A Comparison of Deep Neural Network Training Methods for Large Vocabulary Speech Recognition

László Tóth* and Tamás Grósz

MTA-SZTE Research Group on Artificial Intelligence
Hungarian Academy of Sciences and University of Szeged
tothl@inf.u-szeged.hu,groszt@sol.cc.u-szeged.hu

Abstract. The introduction of deep neural networks to acoustic modelling has brought significant improvements in speech recognition accuracy. However, this technology has huge computational costs, even when the algorithms are implemented on graphic processors. Hence, finding the right training algorithm that offers the best performance with the lowest training time is now an active area of research. Here, we compare three methods; namely, the unsupervised pre-training algorithm of Hinton et al., a supervised pre-training method that constructs the network layer-by-layer, and deep rectifier networks, which differ from standard nets in their activation function. We find that the three methods can achieve a similar recognition performance, but have quite different training times. Overall, for the large vocabulary speech recognition task we study here, deep rectifier networks offer the best tradeoff between accuracy and training time.

Keywords: deep neural networks, TIMIT, LVCSR.

1 Introduction

Recently there has been a renewed interest in applying neural networks (ANNs) to speech recognition, thanks to the invention of deep neural nets. As the name suggests, deep neural networks differ from conventional ones in that they consist of several hidden layers, while conventional ANN-based recognizers work with only one hidden layer. The application of a deep structure can provide significant improvements in speech recognition results compared to previously used techniques [1]. However, modifying the network architecture also requires modifications to the training algorithm, because the conventional backpropagation algorithm encounters difficulties when training many-layered feedforward networks [2]. As a solution, Hinton et al. presented a pre-training algorithm that works in an unsupervised fashion [3]. After this pre-training step, the backpropagation algorithm can find a much better local optimum. The first tests of deep networks for speech recognition were performed on the TIMIT database [4], which is much smaller than the corpora routinely used for the training of industrial-scale speech recognizers. Hence, since their invention, a lot of effort has been gone into trying to

* This publication is supported by the European Union and co-funded by the European Social Fund. Project title: Telemedicine-focused research activities in the fields of mathematics, informatics and medical sciences. Project number: TÁMOP-4.2.2.A-11/1/KONV-2012-0073.

I. Habernal and V. Matousek (Eds.): TSD 2013, LNAI 8082, pp. 36–43, 2013.

scale up deep networks to much larger datasets and large vocabulary tasks [5,6,7]. The main problem here is that Hinton's pre-training algorithm is very CPU-intensive, even when implemented on graphic processors (GPUs). Several solutions have been proposed to alleviate or circumvent the computational burden of pre-training, but the search for the optimal training technique is still continuing.

Here, we compare three different technologies for the training of deep networks. One is the original pre-training algorithm of Hinton et al.[3]. It treats the network as a deep belief network (DBN) built out of restricted Bolztmann machines (RBMs), and optimizes an energy-based target function using the contrastive divergence (CD) algorithm. After pre-training, the network has to be trained further using some conventional training method like backpropagation.

The second algorithm is called 'discriminative pre-training' by Seide et al. [5]. This method constructs a deep network by adding one layer at a time, and trains these subnetworks after the addition of each layer. Both the pre-training of the partial nets and the final training of the full network are performed by backpropagation, so no special training algorithm is required.

As for the third method, it is different from the two above in the sense that in this case it is not the training algorithm that is slightly modified, but the neurons themselves. Namely, the usual sigmoid activation function is replaced with the rectifier function $\max(0, x)$. These kinds of neural units have been proposed by Glorot et al., and were successfully applied to image recognition and NLP tasks [8]. Rectified linear units were also found to improve restricted Boltzmann machines [9]. It has been shown recently that a deep rectifier network can attain the same phone recognition performance as that for the pre-trained nets of Mohamed et al. [4], but without the need for any pre-training [10].

