Artificial Neural Network Technologies Applied to Road Condition Classification Using Acoustic Signals

Kevin McFall
Dalarna University
Department of Computer Engineering and Informatics

Abstract
The goal of the ARCANA project is to produce a road condition sensor that can be used in road weather information systems. ARCANA implements signal processing and artificial neural network (ANN) technology to classify signals recorded with a microphone. This paper is a review of ANN technologies and their application to ARCANA. The entire process is discussed from data acquisition, feature extraction, and ANN design to the actually training of the network. Preliminary classification results in an independent validation set yielded 88% correct classification. This was improved to a full 100% with the addition of a confidence level threshold.

Table of Contents

Introduction

Background

Road weather information systems (RWIS) are an integral part of winter road maintenance schemes. RWIS stations provide a base of information as support to making decisions about winter road maintenance. The road surface is affected by many factors and an RWIS should be able to provide the desired information as in Figure 1. The majority of today's RWIS sensors are based on traditional measurements such as temperature, humidity, wind speed, etc. and none of them can directly classify a road into one of the following classes: dry, wet, snow, and ice. The project Acoustic Road Condition ANAlysis (ARCANA) was developed to provide such a road condition sensor.

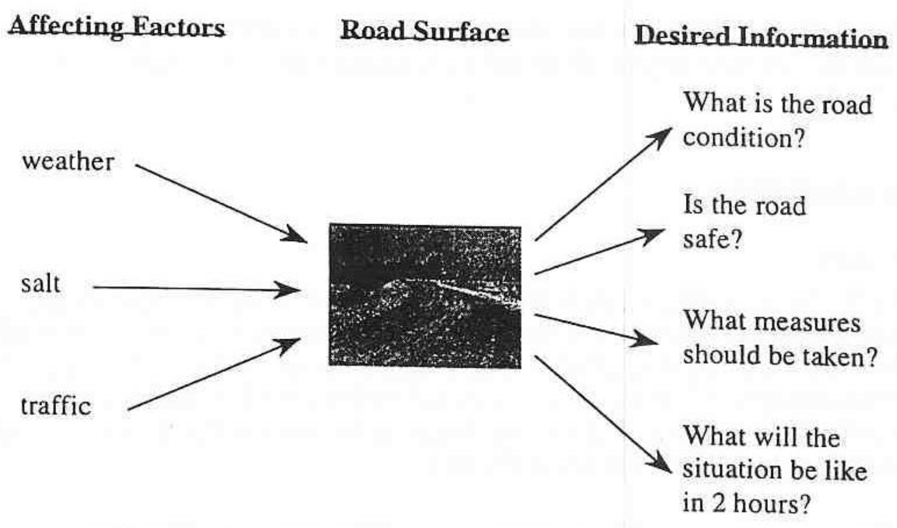
Affecting Factors
Road Surface
Desired Information
weather
salt
traffic
What is the road condition?
Is the road safe?
What measures should be taken?
What will the situation be like in 2 hours?

Figure 1: Schematic diagram of the road environment and information about it an RWIS should support.

ARCANA is inspired by signal processing and neural network technologies and attempts to mimic a human's ability to classify road conditions simply by listening to cars traveling on them. This paper is a survey of neural network classifiers and preliminary results of applying such technology to road condition classification.

Related Work

A project related to ARCANA was carried out in 1992 which attempted to build an acoustic road condition sensor which would be mounted in the wheel well of a moving vehicle [1]. This project ended with unconvincing results mainly due to complications with hardware and underdeveloped feature extraction. Other work in the field of road condition classification has not been found, although acoustics and neural networks have often been applied to vehicle classification [2,3].

Current Technology in Artificial Neural Networks

General

Artificial Neural Networks (ANNs) have become popular in recent years for solving pattern recognition problems. Pattern recognition is the research area that studies the operation and design of systems that recognize patterns in data; the patterns in this case are the various road condition classes. ANNs compete with other more traditional pattern recognition techniques based on techniques such as curve fitting, *k*-nearest neighbor methods, and Bayesian statistics [4,5]. These traditional techniques are plagued with problems such as limited ability to map input patterns to output classes, large requirements in memory, requiring estimations of unknown probabilities, poor generalization, etc. ANNs are compact and powerful, and have the ability to learn relationships of arbitrary complexity between input patterns and output classes.

Design of an ANN classifier involves many steps that are all interrelated. The following sections discuss these steps in detail and examine the various technologies applicable for use in ARCANA.

Data Acquisition

Hardware

ARCANA uses acoustic signal as input to classify road condition. The first step is to record passing vehicles and load the digitized signal into a computer. There are essentially two methods for digitizing the signal: 1) using a digital recording device such as a mini-disc or digital audiotape, or 2) using an analog recording device and digitizing with a computer sound card. After the signal is in digital form, the computer can classify the road condition. A schematic of this process appears in Figure 2.

