**Project 2 Paper Review**

**Title:** Inception-V4, Inception-ResNet and The Impact of Residual Connections on Learning

**Authors:** Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke

**Abstract:** The authors of the paper attempt to find out whether adding residual connections to the Inception model architecture will perform better than the state-of-the-art inception networks. They created 2 different inception networks, one called Inception-V3 which is the most recent inception model, and their own modified inception model which they called Inception-V4. They also created 2 different Inception ResNet models which they called Inception-ResNet-V1 and Inception-ResNet-V2. They experimented using the ImageNet Classification challenge dataset. They found that both inception models implemented with residual networks outperformed the state-of-the-art inception models by significantly reducing training time and by marginally improving accuracy.

**Introduction:**

The problem that the paper seeks to solve is whether inception networks implemented with residual networks will perform better than the most state-of-the-art inception networks. The authors sought out to see whether the most recent version of inception networks would perform compared to their modified inception networks with residual networks built into them. As mentioned in the abstract the authors created and tested 2 different variants of inception networks. They created many residual inception networks but chose only 2 for testing and documentation. The first which they dubbed as Inception-V3 was the most recent state of the art inception network they could build. It could be considered the plain vanilla inception network. The second inception network that they implemented was dubbed Inception-V4. Inception-V4 was different from Inception-V3 in that the authors thought the most recent state of the art of model could use some modifications to perform better. They went on to state that both models had the same computational cost and the question was which one of the two is more efficient for the pure inception networks. In short, it could be argued that Inception-V3 is the “control” group in this experiment and the other 3 models were what they were testing.

Continuing on, the next 2 models that the authors created where inception networks with residual networks or residual blocks built into them. Why implement ResNets into an inception network? Well to answer that question, interestingly enough when ResNets were first introduced by He et al [1], they found that they helped solve the issue vanishing/exploding gradient. What was interesting about this is that, in theory, the more layers/deeper your network, the more accurate it should be. However, the reality was that as you increased the number of layers, the gradient for each layer’s calculation would either become exponentially small or exponentially large leading to decreasing accuracy and increasing cost. Normally in a plain deep neural network, each layer would calculate some value and you’d have a batch normalization layer to help stabilize the values so that the model would better be able to learn and not become confused. ResNet’s come into play now by taking all of the layers and bundling them up into these “residual blocks.” Now here is where the interesting bit takes place, how these residual blocks work, is there is a block with n amount of hidden layers. The blocks work like any normal plain NN layer however, in a ResNet the input for the block doesn’t just go into the block and gets forgotten. The input to the layer gets added back to the output of the block so as to solve the vanishing/exploding gradient problem. It is imperative to understand that previous convolutions information not be forgotten, because the features that were extracted prior to a residual block are incredibly important and can help the model better learn the features of a given input.

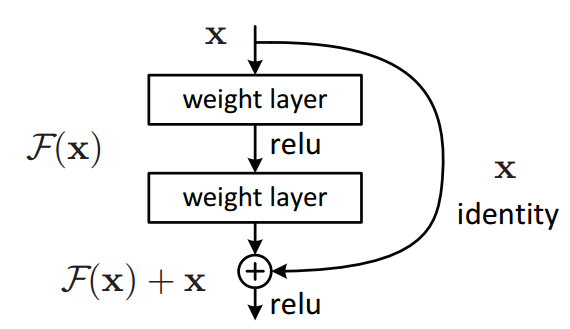


Figure 1: A simple diagram of a residual block

The two Inception ResNet models differed vastly, they only shared 1 architectural block, which was their Reduction-A block. Inception-ResNet-V1 was created and meant to be tested against the most recent inception network which in the study was dubbed Inception-V3. Inception-ResNet-V1 was tested against Inception-V3 since they both had comparable computation costs. Inception-ResNet-V2 on the other hand, was made to compete against the researchers novel inception model in which they dubbed as Inception-V4. What must be noted for both of the residual networks is that they are both deeper than the inception networks so as a result, they attributed some of the success to the residual networks due to being deeper.

