Spiking Neural Networks and the Generalised Hough Transform for Speech Pattern Detection

Jonathan Dennis, Huy Dat Tran, Haizhou Li

Abstract

This paper proposes a novel spiking neural network (SNN) architecture that integrates with the generalised Hough transform (GHT) framework for the task of detecting specific speech patterns such as command words. The idea is that the GHT can model the geometrical distribution of speech information over the wider temporal context, while the SNN is used to learn the discriminative prior weighting in the GHT to provide a spike output indicating a detection decision. The SNN therefore enhances the projection of the GHT from the input acoustic information into the sparse Hough accumulator space for detecting specific sound patterns. Compared using conventional neural network architectures for this task, the GHT-SNN system has the advantage that it does not require a voice activity detection module or an explicit noise model to reject non-target frames. Instead, the output of the SNN is a voltage that is trained to exceed a threshold for positive instances of the sound pattern while remaining below this threshold otherwise, requiring no explicit noise model. Experiments are carried out on the challenging Chalearn gesture recognition task where spoken commands must be detected against variable background noise while rejecting a range of out-of-vocabulary words.

Index Terms: spoken word detection, generalised Hough transform, codebook activation map, spiking neural networks

1. Introduction

Current deep neural network (DNN) architectures in automatic speech recognition (ASR) typically have an input layer capturing a context window of features, followed by multiple hidden layers with non-linear activation functions, combined with a softmax output layer to make a probabilistic classification decision of the underlying features [1–4]. When using such a system for detecting speech patterns in challenging conditions, this architecture has two drawbacks. Firstly, the input context window of features may not be the best approach for capturing the speech information when extending the input context to cover a longer temporal context. Secondly, using a frame-based softmax classification output requires an explicitly trained noise class to capture frames not belonging to the specified targets, which may perform poorly in the presence of variable noise and out-of-vocabulary (OOV) words.

The first of these problems is addressed in this paper using the generalised Hough transform (GHT), which is a sophisticated object detection system from machine vision [5–7]. In our previous works we have successfully utilised machine vision techniques for challenging audio processing tasks [8], including adapting the GHT for both speech phoneme classification [9] and overlapping sound event recognition [10]. The idea is this paper is that the GHT is to learn the two-dimensional geometrical distribution of codebook activations to create a template model of the underlying shape information for each target class. An instance of the target class will then produce a sparse peak in the output if the input codebook distribution is similar to the template learnt in the training.

The second problem is the issue of speech pattern detection in the presence of variable noise and OOV words. In this paper, we propose to integrate the GHT into a spiking neural network (SNN) framework. The idea is that the SNN can learn the prior weighting component of the GHT in a discriminative manner, which enhances the sparse output of the GHT to produce an output spike indicating the detection of a particular sound pattern. We focus on the biologically plausible Tempotron learning approach for this purpose [11, 12]. This uses a cost function that reinforces the strongest output voltage over the reference segment from the positive class until a spike is produced, while penalising negative classes that associate strongly with the given input. This only requires labelling of the target speech patterns in training, and simply learns to produce a low output voltage for non-target speech and noise patterns and hence does not require an explicit noise model.

Experiments are carried out on a challenging spoken word recognition task based on the recent Chalearn multi-modal gesture recognition challenge [13]. In this dataset, the task is to detect a variety of command words which are spoken by a range of speakers against variable background noise and including an unknown number of OOV gesture commands. This highlights the ability of the system to capture the geometrical distribution of the phonetic information in each word using the GHT, and tests the performance of the SNN in detecting the target words while rejecting both noise and OOV words.

2. GHT-SNN System Design

In this section we describe in detail the components of the proposed system. We begin by giving an overview of the three key processing steps, as shown diagrammatically in Fig. 1, followed by a more detailed analysis: (1) codebook activation map (CAP) representation of the spectral information; (2) probabilistic Hough voting using the GHT; and (3) learning a prior output weighting using an SNN to enhance the GHT mapping and produce a spike indicating the detection of a speech pattern.

2.1. Overview of the Proposed System

The framework for the GHT-SNN approach is based on the GHT, which is an extension of the original Hough transform that allows the detection of arbitrary shapes that cannot be represented by an analytical equation [14]. In particular we focus on the probabilistic formulation of the GHT, specifically the implicit shape model [6, 7], that is among the state-of-the-art for...
object detection in image processing. According to this framework, the Hough accumulator score is obtained by adding up the individual voting probabilities over all local features, with a marginalisation over the codebook entries as detailed in [7] to make the solution tractable. For a fixed range of local feature context the formulation is as follows:

\[
S(O_n, x) = \sum_{i=1}^{t} \sum_{j=x-M}^{x+M} p(C_i|f_j)p(x|O_n, C_i, j)p(O_n|C_i, j)
\]

(1)

where \(f, t\) are the feature and location respectively observed at relative location index \(j\) within the context window of size \(M\), \(C_i\) denotes the codebook entry \(i = 1 \ldots t\) such that \(p(C_i|f_j)\) is the posterior probability of the codebook activations, and \(S(O_n, x)\) is the probability score for class \(O_n\) at location \(x\). Note that when considering frame-based features, such as the case with the MFCC features in this paper, the location parameter \(x\) can be replaced by the time-frame index, \(t\).

