Phoneme Recognition Using Hierarchical Temporal Memory with Articulator Correlation

Mishal Awadah
emish
@seas.upenn.edu
Univ. of Pennsylvania
Philadelphia, PA

Robert Hass
hassr
@seas.upenn.edu
Univ. of Pennsylvania
Philadelphia, PA

John P. Mayer
jpmayer
@seas.upenn.edu
Univ. of Pennsylvania
Philadelphia, PA

Mark Liberman
myl
@cis.upenn.edu
Univ. of Pennsylvania
Philadelphia, PA

ABSTRACT

The aim of this project is to develop a biologically-motivated approach to speech recognition. Hierarchical Temporal Memory systems will be used to recognize words in recorded speech aided by tracking the articulators in speakers' mouths.

This system employs scientific knowledge of human speech perception at every step. Hierarchical Temporal Memory networks compute as neural circuits in the neocortex do. The audio inputs to the HTM are transformed in a way designed to mimic the processing done by the ear. By training this system on both speech sounds and articulator motions, the data available to it is made very similar to that which humans use when processing speech. Using phoneme classes as an intermediate step in the path from sound wave to word is another feature shared by this system and the human brain. It is hoped that this method, by approximating as much as possible the best known approach to speech recognition, will enjoy similar success.

1. INTRODUCTION

Speech recognition today relies on general-purpose machine learning algorithms as opposed to biologically-motivated models. By failing to incorporate scientific knowledge of human speech perception, current approaches ignore valuable insight into a difficult computational problem. It is hoped that the following approach, which applies a neurological model to biologically plausible inputs, can match or exceed the performance of probabilistic formulations.

Speech recognition technology is increasingly prevalent. Automatic transcription packages, as well as voice controlled applications and appliances, are becoming more and more common. However, refinements to current speech recognition methods yield ever-smaller gains in performance; while standard methods are quite accurate they seem unlikely to improve much beyond their current state. Also, persistent problems like sensitivity to noise continue to impede performance.

Hierarchical temporal memory (HTM) is a recent machine learning model developed by Numenta, Inc., that replicates the structural and algorithmic properties of the neocortex [2]. HTMs build a model of some domain by learning from experience. This experience comes from exposure to data received from an array of sensors.

HTMs are organized as a tree-shaped hierarchy of nodes, where each node implements a common learning algorithm. The input to every node, regardless of its position in the hierarchy, is a temporal sequence of patterns in vector form. Each node contains two components referred to as "poolers". The spacial pooler maps incoming vector data to currently stored vectors that have already been seen, and the temporal pooler groups vectors together based on how close they occur to each other in time.

Every node, and as a result the HTM as a whole, operates in two modes: training and inference. From the perspective of a node during the training mode, only the spatial pooler is active. Incoming vector data is stored as quantization points which represent different data sets. These points are defined by setting a Euclidean distance D for which the spatial pooler uses to compare currently stored quantization points to the incoming data vectors. If incoming data differs from currently stored data by at least D, then a new quantization point is created to represent that data. Otherwise, no new data is added to the spatial pooler. Once this operation has gone on for long enough such that quantization points are no longer being added, or the pooler has reached an acceptable threshold, the temporal pooler is activated.

The temporal pooler identifies patterns in time, and so begins to group quantization points together based on their temporal proximity. As new vector data is received by the node, quantization points are outputte from the spatial pooler in a sequential manner. So if event A is continuously followed by event B, for instance, the temporal pooler will create a group that contains A followed by B. Ignoring much of the complicated logic governing this learning scheme; the temporal pooler creates a first-order time-adjacency matrix where time coherent group creation can be viewed as finding the most highly connected sub-graphs from the graph represented by the adjacency matrix [1]. Once these temporal transitions are learned and corresponding groups of centers created, the temporal pooler starts producing output in terms of its learned temporal groups. The output is a vector of size equivalent to the number of temporal groups created, and can be considered as a probability distribution over the space of the temporal groups [1].

The operation of nodes in the hierarchy follows a level by level strategy. Preliminarily, nodes at the bottom level are trained while the rest of the nodes are disabled. Once all the nodes at level i are finished learning, they begin to output to level $i + 1$ nodes. The level i nodes are said to have reached the "inference" stage, and the level $i + 1$ nodes are enabled and are now in the "learning" stage. This process repeats until all nodes are fully trained. When this finally occurs, the HTM as a whole can begin making inferences about the entire data set that it is being tested on.

2. RELATED WORK

Most automated speech recognition projects rely on general-purpose probabilistic models, of which the most prominent has been the Hidden Markov Model. While these models are excellent tools for probabilistic analysis, they are much broader in their analyses than HTMs because they do not make as many assumptions about the nature of the world [site: htm_comparison]. The Hierarchical Hidden Markov Model is a modified general-purpose model that most closely resembles HTMs in the way it models time. The difference between them, however, is that HMMs exploit hierarchy in only one dimension, namely time. HTMs on the other hand, contain a hierarchy in both time and space, which gives them unique advantages while learning [5].

