Improving Deep Neural Networks Using Softplus Units

Hao Zheng, Zhanlei Yang and Wenju Liu
National Laboratory of
Pattern Recognition,
Institute of Automation,
Chinese Academy of Sciences, Beijing,
P.R.China
Email: {hzheng, zhanlei.yang, lwj}@nlpr.ia.ac.cn

Jizhong Liang and Yanpeng Li
Electric Power Research
Institute of ShanXi
Electric Power Company,
China State Grid Corp
{jzhliang.csgd, ypli.csgd}@gmail.com

***Abstract*—Recently, DNNs have achieved great improvement for acoustic modeling in speech recognition tasks. However, it is difficult to train the models well when the depth grows. One main reason is that when training DNNs with traditional sigmoid units, the derivatives damp sharply while back-propagating between layers, which restrict the depth of model especially with insufficient training data. To deal with this problem, some unbounded activation functions have been proposed to preserve sufficient gradients, including ReLU and softplus. Compared with ReLU, the smoothing and nonzero properties of the in gradient makes softplus-based DNNs perform better in both stabilization and performance. However, softplus-based DNNs have been rarely exploited for the phoneme recognition task. In this paper, we explore the use of softplus units for DNNs in acoustic modeling for context-independent phoneme recognition tasks.The revised RBM pre-training and dropout strategy are also applied to improve the performance of softplus units. Experiments show that, the DNNs with softplus units get significantly performance improvement and uses less epochs to get convergence compared to the DNNs trained with standard sigmoid units and ReLUs.**

***Index Terms*—softplus, dropout, deep neural networks, TIMIT**

I. INTRODUCTION

In the past few years, deep neural networks (DNNs) have been introduced to speech recognition tasks and gained a great success. It is believed that the hierarchical nonlinear feature extraction capability of DNNs is one of the most important contributing factors for the success [1]. Thus, in order to extract more semantic features and further to achieve more encouraging results, the models are designed very deeply in general [2].

However, training acoustic model becomes intractable as the depth of DNNs increasing. Besides, the increasing need of the training data, which is called vanishing gradients problem, is a big obstacle to effectively training of deeper networks. The gradients of traditional sigmoid function damp sharply while back-propagating from deep hidden layers to input layer, which restricts the usage of DNNs. Although the negative effect of the vanishing gradient has been eliminated by the RBM based algorithms in the pre-training stage [3], it still exists in the following fine-tuning step. Besides, some unbounded activation functions, such as rectified linear unit (ReLU) and softplus unit, have been proposed to preserve sufficient gradients [3]. ReLU-based DNNs are now widely used in auto speech recognition (ASR) tasks because of its simple format for computation and that it needs less epochs to get convergence [4][5]. However, the discontinuity at the zero point and zero-gradient at negative shaft of input restrict the performance of ReLU. As a smooth version of ReLU, the softplus unit overcomes those shortages to some extent and however, the performance of using softplus for ASR tasks have rarely been studied.

In this paper, we adopt softplus as the activation function for hidden layers of DNNs to avoid extreme small gradients in the DNN training stage. In addition, a revised RBM is applied for pre-training, and the fine-tuning procedure is realized by standard back-propagation algorithm with dropout.

II. THE SOFTPLUS FUNCTION

The widely used sigmoid function often suffers from the problem that, the weight gradients, which are used for updating the weights in hidden layers in the back-propogation algorithm, trend to vanishing gradient problem when models become deeper. To deal with this problem, some unbounded activation functions have been proposed to preserve sufficient gradients, such as the ReLU [6] and softplus [7].