Here, we first compare the three methods on the TIMIT database, but the main goal of the paper is to obtain results for a large vocabulary recognition task. For this purpose, we trained a recognition system on a 28-hour speech corpus of Hungarian broadcast news. This recognizer is a hybrid HMM/ANN system [11] that gets the state-level posterior probability values from the neural net, while the decoder is the HDecode program, which is a part of the HTK package [12]. As Hungarian is an agglutinative language, our system runs with a relatively large dictionary of almost five hundred thousand word forms.

2 Training Algorithms for Deep Neural Networks

2.1 DBN Pre-Training

This efficient unsupervised algorithm, first described in [3], can be used for learning the connection weights of a deep belief network (DBN) consisting of several layers of restricted Boltzmann machines (RBMs). As their name implies, RBMs are a variant of Boltzmann machines, with the restriction that their neurons must form a bipartite graph. They have an input layer, representing the features of the given task, a hidden layer which has to learn some representation of the input, and each connection in an RBM must be between a visible unit and a hidden unit. RBMs can be trained using the one-step contrastive divergence (CD) algorithm described in [3]. An RBM assigns

the following energy value to each configuration of visible and hidden state vectors, denoted by v and h, respectively:

$$E(v,h|\Theta) = -\sum_{i=1}^{V}\sum_{j=1}^{H} w_{ij}v_i h_j - \sum_{i=1}^{V} b_i v_i - \sum_{j=1}^{H} a_j h_j. \tag{1}$$

Derived from the gradient of the joint likelihood function of data and labels, the one-step contrastive divergence update rule for the visible-hidden weights is

$$\Delta w_{ij} \propto \langle v_i h_j \rangle_{input} - \langle v_i h_j \rangle_1, \tag{2}$$

where $\langle . \rangle_1$ represents the expectation with respect to the distribution of samples got from running a Gibbs sampler initialized on the data for one full step.

Although RBMs with the energy function of Eq. (1) are suitable for binary data, in speech recognition the acoustic input is typically represented by real-valued feature vectors. For real-valued input vectors, the Gaussian-Bernoulli restricted Boltzmann machine (GRBM) can be used, and it requires making only a minor modification of Eq. (1). The GRBM energy function is given by:

$$E(v,h|\Theta) = \sum_{i=1}^{V} \frac{(v_i - b_i)^2}{2} - \sum_{i=1}^{V}\sum_{j=1}^{H} w_{ij}v_i h_j - \sum_{j=1}^{H} a_j h_j \tag{3}$$

Hinton et al. showed that the weights resulting from the unsupervised pre-training algorithm can be used to initialize the weights of a deep, but otherwise standard, feed-forward neural network. After this initialization step, we simply use the backpropagation algorithm to fine-tune the network weights with respect to a supervised criterion.

2.2 Discriminative Pre-training

'Discriminative pre-training' (DPT) was proposed in [5] as an alternative to DBN pre-training. It is a simple algorithm where first we train a network with one hidden layer to full convergence using backpropagation; then we replace the softmax layer by another randomly initialized hidden layer and a new softmax layer on top, and we train the network again; this process is repeated until we reach the desired number of hidden layers. Seide et al. found that this method gives the best results if one performs only a few iterations of backpropagation in the pre-training phase (instead of training to full convergence) with an unusually large learn rate. In their paper, they concluded that this simple training strategy performs just as well as the much more complicated DBN pre-training method described above [5].

2.3 Deep Rectifier Neural Networks

In the case of the third method it is not the training algorithm, but the neurons that are slightly modified. Instead of the usual sigmoid activation, here we apply the rectifier function $\max(0, x)$ for all hidden neurons [2]. There are two fundamental differences

between the sigmoid and the rectifier functions. One is that the output of rectifier neurons does not saturate as their activity gets higher. Glorot et al. conjecture that this is very important in explaining their good performance in deep nets: because of this linearity, there is no gradient vanishing effect [2]. The other difference is the hard saturation at 0 for negative activity values: because of this, only a subset of neurons are active for a given input. One might suppose that this could harm optimization by blocking gradient backpropagation, but the experimental results do not support this hypothesis. It seems that the hard nonlinearities do no harm as long as the gradient can propagate along some paths.

The main advantage of deep rectifier nets is that they can be trained with the standard backpropagation algorithm, without any pre-training. On the TIMIT database they were found to yield phone recognition results similar to those of sigmoid networks pre-trained with the DBN algorithm [10].

