Available online at www.sciencedirect.com

ScienceDirect

AASRI Procedia 6 (2014) 89 – 94

AASRI Procedia

www.elsevier.com/locate/procedia

2013 2nd AASRI Conference on Computational Intelligence and Bioinformatics

Comparison of Regularization Methods for ImageNet Classification with Deep Convolutional Neural Networks

Evgeny A. Smirnov*, Denis M. Timoshenko, Serge N. Andrianov

Department of Computer Modelling and Multiprocessor Systems, Saint Petersburg State University, Universitetskii prospekt 35, Petergof, Saint-Petersburg 198504, Russian Federation

Abstract

Large and Deep Convolutional Neural Networks achieve good results in image classification tasks, but they need methods to prevent overfitting. In this paper we compare performance of different regularization techniques on ImageNet Large Scale Visual Recognition Challenge 2013. We show empirically that Dropout works better than DropConnect on ImageNet dataset.

Keywords: Deep Neural Networks; Convolutional Neural Networks; Dropout; DropConnect; ImageNet

1. Introduction

Visual object recognition is one of the most challenging problems in Computer Vision, especially in large scale and realistic settings, with high resolution images and thousands of object categories. Until recently neural networks were not widely used for this task, because they need a lot of labeled data and computational power to train. Now, with the advance of fast GPUs and big labeled image datasets, they can be used efficiently, and, moreover, they can beat other methods. Neural networks potentially have fairly large learning

* Corresponding author. Tel.: +7-960-238-19-58
E-mail address: Evgeny.Versus.Smirnov@gmail.com.

doi:10.1016/j.aasri.2014.05.013

capacity, which can be controlled by the number and size of layers, so they can adapt to very big problems. Best results can be obtained with deep neural networks, because depth is essential for learning good internal representations of input data. Large neural networks suffer from the problem of overfitting, so there is a need for powerful regularization techniques like data augmentation (Krizhevsky et al., 2012), Dropout (Hinton et al., 2012) or recently introduced DropConnect (Wan et al., 2013). Another way to improve performance of neural networks is inserting some prior knowledge like awareness of 2D structure of input data. One type of such networks is Convolutional Neural Networks. Because of their structure they have fewer learnable parameters than standard fully-connected neural networks, so they are easier to train and less suffer from overfitting.

The specific contribution of this paper is comparing performance of DropConnect (which, to our knowledge, was evaluated only on small datasets) and Dropout and improved Data Augmentation in large scale settings - on ImageNet Large Scale Visual Recognition Challenge 2013 (ILSVRC2013). We started with architecture close to proposed by (Krizhevsky et al., 2012), which is a winner of ILSVRC2012, and explored how it works with other regularization methods instead of Dropout. Also we proposed several ways to improve results.

2. Related Work

In recent years Convolutional Neural Networks (CNN) gained incredible popularity in many different domains like image classification (Krizhevsky et al., 2012., Zeiler and Fergus, 2013a., Donahue et al., 2013.) object (Toshev et al., 2013) and face (Timoshenko and Grishkin, 2013) detection, speech recognition (Sainath et al., 2013), Bioacoustics (Smirnov, 2013) and others (Frome et al., 2013). Models, based on CNN, improve the state-of-the-art on many important datasets and for some of them also overcome estimated human performance (Wan et al., 2013). One of the most notable successes is their performance on ImageNet dataset (Krizhevsky et al., 2012), where they have no proper competitors and win second year in a row. Big CNNs perform the best, so most of the recent work in Deep Neural Networks focuses on the ways to avoid overfitting to be able to train bigger models (Hinton et al., 2012., Wan et al., 2013., Tomczak, 2013., Ba and Frey, 2013., Goodfellow et al., 2013., Zeiler and Fergus, 2013b., Gulcehre et al., 2013., Wang and Jaja, 2013). Training is usually done on fast GPUs. We used several of these improvements in our work.