The first term, \(p(C_i|f_j)\), in the above equation is referred to here as the codebook activation map (CAP), and represents the activation strength of the feature at the current location against a codebook of representative feature patterns. The second term, \(p(t|O_n, C_i, j)\) is the probabilistic Hough vote for a target class position, given the class label and the codebook activations, that is learned by modelling the CAP relative to target class occurrences in the training.

The final term specifies a confidence that the codebook entry \(C_i\) at location \(j\) is really matched on the target class, \(O_n\). This prior weighting is replaced by the SNN as part of our proposed GHT-SNN framework, which can learn a discriminative mapping to enhance the GHT output and produce a spike indicating the detection of a learned speech pattern. This aspect is described later in Section 2.4. Next we provide details of the CAP and probabilistic Hough voting which are important aspects of the framework.

2.2. Codebook Activation Map

The CAP is generated in this paper using an unsupervised Gaussian mixture model (GMM) clustering to form a universal background model (UBM) of the low-level features [9]. During training, the size of the UBM with \(i = 1 \ldots t\) entries is first fixed, before initial clusters are generated using k-means, followed by fine-tuning using the expectation-maximisation algorithm. Denoting the input MFCC features as \(f\) and the codebook as \(C\), the CAP is based on the distance between the feature and codebook, taken here as the log-likelihood of each GMM component as follows:

\[
d(f|C_i) = \log \left[ \mathcal{N}(f; \mu_i, \Sigma_i) \right] = \log \left[ p(f|C_i) \right]
\]

(2)

where \(\mathcal{N}(f; \mu, \Sigma)\) is the Gaussian density with mean, \(\mu\), and covariance, \(\Sigma\). Note that we use \(d(f|C)\) to denote the feature-codebook distance, since the codebook is not limited to a particular clustering scheme.

The CAP is the normalised posterior probability estimates of the codebook activations, achieved through using Bayes’ formula as follows:

\[
p(C_i|f) = \frac{p(f|C_i)p(C_i)}{p(f)}
\]

(3)

Assuming a uniform prior, \(p(C_i)\), and substituting in the distance from above, we get:

\[
p(C_i|f) = \frac{\exp \left( d(f|C_i) \right)}{\sum_{j} \exp \left( d(f|C_j) \right)} \approx \frac{\exp \left( \gamma d(f|C_i) \right)}{\sum_{j} \exp \left( \gamma d(f|C_j) \right)}
\]

(4)

where \(\gamma\) is a positive constant, typically less than one, that is introduced to control how strongly the scores are biased towards the best-matching codebook entry [6].
2.3. Probabilistic Hough Voting

The fundamental contribution of the GHT is the transformation of the feature activations in the CAP into the sparse and separate Hough accumulator space that enables detection of the target object classes [14]. It achieves this by learning a set of voting functions for each class, \( p(t(O_n, C_i, j)) \) from (1), that model the geometrical distribution of the activations of each codebook independently. The geometrical location information, represented by \( j \), is measured relative to a reference point defined for each instance of the class during training. These voting functions then effectively form a template of the codebook activations for each class, such that the accumulator will receive a high score when the geometrical distribution of the observed CAP is similar to that observed during training. As in [7], the geometry modelling of the probabilistic Hough vote is achieved though a binned estimate of \( p(t(O_n, C_i, j)) \) that is generated during training by averaging the codebook activations within a temporal context window centred on the target class locations. This is normalised to sum to one within the context window, such that the value is equal to zero at all locations outside of the context size.

Note that although the GHT is typically interpreted as a feature-level voting scheme, it has previously been observed that the formulation is equivalent to a feature-weighting scheme within a sliding window, given that a fixed range of context is allowed for the features [15]. This makes an interesting comparison with current DNN architectures, where a context window of input features are concatenated to form an input vector representing information from the surrounding frames. In addition, the probabilistic Hough vote in (1) of the GHT is a multiplication and summation in the same way as conventional hidden layer neurons in a DNN. Therefore in our SNN architecture, as shown in Fig. 1, the GHT is implemented as a special layer within the neural network, albeit with linear activation function and fixed weights that are learned according to the geometrical modelling of the CAP as described above.

