Convolutional Neural Networks: A Brief History of their Evolution

Brajesh Kumar · Follow

Published in AppyHigh Blog · 8 min read · Aug 31, 2021

In the world of deep learning, Convolutional Neural Network (CNN) is a class of artificial neural network, most commonly used for image analysis. Since inception, CNN architectures have gone through rapid evolution and in recent years have achieved results which were previously considered possible only via human execution/intervention. Depending on the task at hand, and the corresponding constraints, a wide variety of architectures are available today. These are too deep to be completely visualized and are often treated as black boxes. But were they always like that? Isn't it interesting to delve down into the history of CNN architectures? Tie your seatbelts for a quick trip into this history.

1. Neocognitron (1980)

Neocognitron was the first architecture of its kind, perhaps the earliest precursor of CNNs. The concepts of feature extraction, pooling layers, and using convolution in a neural network were introduced and finally recognition or classification at the end was proposed in the Neocognitron (paper). The structure of the network was inspired by that of the visual nervous system of vertebrates. In the whole network, with its alternate layers of S-cells (simple cells or lower order hypercomplex cells) and C-cells (complex cells or higher order hypercomplex cells), the process of feature-extraction by S-cells and toleration of positional shift by C-cells was repeated. During this process, local features extracted in lower stages are gradually integrated into more global features. It was used for handwritten (Japanese) character recognition and other pattern recognition tasks, and further paved the way for convolutional neural networks.

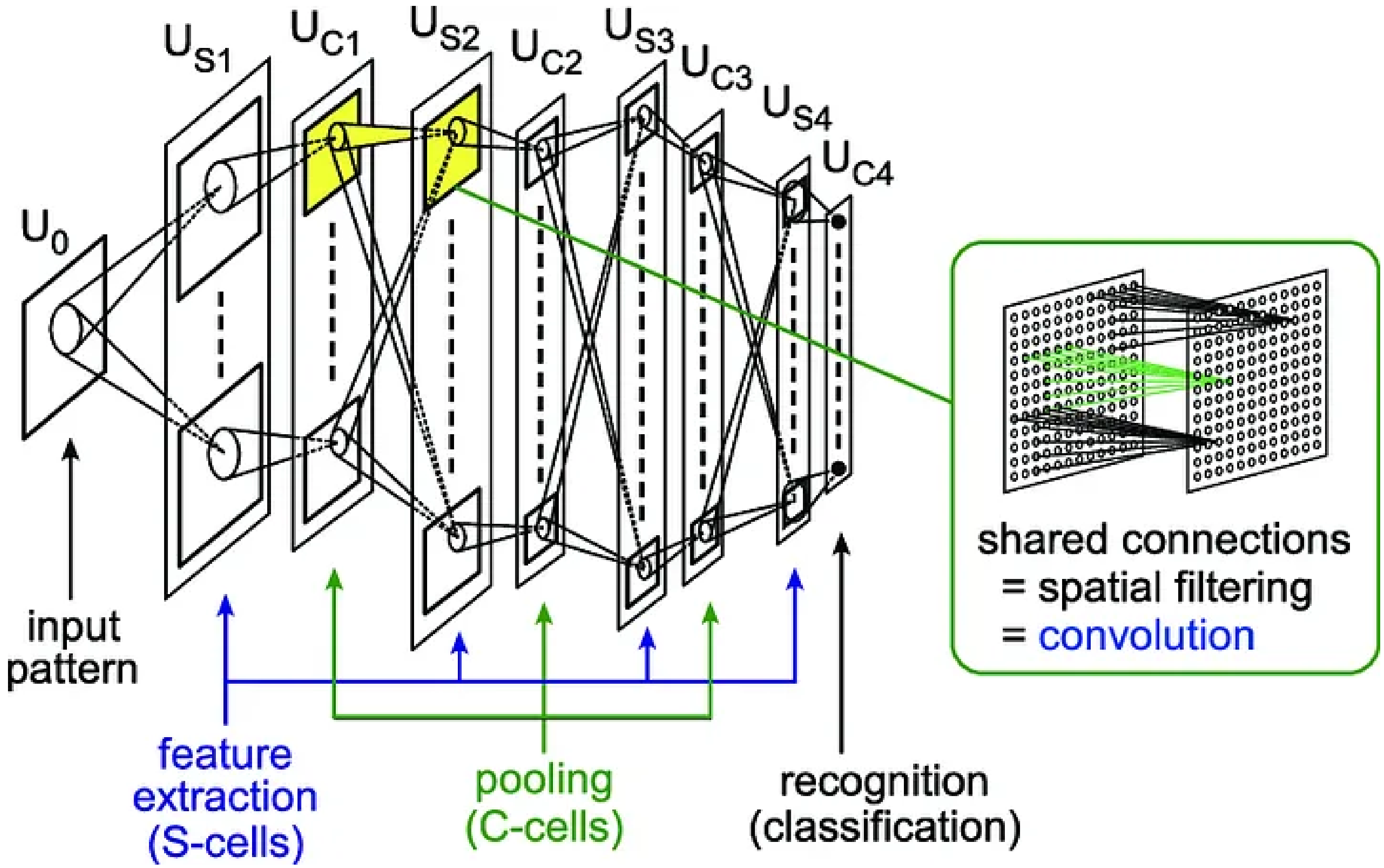


Fig. 1: The architecture of Neocognitron

2. LeNet-5 (1989–1998)

The name convolutional neural networks actually originated with the design of the LeNet by Yann LeCun and team (paper). It was largely developed between 1989 and 1998 for the handwritten digit recognition task.

