IMPROVING THE CAPACITY OF VERY DEEP NETWORKS WITH MAXOUT UNITS

Oyebade K. Oyedotun*, Abd El Rahman Shabayek *◊, Djamila Aouada *, Björn Ottersten *

* Interdisciplinary Centre for Security, Reliability and Trust - University of Luxembourg, L-1855 Luxembourg

◊ Computer Science Department, Faculty of Computers and Informatics, Suez Canal University, Egypt

Abstract

Deep neural networks based on rectified linear units (ReLUs) can suffer reduction in representation capacity due to dead units. Moreover, approximating very deep networks trained with dropout at test time can be more inexact due to the several layers of nonlinearities. To address the aforementioned problems, we propose to learn the activation functions of hidden units for very deep networks via maxout. However, maxout units increase the model parameters, and therefore model may suffer from overfitting; we alleviate this problem by employing elastic net regularization (ENR). We perform extensive experiments and reach state-of-the-art results on the USPS and MNIST datasets.

Motivation

- Preserve the representation capacity of very deep networks due to dead ReLUs by learning the activation functions of units via maxout.
- Learn features that are quite linearly separable such that a linear SVM can successfully replace the fully connected layers of the model.

Proposed Approach

- Replace ReLUs with maxout units for improved model capacity.
- Employ ENR for regularizing the S-ResNet.
- Employ Feature Standardization (FS) for improved optimization and regularization of the classifier SVM.

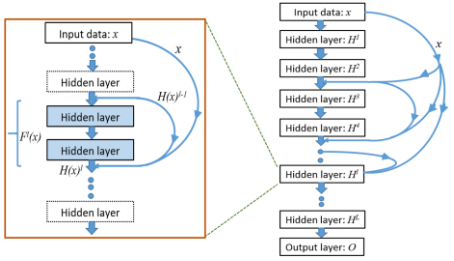

Fig.1. A hypothetical block of two hidden layers

Proposed approach

- Maxout: The output of a maxout unit *k* at layer *l*, $h(x)_k^l$, can be written as follows

$$h(x)_k^l = max_{j\in[1,c]} O(x)_{kj}^l$$

where $O(x)_{kj}^l$ is the output of a linear regressor *j* at layer *l*, and *c* is the number of feature extractors or channels across which we max pool.

- ENR: The elastic net regularization can be seen as a linear combination of *L1*-norm and *L2*-norm regularizations where we optimize for *W* in the following

$$-\arg\min_{W} \sum_{n=1}^{N} \log P\left(y^{(n)}|x^{(n)};W\right) + \lambda_1\|W\|_1 + \lambda_2\|W\|_2^2$$

- FS: Transforms features, $x^{(i)}$, to the same scale, $x_z^{(i)}$, such that features which vary less are not dominated by features that vary more.

$$x_z^{(i)} = \frac{x^{(i)} - \bar{x}^{(i)}}{\sqrt{Var[x^{(i)}]}}$$

- **Learning algorithm**

Algorithm 1: ENR S-ResNet+FS+SVM

1. Set P_s, P_h, λ_1, λ_2, η, α & N_e using a validation set
2. Train S-ResNet via SGD+BN: feed data in mini-batches
3. If stopping condition for (2) is reached, discard the fully connected layers and do (4), else do (2)
4. Save model parameters
5. Forward propagate input data through model
6. Standardize the features obtained from the model's last layer
7. Train a linear SVM on the standardized features

Results

Experimental results for the USPS dataset are in Table 1.

Models	Test error
Invariant vector supports [21]	3.00
Neural network (LeNet) [12]	4.20
Neural network + boosting + data aug. [12]	2.60
Manifold constraint transfer (MCT) [22]	2.99
Evolutionary compact embedding (ECE) [23]	3.90
Polynomial kernel SVM [24]	3.20
Tangent distance + data aug. [13]	2.50
Human performance [13]	2.50
Nearest neighbour [25]	5.60
Residual network (ResNet) - 54 hidden layers [8]	3.34
Baseline: 54 hidden layers S-ResNet [8]	2.69
Ours: 54 layers Maxout S-ResNet+ENR+SVM	**2.34**
Ours: 54 layers Maxout S-ResNet+ENR+FS+SVM	**2.19**

Table 1. Error rate (%) on the USPS dataset

Results

Experiment results for the MNIST dataset are in Table 2.

Models	Test error
Polynomial kernel SVM [24]	0.56
Highway net-32 [7]	0.45
Maxout net [11]	0.45
Deep fried convet [26]	0.71
PCANet [27]	0.62
Network in network (NIN) [14]	0.47
Deeply supervised Network (DSN) [28]	0.39
ConvNet + L-BFGS [29]	0.69
Neural network ensemble + DropConnect [4]	0.52
Neural network ensemble + DropConnect + data aug. [4]	0.21
Stochastic pooling [15]	0.47
Residual network (Resnet) - 54 hidden layers [8]	0.76
Baseline: 54 hidden layers S-ResNet [8]	0.52
Ours: 54 layers Maxout S-ResNet+ENR+SVM	**0.40**
Ours: 54 layers Maxout S-ResNet+ENR+FS+SVM	**0.36**

Table 2. Error rate (%) on the MNIST dataset

Conclusion

In this paper, we propose to learn the activation function of units in the S-ResNet via maxout units since ReLU units can die out in training and impact representation capacity of very deep networks. We employ ENR for further regularization of the S-ResNet model due to increased parameterization based on the maxout units. Our experiments show that we outperform all earlier reported results, including human performance on the USPS dataset. On the MNIST dataset, we obtain very competitive results.

References

[1] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," arXiv preprint arXiv:1302.4389, 2013

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[3] O. K. Oyedotun, A. E. R. Shabayek, D. Aouada et al., "Training very deep networks via residual learning with stochastic input shortcut connections," in International Conference on Neural Information Processing. Springer, 2017, pp. 23–33.

[4] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67(2), 2005, pp. 301–320.

[5] D. Bollegala, "Dynamic feature scaling for online learning of binary classifiers," Knowledge-Based Systems, 129, 2017, pp.97–105.

This work was funded by:

The National Research Fund (**FNR**), Luxembourg, under the project reference R-AGR- 0424-05-D/Bjorn Ottersten.

Fonds National de la Recherche Luxembourg