or bad. The verdict reached by each neural net is then fed back

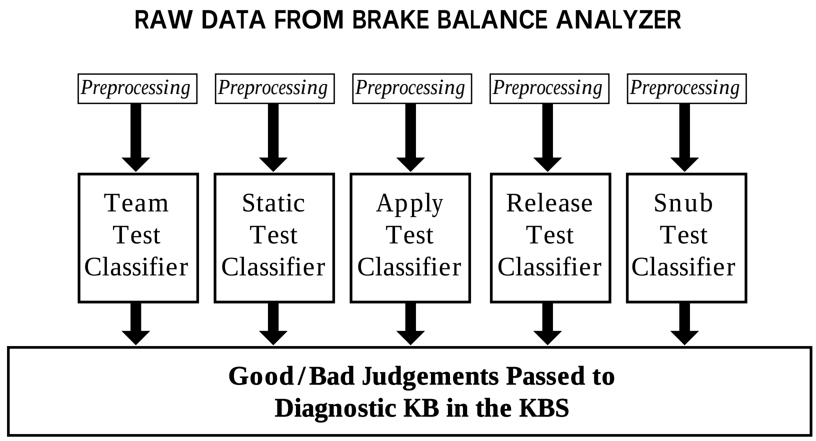

Figure 5. The Role of the Neural Network Classifiers in Eaton's Brake-Balance System.

to the expert system. Figure 5 shows the role of neural networks in the system.

I decided that a 90-percent success rate for each type of graph in the test suite would yield excellent results because of inherent redundancy in factors measured by the tests and subsequent cross-checking by the knowledge-based system. To meet our 15 December 1989 deadline for successful deployment, little time was spent analyzing the data to decide on an optimal neural network paradigm. It was decided that a simple back-propagation scheme with feed forward would be used. If time and results allowed, an alternate algorithm would be tried.

Many of the interesting problems we encountered resulted from the fact that although we had a five-year history of the expert's reports, involving dozens of truck fleets and thousands of graphs, to train and test the networks, we still had relatively small data sets for training and testing neural networks using the back-propagation scheme. This information on data set sizes and training is illustrated in figure 6.

An approach to overcoming small sample size was used in the cases of the apply and release tests. Only 25 and 91 examples of bad graphs are available, respectively, but a relatively huge number of good examples exist. This technique, known as *oversampling*, involves making the training set larger by duplicating its members. One keeps adding examples to the training set until training produces compatible accuracy on both the good and bad graphs. Oversampling did the trick under circumstances where it seemed impossible to train the apply and release classifiers. It was fortunate that our samples were typical exam-

Number of Test and Training Examples Available

Test	Good	Bad	Total
Temperature	42	50	92
Static Pressure	144	131	275
Apply	338	25	363
Release	293	91	384
Snub	92	105	197

Figure 6. The Number of Examples Used to Train and Test Each of the Five Neural Network Classifiers Was Relatively Small.

Classification Results (%)

Test	Linear	Heuristic	Best Network
Temperature	64	85	91
Static Pressure	72	73	91
Apply	93 (0% on BADS)	90 (28% on BADS)	88
Release	76 (0% on BADS)	82	89
Snub	47 (0% on BADs)	76	92

Figure 7. The Performance of the Neural Networks Compares Favorably with Two More Traditional Approaches.

ples, or training would have emphasized idiosyncrasies in the data.

To gauge the performance of network classifiers relative to conventional techniques, two sets of baseline classifiers were developed. The first baseline consisted of a set of linear discriminant classifiers that were provided with the same input as the network classifier. In general, these classifiers performed badly, as indicated in figure 7. Note in particular that they had a hard time in correctly classifying bad graphs. Unfortunately, these graphs are the ones we can least afford to misclassify.

The second set of baseline classifiers was derived from heuristics used by Eaton in evaluating test data. Essentially, the expert's rules of thumb were represented by statistical manipulations of the raw data. The overall performance is better than that of the linear baseline, as one would expect if heuristics do indeed have value.