2.4. Prior Weighting using Spiking Neural Network

The last step of the GHT is a prior weighting that captures how confident we are that the codebook entry \( C_i \) corresponds to the class \( O_n \) as opposed to the rest. Assuming that \( p(O_n|C_i) \) is independent of the relative time-frame location, \( j \), the conventional approach is based on the relative frequency of the codebook entry across the different target classes as follows:

\[
p(O_n|C_i, t) = p(O_n|C_i) \times \frac{p(C_i|O_n)}{p(C_i)}
\]

where \( p(C_i|O_n) \) is the relative frequency of codebook \( C_i \) in target class \( O_n \), while \( P(C_i) \) is the relative frequency across all of the training data.

However, by factoring \( p(O_n|C_i) \) as a separate summation in (1), the equation can be written as:

\[
S(O_n, t) = \sum_{i=1}^{T} p(O_n|C_i) \sum_{j=t-M}^{t+M} p(t(O_n, C_i, j))p(C_i|f_j)
\]

\[
= \sum_{i=1}^{T} w_{i,n} \times v_{i,n}(t) = w^T V_n(t)
\]

where \( V^T = [v_{1,n}^T, v_{2,n}^T, \ldots, v_{I,n}^T] \) is the concatenation across the \( i = 1, 2, \ldots, I \) codebook entries of \( v_{i,n} \), which is the factored multiplication and summation within the context window performed by the first probabilistic Hough voting layer of the SNN architecture.

In the proposed GHT-SNN architecture, the hidden and output layer of the neural network learn the discriminative weighting \( w \) in (6). The chosen SNN system is the Tempotron [11], which is a biologically plausible architecture that learns to produce an output spike representing a detection of a given class. It does this by using a novel cost function that is unlike the conventional cross-entropy cost function for classification, or the mean-square-error cost function for regression. The Tempotron cost function is defined based on the difference between a threshold voltage \( S_{thr} \), and the maximum output voltage over the target class segment \( S(t_{max}) \), while the delta is as follows:

\[
\Delta t_{max} = S(t_{max}) - S_{thr}
\]

Error back-propagation for weight modification is only required for erroneous patterns at the time instance \( t_{max} \), where erroneous patterns are defined as positive patterns with no spike produced, or negative patterns producing a spike. The procedure for error back-propagation through the hidden layer is equivalent to that of a conventional DNN system.

An example of the output from the system, both before and after the enhancement using the SNN, is shown in Fig. 2. It can be seen that the SNN is able to enhance the already sparse output of the probabilistic Hough voting by learning the prior output weighting to increase the discrimination between the target and non-target classes. Note that unlike the conventional GHT, the voting scores for each target class are concatenated together prior to the hidden layer of the SNN, giving the SNN additional information to optimise the weights and increase the discrimination of the target class scores.

Comparing the SNN to a conventional DNN system, the Tempotron cost function in the SNN has two clear advantages. Firstly, it does not require an explicit noise model to be learnt, since the SNN it trained to produce zero output if none of the target patterns are present in the segment. Secondly, it has the advantage that it is only required to produce a single spike to identify the detection of one of the target speech patterns. This allows for much sharper and more sparse weighted mapping to be learnt, compared to a DNN that must learn weights to perform classification of every frame, including an explicit noise model, against a hard pre-defined label.

![Figure 2: Example demonstrating the output scores of the GHT before and after SNN mapping for the example in Fig. 1. It can be seen that the sparsity of the scores is being significantly increased by the SNN in (b), making the peak of the target class in green much easier to discriminate.](image-url)
3. Experimental Evaluation
The dataset chosen for this task is the recent Chalearn multimodal gesture recognition challenge [13]. The task focuses on user independent learning of gestures from multi-modal data, where participants speak and act out one of 20 gestures in the Italian language. Here we utilise only the audio data, such that the task becomes solely a speech pattern detection task of the spoken gestures against variable background noise. For training and development there are a total of 7754 and 3362 manually labelled gestures respectively, while the testing set contains 2742 unsegmented gestures to be detected. The development and testing clips contain additional non-target distractor words which makes the task significantly more challenging as these OOV words need to be rejected by the detection system.

3.1. Experimental Setup
Our proposed GHT-SNN first generates a CAP using a UBM with \( I = 200 \) entries generated through unsupervised clustering of the low-level spectral features. Throughout these experiments, MFCC features with their first and second deltas are used to represent the spectral information in each local time frame. The gamma parameter from (4) is set to \( \gamma = 0.2 \), and the context window size is set at \( M = 50 \), which gives a total context window size of 101 frames that is large enough to capture the distribution of information within each of the target gesture words. Both of these parameters were selected as they were found to produce good results in preliminary experiments and to enable a direct comparison with our previous work in [9].

