Bidirectional LSTM Networks for Improved Phoneme Classification and Recognition

Alex Graves[1], Santiago Fernández[1], Jürgen Schmidhuber[1,2]

[1] IDSIA , Galleria 2, 6928 Manno-Lugano, Switzerland
{alex,santiago,juergen}@idsia.ch
[2] TU Munich, Boltzmannstr. 3, 85748 Garching, Munich, Germany

Abstract. In this paper, we carry out two experiments on the TIMIT speech corpus with bidirectional and unidirectional Long Short Term Memory (LSTM) networks. In the first experiment (framewise phoneme classification) we find that bidirectional LSTM outperforms both unidirectional LSTM and conventional Recurrent Neural Networks (RNNs). In the second (phoneme recognition) we find that a hybrid BLSTM-HMM system improves on an equivalent traditional HMM system, as well as unidirectional LSTM-HMM.

1 Introduction

Because the human articulatory system blurs together adjacent sounds in order to produce them rapidly and smoothly (a process known as co-articulation), contextual information is important to many tasks in speech processing. For example, when classifying a frame of speech data, it helps to look at the frames after it as well as those before — especially if it occurs near the end of a word or segment. In general, recurrent neural networks (RNNs) are well suited to such tasks, where the range of contextual effects is not known in advance. However they do have some limitations: firstly, since they process inputs in temporal order, their outputs tend to be mostly based on *previous* context; secondly they have trouble learning time-dependencies more than a few timesteps long [8]. An elegant solution to the first problem is provided by bidirectional networks [11,1]. In this model, the input is presented forwards and backwards to two separate recurrent nets, both of which are connected to the same output layer. For the second problem, an alternative RNN architecture, LSTM, has been shown to be capable of learning long time-dependencies (see Section 2).

In this paper, we extend our previous work on bidirectional LSTM (BLSTM) [7] with experiments on both framewise phoneme classification and phoneme recognition. For phoneme recognition we use the hybrid approach, combining Hidden Markov Models (HMMs) and RNNs in an iterative training procedure (see Section 3). This gives us an insight into the likely impact of bidirectional training on speech recognition, and also allows us to compare our results directly with a traditional HMM system.

2 LSTM

LSTM [9,6] is an RNN architecture designed to deal with long time-dependencies. It was motivated by an analysis of error flow in existing RNNs [8], which found that long

time lags were inaccessible to existing architectures, because the backpropagated error either blows up or decays exponentially.

An LSTM hidden layer consists of a set of recurrently connected blocks, known as memory blocks . These blocks can be thought of a differentiable version of the memory chips in a digital computer. Each of them contains one or more recurrently connected memory cells and three multiplicative units - the input, output and forget gates - that provide continuous analogues of write, read and reset operations for the cells. More precisely, the input to the cells is multiplied by the activation of the input gate, the output to the net is multiplied by the output gate, and the previous cell values are multiplied by the forget gate. The net can only interact with the cells via the gates.

Some modifications of the original LSTM training algorithm were required for bidirectional LSTM. See [7] for full details and pseudocode.

3 Hybrid LSTM-HMM Phoneme Recognition

Hybrid artificial neural net (ANN)/HMM systems are extensively documented in the literature (see, e.g. [3]). The hybrid approach benefits, on the one hand, from the use of neural networks as estimators of the acoustic probabilities and, on the other hand, from access to higher-level linguistic knowledge, in a unified mathematical framework.

The parameters of the HMM are typically estimated by Viterbi training [10], which also provides new targets (in the form of a new segmentation of the speech signal) to re-train the network. This process is repeated until convergence. Alternatively, Bourlard *et al.* developed an algorithm to increase iteratively the global posterior probability of word sequences [2]. The REMAP algorithm, which is similar to the Expectation-Maximization algorithm, estimates local posterior probabilities that are used as targets to train the network.

In this paper, we implement a hybrid LSTM/HMM system based on Viterbi training compare it to traditional HMMs on the task of phoneme recognition.

4 Experiments

All experiments were carried out on the TIMIT database [5]. TIMIT contain sentences of prompted English speech, accompanied by full phonetic transcripts. It has a lexicon of 61 distinct phonemes. The training and test sets contain 4620 and 1680 utterances respectively. For all experiments we used 5% (184) of the training utterances as a validation set and trained on the rest.

We preprocessed all the audio data into frames using 12 Mel-Frequency Cepstrum Coefficients (MFCCs) from 26 filter-bank channels. We also extracted the log-energy and the first order derivatives of it and the other coefficients, giving a vector of 26 coefficients per frame in total.