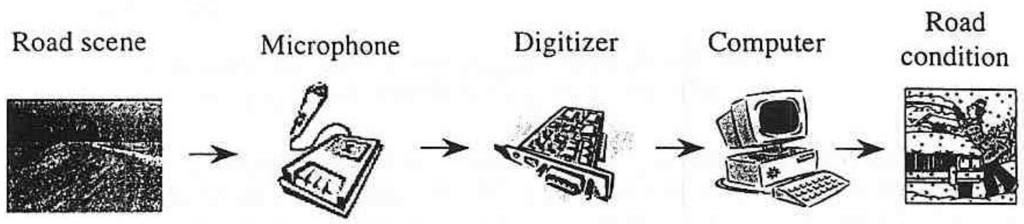

Figure 2: Schematic of signal acquisition and classification

It is importance that the same equipment is consistently used. An ANN trained with signals recorded and digitized with one device can fail completely in classifying signals from another device. Although the same signal captured with different devices may at first appear identical, close examination reveals significant differences. Different microphones have different frequency responses that affect later feature extraction. Additionally, small differences in the analog to digital conversion can also prove problematic.

Data Examination

After digitizing the signal, its quality should be examined. The importance of this step should not be underestimated since signals collected to develop the ANN must be representative of the various road condition classes the system is to detect. By viewing the raw signal or plots of the data, a first estimate to the quality of the data can be obtained. Perhaps the signal had been incorrectly recorded or there was disturbing background noise. Signals containing little or no useful information or those significantly different from the norm, called outliers, must be rejected.

Feature Extraction

Feature extraction calculates characteristics, or features, from the input signal that may be useful for classification. During design of the classifier, there are generally many features as it is unknown from the outset which features contain information capable of separating the desired classes from one another. For example, two hypothetical classes, A and B, are shown in Figure 3. The 2-dimensional plane made by the 2 features is called the feature space, which can be of higher dimension if more features are considered. The regions in the features space where data points of the same class appear are called clusters. As can be seen, feature 1 provides no information useful in discriminating clusters of one class from the other. Much of the difficulty in feature extraction is finding useful features.

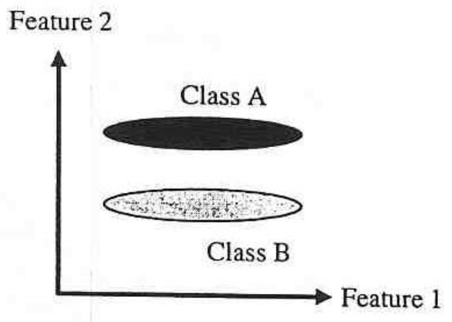

Figure 3: Graph displaying the inability of feature 1 to discriminate class A from class B.

Feature extraction is part art and part intuition, where experience and relentless examination of the acquired data play important roles. Features can be developed which mirror observations of the road classes: the sound from cars traveling on snowy roads is damped, and the splash from a wet road is heard as a car passes. Additionally it might be possible to identify or model the physical phenomena that generate the acoustic signal. For instance measuring Doppler shifts could indicate the speed of the vehicle.

Feature extraction generates a relatively small set of values compared to the original signal. The reduction in data can be very significant: a 5 second signal sampled at 44.1 kHz contains over 220,000 data points while features developed to represent a signal would likely number somewhere between 10 and 100. Feature extraction can thus be viewed as a data compression method that removes unimportant information from the original signal.

Features can characterize signals in different ways. Generally, features can be extracted from signals using different domains: amplitude in the time domain, spectral content in the frequency domain, and similarity coefficients in the scale space domain. In the time domain, such techniques as linear predictive coding or autoregressive moving average models can be applied. Frequency domain features include periodograms, spectrograms, and Gabor transforms [6]. Scale space domain incorporates the use of wavelets [7,8], which contain information from both the time and frequency domains.

Neural Network Design

ANN Layer Structure

In general the structure of an ANN consists of various layers: input, hidden, and output as illustrated in Figure 4. Each line in the figure represents a weight and each circle a node, or artificial neuron. There are i features, j hidden nodes, and m outputs. The value for i is decided by how many features are calculated, but j and m are design parameters.