The use of HTMs in speech processing is limited, but it seems to be spreading slowly. A primary example that demonstrates proof of concept is the speech processing demonstration bundled with NuPIC, the Numenta Platform for Intelligent Computing. In this demonstration, Numenta have trained an HTM to process speech and solve two problems: gender classification and speaker identification. The demonstration uses digital audio recordings of human speakers that undergo required signal pre-processing such that inputs to the HTM is in the log Mel spectrum. The experiment shows promising accuracy results for the inferring HTM [6].

Students in a Stanford Machine Learning class have also demonstrated the use of HTMs for spoken language identification [7]. Robinson, Leung, and Falco reported high utterance-by-utterance accuracies for three English and one French classifications. The HTM was able to reach near perfect classification between English and French languages with fewer than fifty training examples.

Finally, at the Center for Language and Speech Technology at the University of Nijmegen in the Netherlands, Doremalen and Boves have shown that using HTMs to recognize spoken digits holds promise [3]. The results of this study show error rates below 20%, despite the HTM system not being fully optimized for processing audio signals. Doremalen and Boves also suggest that the present implementation of HTM learning algorithms may be suboptimal for processing signals that encode information mainly in dynamic changes due to lack of propagating learned input patterns as top-down feedback.

Other studies have shown the benefits in general of incorporating articulatory data in continuous speech recognition [4]. A phoneme recognition system achieved higher accuracy on both articulator and sound data than on either alone.

Another approach, the BeBe system, used parallel detectors in an attempt to emulate speech perception in humans [8]. This study was a preliminary effort – only a small number of phonemes were analyzed, all of the detectors were hand-coded, and the system was trained and tested on a very small speech sample.

3. PROJECT PROPOSAL

3.1 Anticipated Approach

At a high level, HTMs will be employed to recognize and classify incidences of phonetic classes. This will be done in several steps with a variety of HTM configurations – successive stages will incorporate both audio and articulatory data on small and large corpora (See Figure 1).

In the first configuration, a series of HTMs will each be trained to identify occurrences of a single phonetic class. The set of classes that will be implemented includes stops, fricatives, vowels and nasals, although other

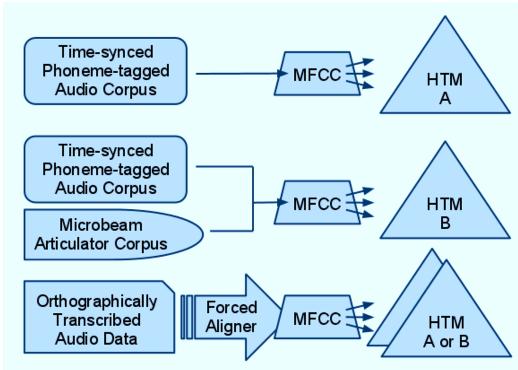

Figure 1: Training Pipelines

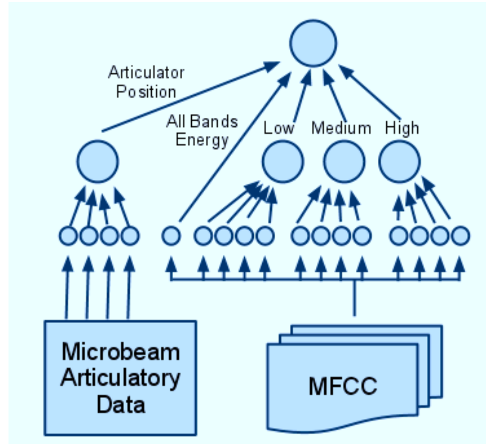

Figure 2: HTM Node Configuration

classes may be investigated. Initially, individual HTMs will each be trained to recognize a particular phoneme class; these discrete detectors will then be joined into a single model.

Detector HTMs will first be trained on audio samples alone. Using data from the TIMIT corpus, occurrences of a given phonetic class will be identified. This subset of the samples will then go through a series of pre-processing operations to reduce noise and to represent the audio data in a distributed format. Various transforms will be tested and compared, including Log-spectrum, Mel-frequency Cepstrum Coefficients (MFCC), and Perceptual Linear Predictive (PLP) Model. One of these parallel representations will serve as input to bottom level of the HTM tree.

Crucial to the functionality of each node is the notion of distance between distinct instances of data from the input streams. It is anticipated that tuning this parameter will require a large number of iterations. Additionally, various techniques of determining distance will be compared and assessed; currently Euclidean and Mahalanobis metrics will be compared.

