NEURAL NETWORK CLASSIFIERS TO GRADE PARTS BASED ON SURFACE DEFECTS WITH SPATIAL DEPENDENCIES

Daniel L. Schmoldt
USDA Forest Service
Brooks Forest Products Center
Virginia Tech University
Blacksburg VA 24061-0503

INTRODUCTION

In many manufacturing operations, unfinished or unassembled parts are not necessarily accepted or rejected in their entirety. Oftentimes, they are given some qualifying grade that indicates their worth to potential buyers or their suitability for particular processing operations. The manufacture of hardwood lumber in sawmills follows this general procedure. Lumber is graded based on appearance because its end use is primarily furniture and other goods that have a large aesthetic aspect to their value. Lumber grades are used by sawmill operators (sellers) and furniture plants (buyers) as an estimate of how much clear wood material is in the lumber being traded, hence the lumber's value. Also, certain sawmill operations treat a board differently based on actual or potential lumber grade. Therefore, accurate lumber grading is important to mill operators.

Very detailed and specific rules have been developed to grade hardwood lumber. They set minimum requirements that must be met for each board grade. Despite the complexity of these grading rules, a computer implementation of the formal grading rules has been created [1, 2]. Because this computer grader is a fairly complete representation of the grading rules, it is exhaustive in its grading procedures, even where some of the calculations are computationally expensive. Therefore, it is not guaranteed to complete in a fixed amount of time. This is not acceptable for some real-time processing operations.

To avoid this limitation, a pattern classifier might be used to perform nearly constant-time grading, while allowing for some loss in grading accuracy. Artificial neural networks (ANNs) were selected as the classifier architecture because: (1) they can simulate any differentiable mathematical mapping, (2) many network types (topologies) are available, and (3) once developed, the final ANN mapping can be coded as a subroutine that executes in a small, fixed amount of computational time.

This report details initial efforts to develop a real-time lumber grader. Classifier development is introduced conceptually, spatial features are described, and the use of training and testing data sets is discussed. A utility function for comparing different classifiers is derived and demonstrated. Comparison tests using three different ANN topologies indicate where improvements can be made in classifier design and in training.

APPLICATION OVERVIEW

At a sawmill, boards are sawn from a hardwood log in a fashion similar to the illustration in Figure 1. Flat faces are produced on each side by removing the curved portion of the log cylinder. The highest quality wood in the log lies immediately beneath these pieces. Several boards are removed from each side (only one per side is displayed in

Review of Progress in Quantitative Nondestructive Evaluation, Vol. 14
Edited by D.O. Thompson and D.E. Chimenti, Plenum Press, New York, 1995

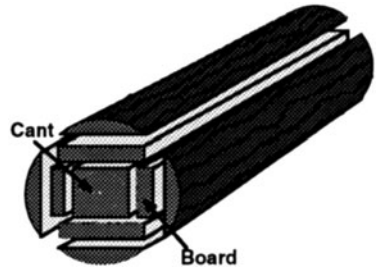


Figure 1. Boards are removed from the outer, high-quality regions of a log.

Fig. 1 for graphical simplicity). The remaining center cant of the log contains the lowest quality wood of the log, and may be resawn to salvage additional boards or may be sold for the manufacture of pallet material.

Logs enter the sawmill, are debarked, and pass through a band headrig several times to remove boards from around the outside of the log. Boards then proceed to edger and trimmer operations, after which they are graded and sorted. Because many processing steps in this type of mill are controlled by human operators, sub-optimal processing decisions, productivity losses, and raw material waste result. However, many mill inefficiencies can be reduced by introducing scanning systems, and the software to make processing decisions based on scanner information. Installation of scanning systems, however, requires that human operator tasks, e.g., lumber grading, are also automated.

Lumber Grading

There are 4 primary grades of hardwood lumber: FAS, #1 Common, #2 Common, and #3 Common. Figure 2 contains some examples of these. Lumber that does not minimally meet 3C spec's is below grade (or cull). Higher quality lumber: (1) has maximum limits for the size and numbers of unsound defects, (2) must be larger dimension material, and (3) must have a higher percentage of its surface area in clear wood cuttings of a larger size. Final lumber value is based on the surface measure of the board and its grade, where grade determines the value per square foot of surface measure.

Lumber grading impacts an automated mill in several ways. First, an optimal log breakdown pattern depends on the final value of each board produced. Second, edging and trimming operations attempt to create the highest value piece of lumber from each board. This operation may remove substantial amounts of wood to increase grade, and hence the board's total value. This grade then becomes the final grade of the piece of lumber for sale. Finally, lumber is graded prior to sale to secondary manufacturers.