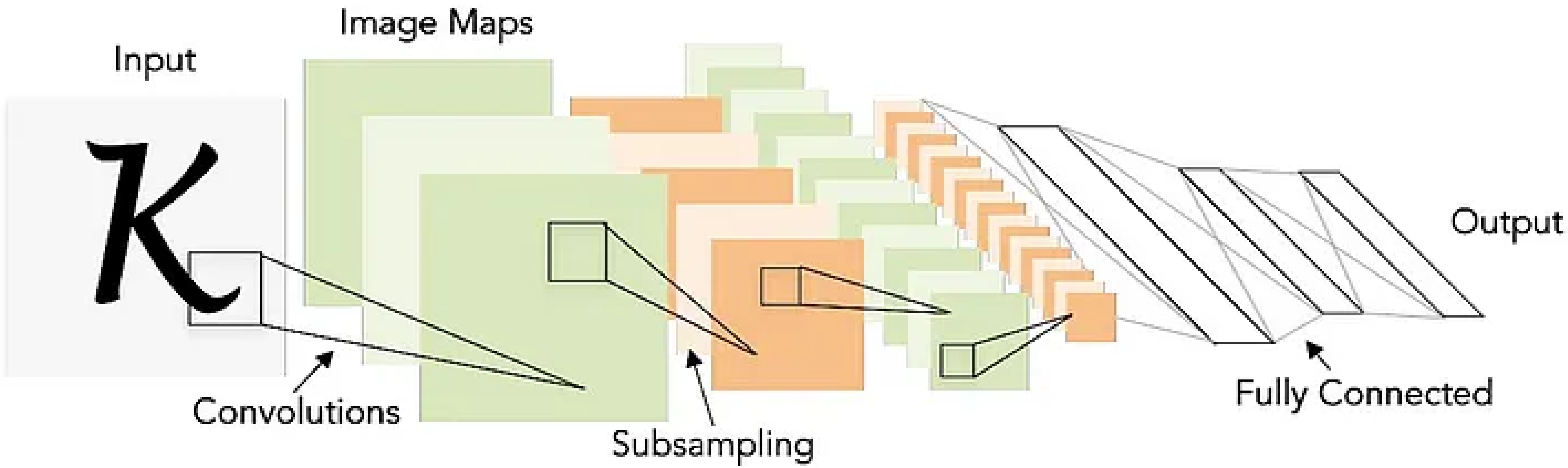


Fig. 2: The architecture of LeNet-5 [7]

The overall architecture was [CONV-POOL-CONV-POOL-**FC**-**FC**]. It used 5x5 convolution filters with a strike of 1. The pooling (subsampling) layers were 2x2 with a stride of 2. It has about 60 K parameters.

The credit for newer architectures of CNNs goes to ImageNet (a dataset) classification challenge named 'ImageNet large scale visual recognition challenge (ILSVRC)'. It was started in 2010 which led to a significant effort across researchers to benchmark their machine learning and computer vision models, in particular for image classification, on a common dataset. Performance was measured in Top-1 error and Top-5 error. In 2010, the winning error rate was 28.2% and it was done without neural networks. In 2011 researchers improved the score from 28.2% to 25.8% error rate. Fig. 3 shows all the winners with the corresponding errors as bar.