3. Challenge

ImageNet dataset has over 15 million labeled high resolution images of 22,000 categories. Annual competition called the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) uses a subset of ImageNet with 1.2 million variable-resolution labeled images of 1000 categories. There are also 50,000 labeled images used for validation and 100,000 unlabeled images used for testing. Two error rates are reported: top-1 and top-5, where the top-5 error rate is the fraction of test images for which the correct label is not among the five labels considered most probable by the model. Winner's result of ILSVRC2012 is 16.4% top-5 error for 5 averaged CNNs (18.2% top-5 error for single CNN). Winner's result of ILSVRC2013 is 11.7% top-5 error.

4. Approach

4.1. Data preprocessing

At the preprocessing stage we rescaled input images such that the shorter side was of length 256, then cropped central patch of size 256x256x3, subtracted the mean activity over the training set from each pixel.

Later, as a part of data augmentation method, we cropped random patches of size 224x224x3 from this central patch. Data preprocessing was done on the CPU, while in parallel performing training on GPU, so it didn't take much computation time. We didn't use PCA in contrast with (Krizhevsky et al., 2012).

4.2. Architecture

We used Deep Convolutional Neural Network architecture, similar to (Krizhevsky et al., 2012), but trained on one GPU. It has 8 trainable layers, the first five of which are convolutional and other three are fully-connected (see Fig. 1). First, second and fifth convolutional layers are followed by max-pooling layers. First and second max-pooling layers are followed by local response normalization layers. We used Rectified Linear Units (ReLU) as neurons. First convolutional layer has 96 kernels of size 11x11x3 with a stride of 4 pixels. The second layer takes as input the max-pooled and response-normalized output of first layer and filters it with 256 kernels of size 5x5x48. The third convolutional layer takes as input the max-pooled and response-normalized output of the second layer and filters it with 384 kernels of size 3x3x256. The fourth layer has 384 kernels of size 3x3x192, and the fifth layer has 256 kernels of size 3x3x192. The fully-connected layers have 4096 neurons each. Max-pooling layers have size of 3x3 and stride of 2. The final layer is 1000-way Softmax.

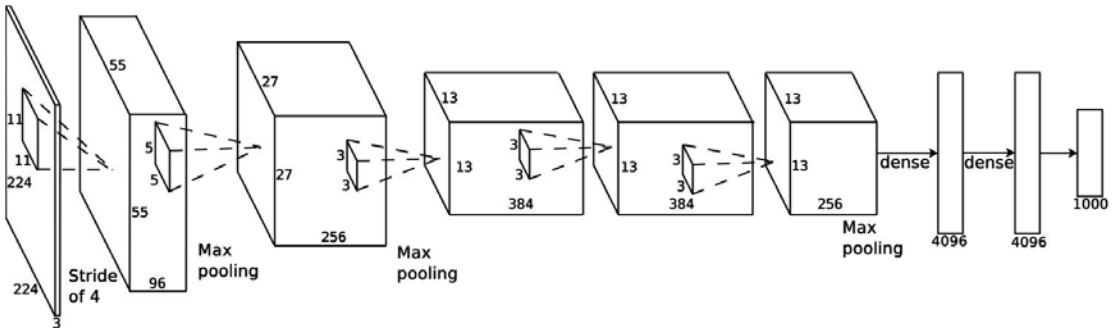

Fig. 1. Deep Convolutional Neural Network architecture

4.3. Training

For training we used stochastic gradient descent with a batch size of 128, momentum of 0.9 and weight decay of 0.0005. We started to train with learning rate of 0.01 for all layers, and then decreased it manually every time when the validation error rate stopped improving. Final learning rate was 0.0001. We trained our network for about 30 epochs. It is 3 times less than in (Krizhevsky et al., 2012), but we didn't have enough time to wait all 90 epochs. At the end we got 96 learned kernels of size 11x11x3 in the first layer as in Fig. 2.