CLASSIFIER DEVELOPMENT

Classifiers in General

Many different types of classifiers exist, but they are all similar conceptually. The basic goal is to automatically train a classifier to associate patterns with categories. A set of examples, typical of real-world categories, are used for training. These examples consist of feature patterns paired with categories. Once a classifier has been trained, it is tested on other data which it has not seen before. A useful classifier performs well on those new patterns. The ability to accurately classify novel patterns is the goal of the training process. Learning, in this context, is considered to be supervised, because desired results are paired with patterns in the training phase.

Training and Testing

Each example in a training set contains a vector of features that, in some sense, are intended to unambiguously describe that example as belonging to one of the important

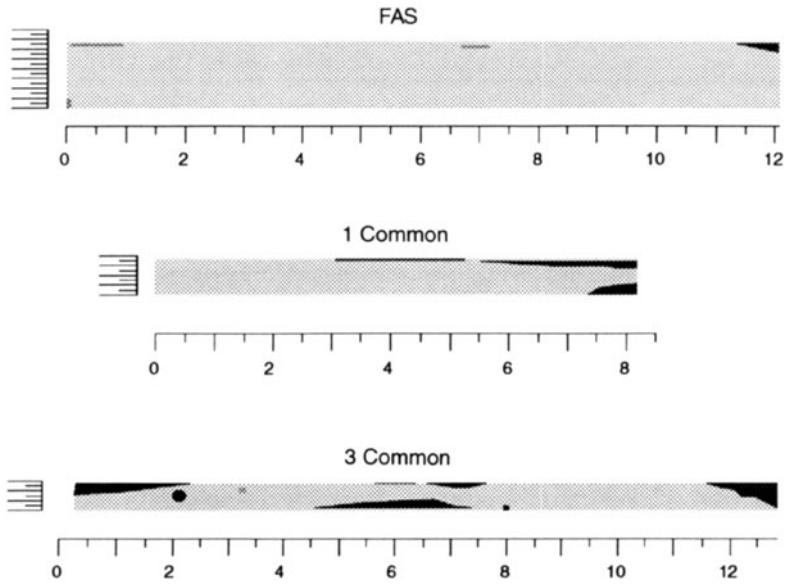


Figure 2. Different grades of lumber allow different numbers and sizes of defects.

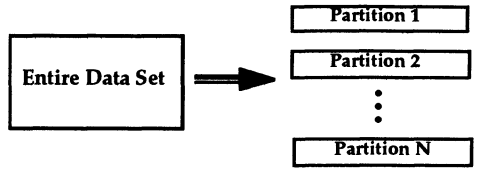
categories. Some features are very discriminating with respect to selecting a single category for the examples, while other features are not, they contain noise. It is relatively easy to create a classifier that works well on training data, i.e., one that has a low *apparent* error rate. In the extreme case we can imagine a table lookup procedure, which has an apparent error rate of zero. What we want, on the other hand, is a classifier that has a low *true* error rate [3]. That is, one that has the lowest error rate if presented with all possible real world cases. Because it is impractical to train a classifier on all possible samples, an estimate of the true error rate must be determined from a relatively small set of training data.

A good classifier: (1) extracts the maximum amount of information from the training set, (2) is not selected based on its ability to accurately distinguish between cases in the training set, and (3) is able to extrapolate to new cases. The former may be difficult to attain, but the latter two can be accomplished through appropriate classifier training and testing methods.

One statistically accepted procedure for obtaining good estimates of the true error rate during training and testing is cross validation. In this technique, the training data are partitioned into N exclusive sets. An N step process is then used, where at each step i, partition i is used for testing and the remaining partitions are used for training (Figure 3). The true error rate estimate is then the average of the test values for the N partitions. When N equals the number of cases in the full training set, then this type of cross validation is referred to as *leave-one-out*. Else, it is referred to as N-fold cross validation. The data set for our study consisted of approximately 1600 boards [4]. These were randomly partitioned into 10 subsets. Each subset contained the same frequency distribution of boards from the different classes.

CLASSIFIER FEATURES

Board outlines and defects had been manually mapped and values recorded [4]. Using these numerical representations of the boards, each was graded with the computer grader [2] and visually verified. A total of 19 different features were calculated for each board in the data set. These features fall into the four feature types presented below: traditional, moment-mean ratios, defect packing, and information gain.



$$\text{True Error Rate} = \sum_i \frac{Partition_i}{N}$$

Figure 3. Cross validation estimates the true error rate by testing on separate partitions of the data set. During each iteration, training occurs on the entire data set, less the test partition.