The popular ReLU can be formulated as $f(x) = max(0, x)$. As a smoothing version of the ReLU function, the softplus function is defined as:

$$s(x) = log(1 + e^x). \tag{1}$$

Compared with ReLU, the softplus has a number of advantages. First of all, it is smooth on the definition domain. This property makes the softplus function more stable no matter when being estimated from the positive and negative directions, while ReLU has a discontinuous gradient at point 0. Another advantage is that the softplus unit has a non-zero gradient while the input of unit is negative. Unlike ReLU propagates no gradient in $x < 0$, the softplus function can propagate gradients throughout all real inputs. The softplus unit also outperforms sigmoid unit with the following aspects. The derivative of softplus is a sigmoid function. It means that

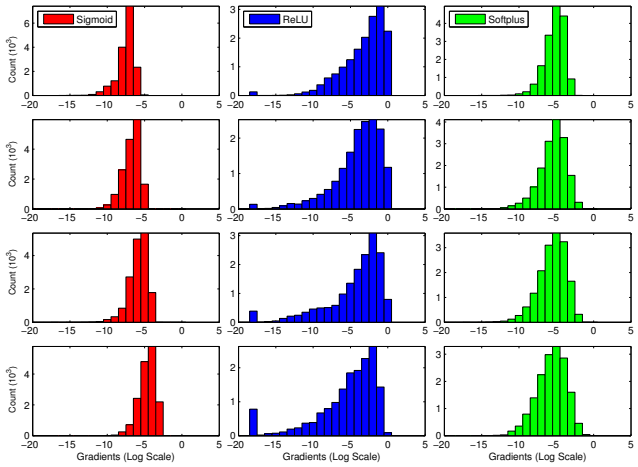

Fig. 1. The counts of logarithmic gradients of weights. Histograms in figures from top to bottom separately denote weights nearest to input layer to nearest to output layer. Dataset: TIMIT; Network configuration: 5 hidden layers and 128 nodes each layer; Learning rate: 8×10^{-4}

the gradient of softplus unit approaches 1 when the input increases, which largely reduces the bad effects of vanishing gradient problem. Additionally, larger gradient benefits dropout fine-tuning. In the dropout procedure, gradient cannot back-propagate through dropped unit [8], which may enhance the impact of gradient damping [3]. Larger gradient is prone to retain a sufficient amount for learning while the smaller one, like sigmoid units, may suffer vanishing gradient more. As a result, softplus unit with dropout will gain more improvement than both ReLU and traditional sigmoid unit.

To better understand the effect of vanishing gradients, Fig. 1 shows the logarithmic histograms pf weights' gradients by using sigmoid, ReLU, and softplus as activation functions respectively in the back-propagating. We use TIMIT as dataset, which is described in section V in detail. The network contains 5 hidden layers and 128 nodes each layer. Logarithmic gradients of weights between hidden layers are counted, only.

From Fig. 1 we can see that the gradients of sigmoid units have a distinct damping, however, those from ReLUs and softplus units keep almost the same throughout 4 layers. Because no gradient is propagated in $x < 0$, a part of gradients with ReLUs are isolated to be 0 (In order to meet the demands of drawing, we set a tiny number 10^{-8} for value 0), which means that weights with these 0 gradients are not updated. It also can be seen that the ReLU and softplus generally produce larger gradients than the traditional sigmoid units.

III. Revised RBM pre-training

A. Traditional RBM pre-training for sigmoid units

In speech recognition tasks, features such as MFCC or PLP generally follow standard Gaussian distributions after cepstral mean and variance normalization (CMVN). To deal with the continuous-valued data and sigmoid units in the hidden layer of DNN, a Gaussian-Bernoulli RBM (GBRBM) [9] is used for pre-training the weights between input and first hidden layer. In GBRBM, each unit is connected to all units in the following layers. It has been shown that the latent variables of a learnt GBRBM can be used as meaningful unsupervised features [9]. The energy function of GBRBM is defined as:

$$E(\mathbf{v}, \mathbf{h}) = \sum_{i \in \mathbf{v}} \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_{j \in \mathbf{h}} c_j h_j - \sum_{i,j} \frac{v_i}{\sigma_i} h_j w_{ij} \quad (2)$$

where σ_i is the standard deviation of the Gaussian distribution for visible unit i, w_{ij} is the weight between the ith visual unit v_i and the jth hidden unit h_j, b_i and c_j are biases of visible unit v_i and hidden unit v_j respectively. The variance of each input unit becomes 1 after CMVN, which means that the σ_i in equation 2 is equal to 1.