The best performance came from the back-propagation networks (BPNs). BPN is a multilayer mapping network that minimizes the mean square mapping error between desired output and actual system output. BPN is typically a three-layer feed-forward neural network. The three lay-

Neural Network Configuration Data

Test	Size of Set Training	Test	Number of Nodes In Hidden Layers 1/2	Iterations To Convergence
Temperature	58	34	3-5 / No layer 2	1,000
Static	189	79	11/11	10,000
Apply	218	145	10 / No layer 2	10,000
Release	238	157	3-5 / No layer 2	1,000
Snub	129	69	11/11	50,000

Figure 8. The Critical Characteristics of the Five Neural Network Classifiers. Note that the classifiers can have one or two hidden layers.

ers (slabs) are referred to as the input layer, the hidden layer, and the output layer, each consisting of a number of processing elements.

Figure 8 gives the relevant characteristics of the networks. The results achieved with these classifiers are attributed to experimentation. Of course, not all parameters can freely be adjusted. The training set and test examples must randomly be chosen from the population of examples to ensure that each is typical of its kind. The total number of good and bad examples in the test set must exist in fair proportion to one another too.

Note that two of the networks have two hidden layers. I want to emphasize that multiple hidden layers are artifacts borne from the need to quickly get results. We were able to monitor each network's training performance in a convenient graphic format that worked in real time—an indispensable tool provided by HNC. If results weren't forthcoming before our patience wore thin, we found that adding another hidden layer usually produced quicker results. That is, the real time required for training to converge was reduced to acceptable levels. Therefore, the stubborn trainers are readily identified from figure 8. They use two hidden layers, have more nodes for each hidden layer, and require more iterations to converge.

The input propagate through the hidden layer, which forms nonlinear combinations of the input values. The output of the hidden layer are subsequently combined by the single processing element to produce an output value that is compared against a threshold to provide a classification into one of two categories. These categories are identified in our expert's technical vernacular as good and bad.

In general, the output layer can contain an arbitrary number of processing elements, as determined by the dimensionality of the target space for the network. The weights of the network cannot directly be calculated; rather, the weights are adjusted by training the network to perform the desired classification of the input patterns. In this procedure, the network output and the desired output are computed. The *output error*, that is, the difference between desired and output values, is then propagated backwards through the network, and the weights are updated according to the generalized delta rule.

The choice of an input representation is a compromise between giving the network enough information to work with and finding a compact representation of the data. Our approach is to give the network input that is as close as possible to the raw data but to pare down the amount of data it sees by considering symmetries, regularities, and tendencies. With this approach, if one of the baseline statistics suggested by Eaton heuristics is useful, then the network is able to derive it; however, the network has enough additional information to model regularities in the data that are not predicted by the baseline statistics.

Each set of data (graph examples) was split into two sets, one used for training and the other used to test the performance of the trained network on novel examples. Several experiments were performed with networks of various topologies. The best results were consistently obtained with networks of 1 or 2 hidden layers and 3 to 11 hidden nodes and with connections from the input neurons to the output neurons enabled. After training for a few thousand iterations, the networks reliably learned to classify the training set with 100-percent accuracy, indicating that the network memorized the training examples. Figure 8 summarizes this information.

Diagnostic Knowledge Base

The second knowledge base accepts data from the networks and the preprocessing knowledge base and the truck model to arrive at a pinpoint analysis of the brake system. It produces one or more recommendations for putting it right again.

The knowledge-based system is 100-percent successful at this task, provided it is given valid classifications of the graphs. It tries to explain its reasoning as it renders its decisions. An example of an explanation is, “The mismatched coefficients of the brake pads, combined with the oversized brakes and generally slow reaction time of the brake system, will tend to cause the trailer brakes to do more work than the tractor brakes, creating excessive operating temperatures. In addition to wearing the brake pads prematurely, the brakes will quickly go out of adjustment, exaggerating the problem of a tardy brake system.”