Traditional

Based on what is known about lumber grades and the distribution of defects on boards [5] several board features, known to be correlated with lumber grade, were selected. This set included: percent of defect area containing knots, total length of wane, average size of all defects, number of defects per area, and total number of defects. Each of these is very computationally inexpensive to calculate. Nevertheless, they provide only general descriptions of a board and its defects, without any indication of spatial arrangement.

Moment-Mean Ratios

For a Poisson spatial distribution, complete spatial randomness (CSR) means that the variance of the population and the mean of the population are equal. In the context of board surfaces, the "population" is the set of spatial intervals on the board. In fact, the 2nd, 3rd, and 4th moments about the mean have the expected values appearing in (1).

$$\begin{aligned} M_{2,\mu} &= \mathrm{E}[(X-\mu)^2] = \mu \\ M_{3,\mu} &= \mathrm{E}[(X-\mu)^3] = \mu \\ M_{4,\mu} &= \mathrm{E}[(X-\mu)^4] = \mu + 3\mu^2 \end{aligned} \tag{1}$$

Therefore, for randomly sampled quadrats on a board, the ratio of the variance to the mean, the ratio of the 3rd moment to the mean, and the ratio of the 4th moment to the mean should indicate differences across boards in spatial distributions of defects. Sample moments k about the mean are calculated as in (2), where X_i is the number of defected units in a quadrat. A unit is a 1/4" square for this application. The true mean μ is easily calculated as the product of the size of the sampling quadrat and the ratio of the total number defect units to the total number of units on the board. Some initial examination of sampling histograms suggested that 1000 sampling quadrats should be sufficient to obtain fairly consistent estimates of these features. Quadrats were randomly located on a board.

$$\hat{M}_{k,\mu} = \sum_i (X_i - \mu)^k / n - 1 \tag{2}$$

High moment-mean ratios can be due to either a few scattered defects or a large number of clustered defects. To distinguish between those cases, we can normalize these ratios by the number of defects on the board. These normalized ratios provide a more representative description of defect scatter and clustering.

Defect Packing

While the point estimators above provide some information about the distribution of defected areas on a board, they provide little information about the arrangement of those

defects. Consequently, I have created a board defect feature, D, that provides a measure of how closely defects are packed on the board and how near the board edges defects lie. The distance D in (3) is normalized by the board area A, and is composed of two parts: (1) the nearest neighbor distance NN_i, using defect centers (X_i, Y_i) in the calculation and (2) the distance to the closest edge NE_i. The values *xmax* and *ymax* are the X and Y coordinates of the board's right end and upper edge, respectively ((0, 0) lies at the lower left corner of the board). A smaller value for D (zero in the case of no defects) would indicate that there is more, relative clear wood area present, and hence the board should grade as higher quality.

$$
\begin{aligned}
&D = \sum_i (NN_i + NE_i) / A \\
&\text{where,} \\
&NN_i = \min_{j \neq i} \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} \\
&NE_i = \min(X_i, xmax - X_i, Y_i, ymax - Y_i)
\end{aligned}
\tag{3}
$$

Information Gain

Entropy is the loss of energy in a closed system. This loss of energy is reflected in a concomitant accumulation of mass, or clumping of cooled energy. Shannon used this idea as a basis for information theory [6], where information corresponds to lower entropy, or reduced randomness. Entropy H is defined in (4), where p_i is the observed proportion of quadrats with outcome i, i.e., with a certain number of defect units. By examining the relative amount of information gained (or entropy lost) $I(p:q)$ in going from CSR to a more clumped pattern of defect arrangement, we have another estimate of defect contagion [7].

$$H = \sum_i p_i \ln(1 / p_i) \tag{4}$$

Information gain $I(p:q)$ is calculated as the difference between the entropy of a CSR pattern and the observed pattern (5). The expected number of quadrats q_i with i points in a CSR pattern is calculated from the Poisson p.d.f. (6). The Poisson parameter λ is the ratio of the total number of units on the board to the number of defected units on the board. The value of $I(p:q)$, the weighted average expected information gain, should be lowest for CSR patterns and should increase for clumped and regular patterns.

$$
\begin{aligned}
I(p:q) &= \sum_i p_i [\ln(1 / q_i) - \ln(1 / p_i)] \\
&= \sum_i p_i \ln(p_i / q_i)
\end{aligned}
\tag{5}
$$

$$q_i = (e^{-\lambda} \lambda^i) / i! \tag{6}$$

NEURAL NET CLASSIFIERS

Some initial tests were conducted with a variety of ANN topologies. Three of those, back-propagation, learning vector quantization, and radial basis function, were selected for further study. Each of the ANNs contained 19 input nodes, 1 hidden layer, and 5 output nodes, corresponding to the five classes of lumber grade. For each ANN classifier, classification rate was used as the termination criterion that determined what network weights were saved as the "best."