B. Revised RBM pre-training for softplus units

In this paper we use softplus units as hidden units instead of sigmoid ones. Gaussian noise with sigmoid variance [4] is used for sampling hidden units, then given visual units, the Gibbs sampling of hidden units in the jth hidden layer is expressed as:

$$h_j = s(x + \delta) \quad (3)$$

where $s(\cdot)$ represents the softplus function, δ is the Gaussian noise which follows a normal distribution with zero mean and $\frac{1}{1+e^x}$ variance, and x denotes the net input to hidden unit which equals to $w_{ij}v_i + b_j$.

Let us put aside the original derivation of this procedure from the RBM formula. This procedure can be considered as a generative model which tries to minimize the error between positive visual inputs v and negative visual inputs $\tilde{v}$ [10]. Since the distributions of softplus units and visual units are quite different, it is strict for connection weights to minimize $\|v - \tilde{v}\|$ by traditional way of $\tilde{v_i} = \sum_j w_{ij} h_j + b_i$. In this paper, we apply a revised RBM procedure in practice. A scaling factor s_j is multiplied with the hidden layer node. This modification recovers the big difference of distributions between Gaussian and softplus units, which helps to reduce the reconstruction error. The reconstruction step is then modified as

$$\tilde{v_i} = \sum_j w_{ij} s_j h_j + b_i. \quad (4)$$

Finally, the hidden units are re-estimated by using the reconstructed inputs $\tilde{v_i}$ in equation 5:

$$\tilde{h_j} = s(\tilde{x} + \delta) \quad (5)$$

where $\tilde{x}$ is the net input to hidden unit with reconstruction units v_i which equals to $w_{ij}\tilde{v_i} + b_j$.

The tuning of the scaling factor s_j is done by taking the gradient direction which minimizes the difference between v and $\tilde{v}$:

$$\triangle s_j = \langle h_j \sum_i (v_i - \tilde{v_i}) w_{ij} \rangle. \quad (6)$$

The revised RBM can be seen as a simple version of the Spike and Slab Restricted Boltzmann Machine (ssRBM)[11]. With multi-dimensional slab variables, the ssRBM becomes

more powerful to model complex densities, such as sharp edges of images in particular. In ssRBM, if we set the dimension of slab variable s, the penal matrix Λ and the scaler α to 1, it is almost the same as revised RBM. We estimate s not with Gibbs sampling, but gradient descent to reduce the computation time. Obviously, the revised RBM performs much faster than ssRBM. Compared to standard RBM, the revised RBM reduces reconstruction error by about 10.5% in average. The classification error reduction is discussed in section V.

IV. DROPOUT

Neural networks with deep size and powerful learning skills can model training data well enough. However, the models are prone to be over-fitted because of the different distributions between training and test sets, especially when there is only a limited amount of labeled training data. In this situation, the relations between units are always strong. To decrease the relations of connected units, the dropout strategy[8] is proposed to improve neural networks by preventing co-adaptation of feature detectors.

The computation for propagation of an L-layer network in training stage can be written as:

$$\mathbf{x}^{(l+1)} = s(\frac{1}{1-r}\mathbf{x}^{(l)} * \mathbf{m}\mathbf{w}^{(l)T} + \mathbf{b}^{(l)}) \;\; 1 \leq l \leq L-1 \quad (7)$$

where $w^{(l)}$ is the weight matrix between $l-1$-th and l-th layers and $b^{(l)}$ is bias vector of l-th layer. $*$ denotes the element-wise multiplication, while r means the probability of dropout, $\mathbf{m}$ is a binary mask with entries drawn i.i.d. from $Bernoulli(1-r)$ indicating which activations are not dropped out. The factor $\frac{1}{1-r}$ is used for scaling the dropping lose during training stage, which ensures that no extra step is needed in test stage.