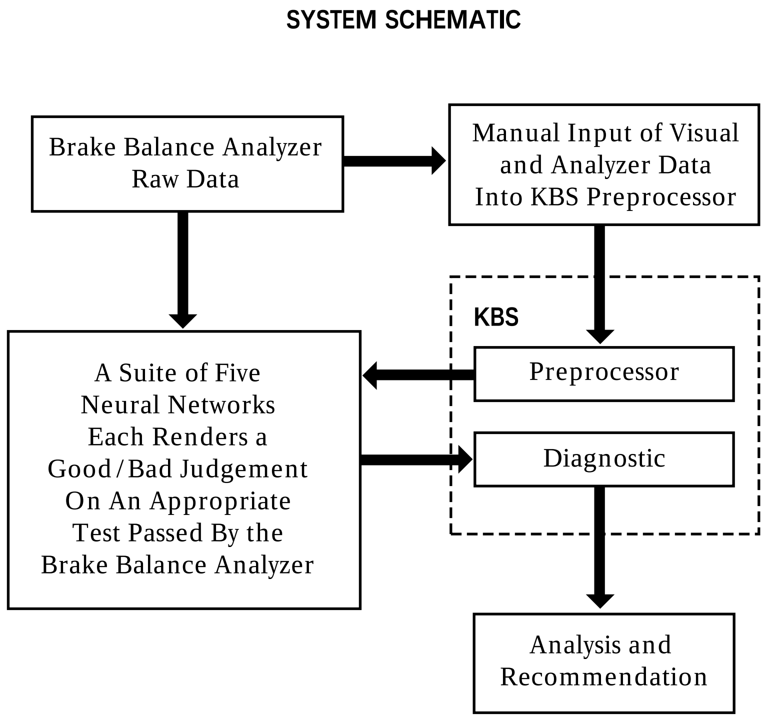

Figure 9. An Overview of the Brake-Balancing Process, Including Raw Data Acquisition, Knowledge-Based System Processing, and Classification by Neural Networks.

Innovative System Qualities

The brake-balance system is innovative in that it uses a rare approach to a real-world problem by featuring cooperative diagnostics and problem solving between a knowledge-based system and a suite of neural networks. Figure 9 illustrates the process, which is described in the following paragraph:

The cooperation proceeds according to these steps: First, observational data and data from the brake-balance analyzer are manually input to the knowledge-based system. Observational data include information that is stamped into steel underneath the truck, requiring a flashlight and the removal of sludge to read. Second, preprocessing is performed by the knowledge-based system. The knowledge-based system assures that the tractor and trailer are each balanced. Third, preprocessing is performed by the neural networks. The classifiers can detect some of the same inconsistencies as the knowledge base. This result suggests great potential for an expanded role for neural networks in the brake analyzer system. The classifiers also locate and identify interesting features and perform some smoothing to the raw data. Fourth, judgments are rendered by the neural network classifiers. Fifth, these judgments are passed to the knowledge-based system. Sixth, a *reality check* is performed by the knowledge-based system to discover and correct isolated errors by the neural networks. Seventh, diagnosis and a recommended solution are given by the knowledge-based system.

Two additional features are worthy of note: After the system was designed, much of the labor to construct the knowledge-based system was produced by juniors and seniors earning college credit from local universities while working at Eaton. They were relatively unfamiliar with programming, let alone Lisp, KEE, or AI. Still, they were useful in building the system. This ability speaks well for the maturity of the technology used in this application.

During the knowledge-based system development phase, several former brake-balance "experts" came by for demonstrations. Each used the fledgling system to try out a favorite "chestnut" (real-life experience). Realizing we were wasting knowledge-acquisition opportunities, these visits prompted the development of an automatic rule-generation feature. If the system had no recommendation, they could provide one, essentially teaching the system a new rule. These rules were flagged for approval by the expert before inclusion in the permanent knowledge base.

Criteria or Deployment

The deadline for system delivery was 15 December 1989 and was successfully met.

The knowledge-based system is successful only if an accurate diagnosis and recommendation is given nearly 100 percent of the time. Two assumptions are required, however. The data from the brake-balance analyzer must be accurate, and the classifications of the graphs must be within the accuracies given below.

Two levels of success were set for the neural net portion of the system: The easiest criterion for success required that all five networks operate at 80-percent accuracy. The system is a legitimate success at this level of performance because of natural relationships between the graphs. Thus, the knowledge-based system will identify and correct any single error by one of the neural networks. Therefore, a usable and correct solution will be reached when isolated errors occur in classifying the graphs. Further, occasional errors on specific truck rigs are of no consequence because we are recommending brake-balance procedures for entire truck fleets

The preferred level of success was to reach 90-percent accuracy on all five of the networks. Current success rates are 88 percent, 89 percent, 91 percent, 91 percent, and 92 percent. This result is an unqualified success. It is important to note that relatively small data sets for training and testing neural networks using the back-propagation scheme were available, even though these data sets constituted a five-year history of the expert's reports, involving many truck fleets and thousands of samples.