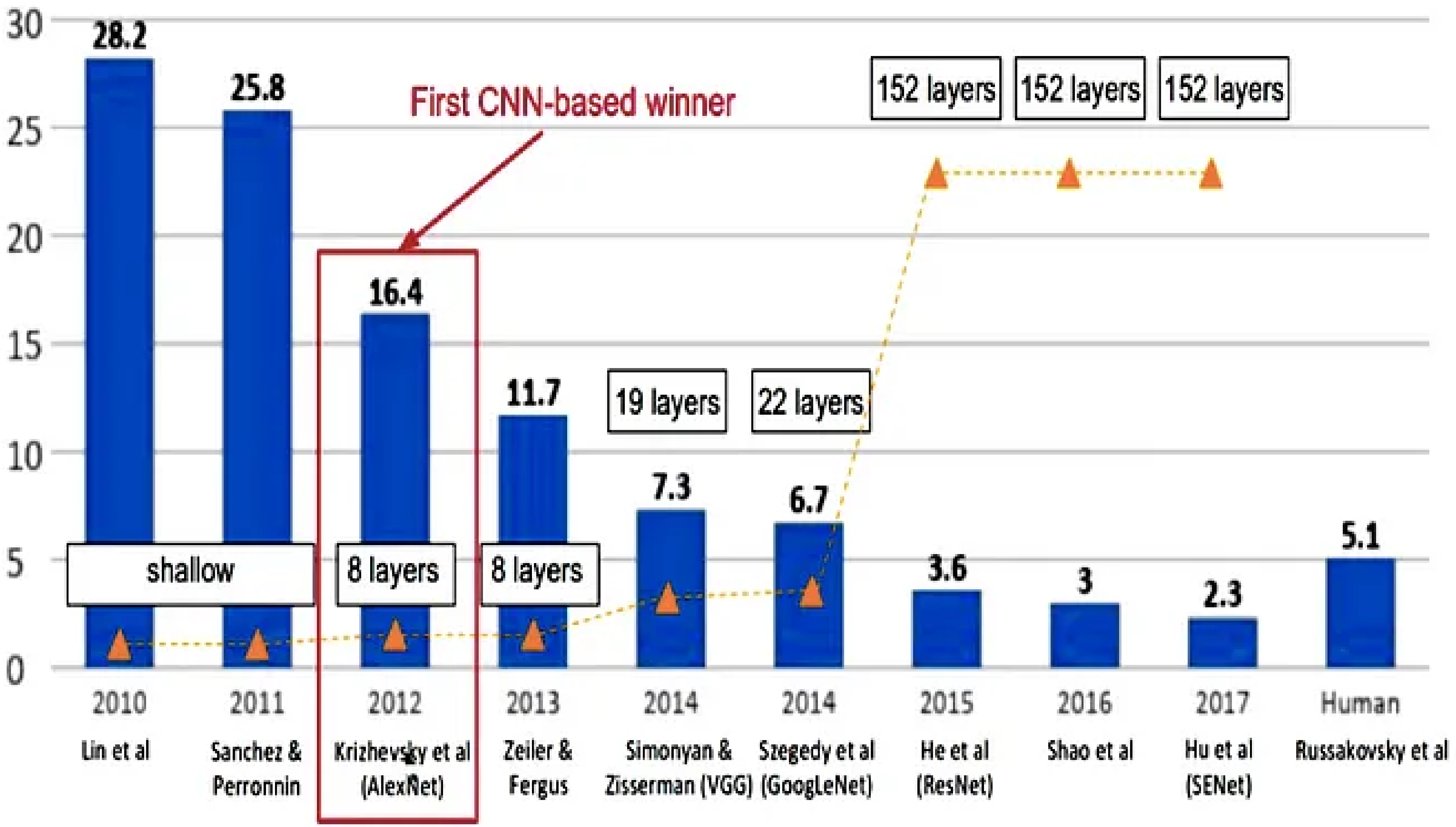


Fig. 3: Winners of ImageNet Classification Challenge [7]

Finally in 2012, Alex Krizhevsky and Geoffrey Hinton came up with a CNN architecture popular to this day as AlexNet, which reduced the error from 25.8% to 16.4% which was a significant improvement at that time.

3. AlexNet (2012)

AlexNet (paper) was the first winner of the ImageNet challenge and was based on a CNN, and since 2012, every year's challenge has been won by a CNN; significantly outperforming other deep and shallow (traditional) machine learning methods.

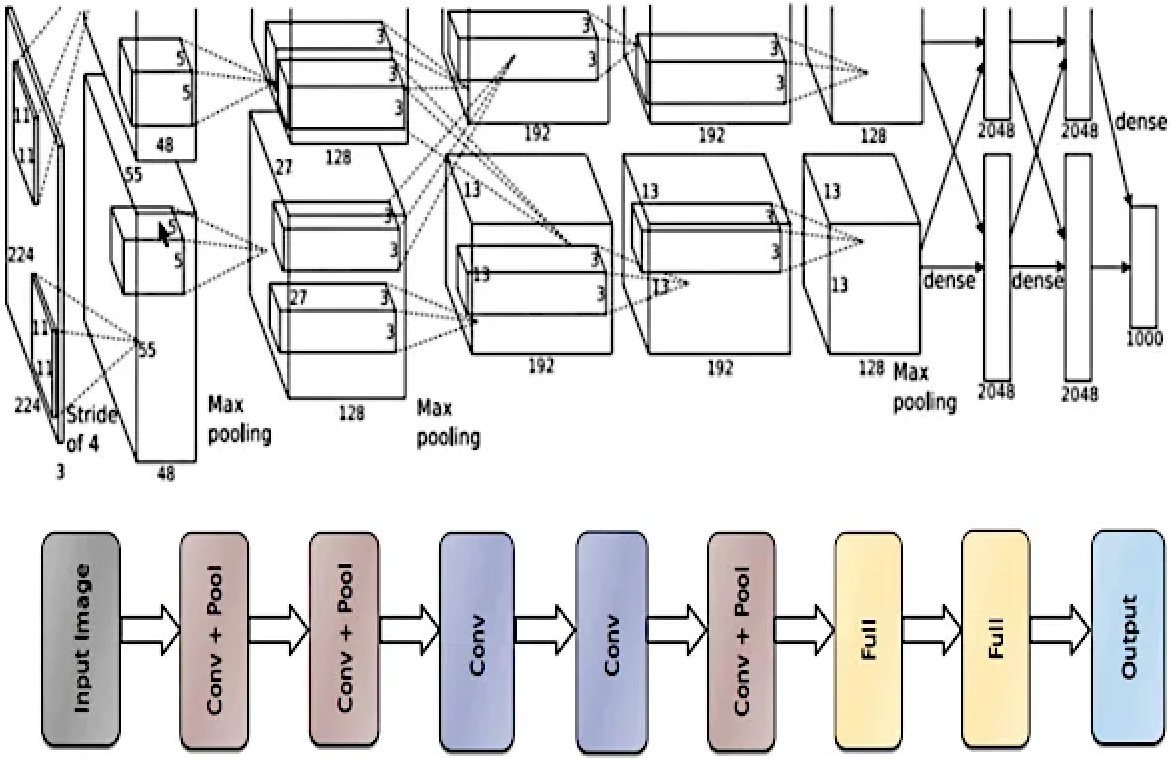


Fig. 4: The architecture of AlexNet

AlexNet has 8 layers in total (5 convolutional layers plus 3 fully connected layers), obviously trained on ImageNet Dataset. A normalization layer called the response normalization layer was first introduced. It normalized all the values in a particular location across the channels in a given layer. Further, it also introduced the rectified linear unit (ReLU) as an activation function. It has about 60 M parameters (Can you recall the number of parameters in LeNet?). Interestingly the convolutional layers cumulatively contain about 90–95% of computation but only about 5% of the parameters.

