ImageNet Winning CNN Architectures – A Review

Rajat Vikram Singh
rajats@andrew.cmu.edu

I. Introduction

Since a convolutional neural network won the ImageNet challenge in 2012, research in CNNs has proliferated in an attempt to improve them with progress being made every year. In this report I will be discussing the major improvements made in CNNs since the advent of modern CNNs. I will be briefly discussing some of the path-breaking approaches and the value it adds to the previous approaches.

II. ImageNet Challenge and CNN Architectures

Since 2010, the annual ImageNet Large Scale Visual Recognition Challenge (ILSVCR) [1], commonly called the ImageNet challenge, is a competition where research teams submit programs that classify and detect objects and scenes. Over the years, various approaches and architectures have been used to compete in the ImageNet challenge and every year many new and exciting architectures make it to the competition. I will be covering the winners of the ImageNet challenge from 2012 to 2015 in this report.

III. AlexNet

AlexNet [2] is considered to be the break-through paper which rose the interest in CNNs when it won the ImageNet challenge of 2012. AlexNet is a deep CNN trained on ImageNet and outperformed all the entries that year. It was a major improvement with the next best entry getting only 26.2% top 5 test error rate. Compared to modern architectures, a relatively simple layout was used in this paper. The architecture from their paper is as follows:

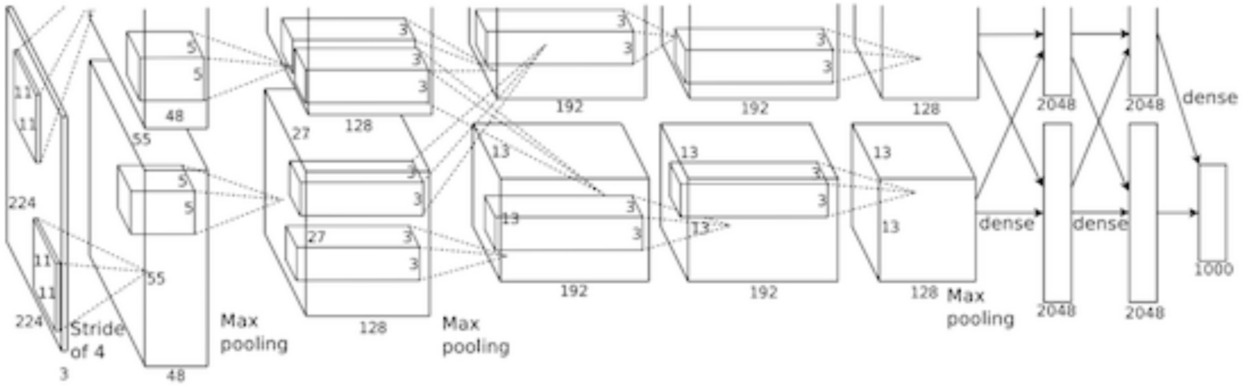


Fig. 1: AlexNet architecture

The network was made up of 5 conv layers, max-pooling layers, dropout layers, and 3 fully connected layers at the end. AlexNet used ReLU for the nonlinearity functions, which they found to decrease training time because ReLUs are much faster than using tanh functions. They also did image translations, horizontal reflections, and patch extractions as a way of augmenting the data before using it to train their network. They also used dropout layers to prevent over-fitting to the training data. They used a batch stochastic gradient descent optimization. Their implementation was trained on GPUs for five to six days.

IV. ZF Net

The ZF paper [3] has a slightly modified AlexNet model which gives better accuracy. The paper also describes a very interesting way of visualizing feature maps (deconvnets). The architecture of the ZF Net as described in their paper is as follows:

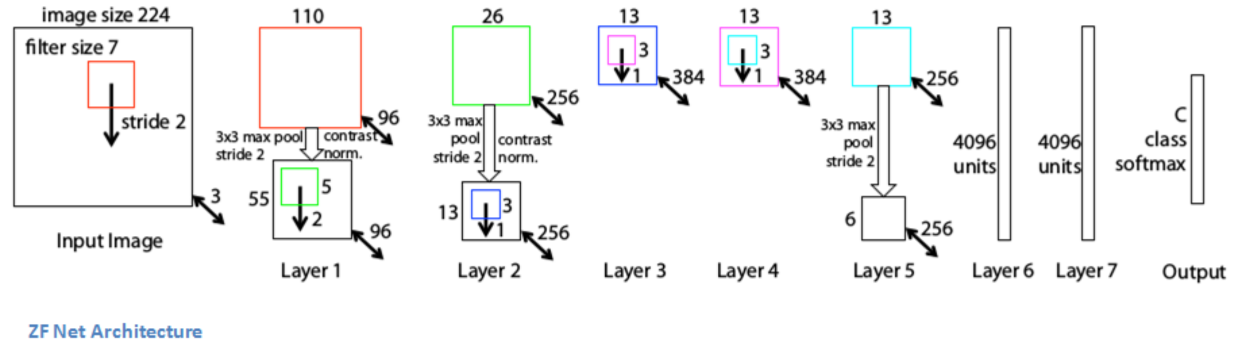


Fig. 2 – ZF Net Architecture

ZF Net used 1.3 million images for training, compared to 15 million images used by AlexNet. One major difference in the approaches was that ZF Net used 7x7 sized filters whereas AlexNet used 11x11 filters. The intuition behind this is that by using bigger filters we were losing a lot of pixel information, which we can retain by having smaller filter sizes in the earlier conv layers. The number of filters increase as we go deeper. This network also used ReLUs for their activation and trained using batch stochastic gradient descent. It trained on a GPU for twelve days. Another interesting approach from the paper was this idea of a DeConvNet which can be used to see which image pixels excite each filter and provides great intuition in how CNNs work.

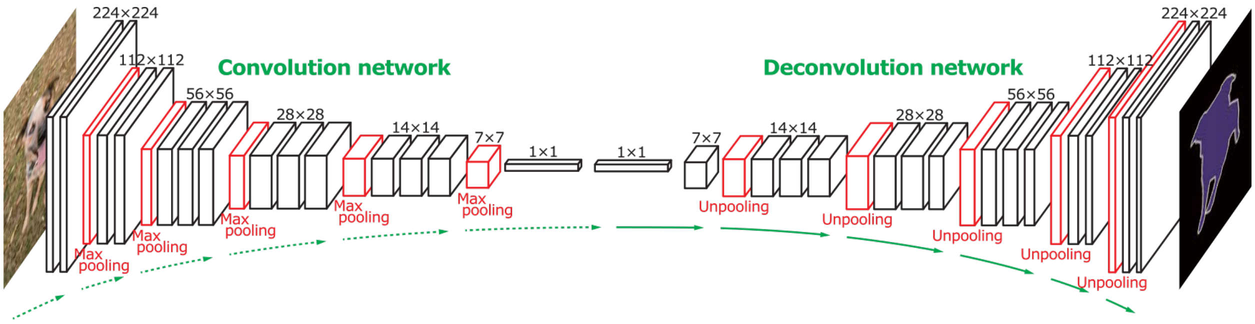
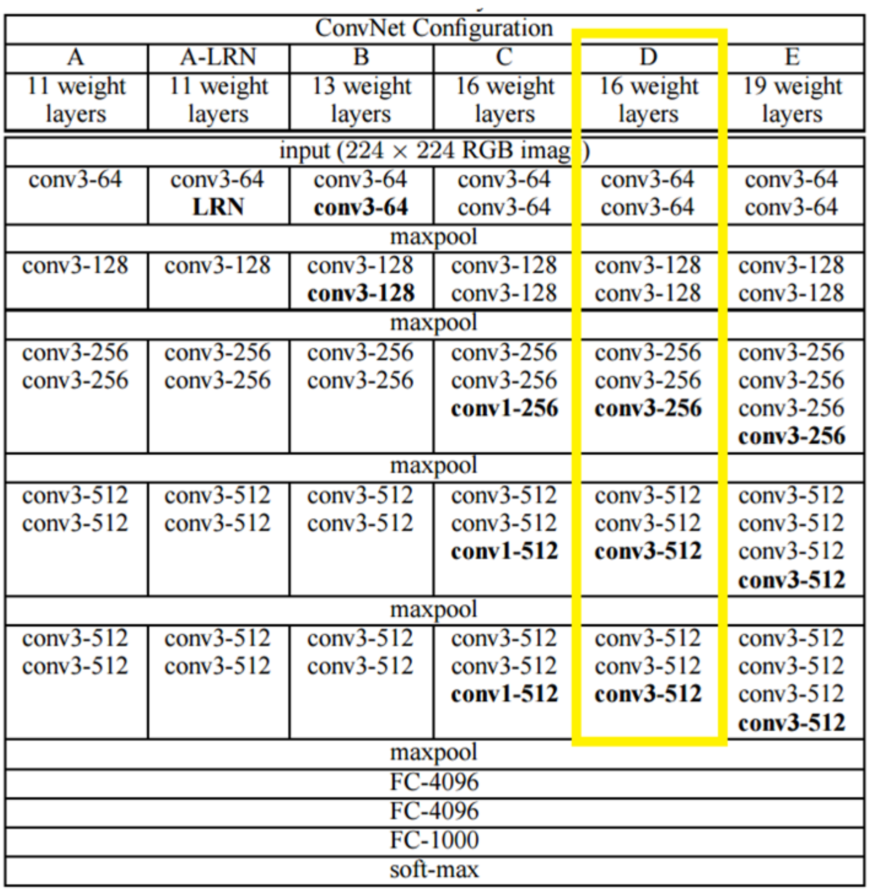