4.1 Experiment 1: Framewise Phoneme Classification

Our first experimental task was the classification of frames of speech data into phonemes. The targets were the hand labelled transcriptions provided with the data, and the recorded

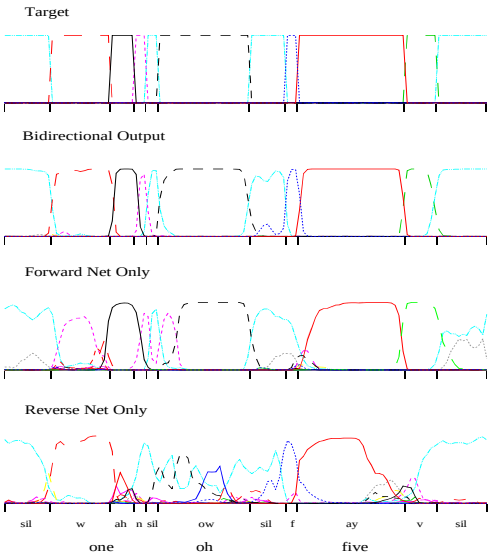

Fig. 1. A bidirectional LSTM net classifying the utterance ”one oh five” from the Numbers95 corpus. The different lines represent the activations (or targets) of different output nodes. The bidirectional output combines the predictions of the forward and reverse subnets; it closely matches the target, indicating accurate classification. To see how the subnets work together, their contributions to the output are plotted separately (“Forward Net Only” and “Reverse Net Only”). As we would expect, the forward net is more accurate. However there are places where its substitutions (‘w’), insertions (at the start of ‘ow’) and deletions (‘f’) are corrected by the reverse net. In addition, both are needed to accurately locate phoneme boundaries, with the reverse net tending to find the starts and the forward net tending to find the ends (‘ay’ is a good example of this).

scores were the percentage of frames in the training and test sets for which the output classification coincided with the target.

We evaluated the following architectures on this task: bidirectional LSTM (BLSTM), unidirectional LSTM (LSTM), bidirectional standard RNN (BRNN), and unidirectional RNN (RNN). For some of the unidirectional nets a delay of 4 timesteps was introduced between the target and the current input — i.e. the net always tried to predict the phoneme of 4 timesteps ago. For BLSTM we also experimented with duration weighted error, where the error injected on each frame is scaled by the duration of the current phoneme.

We used standard RNN topologies for all experiments, with one recurrently connected hidden layer and no direct connections between the input and output layers. The LSTM (BLSTM) hidden layers contained 140 (93) blocks of one cell in each, and the RNN (BRNN) hidden layers contained 275 (185) units. This gave approximately 100,000 weights for each network.

All LSTM blocks had the following activation functions: logistic sigmoids in the range $[-2, 2]$ for the input and output squashing functions of the cell , and in the range $[0, 1]$ for the gates. The non-LSTM net had logistic sigmoid activations in the range $[0, 1]$ in the hidden layer.

All nets were trained with gradient descent (error gradient calculated with Back-propagation Through Time), using a learning rate of 10^{-5} and a momentum of 0.9. At the end of each utterance, weight updates were carried out and network activations were reset to 0.

As is standard for 1 of K classification, the output layers had softmax activations, and the cross entropy objective function was used for training. There were 61 output nodes, one for each phonemes At each frame, the output activations were interpreted as the posterior probabilities of the respective phonemes, given the input signal. The phoneme with highest probability was recorded as the network's classification for that frame.

4.2 Experiment 2: Phoneme Recognition

A traditional HMM was developed with the HTK Speech Recognition Toolkit (http://htk.eng.cam.ac.uk/). Both context independent (mono-phone) and context dependent (tri-phone) models were trained and tested. Both were left-to-right models with three states. Models representing silence (h#, pau, epi) included two extra transitions: from the first to the final state and vice versa, in order to make them more robust. Observation probabilities were modelled by eight Gaussian mixtures.

Sixty-one context-independent models and 5491 tied context-dependent models were used. Context-dependent models for which the left/right context coincide with the central phone were included since they appear in the TIMIT transcription (e.g. "my eyes" is transcribed as /m ay ay z/). During recognition, only sequences of context-dependent models with matching context were allowed.

In order to make a fair comparison of the acoustic modelling capabilities of the traditional and hybrid LSTM/HMM, no linguistic information or probabilities of partial phone sequences were included in the system.

For the hybrid LSTM/HMM system, the following networks (trained in the previous experiment) were used: LSTM with no frame delay, BLSTM and BLSTM trained with weighted error. 61 models of one state each with a self-transition and an exit transition probability were trained using Viterbi-based forced-alignment. Initial estimation of transition and prior probabilities was done using the correct transcription for the training set. Network output probabilities were divided by prior probabilities to obtain likelihoods for the HMM. The system was trained until no improvement was observed or the segmentation of the signal did not change. Due to time limitations, the networks were not re-trained to convergence.