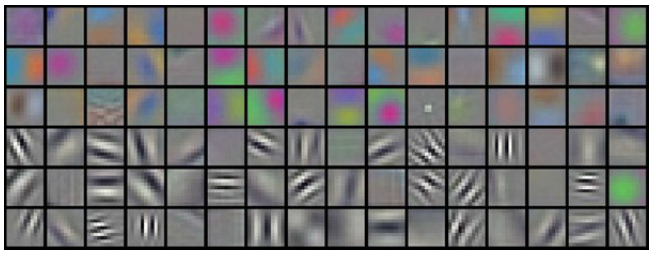

Fig. 2. Convolutional kernels of size 11x11x3 learned by the first convolutional layer

5. Reducing Overfitting

Our network has many parameters, so without good regularization it suffers from overfitting. To avoid it we used several methods. First, we used standard data augmentation: we cropped random patches of size 224x224x3 from input images of size 256x256x3, and then randomly flipped some of them horizontally. This method is described by (Krizhevsky et al., 2012). It helps to increase the size of training dataset. We used it in all our networks. Other methods are Dropout (Hinton et al., 2012), DropConnect (Wan et al., 2013) and Improved Data Augmentation. In order to compare regularization abilities of these methods, we trained three neural networks in parallel. First, we trained for 25 epochs two networks: one with Dropout regularization and another one with DropConnect regularization (each on its own GPU). Then we took best of trained networks (Dropout-trained network), added Improved Data Augmentation and trained it for 5 more epochs on third GPU, while continuing training first two networks without Improved Data Augmentation.

5.1. Dropout

Introduced by (Hinton et al., 2012), this method is now very popular. It consists of setting to zero the output of each hidden neuron in chosen layer with some probability (usually 50%), and is proven to be very effective in reducing overfitting. We trained one of our networks with Dropout on 6's and 7's fully-connected layers with probability of 50%.

5.2. DropConnect

Recently introduced by (Wan et al., 2013), this method is very new. To our knowledge, it was used only on small datasets, and performed good, but not always better than Dropout. It consists of setting to zero not the outputs of neurons, but weights (See Fig. 3) in chosen layer with some probability (usually 50%). We decided to compare its performance on large dataset like ILSVRC2013 and trained second of our networks with DropConnect on 6's and 7's fully-connected layers with probability of 50%.

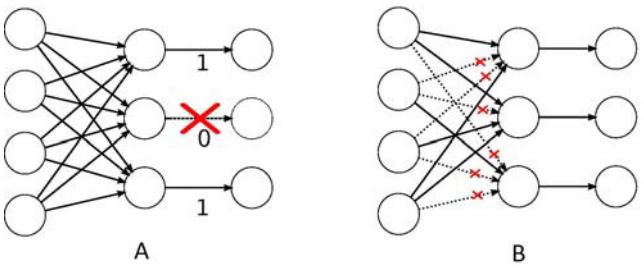

Fig. 3. Regularization methods: (a) Dropout, (b) DropConnect

5.3. Improved Data Augmentation

Instead of just cropping and randomly horizontally flipping patches, we decided to try more sophisticated data augmentation technique – random scaling and rotation. Sadly, we didn't get improvement from this technique – error rate only increased. We think that it is because our network is too small, and we need larger neural network to use this technique efficiently.

6. Results

After we trained three neural networks, we used them to classify validation and testing parts of the dataset. To get better results, we used multiview testing technique, proposed by (Krizhevsky et al., 2012), which consists of averaging predictions of 10 patches for every classified image: 4 corner patches, central patch and their horizontal reflections. Also we averaged predictions of two our best networks. Our results are presented in Table 1.

Table 1. Comparison of error rates on ILSVRC2013 validation and test sets.

Model	Top-1 (val)	Top-5 (val)	Top-5 (test)
CNN + Dropout	45.2%	21.6%	21.5%
CNN + DropConnect	47.6%	23.9%	23.7%
Two models combined	**44.2%**	**21.0%**	**20.9%**

Results on training set for networks with Dropout and DropConnect regularization are roughly the same (35% Top-1), and they were trained pretty identically (except of regularization method), so we think that difference in validation and test set performance shows us empirically that for ImageNet dataset Dropout is better regularization technique than DropConnect. Results of ILSVRC2013 competition are presented in Table 2.

