Nonlinear activations for convolutional neural network acoustic models

Clara Fannjiang (clarafj@stanford.edu), Michael Fang (mjfang@stanford.edu)

January 5, 2016

Abstract

Following their triumphs in visual recognition tasks, convolutional neural networks (CNNs) have recently been used to learn the emission probabilities of hidden Markov models in speech recognition. The key distinction of CNNs over deep neural networks (DNNs) is that they leverage translational invariance in the frequency domain, such that weights are shared and there are significantly fewer parameters to train. Since the acoustics of speech indeed display some translational invariance, CNNs could provide more powerful models than DNNs for various speech recognition tasks. Here, we compare the per-frame state classification accuracy of several popular nonlinear activation functions, both sigmoidal and rectified, for a small-scale CNN: the logistic function, the hyperbolic tangent function, the rectified linear unit, the leaky rectified linear unit, and the soft-plus function. We find that the leaky rectified linear unit and soft-plus function perform best by far, which suggests their potential in full-scale CNN acoustic models.

1 Introduction

Since the emergence of the expectation-maximization algorithm for training hidden Markov models (HMMs) nearly four decades ago, most speech recognition systems have used Gaussian mixture models (GMMs) to represent the complex distribution of the acoustic features emitted by an HMM phone state. Given enough Gaussian components, GMMs in theory can model probability densities with arbitrarily high accuracy, which has made these “GMM-HMMs” highly successful. However, GMMs are not efficient for modeling data that lies on non-linear manifolds [3].

Deep neural networks (DNNs) have recently become popular as an upgrade over GMMs in modeling acoustic features. In contrast to GMMs, DNNs are in theory capable of learning arbitrarily complex non-linear representations of data. Indeed, they have shown significant improvements over GMMs, even when given less training data [3].

Convolutional neural networks (CNNs) have displayed recent success in many image processing and computer vision tasks. Unlike DNNs, CNNs have an in-built invariance to
translations in the spectral domain and thus do not rely on carefully centered and normalized data [4]. Whereas DNNs connect each unit to all units in the previous layer, CNNs connect each unit to only a local subset of units from the previous layer. This allows the CNN to both use fewer parameters than the DNN and become sensitive to local features of the data. Translational invariance in speech naturally arises from different vocal pitches and speaking styles, and CNNs have begun to show improvements over traditional GMMs as well as DNNs in some speech processing tasks [7].

Neural networks are constructed from units that are simplified abstractions of the neuron. Each unit takes as input a weighted sum of the outputs from the units feeding into it, then outputs a nonlinear transformation of this input. The choice of this nonlinear activation function can play a key role in the performance of the neural network. Traditionally, sigmoidal functions such as the logistic function or hyperbolic tangent have been used. However, nondifferentiable functions such as rectified linear units (ReLUs) have recently been shown to significantly outperform them in DNNs on LVCSR tasks [5].

Here, we compare the performance of different nonlinear activation functions for a small CNN acoustic model. We characterize the different nonlinear activation functions, discuss the architecture and training method for our CNN, and report and discuss our results.

2 Nonlinear activation functions

We compare the following non-linear activation functions: the hyperbolic tangent \( f(x) = \tanh(x) \), the logistic function \( f(x) = \frac{1}{1+e^{-x}} \), the rectified linear unit (ReLU) \( f(x) = \max(x, 0) \), the leaky rectified linear unit (LReLU) \( f(x) = \max(x, 0.01x) \), and the soft-plus function \( f(x) = \log(1+e^x) \).

Of the two sigmoidal functions, the hyperbolic tangent and logistic function, the hyperbolic tangent has generally shown to provide more robust performance with DNNs [5]. Both, however, suffer from the vanishing gradient problem [1]: when the upper-layer units become nearly saturated, they back-propagate near-zero gradients to lower layers. These “vanishing gradients” can slow training convergence, or cause convergence to a poor local minimum [5]. Rectified linear units do not have this problem, since an activated unit gives a constant gradient of 1. We consider both the hard and leaky variants of ReLUs, which differ in their behavior for activations below 0. The hard ReLU has a zero gradient while the leaky variant has a small non-zero gradient, which may help accelerate learning. The zero gradient of the hard ReLU provides model sparsity, which lends itself to elegant conceptual connections to sparse coding, but may limit the actual performance of the model [2]. We also consider the soft-plus function, which can be understood as a smoothed variant of the ReLU with a non-zero gradient everywhere.
3 CNN architecture and dataset

For proof of concept and tractability, we built a relatively small CNN. Our CNN has a single convolutional layer, followed by a max-pooling layer and two fully connected layers. There are 16 convolved feature maps, whose shared weights correspond to a patch size of 20 (frequency bands). The max-pooling layer has a pool size of 4, and there are 1184 hidden units in each of the fully connected layers.