Nature and Estimate of Payoff

Expenses amounting to $100,000 that were incurred by the expert are saved annually. Ninety round trip air fares, 90 car rentals, about 225 hotel room rentals, and 675 meals are eliminated. These numbers do not include the savings of the expert's labor costs.

Although concrete dollar figures are difficult to come by, the largest payoff directly relates to attracting additional business for Eaton Corporation. Brake balance is offered as a free service to our customers who operate large truck fleets. It is assumed that our customers react positively to this practice and, in return, do more business with us. The system allows the field support staff to conduct tests and generate solutions immediately. Brake balancing conducted by local engineers yields the following benefits: First, six times the former number of fleet appraisals can be conducted each year, effectively spreading six times the good will. Second, brake-balance test results no longer need desk analysis. Extensive analysis can delay the brake-balance report to the customer by one to three months. Speedy analysis increases good will. Third, the expert is freed from a burden. It is likely the expert will be promoted to pursue other more challenging and rewarding work, which is a professional perk. It's certainly preferable that the expert not be put out of a job after providing his(her) expertise. Such an expert is usually more cooperative than one who is losing his(her) job or is not being promoted. Fourth, as with many diagnostic systems, it has shown value as a training tool for inexperienced maintenance people. Fifth, AI technologies are put into the hands of people who sell and maintain trucks. They are used on a daily basis to perform tasks that are basic, easily understood, and not secret. The brake-balance application is down to earth. The average person can understand it and feel good about it.

Outline of System Use

The development time and costs were moderate, and deployment went smoothly. Further deployment, however, requires working out some details.

Development Time and Costs

The knowledge-based system required two calendar months for knowledge acquisition (including truck brake school), which is four person-weeks of labor. The design, development, and testing of the system took an additional six weeks. Total development cost was $30,000. The work was conducted during the period from October 1988 to February 1989. The neural nets cost approximately $50,000 to develop; this work occurred from August 1989 to December 1989.

A development copy of KEE existed in the knowledge-based system laboratory. Each delivery copy put to the field for personal computer (PC) deployment will cost about $2500. Thus, it would cost $17,500 for seven systems.

The neural net Anza Plus development board and computer cost $27,000. Field deployment costs are considerably less. The compiled software simulation of the networks is less than $1000 for each deployed PC. The total cost for development and deployment is, thus, $105,500, when software discounts are applied.

Deployment

Although it was completed two calendar months earlier, holidays and schedule conflicts postponed the deployment of the system until 13 February 1990. The system was delivered to the expert, and the deployment consisted of a brief training episode followed by two field tests. The field tests entailed recommendations for two truck fleets; judgments were rendered on 38 graphs.

Issues of Further Deployment

Before deploying the remaining systems, the following items need attention: First, the trucking industry experienced a downturn in the first quarter of 1990. Fleet testing was curtailed until the situation reverses. Thus, additional deployed systems will not be useful until later in 1990. Second, only one-third of the field stations have adapted to using the proprietary brake-balance analyzer equipment, delaying deployment opportunities. Third, the cost of delivery for KEE systems is too high for our customer, in part because 80386-based PCs must be purchased. Fourth, the customer would like more features to be added before initiating full-scale deployment.

Results and Future Work

The knowledge-based system performed flawlessly both on five years of field data and in two recent field tests. The neural network classifiers were 90-percent effective on the archived data and perfect on the field tests. These results are encouraging, especially because the choice of networks is not yet optimized for the task.

However, the classifiers shouldn't be doing this well. The expert suspects the field-test success of the classifiers is partially explained by the fact that the system encountered the most challenging problems during its development and that today's problems are simply less interesting. The brake systems of today's trucks are in better shape than those of a couple years ago, and some of the really awful tests that occurred in the training and test data won't be seen again. To paraphrase, the networks were trained and tested against data that are less well behaved than anything they will see in the field. (This is a great trick if you can do it and opens the question of deliberately training networks on unreasonably difficult examples to ensure greater accuracy in deployed systems that will likely only see better--behaved data.)