Within each HTM, each node of the bottom level will look at only a small subset of the incoming data streams. Streams will first be grouped together by a single frequency band. At higher levels, larger sets of frequency bands will be aggregated, with the root node experiencing the effects of all of the input streams together with the overall energy waveform. This way, lower levels of the hierarchy will group together fast, local patterns, where upper levels will group together slower and more global patterns. The output of the top node of the HTM will represent its confidence of recognising an instance of a particular phonetic event.

A second trial will additionally provide HTM recognizers with articulatory data from the University of Wisconsin X-ray Microbeam Speech Production Database. This data will be integrated as a separate sub-tree in an expanded HTM as shown in Figure 2. The articulatory information in concert with sound should give a more precise indication of phoneme classes than sound alone – this advantage should remain when the same HTM is subsequently trained on audio without articulation data.

While the first generation of detectors will be trained in advantageous circumstances (either by training on the highly sanitized TIMIT corpus or by including articulator information), successive generations will attempt to refine correlations between phonemic classes and acoustic cues by training on additional data sources. These datasets have a tremendous size advantage over either of the others; however, they do not contain phonemic annotation time-synced with audio. To fill this gap, the Penn Phonetics Lab Forced Alignment tool will be used to build a rough tagging of phonemes for audio sources with orthographic transcriptions. From this expanded set of training data, successive generations of detectors will be trained, and their relative performance will be compared as in Figure 1.

3.2 Evaluation Criteria

After each model has been designed and trained on a random selection of the corpus, the model will be tested on samples from the corpus it was trained on, as well as other corpora. For each of the implementation approaches, two major metrics will determine the success of the model. Classification accuracy will measures the number of correctly identified phonetic events per the total. Additionally, the rate of false positives will be recorded. These results will be compared against those of other systems.

4. RESEARCH TIMELINE

There are four major milestones in this project. The first entails training individual HTMs to recognize phoneme class transitions in acoustic data. A general method for configuring an HTM for an arbitrary phonetic category will be developed, and detectors for various acoustic cues will be trained independently, initially without articulator information. This stage of the project is projected to be completed before Thanksgiving.

Second, a new generation of detectors will use the University of Wisconsin X-ray Microbeam Speech Production Database, incorporating articulator information in the classification of speech landmarks.

Third, the detectors will be trained and refined on larger, more varied corpora with no accompanying articulatory data. The associations made by the HTM system that previously analyzed articulator data will continue to inform the workings of the model throughout its further training. This portion of the project will employ HMM forced-alignment on orthographically transcribed samples.

The second and third stages of the project can be developed independently, and it is projected that they will be near completion by February.

The final step of this project is to merge the discrete detectors into a single word recognition system. The previously independent HTMs will feed into higher-level nodes that are connected to a lexicon – this will permit both top-down and bottom-up pathways as are believed to exist in the brain. This stage is primarily exploratory, and is contingent on the timely success of previous components.

5. RESOURCE REQUIREMENTS

Throughout the semester, access to the TIMIT and UBDB (Microbeam Database) corpora must be available for training and testing. Additional corpora will be needed later in the project, but can be secured from any of a variety of sources; the audio samples need only be orthographically transcribed.

Additionally, access to staging machines will be required. It is anticipated that these machines will be running the Ubuntu Server operating system, along with python 2.5.4 and wxPython. The HTK speech recognition toolkit, along with the Penn Phonetics Lab Forced Aligner will be essential components in the training pipeline. Finally, the HTMs will be built using the Numenta Platform for Intelligent Computing (NuPIC).

6. REFERENCES

[1] Dileep George and Bobby Jaros. *The HTM Learning Algorithms*. Numenta Inc., March 2007.

[2] Jeff Hawkins and Dileep George. *Hierarchical Temporal Memory; Concepts, Theory, and Terminology*. Numenta Inc., March 2006.

[3] Interspeech. *Spoken Digit Recognition using a Hierarchical Temporal Memory*, September 2008.

[4] Konstantin Markov, Jianwu Dang, and Satoshi Nakamura. Integration of articulatory and spectrum features based on the hybrid hmm/bn modeling framework. *Speech Communication*, 48(2):161 – 175, 2006.

[5] Numenta Inc. *Hierarchical Temporal Memory; Comparison with Existing Models*, March 2007.

[6] Numenta Inc. *Speech Processing with Hierarchical Temporal Memory*, June 2008. Note: Permission to cite pending.

[7] Dan Robinson, Kevin Leung, and Xavier Falco. Spoken language identification with hierarchical temporal memories. December 2009.

[8] L. Sweeney and P. Thompson. Speech perception using real-time phoneme detection: The bebe system. *Relation*, 10(1.135):5493, 2007.