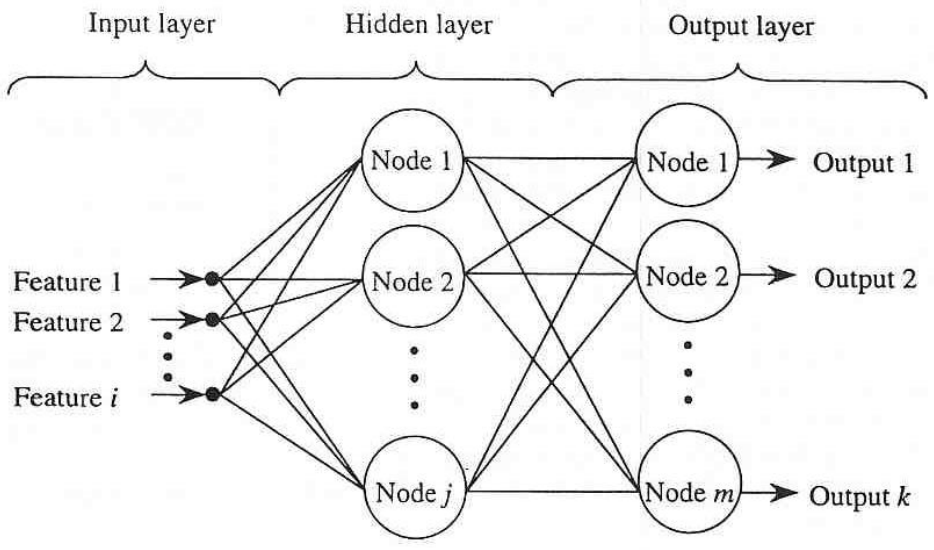

Figure 4: Layer structure of an ANN

The time required to train an ANN is proportional to the number of weights it contains. Equation 1 gives an upper limit for the number of weights, w, needed to completely load the c class patterns using i inputs [10]. The training of the ANNs will be done off-line and thus large numbers of weights will not significantly affect real-time behavior in a final product.

$$w = i\log_2 c \qquad \text{Equation 1}$$

More problematic, however, is the dependence of required number of training examples, t, on network error, e, and number of weights in Equation 2 [10]. If the feature set contained 30 features and all were used to train an ANN to recognize the 5 winter road conditions, a desired error of 2% would require over 2700 training examples. Superfluous features can therefore harm the system and choosing the right features becomes an important step as such a large database of images is not currently available.

$$t \geq \frac{2w}{e} \qquad \text{Equation 2}$$

Equation 1 can be used to set a value for the number of weights the system requires. This value for w is a maximum; often generalization of the network can be improved by reducing w. In any case, Equation 1 can be used as a rule of thumb and then applied along with i and m in Equation 3 to calculate j.

$$j = \frac{w}{im}$$ Equation 3

What remains is to set m. Often this is set equal to c so that there is one output node for every class. However the case of $i = 20$ and $m = c = 5$ (20 input features and 5 outputs for each class) results in only $j = 1$ (rounded up). More hidden nodes than this will most likely lead to overfitting and poor generalization, but experience has shown, especially with ARCANA data, that the relationship between the input features and output classes is quite complicated and 1 hidden node is not sufficient to obtain acceptable results.

A method for resolving this problem is to break the problem into smaller pieces. Instead of 5 outputs, the network can be broken into a system of 5 different ANNs, one dedicated to each of the output classes. For instance one network would become an expert in recognizing dry roads and return a value for whether the signal is a dry road or not. Each network then has 3 hidden nodes with a single output: $j = 3$ (rounded up) for $i = 20$ and $m = 1$. Collecting the outputs from the 5 networks gives a system output similar to the single 5 output network. The main advantage to this is that the system is allowed 15 hidden nodes rather than only 1 without losing ability to generalize. Additionally, each of the 5 networks can utilize different features depending on which are best for distinguishing each particular class.

Temporal Networks

One goal of this project is to produce a short-term prediction of road condition. This requires following trends in road classification and in order to achieve this, a system that can handle temporal data is required. The ANN strategies discussed above deal with static information but can be altered to be temporal. Several neural network structures are designed specifically to be temporal, including time delay and recurrent networks. These networks require features to be calculated in a time series. Time delay networks have multiple inputs for a single temporal variable: n inputs spanning a window with the current and $n-1$ previous values. Figure 5a illustrates an example of a one feature time delay neural network using an $n = 3$ window currently located at the fifth point in the time series. This can, of course be expanded to more than one feature. Rather than being explicitly fed past data, recurrent networks function as a memory by including a feedback loop from the hidden layer of previous time step(s) as input. As a simple example, Figure 5b shows a diagram of a recurrent network with feedback from the last hidden neuron.

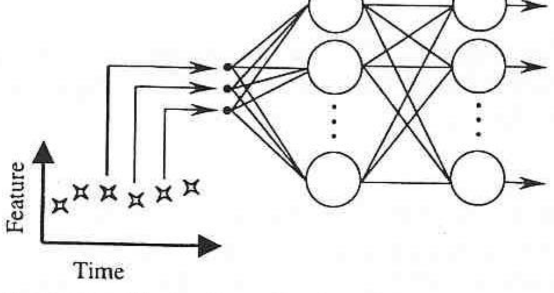

Figure 5a: Illustration of a time delay network.

