* **GoogLeNet : Going deeper with convolutions.**

By, Christian Szegedy, Google Inc. et al.

The main idea of this architecture was the improved utilization of computing resources inside the network. This paper focused on an efficient deep neural network architecture for computer vision named as Inception. It was derived from the Network-in-Network architecture, originally proposed by Lin et al. along with famous “we need to go deeper” internet meme. Here deep word is used in two different meanings i.e. one is the new level of organization in terms of “Inception module” and second is the increased network depth.

Motivation and high-level consideration behind this paper was that one way to improve the performance of the deep neural network is by increasing their size, including both the depth – number of layers and width – number of units per layer. But it has major drawbacks as it increases the number of parameters that makes the network more prone to over-fitting, especially if training data is limited. The other drawback of the increased network is the increased use of computational resources. Therefore, to overcome this problem author introduced a novel architecture named as Inception module in GoogLeNet architecture which makes use of Convolutional Neural Network. The name “GoogLeNet” is the combination of two things i.e. first few characters named as ‘Goog’ since this architecture is developed by Google scientist in their lab; also, they made use of ‘LeNet’ architecture created by Yann LeCun. So architecture is called GoogLeNet.

This paper is important for two reasons. **First**, the model architecture is tiny compared to AlexNet, VGGNet (which is ≈ 28MB for the weights themselves). The authors are able to obtain such a dramatic drop in network architecture size by removing the Fully Connected layer and instead using Global Average Pooling layer while still increasing the depth of the network. Most of the weights can be found in these dense FC layers, if we remove these layers then memory savings are massive. **Second**, the Szegedy et al. paper makes usage of network in network architecture or micro-architecture. If we see, AlexNet, LeNet, VGGNet, they are more sequential neural networks where the output of one layer feeds directly to the next one. But GoogLeNet uses micro-architecture, small building blocks that is used inside rest of the architecture.

Specifically, Szegedy et al. contributed *Inception module* to the AI community, a small network that fits into CNN enabling it to learn convolution layers with multiple filter sizes so that module becomes multi-level feature extractor. This micro-architecture like Inception has inspired other network like Residual module in ResNet and the Fire module in SqueezeNet. Modern state of the art CNN utilizes micro architectures. These are small building blocks designed to enable networks to learn faster and more efficiently.

The general idea behind the Inception module is that it can be hard to decide the size of the filter we need to learn at a convolution layer. Should they be 5\*5, 3\*3 or 1\*1? Instead, why not learn them all and let the model decide? Inside the Inception module, we learn all three 5\*5, 3\*3 and 1\*1 filter sizes. Computing them in parallel and concatenating the resulting feature maps along the channel dimensions. The next layer in GoogLeNet architecture would be another Inception module receives these concatenated output and performs the same process. This process enables GoogLeNet architecture to learn both local features via smaller convolutions and abstracted features with larger convolutions. By learning multiple filter sizes, we can turn the module into multi-level feature extractor. The 5\*5 filter size has the larger receptive field and can learn more abstract feature. The 1\*1 filter size learns the local features and 3\*3 is like balance in between.

**Inception** **Module:** Inception module consists of four branches from the input layer. **The first branch** in the Inception module simply learns a series of 1\*1 local features from the input. **Second branch** first applies 1\*1 convolution, not only as a form of learning local features, but instead as dimensionality reduction. Larger conv filters (i.e. 5\*5, 3\*3) take more computation to perform. Therefore, if we can reduce the dimensionality of the inputs to these larger filters by applying 1\*1 convolutions, we can reduce the number of computations required by our network. Therefore the number of filters learned in the 1\*1 will always be smaller than the number of 3\*3 filters learned directly. **The third branch** applies the same logic as the second branch, only this time with the goal of learning 5\*5 filters. Once again reduce the dimensionality via 1\*1 conv filter and then feed the output into the 5\*5 filters. **The fourth** and final branch of Inception module performs 3\*3 max pooling with a stride of 1\*1 - this branch is known as pool projection branch. Models that perform pooling have demonstrated an ability to have higher accuracy, but through the work of Springenberg et al. in their 2014 paper, “Striving for Simplicity: The All Convolutional Net” showed POOL layers can be replaced with CONV layers with larger strides for reducing volume size. In case of Szegedy et al., this POOL layer was added simply due to the fact that it was thought that they were needed for CNN to perform reasonably. The output of POOL layer is then fed into another series of 1\*1 convolutions to learn local features. **Finally** all four branches of the Inception module converge where they are concatenated together along the channel dimension. Zero-padding is used to ensure the output of each branch has the same volume size, thereby allowing the output to be concatenated. The output of Inception module is then fed into next layer in the network which is again inception network layer. In practice, we often stacked multiple Inception modules on top of each other before performing pooling operation to reduce volume size.

This architecture is the winner of 2014 ILSVRC challenge where they found a move from Fully connected layer to average pooling layer improved the top-1 accuracy by about 0.6%, however author used dropout even after removing the fully connected layer. While training this network, they used CPU based implementation only but suggested that the network could be trained to convergence using high end GPU’s within a week. GoogLeNet performed better in 2014 compared to previous neural networks which obtain top-5 error of 6.67% on both validation and testing data, ranking the first among the other participant for classification task as well as for detection task, this architecture obtains significantly stronger results with the ensemble.

**Conclusions**: The main idea behind this architecture is the improved utilization of the computing resources inside the network. It gives ability to achieve significant quality compared to shallower network. Approach used in this paper shows the evidence that moving to sparser architecture is feasible and useful idea in general. This paper also suggests that future work require sparser architecture as well as applying the inception module to other domain.