Topic 1 - ImageNet Classification with Deep Convolutional Neural Networks[1]

Ayush Mahant[1], Alex Krizhevsky[2], and Geoffrey E. Hinton[2]

[1]IIITDELHI
[2]University of Toronto

27th September 2020

Abstract

The authors have classified 1000 classes from a collection of 1.2 million high-resolution images by training a deep convolutional deep network, which was considerable. This was an initiative for the ImageNet LSVRC-2010. The authors had to train it a little faster, so they used nonsaturated neurons. An efficient GPU had to be implemented for the five convolution layers and introduced a new method called "dropout" for the three fully connected layers in the network. This was done to reduce overfitting. The authors introduced some variances in this model, thus achieving a winning error rate of 15.3 in 2012, **ImageNet LSVRC**.

1 Introduction

For better current approaches towards object recognition, collecting more massive datasets and preventing overfitting techniques is standard. Recently, a large dataset of millions of labeled images was made possible. **Label Me**[2] had full segmented images, and ***ImageNet***[3] had about 25 million high-resolution images but had a problem of complex object recognition. CNN(Convolutional neural network) has prior knowledge of datasets and controls the nature of images. THE current GPU, have advanced 2D connections and can train large CNN's well along with ImageNet. As stated by the authors, the paper's contributions are- largest CNN was used on ImageNet in **ILSVRC-2010** and 2012 competitions[4]. The new neural network contains more features for improved performance and reduced training time. There still was a problem of overfitting because of the network size.

2 Dataset

ImageNet has 15 million labeled high-resolution images, marked using Amazon's Mechanical Turk crowdsourcing tools. The authors have said that they performed most on ILSVRC 2010. Top 5 error rate is a fraction of test images among the five labels considered the correct model is not a part of. The authors' working system requires constant input dimensionality and a fixed 256 x 256 resolution. Thus specific changes like rescaling had to be done.

3 Architecture

3.1 ReLU Nonlinearity

For input 'x' and output 'f' the model of neutrons is $f(x) = tanh(x)$ or $f(x) = (1 + e^{-x})^{-1}$. Saturation nonlinearities are slower than no saturation nonlinearities $f(x) = max(0, x)$. The authors say that they refer to this nonlinearity as rectified linear units**(ReLU)** Figure Fig.1 shows iterations to reach 25% training error on CIFAR -10 dataset.

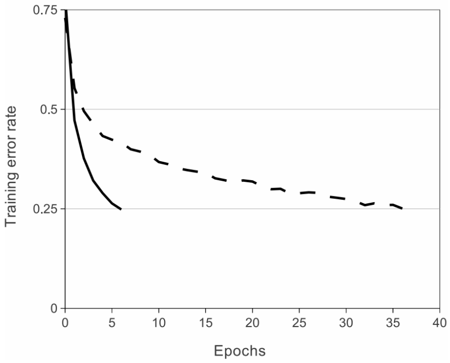

Figure 1: 25% training error on CIFAR

3.2 Training on multiple GNU

GTX 580 GPUs have 3GB of memory, thus restraining the training due to network size. Today GPUs can read and write in one other's memory directly and are suited for GPU parallelization. Also, the authors employed a scheme of communication of GPUs only in certain layers. Level 3 kernels take input from kernel maps in layer 2.

3.3 Local Response Normalization

According to the authors, following normalization helps in generalizing the model. It is given by

$b^i_{x,y} = a^i_{x,y} / (k + \alpha \sum_{j=max(0,i-n/2)}^{min(N-1,i+n/2)} (a^j_{x,y})^2)^{\beta}$

Where $\beta^i_{x.y}$ is the response normalized activity, N is total kernels K,n, alpha and beta are hyper parameters.This model is usually termed as **"brightness normalization"**[5].

3.4 Overlapping pooling

The neighbouring group of neurons have size z x z, and the pooling layer can be considered as a grid of s spaced pixel.When s< z overlapping pooling occurs. For s=2,z=3 it reduces the errors.

3.5 Overall architecture

In Fig. 2 is shown the overall architecture of the CNN.The kernels of second, fourth and fifth convolutional layers are connected to the previous layers. Max pooling follows the layers for response normalization. The first convolution layer filters 224x224x3 input images whose output is input to the second one and this carries on.

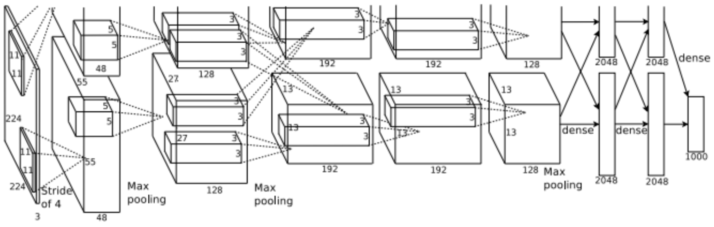

Figure 2: 25% training error on CIFAR

4 Reducing overfitting

There are two ways to reduce overfitting-

4.1 Data augmentation

The schemes the authors used were computationally free as the transformed images in their implementation were in Python on CPU. During testing time 5 224x 224 patches are averaged to them by the network's production.The other alters into RGB channels intensities to each RGB image pixel.

4.2 Dropout

The other technique is **"Dropout"**[6] where every hidden neuron is set to zero with half a probability and reduces adaptation of neurons which is complex.

5 Details of learning

To reduce model's training error the weight decay was used with updated rule for w was-

$$v_{i+1} := 0.9.v_i - 0.0005.\epsilon.w_i - \epsilon.\langle \frac{\partial L}{\partial w} w_i \rangle_{D_i}$$

$$w_{i+1} := w_i + v_{i+1}$$

Here i is the iterative index, v is momemtum, epsilon is learning rate D_i is the derivative wrt to w. The weights in each layer were initialized along with biases where equal learning rate was adjusted.

6 Result

The authors' results on ILSVR-2010 are in Table 1 with error rate- 37.5% for top1 and 17.0 for top5. For ILSVR 2012, Table 2. Displays the results. The author details that error rates of the 2012 competition are not public.

Model	**Top -1**	**Top-5**
Sparse coding[4]	*47.1%*	*28.2%*
SIFT + FVs[7]	*45.7%*	*25.7%*
CNN	37.5%	17.0%

Table 1: ILSVR 2010

Model	**Top -1(val)**	**Top-5(val)**	**Top-5(test)**
SIFT + FVs	-	-	*26.2%*
1 CNN	*40.7%*	*18.2%*	-
5 CNNs	*38.1%*	*16.4%*	16.4%
*1 CNN**	*39.0%*	*16.6%*	-
*7 CNN**	*36.7%*	*15.4%*	15.3%

Table 2: ILSVR 2012

7 Discussion

The author proclaims that supervised learning could prevail over large deep convolutional neural networks. The only problem is with the degrading of the network with one layer removed.The author ends with likeness of using deep nets on video sequences for important information which gets missed in static images.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[2] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, “Labelme: a database and web-based tool for image annotation,” *International journal of computer vision*, vol. 77, no. 1-3, pp. 157–173, 2008.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition.* Ieee, 2009, pp. 248–255.

[4] A. Berg, J. Deng, and L. Fei-Fei, “Large scale visual recognition challenge 2010,” 2010.

[5] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, “What is the best multi-stage architecture for object recognition?” in *2009 IEEE 12th international conference on computer vision.* IEEE, 2009, pp. 2146–2153.

[6] 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.

[7] J. Sánchez and F. Perronnin, “High-dimensional signature compression for large-scale image classification,” in *CVPR 2011.* IEEE, 2011, pp. 1665–1672.