Computer Vision_DiscussionQuiz_05

1. In Wide ResNet we increase the width of the ResNet and decrease the width, in normal ResNet the depth is more and is thin. Wide ResNet with increase in width and decrease in depth outperforms ResNet. We denote a Wide ResNet as WRN-n-k. Here, n is total no.pf convolutional layers and k is widening factor. The architecture difference in Wide ResNet and ResNet is that in Wide ResNet the convolutional layers have more feature maps, and they are wider. In, Wide ResNets we also add a dropout, which shows a little improvement.

2. Identity connection problem for training a network is that due to the gradient flow, while training the only few blocks either learns the whole information or all blocks share and learns very little information. This diminishes the results as they contribute very less to the outputs. This is the main problem of the identity connection while training a network. This problem is called as diminishing feature reuse. To resolve this problem, they came up with adding dropout to the residual blocks by adding an identity scalar to each residual block where we add dropout, to deactivate few blocks while training.

3. The limitations in the ResNet that Wide ResNet tries to solve are, ResNet is slow to train as there are so many depth layers, and ResNet faces diminishing returns issue due to thin depth layers, the outputs get diminishes. The computational costs are high, as the little increase in accuracy leads to the number of layers getting doubled up, and with the number of layers the computational cost gets increased. These are all the limitations Wide ResNet tries to solve with adding a widening factor and reducing the depth.

4. In Wide ResNet, we have seen that even with decreasing the depth the power of residual networks didn't reduce. In Wide ResNet, we decreased the depth and increased the width, and have seen that with this change the power of residual networks didn't change. The power is even increased in Wide ResNets, so we got a conclusion that the depth is not the factor that is being held responsible for the power of residual networks, instead it's the residual blocks that are responsible for the power of residual networks.

5. Ways to increase the representational power of ResNet blocks are:
   i. Adding more convolutional layers per block.
   ii. Widening the convolutional layers by adding more feature maps
   iii. We can increase the filter size in convolutional layers.

6. B(3, 1, 1) denotes that a block with one 3x3 convolutional layer followed by two 1x1 convolutional layers. This is like network-in-network architecture because in network-in-

network architecture there's a convolutional layer which is followed by two 1x1 convolutional layers. So, here in B(3,1,1) it follows the same approach but just that here the first convolutional later is of 3x3, and network-in-network architecture the first could be any convolutional layer. So, we say that B(3,1,1) is similar to the network-in-network style architecture.

7. Let, 'l' be the deepening factor of a model, that means it's the number of convolutional layers per block. Let 'd' be the number of ResNet blocks. And 'k' be the widening factor, that means the number of features in convolutional layers. The relation of number of parameters with these are: With increase in the deepening factor 'l' and the number of ResNet blocks 'd', the number of parameters increases linearly. And with increase in the widening factor 'k', the number of parameters increases quadratically.

8. In Wide ResNet we are adding a widening factor, adding feature planes to increase the width of the convolutional layers, but in ResNext, we are grouping the convolutional layers. Here, we are following split-transform-merge strategy like in inception. So, there is an additional feature called cardinality in ResNext architecture. In the ResNext architecture its value was 32. And in ResNext, we are using a network in neuron architecture, where a neuron does the inner product.

9. In ResNext, simple neuron that performs inner product, operates the split-transform-merge strategy, by splitting the 1x1 convolution layer and transforming to 3x3, 5x5, etc convolutional layers and concatenating these layers. So, let the simple neuron is x, then splitting strategy splits x into x_1, x_1,….,x_n. Then we add weights to transform, then it becomes x_i*w_i, and then the aggregation comes, here, we sum up this w_i*x_i from i=1 to n. So, after aggregation it looks like, Σw_i*x_i, from i=1 to n.

10.
   a. Basic block: In Residual building block the input value is sent to the output through a shortcut connection or skip connection. Let 'X' is the input then in other networks 'Y = F(X)' is the output, but in residual network, if 'X' is the input then 'Y=F(X)+X' is the output, as we send the input to the output and F(X) is the cost function. In addition, in basic block, there are two 3x3 convolutional layers. And in basic wide block, these two 3x3 convolutional layers are widened with a widening factor.

   b. Bottleneck block: In basic block, we used two layers in building block, but in bottleneck block we use three layers instead of two. The first layer is a 1x1 convolutional layers, next is 3x3 convolutional layer and the third one is 1x1 convolutional layer. We use the first 1x1 convolutional layer to reduce the size and the other 1x1 convolutional layer to resize and increase the dimension.

c. Deepening factor: Deepening factor is nothing but the number of depth layers in a block, and these depth layers are indicated by the number of convolutional layers in the block. So, the deepening factor indicates nothing but the number of convolutional layers in the block. And as we described in previous question, with deepening factor, parameters have a linear rise.

   Widening factor: Widening factor describes the width of the residual network. The width is increased by the number of feature planes in the convolutional layers. So, widening factor represents the number of features in the convolutional layers. And, with widening factor, the parameters rise quadratically.

d. Cross-validation: Cross-validation is used to solve over-fitting problem. This is a resampling approach, which splits data into two parts, training set and test set. We use the training set to train the model and test set to test and predict the outcomes. This part of testing and predicting reveals the data accuracy, if we get good accuracy, that means there's no over-fitting and the model is perfectly trained and can be used for further predictions.

e. Cardinality: Cardinality is nothing but the size of set of transformations. It means the number of group convolutions that are being formed is called as Cardinality. In the ResNext paper, the cardinality was 32, as the number of group convolutions are indicated as 32.

Note: https://www.analyticsvidhya.com/blog/2021/05/4-ways-to-evaluate-your-machine-learning-model-cross-validation-techniques-with-python-code/

Papers - Wide Residual Networks
ResNeXt- Aggregated Residual Transformations