V. EXPERIMENTS

A. Corpus and configuration

We conduct phone recognition on the TIMIT dataset to systematically evaluate the performance of the proposed approach and other comparison methods. The training set of TIMIT consists of 3696 utterances from 462 speakers with 8 kind of different dialects. For validation purpose, a dataset of 400 utterances from 50 speakers is chosen to be the development set. We also use the test set with 192 utterances spoken by 24 speakers to examine different apporaches. Conventional 40-dimension FBANK features, along with their first and second order derivatives, are used as features of one frame, and the features of 11 consecutive frames are combined as input features of the DNNs. Each dimension of the input features is normalized to have zero mean and unit variance over the whole training set. Mini-batches of size 256 are used for both pre-training and dropout fine-tuning procedure.

B. RBM pre-training

DBNs are pre-trained by the stacked RBM firstly. Visible biases are initialized to zero, and weights to random numbers sampled from a normal distribution $N(0, 0.1)$. The variance of each visible unit is fixed to 1.0. Learning is done by minimizing contrastive divergence (CD). The momentum starts at 0.5 and is increased linearly to 0.9 in 20 epochs. The DBNs are pre-trained by 120 epochs for the first layer and 40 epochs for the others, and the learning rate decreases from 0.008 to 0.001 linearly in 20 epochs. It has been found that ReLUs are more difficult to be pre-trained than Binary units [4]. Therefore, we pre-train the DBNs with ReLUs only 15 epochs for the first layer and 5 epochs for the others.

When applying revised RBM pre-training to DBNs, the parameter s_j in eq.(4) is estimated by the gradient decent algorithm $s^n = r\triangle s + ms^{n-1}$, where r and m denote the learning rate and momentum of the model, respectively.

Table 1. PERs (%) of model pre-trained with different hidden units and pre-training strategies.

	Sigmoid	ReLU	Softplus
No pre-training	23.69	23.80	23.48
traditional RBM	22.45	23.83	22.19
revised RBM	22.31	23.27	**21.73**

C. Fine-tune stage

The cross-entropy discriminative training is done by back-propagation algorithm to fine-tune the parameters. Each of the 61 phone labels is mapped to 3 HMM states, making the dimension of output layer to be 183. The soft-max function is used in output units. The learning rate is initialized with 0.004 and scaled by a factor of 0.5 if the increase of frame accuracy on validation set is less than 0.5%. The maximum number of training iteration is set to 20. As momentum is very tricky which may affect the final results significantly [12], and we just intend to make a fair comparison between different approaches [13] in the experiments, the momentum is not used as a part of our experiments. The experimental results of phone error rates (PERs) gained by DNNs with 4 hidden layers and 2048 nodes in each layer are shown in Table 1. We also carry out dropout fine-tuning. In this stage, the learning rate is initialized with 0.004 and scaled by a factor of 0.9 in each iteration, and the maximum number of training iteration is set to 100. We empirically choose 0.2 as dropping rate after having tested different values of the dropping rate. The PER results of different approaches are shown in Table 2.

From Table 1 we can see that for all 3 activation functions, DNNs with revised RBM pre-training achieve lower PERs than those with traditional RBM pre-training. The DNNs with softplus units get the best performance of PER.

Fig. 2 shows a tuning experiment where we vary the number of layers with different units. Here rRBM refers to the revised RBM in short. To make a fair comparison, the sparse penalty, which can make a strong improvement to DNNs performance, is not used here. As the results demonstrate in Fig. 2, we can see that DNNs with ReLUs get a higher PER than the other units. This is due to the ReLU-based DNNs are much more likely to get over-fitting without constraint training conditions, such as sparse penalty and dropout strategy. Besides no sparse penalty and dropout applied, another reason for poor performance of ReLU-based DNNs is that training ReLU RBMs

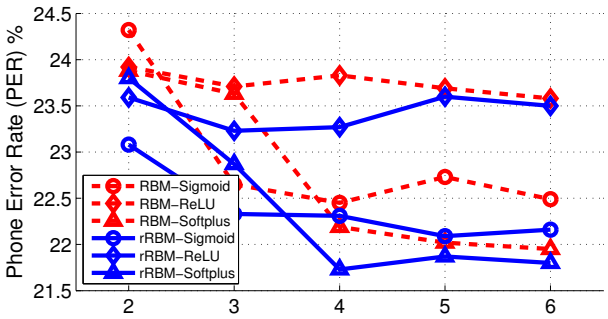

Fig. 2. PERs (%) as a function of the number of hidden layers. Results are shown for the core test set with different kinds of units and pre-training methods.