3 Experimental Setup

Here, we report the results of applying the ANN-based recognizers on two databases. The first one is the classic TIMIT database of English sentences, while the second is a corpus of Hungarian broadcast news. On TIMIT quite a lot of phone recognition results are available, so it is good for comparative purposes. However, TIMIT is quite small and usually only phone-level results are reported on it. Hence, our second group of tests on the Hungarian corpus sought to measure the large vocabulary recognition performance of the methods used.

As regards TIMIT, the training set consisted of the standard 3696 'si' and 'sx' sentences, while testing was performed on the core test set (192 sentences). A random 10% of the training set was held out for validation purposes, and this block of data will be referred to as the 'development set'. The scores reported are phone recognition error rates using a phone bigram language model.

The speech data of Hungarian broadcast news was collected from eight Hungarian TV channels. It contains about 28 hours of recordings, from which 22 hours were selected for the training set, 2 hours for the development set and 4 hours for the test set. The language model was created from texts taken from the [origo] news portal (www.origo.hu), from a corpus of about 50 million words. Hungarian is an agglutinative language with a lot of word forms, hence we limited the size of the recognition dictionary to 486982 words by keeping only those words that occurred at least twice in the corpus. The pronunciations of these words were obtained from the 'Hungarian Pronunciation Dictionary' [13]. Based on the [origo] corpus, a trigram language model was built using the language modelling tools of HTK [12].

As for the acoustic features, we applied the standard MFCC coefficients, extracted from 25 ms frames with 10 ms frame skips. We used 13 MFCC coefficients (including the zeroth one), along with the corresponding Δ and $\Delta\Delta$ values. In each case, the neural network was trained on 15 neighboring frames, so the number of inputs to the acoustic model was 585.

Neural networks require a frame-level labelling of the training data. For this purpose, we first trained a standard hidden Markov model (HMM) speech recognizer, again using

the HTK toolkit. For the TIMIT dataset, monophone 3-state models were created, which resulted in 183 states. For the broadcast news dataset, triphone models were constructed, consisting of 2348 tied triphone states in total. The HMM states were then aligned to the training data using forced alignment. These labels served as training targets for the neural nets.

For the recognition process, we applied the decoders of the HTK package. We used HVite for the phone recognition experiments on TIMIT, while the HDecode routine was applied for the large vocabulary recognition tests on the broadcast news task. In both cases the acoustic modeling module of HTK required a slight modification in order to be able to work with the posterior probability values produced by the neural nets. For the TIMIT dataset, the language model weight and the insertion penalty factor were set to 1.0 and 0.0, respectively. With the broadcast news dataset, these meta-parameters were tuned on the development set. Lastly, for a fairness of comparison, the pruning beam width was set to the same value for each network.

3.1 Training Parameters for the Neural Networks

In the case of the DBN-based pre-training method (see Section 2.1), we applied stochastic gradient descent (i.e. backpropagation) training with a mini-batch size of 128. For Gaussian-binary RBMs, we ran 50 epochs with a fixed learning rate of 0.002, while for binary-binary RBMs we used 30 epochs with a learning rate of 0.02. Then, to fine-tune the pre-trained nets, again backpropagation was applied with the same mini-batch size as that used for pre-training. The initial learn rate was set to 0.01, and it was halved after each epoch when the error on the development set increased. During both the pre-training and fine-tuning phases, the learning was accelerated by using a momentum of 0.9 (except for the first epoch of fine-tuning, which did not use the momentum method).

Turning to the discriminative pre-training method (see Section 2.2), the initial learn rate was set to 0.01, and it was halved after each epoch when the error on the development set increased. The learn rate was restored to its initial value of 0.01 after the addition of each layer. Furthermore, we found that using 5 epochs of backpropagation after the introduction of each layer gave the best results. For both the pre-training and fine-tuning phases we used a batch size of 128 and momentum of 0.8 (except for the first epoch). The initial learn rate for the fine-tuning of the full network was again set to 0.01.