4. ZFNet (2013)

In the very next year 2013, ZFNet (named after its designer Zeiler and Fergus) became the winner of the ImageNet LSRVC. The architecture of ZFNet was the same as that of AlexNet, but there were a few changes in hyper parameters. In Conv layer 1, the filter size was changed from (11x11 with stride 4) to (7x7 with stride 2). In Conv layers 3, 4 and 5 the number of filters was increased from 384, 384, 256 to 512, 1024 and 512 respectively. With this careful selection of hyper parameters there was a significant decrease in the top-5 errors from 16.4% to 11.7%.

5. VGGNet(2014)

Moving on to 2014, one of the major contributions 2014 witnessed was introduction of a new architecture known as the VGGNet (paper). The VGGNet, stands for an (arcade) architecture, invented by Visual Geometry Group (at Oxford University). It was argued that by making CNN deeper, one can solve problems better and get a lower error rate on the ImageNet classification challenge. Multiple architectures of different depths were tried (fig.5). The Group worked on the philosophy that by increasing the depth, one can model more non-linearities in one's function and hence the key contribution was consideration of depth as a critical component in design. Homogeneous architecture and smaller receptive fields were other key features in design. The architecture won the runner-up in the ImageNet challenge in 2014.

Notably VGG-16 (13 Conv plus 3 FC layers) consists of 138M parameters and has a significant memory overhead of 48.6 MB (For a quick comparison AlexNet has 1.9 MB).