The softmax layer, as usual for neural network acoustic models, requires the most parameters due to the vast number of HMM senone states. To prioritize convergent training on our small model, we trained and tested on a subset of 1000 randomly selected states out of the full 8986 states in the Switchboard speech corpus. We chose to use the Switchboard corpus due to its highly diverse data: extracted from about 2,400 telephone conversations from 543 speakers, it lessens the risk of overfitting or poor generalization.

4 Training methods

Our objective function $J$ is the standard cross-entropy on the softmax classifier output. We first randomly initialize the parameters, then train the CNN using stochastic gradient descent (SGD) with momentum, where the gradient is found via standard back-propagation. No pre-training techniques were used.

4.1 Momentum, step size, and batch size

Given the objective function $J$, classical gradient descent updates the parameters $W$ at each iteration as follows:

$$W^{k+1} = W^k - \alpha \nabla_W J(W).$$

A common extension of SGD is to add a momentum term with the coefficient $m$, which biases the descent direction toward the past descent directions:

$$\Delta W^{k+1} = \nabla J + m \Delta W^k$$

$$W^{k+1} = W^k - \alpha \Delta W^{k+1}$$

The momentum term helps accelerate progress toward the minimum when many successive gradients point in the same direction. When the gradient frequently changes directions, it acts as a low-pass filter by smoothing out the variability. For each activation function, we searched over momentum coefficients $m \in \{0.3, 0.5, 0.8\}$ and found that $m = 0.5$ gave fastest convergence for each.

We also searched over constant step sizes $\alpha \in \{0.001, 0.003, 0.005, 0.01, 0.5, 0.1\}$ for each activation function, and report results for the step size that gave the fastest convergence for each. These were 0.1 for the logistic function, 0.05 for the tanh function, 0.001 for ReLU, 0.005 for LReLU, and 0.005 for the soft-plus function.

Whereas classical gradient descent evaluates the gradient $\nabla W J(W)$ over the entire training set each iteration, SGD forms a much faster approximation of the descent direction by
<table>
<thead>
<tr>
<th>Activation Function</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sigmoid</td>
<td>40.05%</td>
</tr>
<tr>
<td>Tanh</td>
<td>28.14%</td>
</tr>
<tr>
<td>ReLU</td>
<td>36.60%</td>
</tr>
<tr>
<td>LReLU</td>
<td>49.00%</td>
</tr>
<tr>
<td>Soft-plus</td>
<td>45.34%</td>
</tr>
</tbody>
</table>

Table 1: Final per-frame classification accuracies on the held-out development set of 1,000 frames.

evaluating over just a “batch” of frames each iteration. We used a batch size of 256 per iteration and ran SGD until convergence or 50,000 iterations, whichever came first. Each iteration, we evaluated the objective and per-frame accuracy on a set of 1,000 randomly selected held-out frames. However, note that we show results for LReLU and soft-plus prior to convergence due to time constraints, so we expect those performances to be much better at convergence.

5 Results and discussion

5.1 Final per-frame accuracy

Accuracy is measured as the average percentage of correctly classified senone states, out of 1,000 possible states, per frame, over a held-out set of 1,000 frames. We report the maximum accuracy achieved throughout training in Table 1.

5.2 Objective and per-frame accuracy throughout training

We plot the objective and per-frame accuracy per iteration in Figs. 1 and 2, respectively. LReLU gives the best performance, though it seems that the soft-plus function will reach similar accuracies with further training. Furthermore, neither LReLU nor soft-plus had converged at the time we reported the results, and we expect them to perform even better they approach convergence. Overall, both are clearly advantageous over the sigmoid functions, consistent with previous results [5].

We look forward to seeing what happens as we scale up the model to accommodate the full set of 8,986 senone states. Stay tuned.

6 Acknowledgements

Many thanks to Andrew Maas and Awni Hannun for their guidance on neural network architecture and training, and for providing Switchboard per-frame filter-bank and context features.
Figure 1: Objective value throughout 50,000 iterations of training. Soft-plus and LReLU are currently still training.

References


Figure 2: Development set per-frame accuracy throughout 50,000 iterations of training. Soft-plus and LReLU are currently still training.