A Speech Recognition Technique Using an Artificial Neural Network

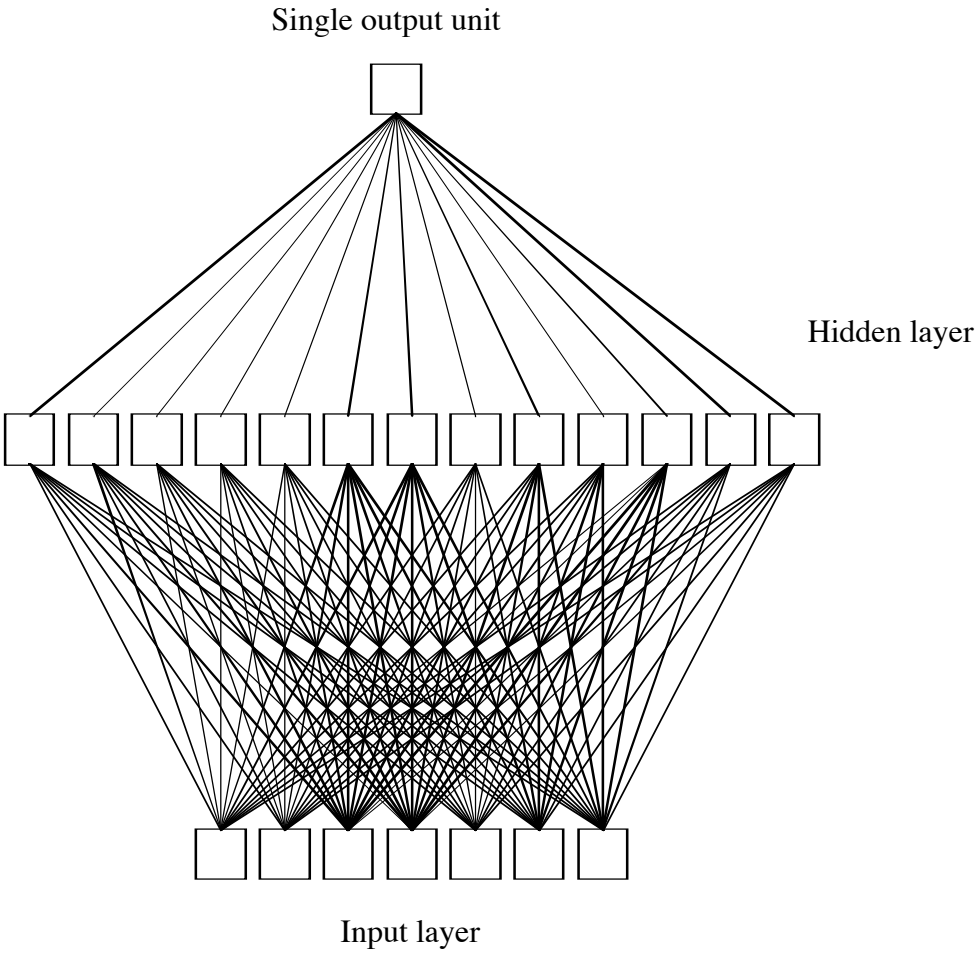
Austin Marshall

CS6320 Natural Language Processing, Fall 2004

Speech recognition is an important discipline that integrates knowledge acquired from the study of computer science and the study of human perception. Endowing computers with the ability to recognize speech can provide numerous benefits in human-computer interaction and artificial intelligence applications. Artificial Neural Networks (ANNs) are a popular means for classification and pattern recognition and are well suited for the problem of speech recognition. This paper will focus on an effective method of speech classification using an ANN model called the perceptron, proposed by Rosenblatt, and will demonstrate how the network can be implemented in Matlab.

Artificial Neural Networks are abstract mathematical models that borrow from concepts unique to human (and otherwise) nervous systems, in particular, the way in which neurons form networks. ANNs are well suited for classification and pattern recognition applications for their ability to generalize data by learning through experience, which information is important in identifying features present in stimuli. The neurons in an artificial neural network compute activation based on the weighted sum of their input. Features that are consistently present in similar samples are weighted more heavily than features that do not appear with as much consistency. After an ANN is trained, the weights are such that unseen stimuli with common features are recognized as stimuli with which the network is familiar.