Fig. 3 – Visualization of a DeConv Net

V. VGG Net

VGG Net [4] was a technique proposed for the ImageNet challenge of 2013. VGG Net didn't win the ImageNet 2013 challenge but it is still used by many people because it was a simple architecture based on the AlexNet type architecture. The architecture is described as below:



ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

The 6 different architecures of VGG Net. Configuration D produced the best results

Fig. 4 – VGG Net – All the approaches tried. Column D gives the best performing architecture.

VGG Net used 3x3 filters compared to 11x11 filters in AlexNet and 7x7 in ZF Net. The authors give the intuition behind this that having two consecutive 2 consecutive 3x3 filters gives an effective receptive field of 5x5, and 3 – 3x3 filters give a receptive field of 7x7 filters, but using this we can use a far less number of hyper-parameters to be trained in the network. As you may notice from the architecture the number of filters double after every max-pooling operation. As a data augmentation technique scale jittering was used. Trained with batch gradient descent and used RelUs. Trained on 4 GPUs for two to three weeks.

VI. GoogLeNet

With submissions like VGG Net ImageNet Challenge 2014 had many great submissions, but the winner of them all was Google's GoogLeNet [5] (The name 'GooLeNet' is a tribute to the works of Yann LeCun in his LeNet [6], widely considered to be the first use of modern CNNs). The GoogLeNet architecture is visually represented as follows:

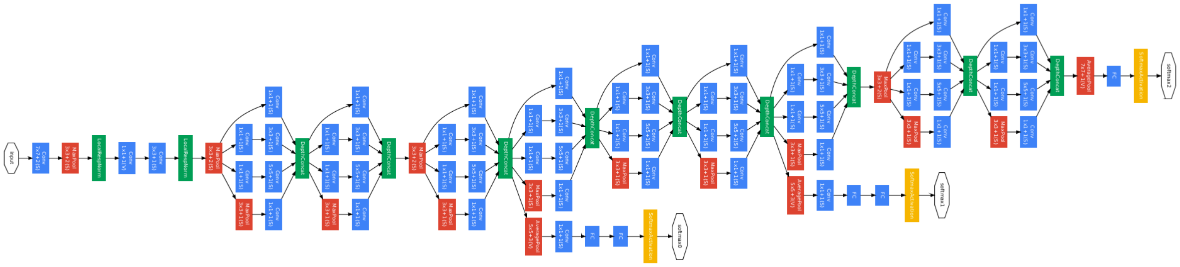


Fig. 5 - GoogLe Net architecture

Inception module:

GoogLeNet proposed something called the inception modules. If you look at the architecture, you can notice some skip connections in the network essentially forming a mini module and that module is repeated throughout the network. Google called this module an inception module. The details of the module are as follows:

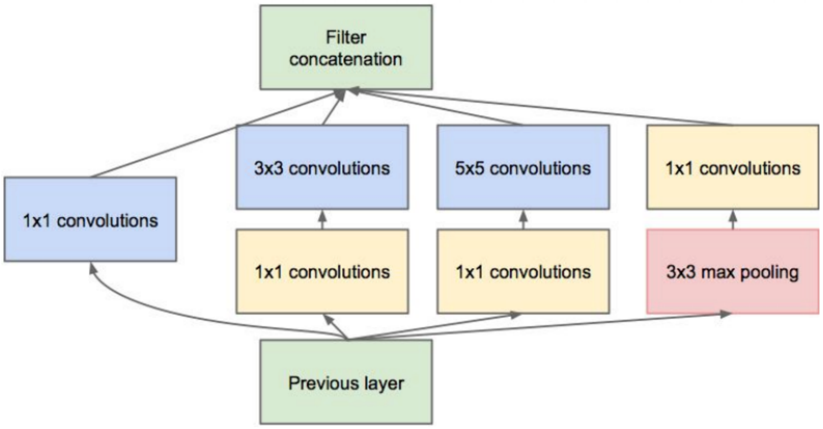


Fig. 6 - Inception module

GoogLeNet uses 9 inception module and it eliminates all fully connected layers using average pooling to go from 7x7x1024 to 1x1x1024. This saves a lot of parameters. As a form of data augmentation, multiple crops of the same image were created and the network was trained on it. Training took less than a week with few high-end GPUs.

VII. Microsoft ResNet

The last CNN architecture I'll discuss here is the Microsoft ResNet (residual network) [7] which won the 2015 ImageNet challenge. The architecture of this CNN is as follows:

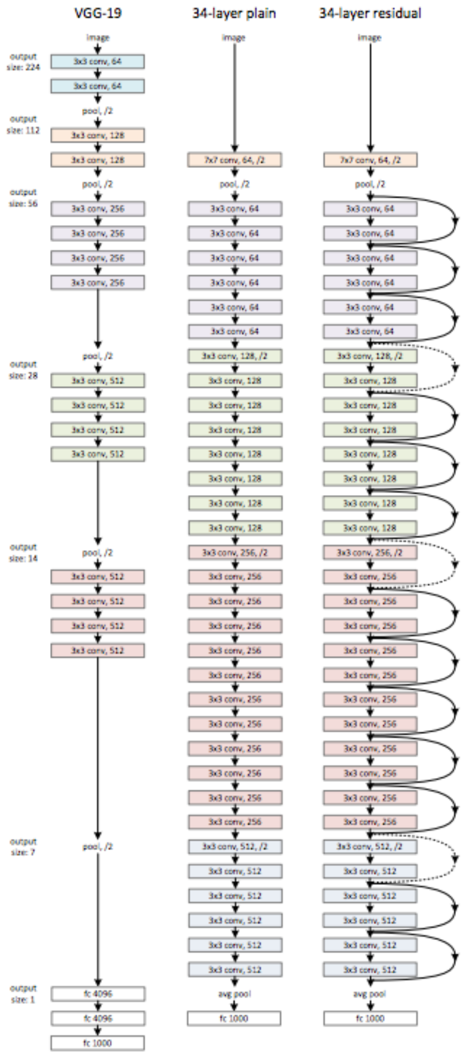


Fig. 7 - Microsoft ResNet visualization compared to a VGG Net

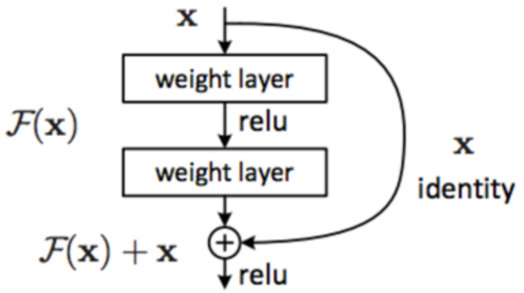


Figure 2. Residual learning: a building block.

Fig. 8 - Residual Block in a ResNet

There are 152 layers in the Microsoft ResNet. The authors showed empirically that if you keep on adding layers the error rate should keep on decreasing in contrast to “plain nets” where adding a few layers resulted in higher training and test errors. It took two to three weeks to train it on an 8 GPU machine. One intuitive reason why residual blocks improve classification is the direct step from one layer to the next and intuitively using all these skip steps form a gradient highway where the gradients computed can directly affect the weights in the first layer making updates have more effect.

VIII. References

[1] Deng, Jia, et al. "Imagenet: A large-scale hierarchical image database." *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009.

[2] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.

[3] Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." *European Conference on Computer Vision*. Springer International Publishing, 2014.

[4] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556*(2014).

[5] Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015.

[6] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998d). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324.

[7] He, Kaiming, et al. "Deep residual learning for image recognition." *arXiv preprint arXiv:1512.03385* (2015).

[8] CS231n Video Lectures: https://www.youtube.com/playlist?list=PLLvH2FwAQhnpj1WEB-jHmPuUeQ8mX-XXG (retrieved: Oct 25th, 2016)

[9] Deep Learning Slides: http://www.slideshare.net/holbertonschool/deep-learning-class-2-by-louis-monier (retrieved: Oct 25th, 2016)