Since the output of both HMM-based systems is a string of phones, a dynamic programming-based string alignment procedure (HTK's HResults tool) was used to compare the output of the system with the correct transcription of the utterance. The accuracy of the system is measured not only by the number of hits, but also takes into account the number of insertions in the output string (accuracy = ((Hits - Insertions) /

Total number of labels) x 100%). For both the traditional and hybrid system, an insertion penalty was estimated and applied during recognition.

5 Results

Table 1. Framewise Phoneme Classification

Network	Training Set	Test Set	Epochs
BLSTM	77.4%	69.8%	21
BRNN	76.0%	69.0%	170
BLSTM Weighted Error	75.7%	68.9%	15
LSTM (4 frame delay)	77.5%	65.5%	33
RNN (4 frame delay)	70.8%	65.1%	144
LSTM (0 frame delay)	70.9%	64.6%	15
RNN (0 frame delay)	69.9%	64.5%	120

Table 2. Phoneme Recognition Accuracy for Traditional HMM and Hybrid LSTM/HMM

System	Number of parameters	Accuracy
Context-independent HMM	80 K	53.7 %
Context-dependent HMM	>600 K	64.4 %
LSTM/HMM	100 K	60.4 %
BLSTM/HMM	100 K	65.7 %
Weighted error BLSTM/HMM	100 K	66.9 %

From Table 1, we can see that bidirectional nets outperformed unidirectional ones in framewise classification. From Table 2 we can also see that for BLSTM this advantage carried over into phoneme recognition.

Overall, the hybrid systems outperformed the equivalent HMM systems on phoneme recognition. Also, for the context dependent HMM, they did so with far fewer trainable parameters.

The LSTM nets were 8 to 10 times faster to train than the standard RNNs, as well as slightly more accurate. They were also considerably more prone to overfitting, as can be seen from the greater difference between their training and test set scores in Table 1. The highest classification score we recorded on the TIMIT training set with a bidirectional LSTM net was 86.4% — almost 17% better than we managed on the test set. This degree of overfitting is remarkable given the high proportion of training frames to weights (20 to 1, for unidirectional LSTM). Clearly, better generalisation would be desirable.

Using duration weighted error slightly decreased the classification performance of BLSTM, but increased its recognition accuracy. This is what we would expect, since its effect is to make short phones as significant to training as longer ones [4].

6 Conclusion

In this paper, we found that bidirectional recurrent neural nets outperformed unidirectional ones in framewise phoneme classification. We also found that LSTM networks were faster and more accurate than conventional RNNs at the same task. Furthermore, we observed that the advantage of bidirectional training carried over into phoneme recognition with hybrid HMM/LSTM systems. With these systems, we recorded better phoneme accuracy than with equivalent traditional HMMs, and did so with fewer parameters. Lastly we improved the phoneme recognition score of BLSTM by using a duration weighted error function.

Acknowledgments

The authors would like to thank Nicole Beringer for her expert advice on linguistics and speech recognition. This work was supported by SNF, grant number 200020-100249.

References

1. P. Baldi, S. Brunak, P. Frasconi, G. Soda, and G. Pollastri. Exploiting the past and the future in protein secondary structure prediction. *BIOINF: Bioinformatics*, 15, 1999.
2. H. Bourlard, Y. Konig, and N. Morgan. REMAP: Recursive estimation and maximization of a posteriori probabilities in connectionist speech recognition. In *Proceedings of Europeech'95*, Madrid, 1995.
3. H.A. Bourlard and N. Morgan. *Connnectionist Speech Recognition: A Hybrid Approach*. Kluwer Academic Publishers, 1994.
4. R. Chen and L. Jamieson. Experiments on the implementation of recurrent neural networks for speech phone recognition. In *Proceedings of the Thirtieth Annual Asilomar Conference on Signals, Systems and Computers*, pages 779–782, 1996.
5. J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, , and N. L. Dahlgren. Darpa timit acoustic phonetic continuous speech corpus cdrom, 1993.
6. F. Gers, N. Schraudolph, and J. Schmidhuber. Learning precise timing with LSTM recurrent networks. *Journal of Machine Learning Research*, 3:115–143, 2002.
7. A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610, 2005.
8. S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In S. C. Kremer and J. F. Kolen, editors, *A Field Guide to Dynamical Recurrent Neural Networks*. IEEE Press, 2001.
9. S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.
10. A. J. Robinson. An application of recurrent nets to phone probability estimation. *IEEE Transactions on Neural Networks*, 5(2):298–305, March 1994.
11. M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45:2673–2681, November 1997.