Back-Propagation

A relatively typical back-propagation network was used. Each of the input nodes was fully connected to the single hidden layer. After some experimentation with the number of nodes in the hidden layer, the number of nodes was fixed at 15.

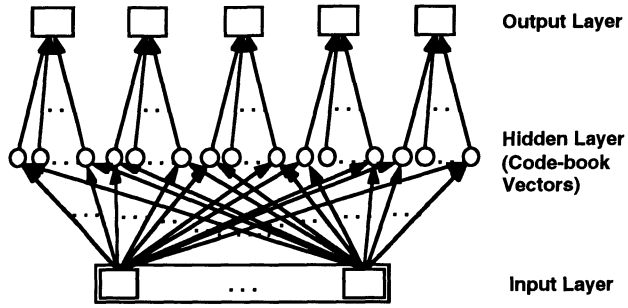


Figure 4. In an LVQ network, prototype vectors belonging to the different classes are trained. For recall, the closest prototype to the test case determines the class assignment.

Learning Vector Quantization

Learning vector quantization (LVQ) maps feature vectors to output classes by the use of code-book (or prototype) vectors (Figure 4). Each output class has the same number of code-book vectors. After some experimentation, 20 code-book vectors per class were used in this application. These vectors are trained to become representative of cases within a particular class. As training cases are presented, the closest vector (using Euclidean distance) is moved toward the training case, if it is in the correct class. It is moved away if it is not. The basic LVQ algorithm has some limitations, but several variants of the original algorithm overcome these problems. Several of these modifications were used here. In test mode, the class with the winning code-book vector is output.

Radial Basis Function

Radial basis function (RBF) networks are actually a class of network topologies. In an RBF (Figure 5), the hidden layer consists of pattern units. Each pattern unit contains a center vector (represented by the weights from the input layer), a distance measure, and a univariate transfer function that maps the distance value to an output value. Such pattern units are radially symmetric. After some initial experimentation, 200 pattern units were used in this network.

A Kohonen-like K-means clustering algorithm is used to adjust the center vector values. Pattern unit transfer functions are adjusted by using nearest neighbor clustering. After this self-organizing phase is complete, the output layer is trained using back-propagation learning.

Classifier Utility

Classification rate can provide a good indication of how well a classifier is performing. However, because error rates for different classes may vary and the importance of correct classification may vary between classes, strict classification rates may not give a true measure of the real utility of a classifier. This is especially important when comparing the performance of several classifiers. The classifier utility metric *CU* described in (7) attempts to deal with these limitations. This metric incorporates: classification rates for each lumber grade C_{ij}, classification utility values for each grade U_{ij}, and frequency of occurrence of each grade p_i. This metric allows us to compare classifier results in a more economically meaningful way. (The confusion matrix that contains the classification utility values U_{ij} is not included or discussed here for space reasons.)

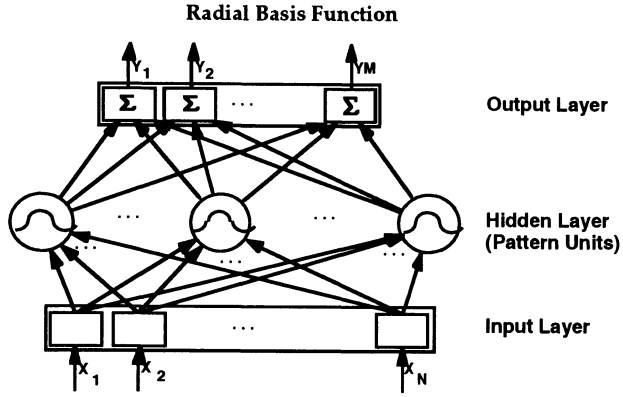


Figure 5. In an RBF network, training occurs in two parts. Pattern units are trained in a self-organizing phase and then the output layer is trained to minimize error.

$$CU = \sum_j \sum_i p_j U_{ij} C_{ij} \tag{7}$$

RESULTS AND CONCLUSIONS

The 10-fold cross validation method uses partitions to conduct 10 training sessions and 10 classifier tests. At this writing only 5 train/test cycles have been completed. However, given the random assignment of cases to the partitions, these 5 cycles probably reflect the final results.