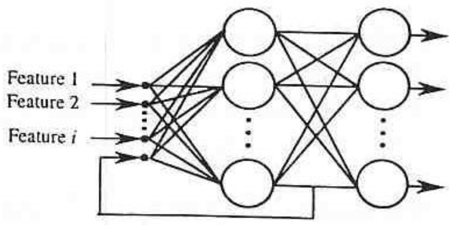

Figure 5b: Illustration of a recurrent network

ANN Training

Supervised versus Unsupervised

Backpropagation is perhaps the most well known learning algorithm, which is applied to a type of ANN known as the multi-layer perceptron (MLP). Backpropagation is a form of supervised training where the weights in the network are altered to reduce the error in classifying the training examples into their predefined output classes. A given weight is altered according to the partial derivative of network error with respect to that weight. Kohonen networks, however, are unsupervised and find locations in the feature space where groups of training examples tend to congregate. Nodes in a Kohonen network are moved so that they correspond to locations in the feature space where clusters of training data exist. While backpropagation networks force classification into user-defined output classes, Kohonen networks indicate the number of classes naturally occurring in the data. For instance, a Kohonen network could notice that a single class for ice could be separated more naturally into thick ice and black ice.

Supervised training schemes are generally more useful for implementation driven projects such as this one where desired output conditions are predetermined. However, Kohonen networks can indicate problems with representativeness of the training data. Additionally, as a continuing development, unsupervised training can be incorporated to continually improve performance during operation without requiring knowledge of the observed road condition.

There also exist methods which combine components of both supervised and unsupervised training. One such is Class Directed Unsupervised Learning (CDUL) [9]. This approach uses Kohonen network nodes where the nodes are moved towards clusters belonging to a certain class, not to the data as a whole.

Training Procedure

When training and evaluating backpropagation MLPs, it is common to split the data examples into 3 sets: training, test, and validation. The training set is used to adjust weights in the ANN, the test set used to avoid overfitting, and the validation set used to evaluate performance.

Before splitting into the 3 sets, the feature data is normalized to zero mean and unity standard deviation. Doing so is not strictly required as the ANN can internalize normalization, but eases the ANN's job by lowering required weight values and eases determination of outliers. At this point, any example with a feature value lying more than 3 standard deviations from the mean are thrown out as an outlier. If included, these outliers could negatively impact performance by affecting training out of proportion with their importance [10].

Ideally, each of the data sets should contain an equal number of examples for each class. An ANN trained with uneven class representation will learn to simply choose those classes with fewer examples with lower frequency rather than learning the true correlation between input and output values.

One danger during training of an ANN is that it will learn to overfit the training data. This occurs when trying to fit exactly to the training set data unnecessarily complicates the network's solution. Figure 2 graphs a hypothetical 2 dimensional feature space where each '+' corresponds to a training example belonging to one class, each 'o' corresponds to a second class, and the line is the ANN's boundary, or discriminant, in the features space between the classes. While both boundaries perfectly separate the training examples, Figure 6a has a desirable discriminant while the discriminant in Figure 6b is overly complicated and will likely result in poor generalization.

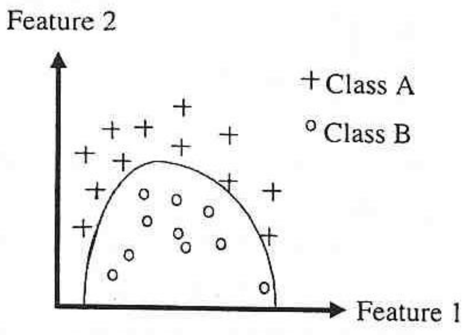

Figure 6a: Desired classification boundary

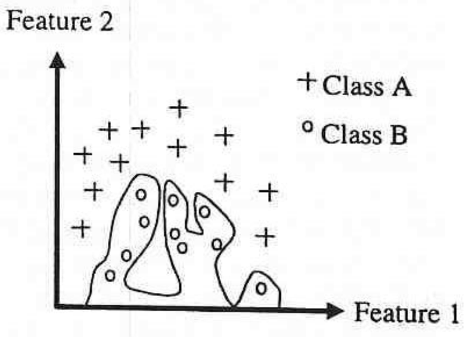

Figure 6b: Overfitted classification boundary

There are several ways to avoid overfitting. One is by limiting the size of weight values in the ANN. The output from each neuron is bounded by a so-called activation function, $g(x)$, and is often chosen to be a sigmoid function. The equation of the commonly used logarithmic sigmoid appears in Equation 4 and is graphed in Figure 7. As can be seen, the value of $g(x)$ rapidly approaches asymptotes as the absolute value of x increases. The value of x is affected by the weights of the input connections to the given neuron, and since high values of x do not affect $g(x)$, there is no need for large weights. Large weights generally signify overfitting and a limit of $W < 10$ is recommended [10].

