Rock Paper Scissors using Deep Residual Learning

Kunal Katiyar

September, 2020

Contents

Chapter 1

Residual Networks

1.1 Problem

Normally, what it seems is error decreases with increase in number of layers but in practical it decreases what I have experienced too, because of problem of vanishing/exploding gradients which hampers convergence from the beginning.

1.2 Solution

Skip connections or Residual Networks are easy to optimize and they easily enjoy accuracy gain from increased depths

The identity shortcuts can be directly used when input and output are of same dimension, or else we consider following methods

1. Padding with zeroes
2. Projection shortcuts for higher dimensions where $y = F(x, W_i) + W_s x, W_s$ is projection shortcut

1.2.1 Identity Block

Used when input activation has same dimension as output activation.

1.2.2 Convolutional Block

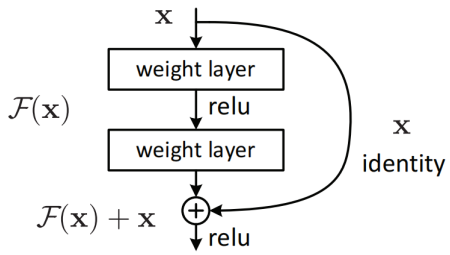

Figure 1.1: Identity shortcuts

Used when the 2 dimensions do not match. Convolutions are performed on $X_{shortcut}$ followed by Batch Normalization. This plays a similar role as W_s (mentioned earlier).

1.3 Implementation and Hyperparameters in original ResNet

- Weights are initialized using He Normal method.
- Optimized by Stochastic Gradient Descent with mini batch size of 256
- Images are resized to $(224, 224)$ with per pixel mean substracted
- Batch Normailzation after each convolution and before activation
- $\alpha = 0.1$, and when error plateaus then α is reduced by a factor of 10
- $\beta = 0.0001$ and momentum $= 0.9$

1.3.1 Using PReLu as activation

$$F(y_i) = \begin{cases} y_i, & \text{if } y_i > 0\,, \\ a_i y_i, & \text{if } y_i \leq 0\,. \end{cases}$$

When $a_i = 0$ it becomes ReLu and when $a_i = k$ (small value) it becomes Leaky ReLu where a_i is learnable

PReLu is inbuilt in Keras and can easily be used as a layer.

$$f(y_i) = max(0, y_i) + a_i min(0, y_i)$$

Leaky ReLu helps in avoiding zero gradients or vanishing gradient issues.

These extra number of parameters in the PReLu layers is equal to the number of channels which is negligible when compared to total number of parameters so no extra risk of overfitting.

Weight decay biases PReLu towards ReLu.

He initialization is used for initializing parameters because Xavier initialization is only for linear activations so it is invalid for Relu and PReLu.

ReLu suffers from problem of dying neurons.

1.3.2 Batch Normalization

When we have features in different ranges we normalize them so that the model can converge faster. If the input layer is benefitting from it why not do the same for values inside the hidden layers ?!

Batch Normalization reduces the amount by what hidden values shift around which is known as covariance shift.

Problem

We train and map $X \rightarrow Y$ and if distribution of X changes we would have to retrain

Solution: Batch Normalization

We can use higher value of α because Batch Normalization makes sure that there is no activation that is too low or high.

It reduces overfitting because it has slight regularization effects. Similar to dropout it adds some noise to each hidden layer's activation. There is ssome evidence that ease of learning an identity function accounts for remarkable performance of Residual Networks even so more than skip connections helping with vanishing gradient problems.

1.4 Conclusion

It was observed that ease of learning an identity function accounts for ResNet's remarkable performance even so more than skip connections helping with vanishing gradient problems.