Table 1 contains averages over 5 test partitions using the input features described above. Weighted classification rates and classifier utility values appear to the right of each sub-table. Desired (or true) grades appear along the top of the tables and ANN classifications appear on the side. Values closer to unity along the diagonal indicate better performance. All ANNs performed best on FAS lumber and worst on BG lumber. Overall classification rate and classifier utility is highest for back-propagation (BP). LVQ has the second highest classification rate, but RBF has the second highest utility value. This difference is due to better performance by RBF on 1C and 2C lumber, which has a relatively high utility value. This demonstrates that the classifier utility function can make a difference in how performance is evaluated.

The actual lumber grading rules are algorithmic and have many discontinuities (branch points). Therefore, they do not translate into a smooth mathematical mapping, which is readily learned by an ANN. Consequently, classification rates at this point are too low to be practically useful. However, many of these branch points require relatively simple and easy calculations from the original board descriptor files that result in binary values–either the board satisfies a constraint for a particular grade or it does not. So, it should be possible to incorporate these binary values into the feature vectors used for classification.

As the number of classes that need to be discriminated decreases, classifier accuracy increases. In many situations, it may not be necessary to distinguish below grade boards, or even 3C, because they are not encountered very often. By eliminating those classes from the training and testing sets, a higher classification rate can be obtained.

Currently, for each ANN topology and each test partition, the best classifier is selected based on overall classification rate. Given the desire to evaluate classifiers based on the overall utility of classifier performance, a better objective function for selecting the best set of network weights would be classifier utility.

Table 1. Averaged classification results for the 3 ANN classifiers are based on completion of 1/2 of the 10-fold cross validation tests.

Avg. LVQ

Overall Avg. = 0.58404

Utility = 1.20

		Desired				
		FAS	1C	2C	3C	BG
Actual	BG	0	0.03274	0.0819	0.2222	0.46668
	3C	0	0.02618	0.08538	0.53334	0.26666
	2C	0.03522	0.24178	0.56566	0.15554	0.15554
	1C	0.16216	0.55208	0.2492	0.08888	0.1111
	FAS	0.80244	0.14722	0.01788	0	0

Avg. BP

Overall Avg. = 0.61476

Utility = 1.28

		Desired				
		FAS	1C	2C	3C	BG
Actual	BG	0	0.00984	0.0176	0.06666	0.33334
	3C	0	0.00984	0.03214	0.51112	0.24444
	2C	0.00714	0.23172	0.69008	0.33332	0.33332
	1C	0.13016	0.67662	0.25658	0.08888	0.08888
	FAS	0.8627	0.072	0.00358	0	0

Avg. RBF

Overall Avg. = 0.55126

Utility = 1.23

		Desired				
		FAS	1C	2C	3C	BG
Actual	BG	0	0	0.00358	0.04444	0.19998
	3C	0	0.01312	0.0391	0.51112	0.24442
	2C	0.01482	0.25188	0.65518	0.39998	0.44444
	1C	0.20264	0.591575	0.28796	0.04444	0.06666
	FAS	0.78254	0.12734	0.0143	0	0.04444

The initial results are encouraging. With the above mentioned improvements and by expanding the types of classifiers considered, e.g., discriminant classifiers, Bayesian decision rule, and decision trees, it seems possible to reach an acceptable level of performance and also have a computational procedure that operates in real time.

REFERENCES

1. P. Klinkhachorn, J. P. Franklin, C. W. McMillin, R. W. Conners, and H. A. Huber, Forest Products Journal 38, (1988).

2. P. Klinkhachorn, R. Kothari, D. Yost, and P. Araman, Forest Products Journal 42, (1992).

3. S. M. Weiss and C. A. Kulikowski, *Computer Systems That Learn.* (Morgan Kaufmann Publishers, Inc., San Mateo, 1991).

4. C. J. Gatchell, J. K. Wiedenbeck, and E. S. Walker, "1992 data bank for red oak lumber", Res. Pap. NE-669, (Radnor PA, U.S. Department of Agriculture, Forest Service, Northeastern Forest Experiment Station, 1992).

5. O. V. Harding, P. H. Steele, and K. Nordin, Forest Products Journal 43, (1993).

6. D. E. Shannon and W. Weaver, *The mathematical theory of communication.* (University of Illinois Press, Urbana, 1949).

7. G. P. Chapman, Economic Geography 46, (1970).