Figure 1 Basic Neural Network graph



Before a perceptron can be trained to recognize speech. Speech must first be "preprocessed" by extracting features in a sound recording sample. Recording samples, rarely, if ever, produce identical waveforms. Waveforms for perceptually similar recording samples vary in length, environmental conditions, background noise, sample rates, pitch, etc… The solution to reducing the dimensionality of and variability between the stimuli is to extract only the features from the signal that are most relevant to speech. A popular method of feature extraction is through the use of cepstral analysis.

A cepstrum refers to the inverse Fourier of the log of the Fourier transform of the original signal, sampled at a regular interval. Cepstral analysis is accomplished by using

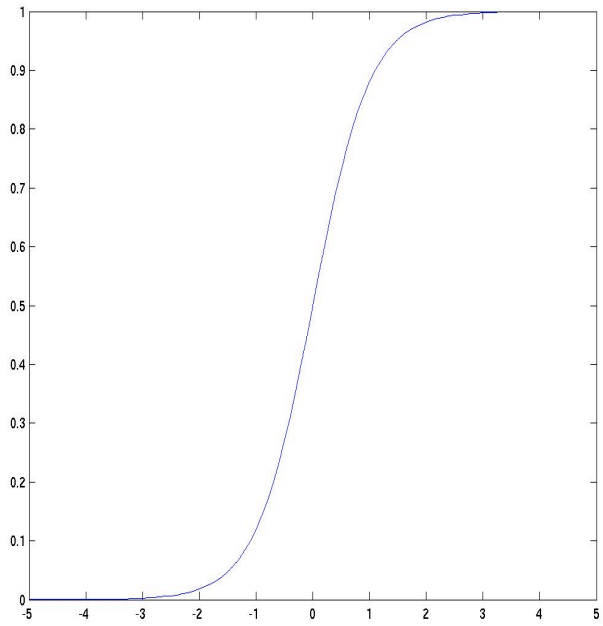
Mel Frequency Cepstrum Coefficients (MFCC), as implemented in Matlab via Malcolm Slaney's Auditory Toolbox. MFCC uses a logarithmic scale based on a series of pitches with a fixed perceptual distance from one another. As such, the Mel Scale is a scale closely related to that of human hearing. The mfcc function returns vectors of 13 coefficients for each time slice specified by the user, reducing the dimensionality of the stimuli from tens of thousands of amplitudes present in the waveform to just 13 coefficients.

A perceptron was trained using stimuli consisting of recorded samples of spoken digits between "one" and "eight" at a sampling rate of 44,100 Hz. Background noise was removed using Audacity, a free software package for audio recording and manipulation, to reduce the likelihood of environmental variability interfering with the training process. Seven samples of each digit were recorded from the same speaker producing a total of 56 training stimuli. A cepstral analysis, of each recording sample was used to represent speech through the use of Mel Frequency Cepstral Coefficients (MFCC). Each training stimulus consisted of a 26-dimensional vector of MFCC coefficients corresponding to the 13 coefficients from the first half of each sample and the 13 coefficients corresponding to their second halves.

A perceptron was used to classify the samples based on a predetermined target. The network architecture consisted of a 26-dimensional input vector, 100 hidden-layer units of McCulloch-Pitts neurons (McCulloch and Pitts) and one output unit. The output of each neuron was limited to values between 0 and +1 using the soft differential sigmoid function such that $S(x)=1/(1+\exp(-x))$. The weights between the input and the hidden layer were random values between –1 and +1 and remained fixed throughout the training

process. The weights between the hidden layer and the output were initially set at zero. A 56-dimensional target vector was used consisting of ones corresponding to stimuli in the target group and zeroes for stimuli not in the target group.

Figure 2 Soft differentiable sigmoid function, S(x)=1/(1+exp(-x))