The training of deep rectifier nets (see Section 2.3) did not require any pre-training at all. The training of the network was performed using backpropagation with an initial learn rate of 0.001 and a batch size of 128.

4 Experimental Results: TIMIT

Fig. 1 shows the phone recognition error rates obtained on the TIMIT development set and core test set with a varying number of hidden layers, each hidden layer containing 2048 neurons. As can be seen, the three training methods performed very similarly on the test set, the only exception being the case of five hidden layers, where the rectifier net performed slightly better. It also significantly outperformed the other two methods on the development set.

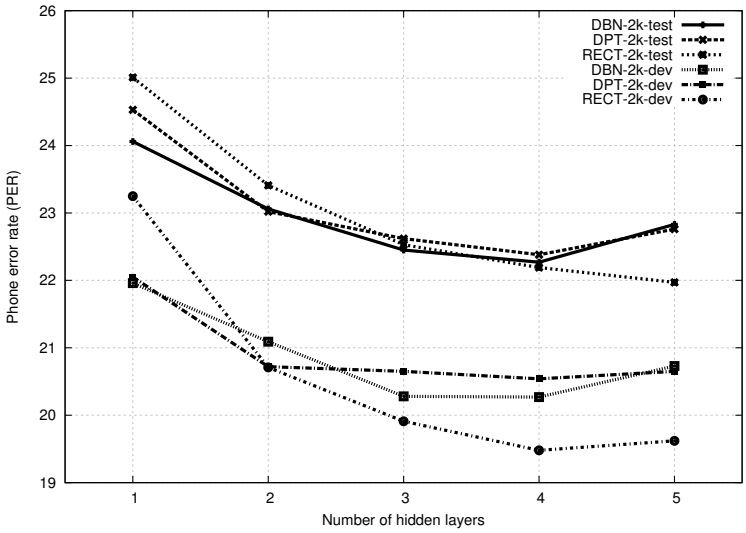

Fig. 1. Phone error rates on TIMIT as a function of the number of hidden layers

Using similar features, training labels and network sizes, Mohamed at al. reported a 22.3% error rate with DBN pre-training [4], while Tóth reported a 21.8% figure with rectifier nets [10]. As our scores fall in the same range, the results also demonstrate the soundness of our methodology.

We mention that a single hidden layer net with the same amount of weights as the best deep net yielded 23.7%. This proves that the better results are due to the deep architecture and not simply due to the increased amount of parameters.

5 Experimental Results: Hungarian Broadcast News

Fig. 2 shows the word error rates got for the large vocabulary broadcast news recognition task. Similar to the TIMIT tests, 2048 neurons were used for each hidden layer, with a varying number of hidden layers. The trends of the results are quite similar to those for the TIMIT database. The error rates seem to saturate at 4-5 hidden layers, and the curves for the three methods run parallel and have only slightly different values. The lowest error rate is attained with the five-layer rectifier network, both on the development and the test sets.

Although their recognition accuracy scores are quite similar, the three methods differ significantly in the training times required. Table 1 shows the training times we measured on an NVIDIA GTX-560 TI graphics card. Evidently, the DBN pre-training algorithm has the largest computational requirements. This algorithm has no clearly defined stopping criterion, and various authors run it with a widely differing number of iterations. The iteration count we applied here (50 for Gaussian RBMs and 30 for binary RBMs) is an average value, and follows the work of Seide et al. [5]. Mohamed applies many more iterations [4], while Jaitly et al. use far fewer iterations [6]. However,