The success of the neural net classifiers implies that today's trucks are showing better brake-balance characteristics. In fact, customers now pay more attention to brake systems when specifying trucks for their fleets. However, antilock brake systems will soon be federally mandated, and international diagnostic standards are being formulated. It is important to preserve today's knowledge and integrate computers into the current environment, thus preparing us to address the needs of the trucking industry for the decade ahead.

I have an alternate explanation for the success of these classifiers: The technicians running the tests are required to perform certain skilled manipulations of the brake pedal. I can attest to the fact that a learning curve exists. It is possible that better-behaved data are produced because the operators have gained experience performing the tests with the brake-balance analyzer. This hypothesis will be easy to verify once rookies begin using the system.

I would like to experiment with other network algorithms to improve the accuracy of the networks. It is possible to expand the diagnostic capability beyond merely good and bad. The diagnosis could include specific information about what features in a graph make it bad or good and what symptoms one expects to see on the vehicle as a result.

Integrating the proprietary brake system, the knowledge-based system, and the neural networks into one platform will ease field deployment and make it cost effective. With regard to the high cost of the KEE system, it is worth noting that IntelliCorp has made two products based on the C programming language available that are less expensive than KEE and have fewer hardware requirements.

I would like to incorporate a truck maintenance expert system that is driven by voice input and synthesizes speech output. This system would keep the operator's hands free for mechanical tasks and eliminate note-taking activity beneath the truck.

Conclusion

The familiar problem of trying to capture expertise and represent it in another form does not go away with neural networks. Choosing network topologies and algorithms and resolving training issues require the same understanding of the domain that a knowledge-based system does. For hybrid applications, pairing the knowledge engineer from the knowledge-based system activity with a neural net specialist produces better results than pairing a neural net specialist with a domain expert. This fact is an indication that neural net technology is not yet ready to be placed in the hands of a lay person. However, artificial neural system and knowledge-based system technologies are complementary in nature and are mature enough to be deployed together in applications that are less than exotic. Further, knowledge-based system tools do not always require highly trained users to be productive and effective. Finally, back propagation can sometimes be successful in cases with relatively small sample sizes, especially when the technique of oversampling can be applied.

Acknowledgments

I would like to thank Eaton Corporation for funding this project and allowing the results to be presented at IAAI-90 and IJCNN-90 and published here. I would like to thank Mark Plutowski, Todd Gutschow, and Dave Shlager of HNC for their support and contributions. Most of all, I would like to thank Eaton Road Ranger George Prokos for being such a fine and articulate expert, without whom this project could never have been conceived.

References

Anderson, J. A., and Rosenfeld, E., eds. 1988. *Neurocomputing: Foundations of Research.* Cambridge, Mass.: MIT Press.

Arbib, M. A. 1989. *The Metaphorical Brain 2.* New York: Wiley.

Barr, A.; Cohen, P.; and Feigenbaum, E., eds. 1981–1982. *The Handbook* *of Artificial Intelligence*, volumes 13, chapters 5, 7, 8, 9, 12, and 15. Los Altos, Calif.: William Kaufmann.

Erman, L., and Lesser, V. R. 1975. A Multi-Level Organization for Problem-Solving Using Many Diverse, Cooperating Sources of Knowledge. In Proceedings of the Fourth International Joint Conference on Artificial Intelligence, 483–490. Menlo Park, Calif.: American Association for Artificial Intelligence.

Erman, L. D.; Hayes-Roth, F.; Lesser, V. R.; and Reddy, D. R. 1980. The HEARSAY-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty. *Computing Surveys* 12:213–253.

Hayes-Roth, B. 1985. A Blackboard Architecture for Control. *Artificial Intelligence* 26:251–321.

McCulloch, W. S. 1988. *Embodiments of Mind.* Cambridge, Mass.: MIT Press.

Rummelhart, D. E.; Hinton, G. E.; and Williams, R. J. 1986a. Learning Internal Representations by Error Propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*, eds. D. E. Rummelhart and J. L. McClelland, 318–362. Cambridge, Mass.: MIT Press.

Rummelhart, D. E.; Hinton, G. E.; and Williams, R. J. 1986b. Learning Representations by Back-Propagating Errors. *Nature* 323:533–536.