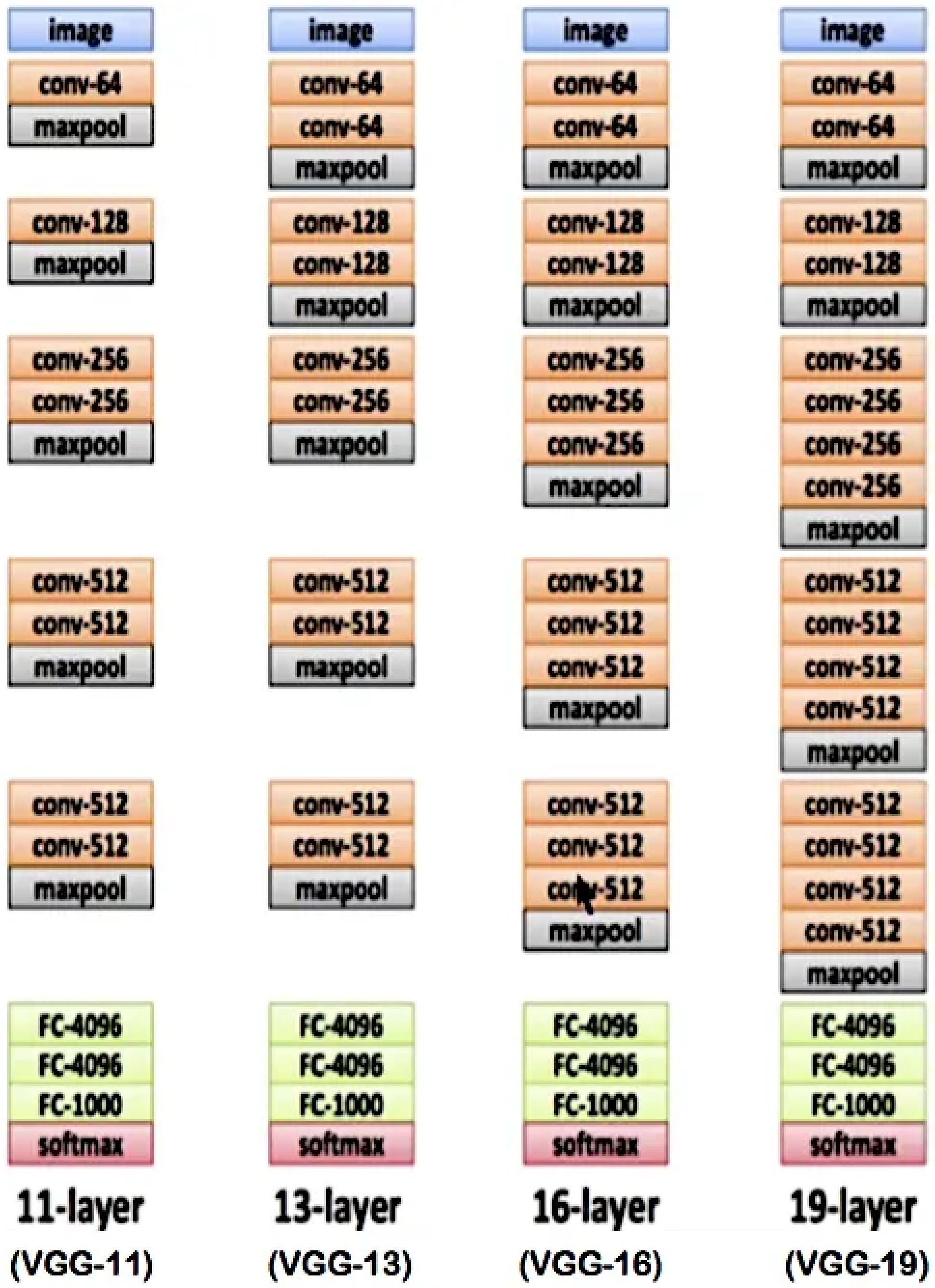
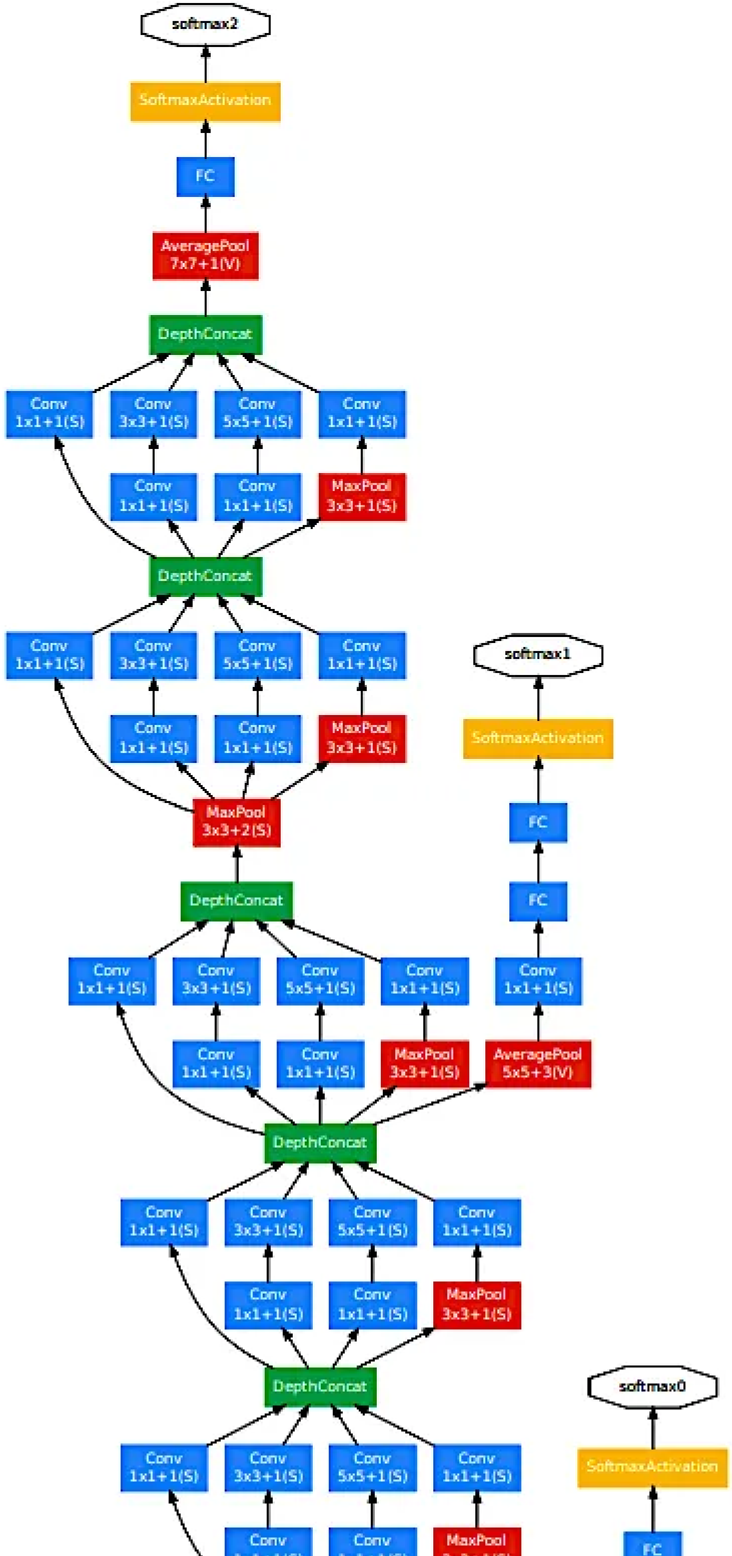