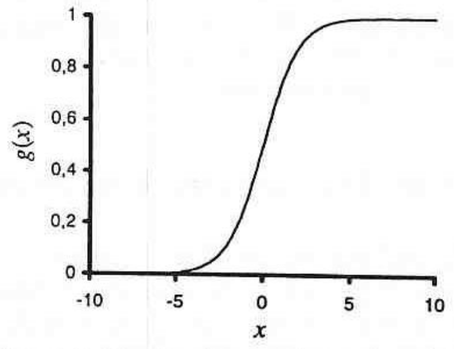

Figure 7: Graph of logarithmic sigmoid

$$g(x)=\frac{1}{1+e^{-x}}$$ Equation 4

A second method for avoiding overfitting is to track performance with respect to the test set during training. Training is an iterative process where weights are adjusted to achieve lower error in the training step during each successive time step, or epoch. In general, error in the test set will also decrease over time. When overfitting occurs, training set error continues to decrease while test set error becomes worse. Figure 8 shows errors in training and test sets during training in an actual run with ARCANA data. Training in this case should be stopped at the eighteenth epoch where error in the test set becomes worse again.

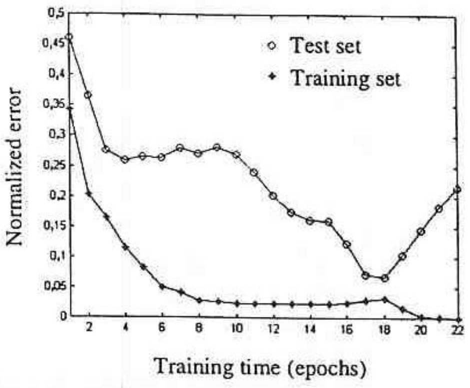

Figure 8: Errors in training and test sets during ANN training

Since ANN training is dependent on both the training and test sets, classification results for examples in those sets are not objective. For this reason, after the ANN has been trained it is tested against the validation set which contains signals the ANN has not previously been exposed to. Classification rates in the validation set are good estimates at how well the system will classify new data.

Representativeness, Generalization, and Confidence

The performance of classification systems depends critically on the quality of the training data. In particular, the issue of representativeness arises, e.g. whether the training data managed to adequately represent all the expecting operating conditions of the final system. Ideally a classifier will be able to extrapolate, or generalize, from its training to correctly classify feature patterns it has not previously been exposed to.

When considering generalization, the first step is to return to data acquisition. Have all the expected operating conditions been captured in the training set? What, if anything, caused outliers to occur in the training data? Are they truly outliers, or is there some other mode of operation that has not been adequately characterized (insufficient sample data)? Poor classifier performance is not always due to poor design of the classifier itself; poor or confusing input data can cause the best classifier to fail.

Much work has been done attempting to measure an ANNs ability to generalize [11-14]. Although rigorous and thorough, these works attempt to connect statistical probabilities with generalization that is not straightforward [5]. Additionally, these statistical investigations require knowledge of probability distributions that are not easily estimated, especially with limited training set data. These statistical techniques measure generalization of the ANN as a whole and do not indicate a confidence in how well a given new signal will be classified. Such a confidence value would be beneficial for making decisions based on results from the classifier.

Development of Confidence Value

A confidence value calculation has been developed using a *k*-nearest neighbor technique [15]. On their own, *k*-nearest neighbor techniques can be used as a classifier. A database of feature patterns is collected and stored with the corresponding output class. When a new feature pattern is encountered, it is classified according to the majority class of the *k* examples in the database that lie closest to it in the feature space. This type of classifier can be quite accurate although it has significant drawbacks: sensitivity to distribution of training examples in the feature space, the database of features patterns must be retained online requiring large amounts of memory, and calculation of the nearest neighbor can be time consuming. In order to reap the benefits of *k*-nearest neighbor techniques and eliminate its drawbacks, research has been done to try and incorporate *k*-nearest neighbor techniques in ANNs [16-17].

The method for calculated confidence value does not combine *k*-nearest neighbor and ANN but rather adds an additional measurement after the ANN has made its classification. When a new signal is to be classified, first the outputs from the ANN system are calculated. Then a certain number, *k*, of examples are chosen from the training and test sets which lie closest in the feature space to the new signal. Each of the neighbors is assigned a weight, either +1 or −1. In order to obtain a positive weight, the neighbor must be classified correctly as well as being classified the same as the new signal; otherwise the neighbor receives a negative weight. The weights of each neighbor are then multiplied by a measure of how close to the new signal they lie in the feature space. Thus neighbors receive a value of ± 1 if they coincide with the new signal and a 0 value if they are infinitely distant. The confidence value is calculated as the mean of the *k* neighbor values.

This confidence value take values between −1 and +1 with a value +/- 1 corresponding to total confidence of correct/incorrect classification and a value of 0 corresponding to lack of confidence in classification results. Those new signals with high confidence values are more likely to be classified correctly. A confidence threshold can be set below which classification is unreliable and therefore not trusted. The system can then wait until it obtains a trustworthy classification.