Looking at both the inception networks are also interesting. Inception networks when first brought on to the scene in 2014 were revolutionary in the image classification field. The reason why inception is such a big deal is because it solved a dilemma that was prevalent in image classification. A typical image dataset has multiple images that vary in dimensions, color, and subjects. If we were to have a dataset filled with images of dogs, we could have multiple images of dogs where we have a golden retriever in one that is very up close and takes up the whole image, another where we have German shepherds running in the image and are pretty far away, and finally a cute image of puppies that are right in the middle of the photo, not too far and not too close. A basic convolutional neural network would struggle to learn all the features of these photos because of how vastly they vary. We can’t build a model that captures features of objects that are deep in the photo while simultaneously capturing objects that take up the whole photo. In other words, a model can only learn specific features at a given rate. Inception solved this problem by creating a model that has multiple convolutional layers, each with specific kernels, filters, strides, and padding to capture those various features simultaneously. In other words, the model learns features that are both deep in the photo, up close, and in the middle. It can be thought of how our human eye can tell what an object despite what distance it is from us. The beauty of this architecture is that that each layer just like the residual network also has a block of convolutions each working independently and then after all of the convolutions are finished, they go through a filter concatenation layer which essentially creates an amalgam of what was learned. The two models that the researchers implemented worked like so. The first model that they created was Inception-V3, which they took from the paper “Rethinking the inception architecture for computer vision.” Inception-V4 was different from V3 by being a much more simpler and efficient model with less tunable parameters but with much less computational cost.

**Experiments and Results:**

Before going into detail about how the models performed against each other in their battle royale, it is important that we go over the dataset used in this experimentation. The dataset that the experiment used was the image net classification challenge set which consisted of over 150,000 images and 1000 classes. Suffice to say this is an incredibly large image classification dataset. The way that models are determined as the best or not is by looking at their top 1 error score and their top 5 error score. What these 2 scores mean is the probability that the correct prediction was not made. So for top 1 error of a model’s performance, if we get a score of 20%, that means that for 20% of the predictions made, the model incorrectly predicted the class. In other words it was correct 80% of the time. Top 5 error in this case means out of the top 5 predictions the model made for one of the test samples, out of all the top 5, the correct prediction was not among them. Another way of understand this is if our model has a top 5 error rate of 2%, that means that 2% of the time, the model completely missed the prediction. So say we have a 10 class dataset consisting of different car brands, [Audi, BMW, Mercedes, Toyota, Hyundai, Ford, Subaru, Infiniti, Nissan, Jeep]. We put in a test sample which we know to be a Jeep and the model predicts [0.3, 0.3, 0.1, 0.1, 0.2, 0, 0, 0, 0, 0], then the model’s top 5 predictions were Audi, BMW, Mercedes, Toyota, Hyundai and not Jeep, showing the Top 5 error. Naturally we’d expect to see Top 5 error to be lower than Top 1 error as it allows for more margin of error in predictions. All of the error rates given are based on the Cross Validation set given by the Image Net dataset.

During experimentation the researchers found the following results. It is important to note that all models were trained for exactly 200 epochs. When comparing Inception-ResNet-V1 with Inception-V3 in regards to the top 1% classification error, Inception-ResNet-V1 started off with a much lower Top 1 error score and continued to decrease it’s error until it reached 100 epochs of training to which it then proceeded to flatten out and stay at a constant 21.3%, which was also Inception-ResNet-V1’s lowest error rate for Top 1 score. While Inception-V3 continued to decline it’s error rate slowly but surely and even beat the Inception-ResNet-V1 model once it reached ~195 epochs of training, right at the end of the training session of 200 epochs. For Top 1 error Inception-V3 reached a minimum score 21.2%.

With respect to the Top 5 error, again Inception-ResNet-V1 beat the Inception-V3 model. Inception-ResNet-V1 started with a ~9% error rate and quickly reached optimality at around 110 epochs, where it plateaued at around 5.5% error rate and reached minimum error of 5.5%. While the Inception-V3 started at a much higher error rate and didn’t reach 5.6% until around 150 epochs of training. Inception-V3’s minimum error rate was 5.6%. Another key thing to note is that Inception-V3 never crossed below Inception-Resnet-V1. These results give a definitive conclusion that it takes far less time to train Inception-Resnet-V1 and is far more efficient with respect to time than the most recent state-of-the-art Inception model, despite it scoring a better Top 1 score towards the very end of training.