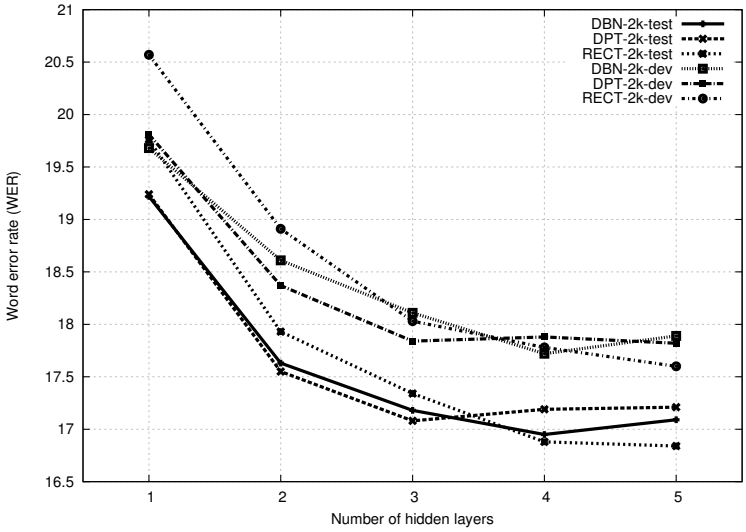

Fig. 2. Word error rates for the broadcast news corpus as a function of the number of hidden layers

no configuration could beat the training time of deep rectifier networks. Discriminative pre-training is also much faster than the DBN-based method, but is still slower than rectifier nets.

Table 1. The training times required by the various methods for 5-layer networks

Training method	Pre-training time	Fine-tuning training time
DBN pre-training	48 hours	14 hours
Discr. pre-training	9 hours	11 hours
Rectifier network	0 hours	14.5 hours

Lastly, although the main goal here was to compare the three deep neural network algorithms, let us now compare the large vocabulary recognition scores with those of a conventional HMM. The same HMM model that was used to generate the training labels attained a word error rate of 20.07% (with maximum likelihood training). However, this result is not fully comparable with those obtained with the hybrid recognizer, because the two systems used different pruning beam widths. Tuning the parameters so that the two systems had a similar real-time factor was also out of the question, as the hybrid model was implemented on a GPU, while the HMM used a normal CPU.

6 Conclusions

It is perhaps no exaggeration to say that deep neural nets have led to a breakthrough is speech recognition. However, they are computationally intensive, and the quest for the

optimal network architecture and training method is still continuing. Here, we compared two training methods and a new type of activation function for deep neural nets, and evaluated them on a large vocabulary recognition task. The three algorithms yielded quite similar recognition performances, but based on the training times deep rectifier networks seem to be the preferred choice. Still, the concept of rectified linear units is quite new, and their behavior requires more theoretical study and practical evaluation. We hope that our study can provide a valuable contribution to this new area of research.

References

1. Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.R., Jaitly, N., et al.: Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. IEEE Signal Processing Magazine 29, 82–97 (2012)
2. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proc. AISTATS, pp. 249–256 (2010)
3. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Computation 18, 1527–1554 (2006)
4. Mohamed, A.R., Dahl, G.E., Hinton, G.: Acoustic modeling using deep belief networks. IEEE Trans. Audio, Speech, and Language Processing 20, 14–22 (2012)
5. Seide, F., Li, G., Chen, X., Yu, D.: Feature engineering in context-dependent deep neural networks for conversational speech transcription. In: Proc. ASRU, pp. 24–29 (2011)
6. Jaitly, N., Nguyen, P., Senior, A., Vanhoucke, V.: Application of pretrained deep neural networks to large vocabulary conversational speech recognition. Technical report, Dept. Comp. Sci., University of Toronto (2012)
7. Dahl, G.E., Yu, D., Deng, L., Acero, A.: Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Trans. Audio, Speech, and Language Processing 20, 30–42 (2012)
8. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier networks. In: Proc. AISTATS, pp. 315–323 (2011)
9. Nair, V., Hinton, G.E.: Rectified linear units improve restricted Boltzmann machines. In: Proc. ICML, pp. 807–814 (2010)
10. Tóth, L.: Phone recognition with deep sparse rectifier neural networks. In: Proc. ICASSP (accepted, in print, 2013)
11. Bourlard, H., Morgan, N.: Connectionist speech recognition: a hybrid approach. Kluwer Academic (1994)
12. Young, S., et al.: The HTK book. Cambridge Univ. Engineering Department (2005)
13. Abari, K., Olaszy, G., Zainkó, C., Kiss, G.: Hungarian pronunciation dictionary on Internet. In: Proc. MSZNY, pp. 223–230 (2006) (in Hungarian)