Word Spotting with the Gamma Neural Model

Jose C. Principe, Larry A. Turner
Computational NeuroEngineering Lab
University of Florida, Gainesville, FL 32611

Abstract
This work reports on the use of the focused gamma model as a word spotter. Several modifications in the
original gamma model implementation are tested. The preliminary results show that they improve the original
model, and demonstrate the importance of appropriate structures for feature representation in complex
classification problems such as speech recognition.

Introduction
The time delay neural network (TDNN) [1] is one of the most widely utilized neural network topologies for
speech recognition. The power of TDNN comes from the fact that it utilizes a memory structure that brings
past information to the classification of speech. Unfortunately, the memory structure in TDNN is not very
versatile. It is built around a finite impulse response structure, and as such the size of the memory dictates
how far into the past the memory can reach. Another characteristic is that the signal past is recalled with
maximal resolution, because every sample of the signal up to the length of the FIR is stored. There are alter-
native memory structures that can potentially improve on the linear combiner characteristics. We have
been working with the gamma memory, a feedforward structure with local recursion [2]. The gamma memory
is a generalization of the FIR filter, and it has the ability to trade “reach” for resolution [3]. In this paper we
compare the performance of the gamma memory with an FIR filter for isolated word recognition. We show
that a focused gamma model performs better than a TDNN with the memory layer restricted to the input.
The gamma model not only learns faster the pattern classes, but also achieves a smaller mean square error,
meaning that it learns more about the input pattern classes. In this work we allowed the tap-to-tap resolution
to vary, effectively creating a memory structure that can handle some forms of time warping.

Preprocessing for the Gamma Model
The gamma memory is a recursive system, therefore the processing can be done one sample at a time. As
such, preprocessing models that do not segment speech are very appropriate. Instead of the more tradi-
tional FFT or cepstral coefficients, we built a bank of 14 constant Q bandpass filters (Q=4) in quarter-octave
distribution from 200 to 3,638 Hz [4]. Each filter output is squared, lowpass filtered by a Kaiser window, and
decimated to obtain a segment of 30 samples [4]. Each word is therefore represented as a time frequency
pattern of size 14x30. In our vocabulary the words sizes were different, so the decimation rate (and the cor-
responding filtering) were dynamically set and in the range [50, 90]. A simple amplitude threshold was used
to determine the beginning and end of each word. No further hand alignment was performed. Each pattern
was normalized over the range [-1, 1] [4].

Gamma Model Implementation
Unlike previous experiments with the gamma memory for speech recognition, here we allowed the feed-
back parameter, $\mu_k$, to adapt locally. The gamma memory has an input to tap $k$ transfer function given by:

$$H_{\mu_{0k}}(z) = \prod_{i=1}^{k} H_{\mu_{i-1}}(z) = C^k \prod_{i=1}^{k} \frac{\mu_i}{z - (1 - \mu_i)} \quad k \to [1, N_{\mu}]$$

(1)

$$h_{\gamma_k}(n) = \frac{C^k (n-1)!}{(k-1)! (n-k)!} (1 - \gamma_k)^{k-1} \gamma_k^{n-k} \frac{\mu_i}{(n-k)!} \quad k \to [1, N_{\lambda}]$$

(2)

Here each stage can have a different $\mu_k$, yielding a composite effective memory depth, $S$, and resolution, $R$. 
\[ \mathcal{S}_{\mu 0k} = -z \frac{d}{dz} H_{\mu 0k}(z) \bigg|_{z=1} = -C^k \frac{d}{dz} \prod_{i=1}^{k} \frac{\mu_i}{z - (1 - \mu_i)} \bigg|_{z=1} \]

\[ \mathcal{S}_{\mu i-1i} = C\mu_i^{-1} \quad i \mapsto [1, N_{\mu}] \]

\[ \mathcal{S}_{\mu 0k} = C^k \mu_1^{-1} + C\mathcal{S}_{\mu 1k} = C^{k-1} \mathcal{S}_{\mu 01} + C\mathcal{S}_{\mu 1k} = C^{k-1} \sum_{i=1}^{k} \mathcal{S}_{\mu i-1i} \]

\[ \Re_{\mu 0N_{\mu}} = N_{\mu} \mathcal{S}_{0N_{\mu}}^{-1} = N_{\mu} C^{-N_{\mu} + 1} \left( \sum_{i=1}^{N_{\mu}} \mathcal{S}_{\mu i-1i} \right)^{-1} \]

The value of \( \mu_k \) controls the location of the pole, hence the phase of each stage, which means also the delay. The parameter \( \mu_k \) is learned during training, using real time recurrent learning (RTML) according to:

\[ \psi_k(n) = (1 - \mu_k) \psi_k(n - 1) + C \mu_k \psi_{k-1}(n - 1) \]

\[ \delta_k(n) = \frac{\partial}{\partial \gamma_k} \psi_k(n) = (1 - \mu_k) \delta_k(n - 1) + C \mu_k \delta_{k-1}(n - 1) + \psi_k(n - 1) - C \psi_{k-1}(n - 1) \]

\[ e_{kj}(n) = (d_{kj}(n) - y_{kj}(n)) N_{Fk}^{-1} \]

\[ \mu_k(n) = -\eta \frac{\partial}{\partial \gamma_k} e_{mse}(n) = -\eta \frac{\partial}{\partial \gamma_k} \sum_{j=0}^{N_j-1} e_{kj}^2(n) = \eta \mu \sum_{j=0}^{N_j-1} e_{kj}(n) \delta_{kj}(n) w_{kj}(n) \]

The step size, \( \eta_\mu \), was kept constant at 10^{-3}. In this experiment the number of taps, \( N_{\mu} \), was experimentally set at 10 (see results). Another modification from previous designs is the fact that an amplification factor C was included in each filter. The reason can be found in the attenuation that each stage produces in the input signal. Experimentally we found that a gain of 1.5 per tap keeps the signal potential at the taps, \( \psi_k \), within reasonable values and gives good results (Figure 2).

