Visualizing the Loss Landscape of Neural Nets

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, Tom Goldstein
Code: github.com/tomgoldstein/loss-landscape

Summary

- We reveal faults in a number of methods for loss landscape visualization, and show existing strategies fail to accurately capture the local geometry.
- We present a visualization method based on filter normalization, which provides accurate visual interpretations for the inscrutability and generalization of neural nets.
- Network architecture choices have visualizable effects on loss functions that could help explain variability.

The Sharp vs. Flat Dilemma

It is widely believed that small-batch SGD produces “flat” minimizers that generalizes better, while large-batch sizes produce “sharp” minimizers with poor generalization[1].

1D linear interpolation

We train a VGG-9 net on CIFAR-10 for a fixed number of epochs using two batch-sizes: 128 and 8192. Let $\mathbf{g}_\theta$ indicate weights of the solutions obtained by small-batch and large batch, respectively. We plot the loss values along the direction $\mathbf{g}_\theta - \mathbf{g}_\theta$ as in [1, 2], i.e.,

$$f(\alpha, \beta) = L(\theta + \alpha \mathbf{g}_\theta + \beta \mathbf{g}_\theta).$$

We plot the 1D and 2D surface of minima obtained by using different optimizers, batch size and weight decay.

Filter Normalized Sharpness Comparison

Create filter normalized random direction(s)

- Each filter of the neural network might live on a different scale. To remove this scaling effect, making the minimizer sharp.
- A smaller batch size results in more weight updates per epoch, causing weights to shrink.
- When weights are small, a small perturbation to the weights has a dramatic effect, making the minimizer sharp.

The Effect of Identity Mapping

- Residual network landscapes are dominated by wide, flat minimizers surrounded by large regions of apparent convexity that capture far-away initializers.
- Without skip connections, loss landscape is populated by many sharp minima with many small regions of convexity, creating strong dependence on initialization.

The Effect of Network Depth

- Skip connections prevent the explosion of non-convexity that occurs when networks get deep.
- ResNet 56 and Wide-ResNet 56 with Wide-ResNets by multiplying the number of filters per layer by $k$.
- Sharpness correlates extremely well with the generalization error.
- Wider models have wider minima and wider regions of apparent convexity.
- Increased width prevents chaotic behavior, and skip connections dramatically widen minimizers.

The Effect of Network Width

- Models with lower width are more sensitive to initialization, while wider models have sharper minima.
- Width effects are less effective for lower dimensional models, which are simply sections of higher dimensional regions.

Wide Models vs. Thin Models

We compare the narrow CIFAR-10 ResNet-56 with Wide-ResNet by multiplying the number of filters per layer by $k$.

Are We Really Seeing Convexity?

Is there “hidden” non-convexity that these visualizations fail to capture?

One way to measure the level of convexity in a loss function is to compute the principal curvatures, which are simply eigenvalues of the Hessian.

- We calculate the min and max eigenvalues of the Hessian ($\lambda_{min}$ and $\lambda_{max}$) and map the ratio ($\lambda_{min}/\lambda_{max}$) across the loss surfaces studied above.
- Convex-looking regions do indeed have insignificant negative eigenvalues, while chaotic regions contain large negative curvatures.

Visualizing Optimization Paths

Effective Trajectory Plotting using PCA Directions

- Random directions fail to capture the variation in optimization trajectories.
- Given $n$ training epochs, we apply PCA to the matrix $\mathbf{J}_n = [\mathbf{J}_1, \mathbf{J}_2, \cdots, \mathbf{J}_n]$ and then select the two most explanatory directions.

- The descent path is very low dimensional: 40% - 90% of the variation in the descent paths lies in a space of only two dimensions.

Refererence