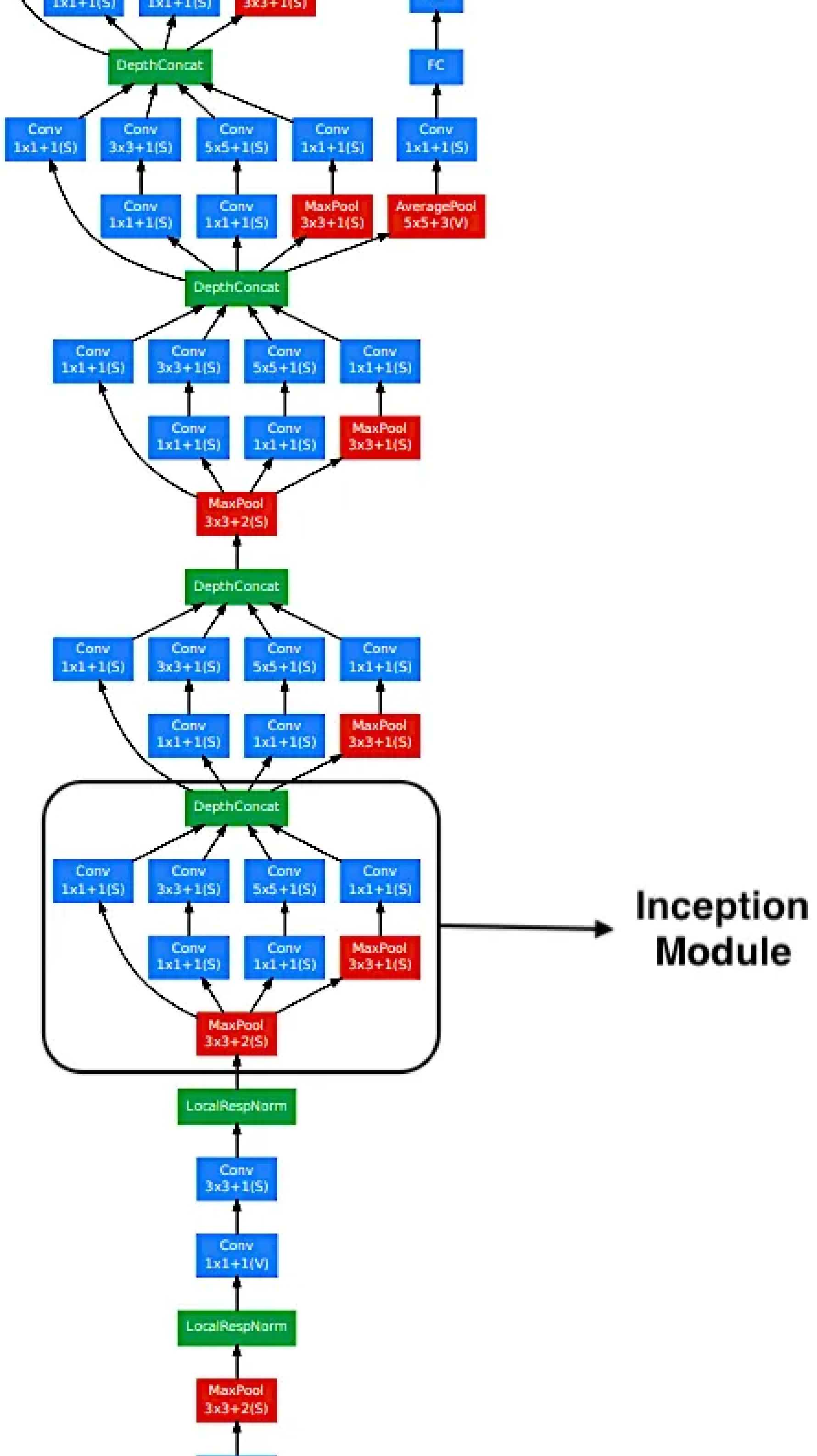
Fig. 5: The architectures of VGG [8]

A deeper variant named VGG-19 was also designed. However, it was only marginally better than VGG16. So, until VGG16 the performance kept improving with depth, but it seemed to saturate after a certain depth. Excited to know why this happened and how it was addressed? — Stay tuned !!!

6. GoogLeNet (2014)

In 2014, Google introduced GoogLeNet (paper). GoogLeNet again focused on deeper networks but with the objective of greater efficiency to reduce parameter count, memory usage, and computation. It went deeper up to 22 layers but without any fully connected (FC) layers. By getting rid of FC layers, the total number of parameters reduced to 5M (12x lesser than AlexNet and around 28x lesser than VGG). A module named 'inception module' was introduced. GoogleNet became ILSVRC-14 classification winner (with 6.7% top-5 error).


softmax2
SoftmaxActivation
FC
AveragePool 7x7+1(V)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
softmax1
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
SoftmaxActivation
MaxPool 3x3+2(S)
FC
DepthConcat
FC
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
AveragePool 5x5+3(V)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
DepthConcat
softmax0
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
SoftmaxActivation
Conv
Conv
MaxPool
FC



Inception Module

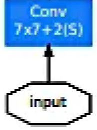


Fig. 6: The architectures of GoogLeNet

The inception module (Local unit with parallel branches) was a key innovation in the approach. This module/block stacked (instead of convolutional layers) many times throughout the network. It served as the foundation to the architecture and hence the name inception. The architecture is also known as Inception-v1. Further, Inception-v3 and Inception-v4 were also introduced in 2015 and 2016 respectively. (Quick Note: Inception-v2 was mainly for experimentation purposes).