![Figure 1. Gamma array continuous dispersive delay kernels for k \rightarrow [1, 10].](image)

The classification part of the network is built from a multilayer perceptron. The hidden layer has 24 processing elements, and the output layer has 11 processing elements, one for each class. Straight backpropagation with adaptive step size was utilized to train the spatial weights [4]. This is possible since the recurrent portion of the net is restricted to the input layer.
Local, neural cell-level step sizes, $\eta_k$, were adapted with $N_0$ and $N_U$ fixed at 5 and 10, respectively.

$$
\Delta \eta_k (n) = \begin{cases} 
-0.1\eta_k (n-1) & \left( \sum_{m=0}^{N_0-1} \text{sgn} (\Delta \delta_k (n-m)) \right) \geq N_0 \\
0.1\eta_k (n-1) & \left( \sum_{m=0}^{N_U-1} \text{sgn} (\Delta \delta_k (n-m)) \right) \leq -N_U \\
0 & \text{else}
\end{cases} 
$$

(11)

The cost function utilized in this study was the inverse hyperbolic tangent as proposed by Fahlman [5] with $\epsilon=0.45$ and hyperbolic tangent activations, $\sigma(\beta x)$, in all neural cells limiting the output layer potential range to $[-1, 1]$. Output and hidden layer cell relations are denoted by subscript $O$ and $H$, respectively.

$$
\delta_{O_k} = \tanh^{-1} (\epsilon (P_k^O - O_k^O)) \quad \epsilon \to [0, 0.5] 
$$

(12)

$$
\delta_{H_k} = \left( \sum_{m=0}^{N_{ck}-1} \delta_{O_m} \omega_{mk} \right) \sigma' \left( \sum_{m=0}^{N_{ck}-1} \omega_{k\mu} \psi_{\mu} - \varphi_{H_k} \right) 
$$

(13)

$$
\Delta \omega_{O_{km}, H_{km}} (n) = \eta \delta_{O_{km}, H_{km}} \omega_{km} + \alpha \Delta \omega_{O_{km}, H_{km}} (n-1) \quad \alpha \to [0, 1] 
$$

(14)

This cost function effectively weights the larger errors, and serves as an approximation to the $L^\infty$ norm. Training also utilized momentum set in the range [0.6, 0.9]. After the net was trained, the patterns for each class were propagated through the net and the maximum error for the class stored. This vector was used to set the recognition threshold during testing, and created automatically a class of unrecognized patterns.

Results

A set of 11 words (up, down, left, right, top, bottom, center, home, end, pause, continue) was spoken 10 times by the same person, digitized at 11.025 kHz, and preprocessed as explained above. Figure 2 shows the mean square error during learning for the gamma model and TDNN with the same number of taps. Observe that the gamma model learns faster and more of the input patterns.

![Figure 2. Gamma memory layer ANN vs. TDNN learning curves.](image)

Table 1: Gamma memory layer ANN vs. TDNN performance.

<table>
<thead>
<tr>
<th>memory</th>
<th>epoch</th>
<th>$E_f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIR</td>
<td>700</td>
<td>0.0321</td>
</tr>
<tr>
<td>Gamma</td>
<td>463</td>
<td>$2.81 \times 10^{-8}$</td>
</tr>
</tbody>
</table>
An increase of the memory size of TDNN to 30 did not improve the performance. We explain this result by the added capability of the gamma memory to span a longer time span without increasing the number of degrees of freedom of the net, or the number of weights. For speech recognition it seems that due to the intrinsic variability of the patterns for each word (time warping) there is no point of providing maximal resolution in the memory traces. A coarser sampling makes the system less sensitive to alignment, and so may avoid the need for full blown dynamic warping algorithms. This is particularly the case when the memory structure has the flexibility to choose the tap-to-tap feedback parameter, because it is already creating a distortion in the time axis. Figure 3 depicts the post-convergence effective memory depth surface and intra-channel effective memory depth, respectively. Notice that there are certain channels where the feedback parameter is uniform, but others where it peaks early on in the memory structure. Another interesting experiment was conducted during learning. We restricted the fan in of the frequency channels, \( N_{bk} \), of the hidden layer neural cells. Only three neighborhood channels connect to each hidden layer neural cell, with one shared channel between adjacent cells. This decreased tremendously the number of net weights, (from 3948 to 449 weights) and did not affect appreciably the performance of the net. It seems that local frequency information is sufficient to classify words, by preserving spatio-frequency mappings in the pattern set. Varying the gamma memory size affects the MSE and rate of convergence. Nine kernels seem to be the best compromise depth/resolution for this set of words. A remarkable compression of 3:1 was achieved. A fully-connected trained ANN was tested with the training set with 100% recognition. A similar sparsely-connected ANN performed at 98.5%. Test set performance was 95% and 90% for the fully-connected and the sparsely-connected ANN, respectively. All errors were reported as unrecognized exemplars, due to the vectored threshold technique employed.

![Graph](image-url)

**Figure 3.** Gamma memory layer surface and intra-channel effective memory depth, \( k \rightarrow [1, 10] \).

**References**