Using the confidence value developed here requires two design parameters: the number of neighbors, *k*, and the confidence threshold. The values of these parameters determine the percentage of validation signals rejected and correct classification rate. In general raising classification rate will also raise the number of rejected signals. It has been found that a 100% correct classification rate can always be achieved by making the threshold arbitrarily high and that using a small *k* decreases the number of signals rejected. If *k* is chosen too small, however, performance will break down. The optimal *k* value is between 8 and 10 nearest neighbors.

There are several advantages to implementing this confidence value. First and foremost is that correct classification rate will rise as the confidence threshold is raised. Additionally, confidence can be used in continually improving a prototype by indicating those signals the system would benefit most from by adding to the training set. Information about signals with low confidence values can be saved to be later classified by hand and added to training. Finally, the confidence value can play an important role in detecting unusual signals in the final system. The system must be able to realize that it will not be able to classify, for example, when cars are passing each other or plowing vehicles are present. Generally an ANN cannot automatically recognize such an events, but a low confidence value should occur under such conditions.

Application of ANN Technology to ARCANA

Data Acquisition

Hardware

The database of signals collected for ARCANA have been recorded with a SONY portable DAT recorder using an AKG C414 B-ULS microphone with a top frequency of 20 kHz. A special adapter was used in order to transfer the digital signal directly from DAT recorder to computer.

Feature Extraction

The features currently used in ARCANA are calculated from the spectrogram of the raw signal. Figure 9 shows spectrograms for typical signals from each of the four classes. As can be seen, each class has a distinguishable spectral signature. For dry roads the spectrogram is quite simple: frequency content increase as the vehicle approaches and then drops sharply as it passes. The signal for wet roads is similar in the beginning, but signal strength takes much longer to decay after the vehicle passes due to water caught in the vehicle's wake. Strong damping of all but the lowest frequencies occurs for signals recorded from snowy roads. Ice on the other hand is similar in shape to the dry signal but is characterized by a more gradual change from the dark signal core to the silence when the vehicle is far from the microphone.

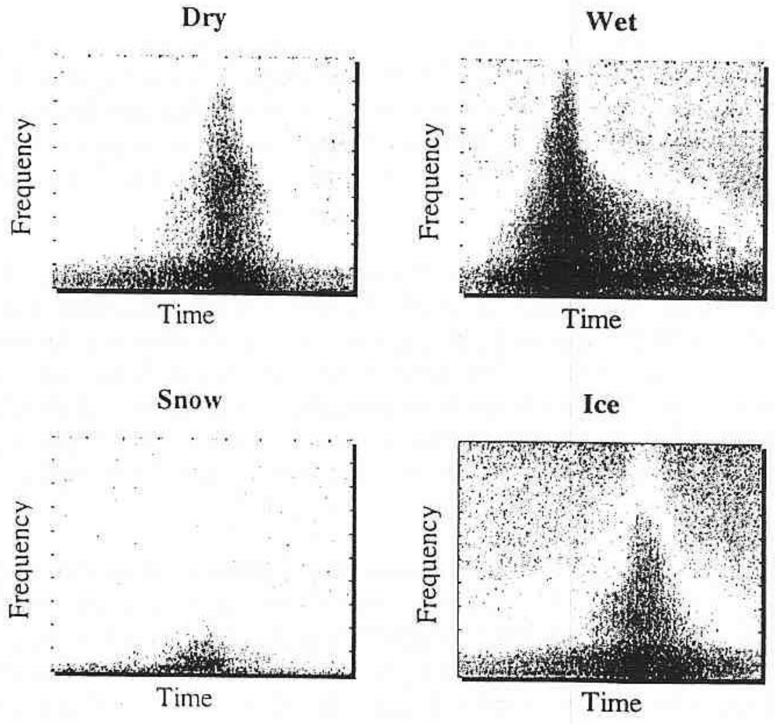

Figure 9: Spectrograms of typical signals for the 4 road classes

Neural Network Design

ANN Layer Structure

The classification results presented in a later section incorporate 20 features calculated from the spectrogram as inputs. As discussed above, a system of 5 one output networks each specializing in a class was used. From the constraint on total number of weights in Equation 1, Equation 3 was used to decide on using 3 hidden nodes. A new signal is classified to the class of the network resulting in the highest output value.

Temporal Networks

Classification and predication using temporal networks requires time series data, which is not yet available. Dalarna University has purchased an RWIS measuring station, which is to be installed in the fall of 1999. This station will include a microphone and ARCANA software for calculating the necessary features. This field station is due to be in full operation by the start of the winter season 1999.

ANN Training

Supervised versus Unsupervised

The original work in ARCANA has involved backpropagation trained MLPs and the majority of results are from using such networks. For comparison, CDUL networks were trained using ARCANA. Results of both network types appear in a later section.