Table 2. PERs (%) of 3 kinds of units with dropout fine-tuning

	Sigmoid	ReLU	Softplus
dev-test	19.73	19.65	19.62
core-test	21.40	21.15	**21.12**

with CD is more difficult than training binary RBMs [4]. The DNNs with softplus units outperform those with sigmoid units when number of layer is larger than 3. The DNNs pre-trained by revised RBM outperform that pre-trained by standard RBM. The best performance is achieved by softplus-based DNN that pre-trained by revised RBM.

The results in Table 2 show that the DNN with softplus units achieve the best PER on both core-test and dev-test sets. In addition, we find that DNNs with softplus units converges faster than those with ReLUs and sigmoid units, which is about 10.5 epochs while the ones with ReLUs and sigmoid units are 11.17 and 12.33.

VI. RELATION TO PRIOR WORK

Softplus was first applied on DNN by Xavier Glorot et al. [6]. Andrew Senior et al.[14] first used softplus for mobile speech recognition and gained better performance than ReLU and sigmoid function. They found that softplus could converge faster than ReLU and outperformed ReLU on word error rate (WER). Aron Courville et al. [11] first proposed ssRBM and compared it to other kinds of RBMs. Dropout was first proposed by Hinton et al. [8] to reduce the influence of over-fitting. According to their work, we first apply softplus function with revised RBM pre-training and dropout on DNN for phone recognition task.

VII. CONCLUSIONS

In this paper, in order to decrease the affect of vanishing gradients problem when training DNNs, softplus units are employed for phone recognition tasks. Results show that the DNNs with softplus units achieve obviously lower phone error rate than ReLUs and sigmoid units. Compared to RBM, a simple version of ssRBM is more helpful to pre-train softplus-based DNNs, and conventional dropout strategy is also useful for softplus units. In future work we will investigate the effectiveness of softplus units for LVCSR. Another possible future task is to carry out more improvement on softplus units, like sparse penalty and discriminative pre-training. Additionally, we will improve softplus-based DNNs by using optimization criterion, such as sparsity criterion as well as discrimination criterion for pre-training.

ACKNOWLEDGMENT

This research was supported in part by the China National Nature Science Foundation (No.91120303, No.61273267, No.61403370 and No.90820011).

REFERENCES

[1] A.-r. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 14–22, 2012.

[2] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks." in *INTERSPEECH*, 2011, pp. 437–440.

[3] P. Swietojanski, J. Li, and J.-T. Huang, "Investigation of maxout networks for speech recognition," in *Proc IEEE ICASSP*, 2014.

[4] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for lvcsr using rectified linear units and dropout," in *Proc. ICASSP*, 2013.

[5] L. Tóth, "Phone recognition with deep sparse rectifier neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6985–6989.

[6] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier networks," in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics. JMLR W&CP Volume*, vol. 15, 2011, pp. 315–323.

[7] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 807–814.

[8] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[9] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[10] K. Zvi and T.-R. Orith, "Audio event classification using deep neural networks," in *INTERSPEECH*, 2013, pp. 1482–1486.

[11] A. C. Courville, J. Bergstra, and Y. Bengio, "A spike and slab restricted boltzmann machine," in *International Conference on Artificial Intelligence and Statistics*, 2011, pp. 233–241.

[12] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013, pp. 1139–1147.

[13] X. Zheng, Z. Wu, H. Meng, and L. Cai, "Learning dynamic features with neural networks for phoneme recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 2524–2528.

[14] A. Senior and X. Lei, "Fine context, low-rank, softplus deep neural networks for mobile speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 7644–7648.