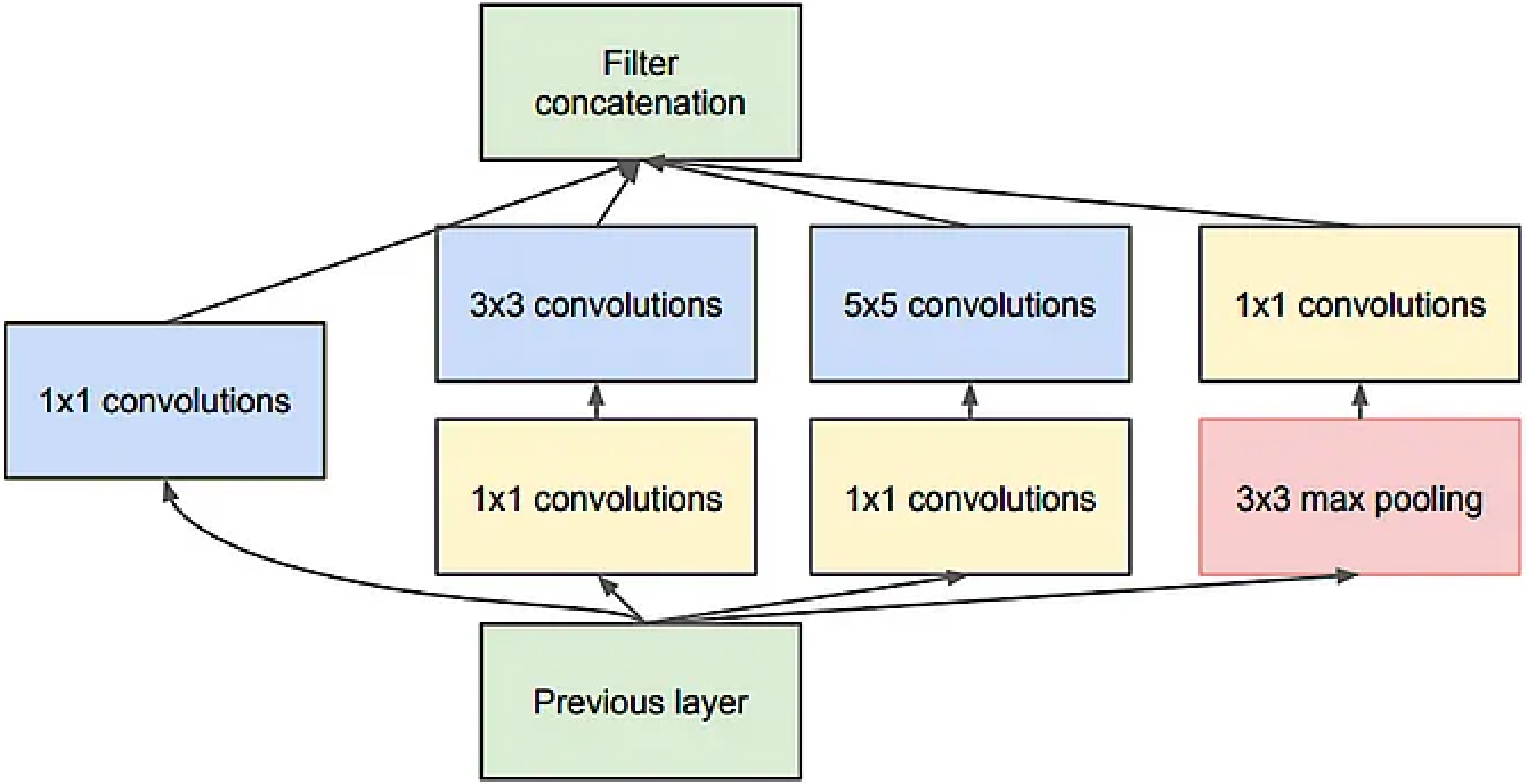


Fig. 7: The Inception Module

Just a wild thought, what happens when we continue to stack deeper layers on a 'plain' convolutional neural network? — Actually the deeper model performs worse than the shallow model. But Why? — The initial guess is that the deeper model is overfitting since it is much bigger than the shallow model. But for this to be true the performance on the training set would have been better for a deeper model. However it was found that a 20 layer deep network performed better than a 56 layer network, both on training and test set. And hence, if we consider the performance on the training set we can say that the deeper (56 layer) network was underfitting. So, what actually happened when the depth of the neural network increased? — *The Vanishing Gradient problem*. Actually, the gradient of a sigmoid function is less than 0.25 and when chain rule is applied, the multiplication of each 0.25 value by each passing layer makes the overall gradient smaller and smaller, approximately 0 for all practical purposes for a sufficiently deep network. And the gradient never reaches the initial layers through backpropagation and the initial layers get stuck with the random initialization, with no weight update and hence no learning. Can we ensure the performance of the deeper model to be at least as well as the shallower model? — ResNet has the answer.

7. ResNet (2015)

Kaiming He et. al. from Microsoft Research came up with an idea of 'residual blocks' which are connected to each other through identity (skip) connections in their architecture ResNet (paper).