Training Procedure

After removing outliers, the signal database contained 201 examples. These examples were then split 60/20/20% into training, test, and validation sets. The numbers of signals in each set belonging to each class are listed in Table 1. A significant problem with an ANN structure consisting of 1 network for each class is the uneven number of training examples in each class. The network trained to recognize whether a signal belongs to the dry class or not has 33 examples of dry and 89 examples of not dry. A network trained with this large of a discrepancy is prone to classify every signal as not dry since there are so few dry examples. To combat this problem, the error is recalculated so that the 33 dry examples carry equal weight as the 89 not dry examples. Thus for example, if an ANN decides to classify every signal as not dry, an error of 50% is obtained rather than only 27% (73% of the validation signals are actually not dry so overall error would be 27%) without the recalculated error measurement.

Table 1: Distribution of signals in data sets and classes

	Dry	Wet	Snow	Ice	Total
Training	33	45	6	38	122
Test	11	15	2	12	40
Validation	10	15	2	12	39
Total	54	75	10	62	201

Training is stopped after one of the following conditions is reached: 100 epochs have passed or the absolute value of a network weight has grown larger than 10. The network is then chosen for the epoch in which the recalculated error is lowest with respect to the test set. ANN training is dependent on the initial weight values and it is common to retrain the network multiple times to be sure the optimal weights are found. In subsequent training runs, error must be better in both test and training sets in order to become the network of choice.

Representativeness, Generalization, and Confidence

Unlike experimental data taken in a laboratory environment, the signals collected for ARCANA were filmed at several locations on 2 lane highways in the vicinity of Kiruna, Sweden. Data was recorded at the end of the winter season 1999 and unfortunately very few snow-covered roads were available. On the other hand, a wide variety of vehicles are represented by the data including passenger cars, pickup trucks, minivans, busses, tractor-trailers, and vehicles towing trailers.

The signal database is unfortunately not particularly large. A desirably sized database would include hundreds rather than tens of signals for each class. Once the ARCANA field station is operational, the database can be quickly expanded. Until then classification results can be quite sensitive to which signals are chosen to be in the validation set. Correspondingly, results below are the average of 10 runs where each run has a different (random) split of the data into the 3 sets. This provides a more accurate view of true performance with new signals.

The quality of the signal data is high, but a database with less than 200 examples is not likely to represent all possible conditions. For this reason, the implementation of the field station is important where large numbers of signals can be recorded with some degree of automation.

The original classification results, see next section, are fairly high but classification rates of over 90% are desirable. In order to raise classification rate, the confidence value measurement was developed. Using the confidence threshold significantly raises correct classification rates. The major drawback is that obtaining very high classification rates requires a threshold, which excludes many validation set signals. In some applications this could be a serious problem, but ARCANA is not overly affected. ARCANA is developed for integration with the RWIS maintained by the Swedish National Road Administration. The RWIS stations report sensor values only once every 30 minutes. Therefore the ARCANA system will have an entire half hour to record a signal having a high enough confidence value. A second drawback, common to all *k*-nearest neighbor techniques, is the need to store many examples. The memory required is proportional to the number of the signals in the database. With the current database, this corresponds to less than 9 kilobytes. An area of future research is to determine whether all training and test set signals are necessary or whether a certain number in given locations in the feature space are sufficient. In any case the memory required for the confidence value is not likely to be prohibitive.

Classification Results

Two types of ANNs, MLP and CDUL, were trained and their performance measured as shown in Table 2. Classification in the validation set is not acceptable; a working ARCANA sensor would need to be at over 90% reliable to be of practical use. Perfect classification can be obtained by implementing a confidence threshold of 0,53 or higher.

Perfect validation set classification comes at the expense of rejecting many of the signals due to low confidence. Figure 10 plots the dependence of classification rate and the number of validation signals accepted for classification on confidence threshold. As the confidence threshold is raised, classification rate increases and accepted signals decreases. At the 0,53 threshold for 100% classification, 23% of the signals are accepted. But as mentioned above, a signal with reliable confidence will likely be recorded in the half hour interval of RWIS station updates.

The results with the CDUL network are lower than with an MLP. CDUL is a new technique and its application to ARCANA has not been completely investigated. CDUL parameters can be further optimized and should be able to obtain performance similar to MLPs. However, there is currently no corresponding confidence value measurement foiCDUL. Without a confidence value, CDUL is not a viable alternative.