Table 2. Results of ILSVRC2013.

Team	Top-5 (test)
Clarifai (winner)	11.7%
NUS	12.9%
ZF	13.5%
Andrew Howard	13.5%
OverFeat - NYU	14.1%
UvA-Euvision	14.2%
Adobe	15.1%
VGG	15.2%
CognitiveVision	16.0%
decaf	19.2%
IBM Multimedia Team	20.7%
Deep Punx (our team)	**20.9%**
Minerva-MSRA	21.6%
MIL	24.4%
Orange	25.1%
BUPT-Orange	25.1%
Trimps-Soushen1	26.2%

7. Discussion

Our results show that Dropout regularization works better than DropConnect for ImageNet classification task. Also we discovered that our network was too small, and for getting better results we need to use larger network with better regularization techniques. We think that results can be improved by using new methods like DropPart (Tomczak, 2013), standout (Ba and Frey, 2013), maxout (Goodfellow et al., 2013), Stochastic Pooling (Zeiler and Fergus, 2013b), DLSVM (Tang, 2013), Lp Units (Gulcehre et al., 2013) or channel-out (Wang and Jaja, 2013) and some data augmentation techniques.

References

[1] A. Krizhevsky, I. Sutskever, G.E. Hinton. ImageNet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems 25 (NIPS'2012), 2012
[2] 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.
[3] L.Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus. Regularization of Neural Networks using DropConnect. In Proceedings of The 30th International Conference on Machine Learning, 2013.
[4] M.D. Zeiler, R. Fergus. Visualizing and Understanding Convolutional Neural Networks. arXiv preprint arXiv:1311.2901, 2013a.
[5] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. arXiv e-prints, arXiv:1310.1531, 2013.
[6] A. Toshev, C. Szegedy, D. Erhan. Deep Neural Networks for Object Detection. In Advances in Neural Information Processing Systems, 2013.
[7] D. Timoshenko, V. Grishkin. Composite face detection method for automatic moderation of user avatars. Computer Science and Information Technologies (CSIT'13), 2013.
[8] T. Sainath, B. Kingsbury, A. Mohamed, G. E. Dahl, G. Saon, H. Soltau, T. Beran, A. Y. Aravkin, and B. Ramabhadran. Improvements to deep convolutional neural networks for LVCSR. arXiv preprint arXiv:1309.1501, 2013.
[9] E. Smirnov, North Atlantic Right Whale Call Detection with Convolutional Neural Networks. In Proceedings of the 1st Workshop on Machine Learning for Bioacoustics, ICML 2013, Atlanta, USA, 2013.
[10] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, and T. Mikolov. DeViSE: A Deep Visual-Semantic Embedding Model. In Advances in Neural Information Processing Systems (pp. 2121-2129), 2013.
[11] J. M. Tomczak. Prediction of breast cancer recurrence using Classification Restricted Boltzmann Machine with Dropping. arXiv preprint arXiv:1308.6324, 2013.
[12] J. Ba and B. Frey. Adaptive dropout for training deep neural networks. Advances in Neural Information Processing Systems. 2013.
[13] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. In ICML'2013.
[14] M. D. Zeiler and R. Fergus. Stochastic pooling for regualization of deep convolutional neural networks. In ICLR, 2013b.
[15] C. Gulcehre, K. Cho, R. Pascanu and Y. Bengio. Learned-norm pooling for deep neural networks. arXiv preprint arXiv:1311.1780. , 2013.
[16] Q. Wang, J. JaJa. From Maxout to Channel-Out: Encoding Information on Sparse Pathways. arXiv preprint arXiv:1312.1909., 2013.
[17] Y. Tang. Deep Learning using Linear Support Vector Machines. ICML 2013 Workshop on Representation Learning, 2013.