Summary of the Paper: Visualizing the Loss Landscape of Neural Nets

There are different techniques to find good minimizers for non-convex loss functions for a neural network. In this paper, the researchers tried to visualize the loss surface, and how different parameters affect loss function. Initially, the authors proposed a method called “filter normalization” to reduce the high dimensional data to a lower dimension so that it can be represented in a 1D or 2D graph. Then using scatter, contour plotting and heatmap, they tried to deduce how weight decay, batch size, skip-connections affect the training loss, accuracy and finally justified why and how these correlates to each other.

For filter-normalization, they used a random Gaussian direction vector **d** which is compatible with the set of parameters $\boldsymbol{\theta}$ by replacing $\boldsymbol{d_{i,j}}$ with $\boldsymbol{d_{i,j} / || d_{i,j} || * || \theta_{i,j} ||}$ where the $\boldsymbol{j}$th filter is in $\boldsymbol{i}$th layer and **||.||** defines the Frobenius Norm. This normalization works both with a convolutional layer and also a Fully Connected layer when the filter is of size 1 x 1.

The authors explored the term Sharp and Flat minimizers and which generalizes the data better. Their experimental results using CIFAR-10 data with a 9-layer VGG network showed that using a small batch, the loss function curve is fairly wide(more flat) but for a large batch it is quite the opposite, it’s sharp. But we get a totally opposite result if we turn on weight decay. Then they do the same thing using filter-normalized plots for Resnet-56 separately for different batches with/without weight decay, and found that larger batches have sharper minima and vice versa for small batches. But using weight decay the sharpness of the curve with large batches decreases.

Then the authors also explored the question using visualization, whether extreme deep architectures render a network not-trainable at all, and if a loss function has significant non-convexity, then they tried to answer why it doesn’t affect all networks. They experimented with ResNet-20/56/110 using Skip and No-Skip connection and another experiment with ResNet-56, Skip and No-Skip connection with k filters per layer. For the first experiment the result shows that as the depth of a network increases, the loss surface turns from convex to chaotic. For shallow networks, the result of skip connection is fairly un-noticeable, but for deeper networks, the chaos turns into a more moderate convex function. Finally, for the second experiment with multiple filters per layer, it is seen that,with more filters per layer, a wider/deeper model has a good loss landscape with hardly any noticeable chaotic behavior at all.

By using filter normalization, as the dimension is dramatically reduced, the authors calculated the convexity of loss function by calculating principal curvatures which are the eigenvalues of the Hessian. As a true convex function has non-negative curvatures, they calculate the max and min$\boldsymbol{(\lambda_{min}}$ $\boldsymbol{and}$ $\boldsymbol{_{max})}$ eigenvalues of Hessian. They plot $\boldsymbol{|(\lambda_{min} / _{max})|}$ across the loss surface using heatmap. The results of previous experiments align with the results of the heatmap. Experimental results show that with skip connection Resnet-56 has less than 1% of eigenvalues that are negative, which determines that the loss surface is less chaotic with skip connection.

Finally they plotted the trajectory of different optimizers using PCA which can capture maximum variance from parameters $\boldsymbol{\theta}$**.** At the beginning of training, the paths tend to move perpendicular to the contours of the loss surface, and when we use weight decay of non zero value with a small batch size, we see the path remain parallel to the contour.

This paper presents different visualization techniques to find insight on choosing how to design a neural network architecture for a practitioner. The paper describes how to select different architecture, optimizer, batch size and when to use skip connection, weight decay etc.