For Inception-ResNet-V2 and Inception-V4, the first thing to note from the experiments is that they both outperformed Inception-V3 and Inception-ResNet-V1, since they were newer modified Inception models. With that being said when looking at the results of the training, a better picture is formed. When looking at Top 1 error, form the very beginning both models started at close error rates, with 30% and 32% for Inception-ResNet-V2 and Inception-V4 respectively. Continuing on, both models steadily lowered their error rates and reached a plateau at around 140 epochs. The end results were 19.9% and 20.0% for Inception-ResNet-V2 and Inception-V4 respectively.

**Summary of Top 1 and Top 5 Errors**

|  |  |  |
| --- | --- | --- |
| **Network** | **Top 1 Error** | **Top 5 Error** |
| Inception-V3 | 21.2% | 5.6% |
| Inception-ResNet-V1 | 21.3% | 5.5% |
| Inception-V4 | 20.0% | 5.0% |
| Inception-ResNet-V2 | 19.9% | 4.9% |

For the Top 5 error rate both models performed impeccably. Although the both started at higher rates than both Inception-ResNet-V1 and Inception-V3, they both achieved their optimal scores that beat out both models. Just as for Top 1 error, both models were very close in their error rates. They both even seemed to plateau at the same number of epochs. In the end, the Top 5 error scores were 4.9% and 5.0% for Inception-ResNet-V2 and Inception-V4 respectively. The conclusion of this experiment is that the Inception-ResNet-V2 after 200 epochs of training is the best model due to having the lowest error rate for both Top 1 and Top 5 errors.

The researchers continued on with other experiments. After seeing which single model performed the best, they then began to experiment with image cropping and the amount of times an image was cropped to see which model would perform the best. The first experiment was with 12 crops as the table below shows.

**Summary of Top 1 and Top 5 Errors with Image Cropping**

|  |  |  |  |
| --- | --- | --- | --- |
| **Network** | **Crops** | **Top 1 Error** | **Top 5 Error** |
| Inception-V3 | 12 | 19.8% | 4.6% |
| Inception-ResNet-V1 | 12 | 19.8% | 4.6% |
| Inception-V4 | 12 | 18.7% | 4.2% |
| Inception-ResNet-V2 | 12 | 18.7% | 4.1% |

**Summary of Top 1 and Top 5 Errors with Even More Image Cropping**

|  |  |  |  |
| --- | --- | --- | --- |
| **Network** | **Crops** | **Top 1 Error** | **Top 5 Error** |
| Inception-V3 | 144 | 18.9% | 4.3% |
| Inception-ResNet-V1 | 144 | 18.8% | 4.3% |
| Inception-V4 | 144 | 17.7% | 3.8% |
| Inception-ResNet-V2 | 144 | 17.8% | 3.7% |

Here the researchers presented the results of their models with image cropping to further show increases in their models’ performance. Given these results the researchers continued to push even further with testing Inception-V3 with an ensemble of Inception-V4 and Inception-ResNet-V2. The ensemble consisted of 1 Inception-V4 model combined with 3 Inception-ResNet-V2 models. The ensemble was an incredibly deep neural network.

**Summary of Top 1 and Top 5 Errors for Ensemble Model**

|  |  |  |  |
| --- | --- | --- | --- |
| **Network** | **Number of Models** | **Top 1 Error** | **Top 5 Error** |
| Inception-V4 | 4 | 17.3% | 3.6% |
| Inception-V4 + 3x Inception-ResNet-V2 | 4 | 16.5% | 3.1% |

Here it is evident that the incredibly deep ensemble of the modified new Inception-V4 and its residual network counterpart is the best performing model in the whole experiment.

**Conclusions:**

From the results of the experiment what we can effectively conclude is that the authors had created a more efficient modified inception architecture. They also successfully showed that a residual network inception network architecture is more efficient than the most state-of-the-art inception architecture, dubbed Inception-V3. Another thing that we’ve seen is that an unfortunate cost of the efficiency of the Inception-ResNet-V2, is that it takes a while to reach optimality of around 140 epochs, versus the Inception-ResNet-V1 which is a residual network version of the state-of-the-art Inception-V3 took around 100 epochs of training. It is no secret that deep neural networks have a high cost attributed to them when it comes to training. Perhaps when future researchers or engineers wish to implement this model or modify it, they won’t have much issue with training time and will care more about performance rather than time to train and reach optimality.