The CAP is used as input to the SNN, where the first layer is configured with fixed weights that are learned using the GHT, as described in Section 2.3. This is followed by a single hidden layer of size 1000 units, and then an output layer with one spiking neuron for each of the target gesture classes.

During testing, a threshold is set for each of the target neurons such that if that threshold is exceed then this represents a detection at that instance in time. The error metric is the Levenshtein distance [16] between the recognised and true gesture sequence. This distance is equivalent to the word error rate (WER) metric commonly used in ASR, hence this term will be used in these results. The WER is controlled by adjusting the thresholds against the output score of the recognition system to achieve the optimal result.

3.2. Baseline Systems
Several baselines audio-only systems are used here for comparison. In each case, the systems are performing frame-level classification, unlike our proposed GHT-SNN which is performing detection with the spiking neural units. Therefore the baseline systems use a speech detector to segment the continuous audio, and the class scores are taken over the detection window, such that the class with the highest score is chosen as the result. They each require 22 class outputs, with one for each of the target classes plus one for the distractor gestures and silence.

The first baseline is the approach of [17], which uses an energy-based system for gesture detection followed by HMM-GMM training to classify the extracted segments. A 6-state HMM word model with 10 Gaussian mixtures per state is trained for each of the output classes. The second baseline system is our previous work in [9], where we examined the combination of the CAP and GHT with a DNN to perform frame-based classification. The CAP-NN system uses the CAP directly as features for DNN classification, without using the GHT transformation. The CAP-GHT system uses the GHT transformation as in this paper, but the neural network performs frame-based classification with the softmax activation function and cross-entropy cost function, unlike the SNN in this paper. In each case the neural network uses the same configuration as in this paper with 1000 hidden units.

3.3. Results and Discussion
The word error rate (WER) results comparing the different approaches are shown in Table 1. It can be seen that the baseline HMM-GMM system in [17] achieves 31.21% WER, which decreases to 25.24% when fusing the results with those obtained on the video data. The CAP-NN and CAP-GHT methods give a strong improvement over this baseline, with the CAP features integrating well with the DNN, and the GHT giving further improvement to reduce the WER to 16.10%. This highlights the ability of the GHT to learn the geometrical distribution of the activations in the CAP over a long context window to capture the shape of the speech information belonging to each word.

The GHT-SNN proposed in this paper outperforms each of these baseline systems, with the best WER of 13.57% achieved without using the video modality. This demonstrates that the SNN is able to provide a mapping from the sparse GHT output to provide a discriminative detection performance. It also has the advantage that it does not require explicit models for the out-of-vocabulary distractor words or silence, as each spiking neuron acts as an independent detector that is trained to exceed the threshold for when their target speech pattern class is presented. The GHT-SNN result of 13.57% is also close to the best performing system in the challenge, that achieved a WER of 12.76% [18], which further demonstrates the effectiveness of the proposed approach.

4. Conclusion
This paper describes the novel combination of the GHT and spiking neural networks for speech pattern detection. The proposed approach is to use the GHT to learn the geometrical distribution of the activations in the CAP, with the distribution information used as a voting function to accumulate evidence for the target classes. The SNN can then take this sparse output from the GHT and learn a discriminative prior weighting to enhance the separation between the classes, and also does not require the use of explicit noise models or a detector. The results on the challenging Chalearn gesture recognition dataset indicate a significant improvement in WER that is close to the best-performing system that included the video modality.

Table 1: Baseline comparison of word error rate on the Chalearn gesture challenge test set (Xent = cross-entropy).

<table>
<thead>
<tr>
<th>Method</th>
<th>Video</th>
<th>GHT</th>
<th>Neural Network</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM-GMM [17]</td>
<td>No</td>
<td>-</td>
<td>-</td>
<td>31.21</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td>-</td>
<td>-</td>
<td>25.24</td>
</tr>
<tr>
<td>CAP-NN [9]</td>
<td>No</td>
<td>-</td>
<td>DNN (Xent)</td>
<td>17.93</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(Tempotron)</td>
<td>16.10</td>
</tr>
<tr>
<td>GHT-SNN</td>
<td>No</td>
<td>Yes</td>
<td>SNN</td>
<td>13.57</td>
</tr>
<tr>
<td>best system [18]</td>
<td>Yes</td>
<td>-</td>
<td>-</td>
<td>12.76</td>
</tr>
</tbody>
</table>
5. References