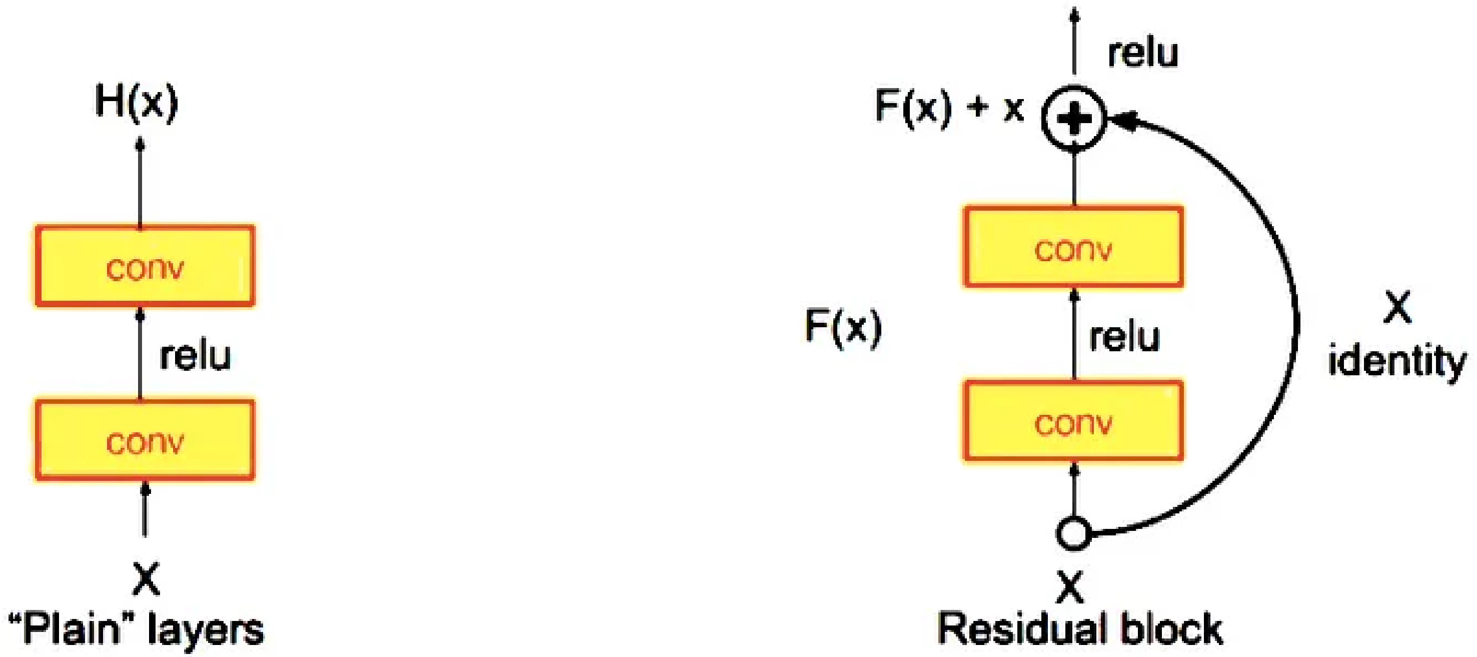


Fig. 8: Plain layers Vs Residual block

A residual network is a stack of many residual blocks (fig.9). Each residual block has two 3 × 3 conv layers.

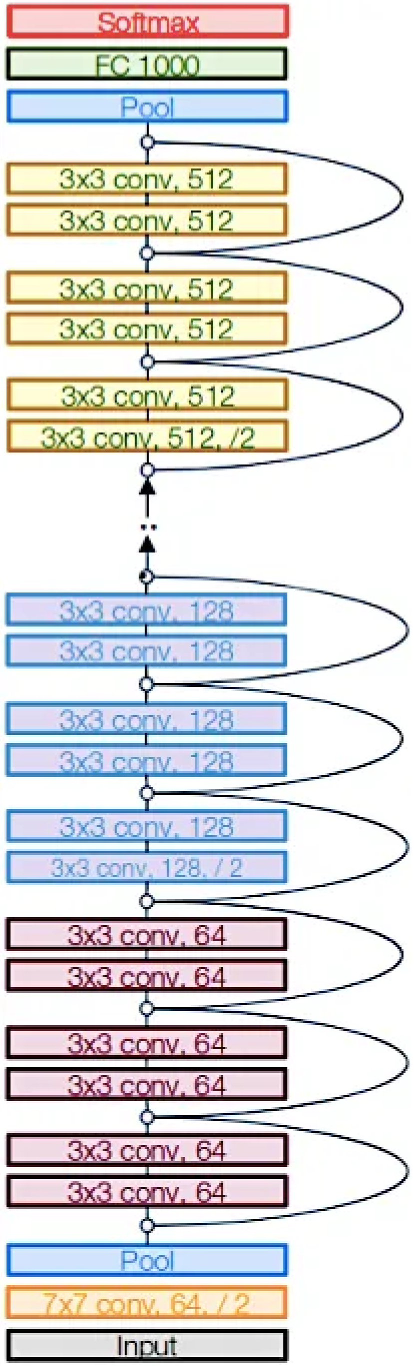


Fig. 9: The architectures of ResNet

The team popularized skip connection and went even deeper up to 152 layers without compromising model's generalisation power. For deeper networks (ResNet-50+), "bottleneck" layer was used to improve efficiency (similar to GoogLeNet).

ResNet won 1st place in all ILSVRC and COCO 2015 competitions and has continued to be a popular choice for several applications. The philosophy used was extended in several recent architectures like Wide Residual Networks (WideResNet), Aggregated Residual Transformations for Deep Neural Networks (ResNeXt), Deep Networks with Stochastic Depth, Densely Connected Convolutional Networks (DenseNets) including more recent ones like MobileNet, EfficientNet, and SENet. With the introduction of newer and newer architecture the story continues like forever and ever...

References

1. Fukushima, K. and Miyake, S., 1982. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets* (pp. 267–285). Springer, Berlin, Heidelberg.

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

3. Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2017. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), pp.84–90.

4. Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

5. C. Szegedy et al., “Going deeper with convolutions,” 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

6. He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

7. Fei-Fei Li, Justin Johnson and Serena Yeung, CS231n course, Stanford, Spring 2019

8. Prof. Vineeth N Balasubramanian, noc20-cs88, IIT Hyderabad, NPTEL, Fall 2020

Computer Vision Appyhigh Artificial Intelligence Cnn Architecture Deep Learning

Written by Brajesh Kumar

Follow