Table 2: Correct classification rate (%) in the three data sets for different ANN schemes

	Training	Test	Validation
CDUL	94	85	74
MLP	99	90	88
MLP with confidence	99	90	100

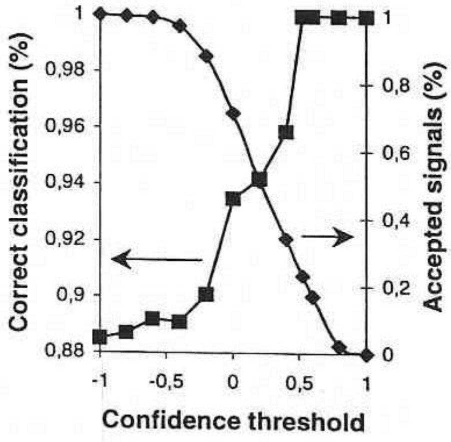

Figure 10: Performance in validation set using confidence threshold.

Discussion

The process of training ANNs for classification of winter road condition is a long process involving data acquisition, feature extraction, design of the ANN, etc. While the actual training of the ANN produces the classifier, the steps leading up to training are equally important. Specifically for ARCANA, several key points need attention: consistently using the same microphone and digitizing hardware, collection of more data, and continued evaluation of the confidence measure.

Original classification results in the validation set were under 90% correct. By implementing a confidence value threshold, 100% correct classification was achieved. This confidence value can also be used to aid in continued training in the ARCANA field station, which will be installed before the coming winter. Additionally, the confidence value can be used to filter unusual signals such as those from unusual vehicles or with significant background noise.

There are also promising areas of future development for ARCANA. One is in the possibility of using time series of output for short-term road condition prediction. The field station will play an important role in this and more research into temporal ANNs is required. Also, ARCANA will hopefully be able to classify mixed road conditions. This may be accomplished using trend data and the newly developed confidence value.

References

1 A. Svärdström, "Klassificering av Väglag med Hjälp av ett Neuralt Nät", Uppsala University Institute of Technology publication, ISSN 0346-8887, June 1993.

2 A. Nooralahiyan, M. Dougherty, D. McKeown, H. Kirby, "A Field Trial of Acoustic Signature Analysis for Vehicle Classification", Transportation Research C, 5(3/4), pp 165-177, 1997.

3 S. Sampan, "Neural Fuzzy Techniques In Vehicle Acoustic Signal Classification", Ph.D. dissertation, Virginia Polytechnic Institute and State University, 1997.

4 Bishop, *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, 1995.

5 M. Smith, *Neural Networks for Statistical Modeling*, International Thomson Computer Press, London, 1996.

6 T. Masters, *Signal and Image Processing with Neural Networks*, John Wiley and Sons Inc., New York, 1994.

7 G. Strang, T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Wellesley, 1997.

8 K. eom, M. Wellman, N. Srour, D. Hillis, R. Chellappa, "Acoustic Target Classification Using Multiscale Methods", Proceeding from the 1997 Sensors and Electron Devices Symposium, http://sensor.sanders.com/public/sensors97/index.html.

9 M. Mackenzie, "CDUL – Class Directed Unsupervised Learning", *Neural Computing & Applications*, Vol. 3, No. 1, 1995.

10 K. Swingler, *Applying Neural Networks*, Academic Press, London, 1996.

11 V. Vapnik, "Principles of Risk Minimization for Learning Theory", *Advances in Neural Information Processing Systems 4*, Morgan Kaufman Publishers, San Mateo, 1992.

12 J. Moody, "The Effective Numer of Parameters: An Analysis of Generalization and Regularization in nonlinear Learning Systems", *Advances in Neural Information Processing Systems 4*, Morgan Kaufman Publishers, San Mateo, 1992.

13 D. MacKay, "Bayesian Model Comparison and Backprop Nets", *Advances in Neural Information Processing Systems 4*, Morgan Kaufman Publishers, San Mateo, 1992.

14 T. Fine, S. Mukherjee, "Parameter Convergence and Learning Curves for Neural Networks", *Neural Computation*, Vol. 11, 1999.

15 K. McFall, "Use of *k*-Nearest Neighbor Based Confidence Value for Improved Classification in Neural Networks", pending publication.

16 Y. Chen, R. Damper, "On Neural-Network Implementation of *k*-Nearest Neighbor Pattern Classifiers", *IEEE Transactions on Circuits and Systems – I: Fundamental Theory and Applications*, vol. 44, No. 7, July 1997.

17 O. Murphy, B. Brooks, T. Kite, "Computing Nearest Neighbor Pattern Classification Perceptrons", *Information Sciences* **83**, 1995.

Biography

Kevin S. McFall graduated from Virginia Polytechnic Institute with a B.Sc. in 1995 and later graduated with an M.Sc. from Massachusetts Institute of Technology (MIT) in 1997, with both degrees in Mechanical Engineering. During studies at MIT he worked with superconducting magnets both at the Francis Bitter Magnet Laboratory and as a visiting researcher at the Japan Atomic Energy Research Institute. During 1997 he worked at the University of Central Florida investigating heat transfer in cryogenic electronics. In 1998 he moved to Sweden where he currently holds an adjunct professor position in the Department of Computer Engineering and Informatics at Dalarna University with interests in artificial intelligence.