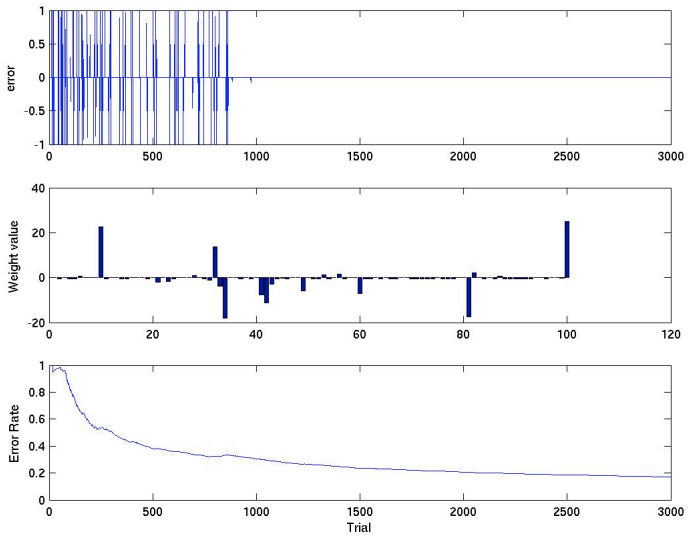


A learning rate (gamma) of 1 and a bias of –1 were used in the activation updating rule. Training stimuli were chosen at random according to a uniform distribution and weights updated according to the error in stimulus/response pairs. The training process occurred until the error rate leveled off; the network had successfully classified all training stimuli.

Eight separate networks were trained, one for each spoken digit, each using the same initially random connection matrix between the hidden layer and input. After each of the eight training sessions ended, the network was tested against unseen stimuli.

By the end of the training process of each of the eight targets, the error rate leveled off sufficiently after 3,000 iterations. After the training process was completed, the network was tested against unseen stimuli. An unseen stimulus was a recording sample from the original speaker, not present in the training stimuli. Eight recording samples were used corresponding to the eight digits used to train the network. The same processes used for training stimuli were used for the unseen stimuli; background noise was removed and MFCC coefficients were calculated. Each network was tested against each of the eight unseen stimuli. After training, each network successfully classified the stimuli, i.e. the network trained to recognize "one" returned full activation (+1) for the "one" stimulus and zero for all else.

Figure 3 Results of training for target "one", showing the error term for each trial, the end result of the weights, and the error rate as a function of time.



Furthermore, preliminary tests were conducted to test the network's response to samples with background noise present and for spoken digits produced by novel speakers, one male and one female. In all three conditions, the networks returned a low

(but greater than zero) activation for their respective targets. Accuracy improved in all three conditions with longer training periods.

The results shown here provide evidence that these Artificial Neural Networks used in conjunction with Mel Frequency Cepstrum Coefficients provide an effective means for speech classification, and the method presented in this paper is useful for classifying familiar speech samples.

Much research is conducted in determining effective speech representation schemes. The representation scheme used in this paper, while based on the popular Mel Frequency Cepstrum Coefficients, is unique insofar as most speech recognition models use MFCCs, or other feature extraction methods, for a fixed window of time as only part of the speech-relevant information, whereas this one uses windows of time from the first and last halves of the samples. The representation scheme used in this way is well suited for recognizing a fixed set of stimuli independent of the set of phonemes used in producing the words. For example, the network could recognize the presence of "one" or "two" in a speech signal, useful for applications like a telephone call center where the listener is presented with a set of options that they must speak into the phone.

The model could be used to recognize phonemes as long as the training stimuli are carefully chosen in such a way that a training stimulus only include information relevant to one phoneme. The network could then be trained to recognize phonemes, and the results could then be applied to a statistical model such as the Hidden Markov Model described in *Speech and Language Processing* by Jurafsky & Martin. Such a model would be useful in continuous speech recognition applications to recognize whole words

based on their phonemic composition. The same model could be extended to recognize whole sentences independent of certain words' presence in the training body.

References

Jurafsky, Daniel and Martin, James H. (2000) *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (1st ed.). Prentice Hall

Golden, Richard M. (1996) *Mathematical Methods for Neural Network Analysis and Design* (1st ed.). MIT Press

Anderson, James A. (1995) *An Introduction to Neural Networks* (1st ed.). MIT Press

Hosom, John-Paul, Cole, Ron, Fanty, Mark, Schalkwyk, Joham, Yan, Yonghong, Wei, Wei (1999, February 2). *Training Neural Networks for Speech Recognition* Center for Spoken Language Understanding, Oregon Graduate Institute of Science and Technology, http://speech.bme.ogi.edu/tutordemos/nnet_training/tutorial.html

Slaney, Malcolm *Auditory Toolbox* Interval Research Corporation

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review* 65: 386-408

Rosenblatt, F. (1962). *Principles of neurodynamics: Perceptrons and the theory of brain mechanisms*. Washington DC: Spartan

McCulluch, W. S., and Pitts, W. (1943). A Logical Calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* 5: 115-133