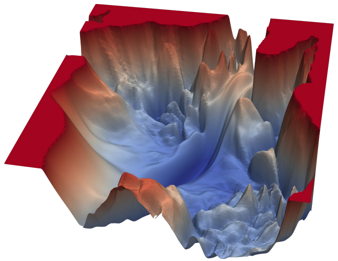
Visualizing Loss Landscapes of Neural Nets

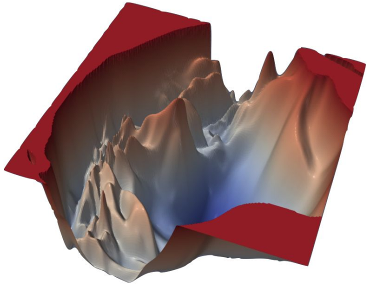
Background

1. 1D Interpolation: $\theta(\alpha) = (1-\alpha)\theta + \alpha\theta'$
    a. Choose 2 sets of parameters;
    b. Plot values of loss function along line connecting these points
    c. Alpha is used to parameterize the line
    d. Drawback: Non-convexities are hard to visualize in 1D
2. Contour Plots $f(\alpha, \beta) = L(\theta^* + \alpha\delta + \beta\eta)$
    a. Choose center in graph, theta*
    b. Choose two direction vectors eta and delta
    c. Drawback: low-res plots that might not capture complexity of loss surface

Motivation/Context

- Is there a significant effect of training parameters (like batch size) on loss landscapes of deep neural nets?
- Effect of loss landscapes on generalization
- Due to size of the weights and high-dimensionality, it is difficult to visualise loss landscapes.
- Previous methods include:
  - 1D Interpolation
  - 2D Random directions (Contour plots)





Proposed Approach: Filter-wise Normalization

- Compute a random gaussian vector d with dimensions same as θ
- Normalize each filter in d such that it has same norm as corresponding filter in θ.
- Applied to Conv and FC layers

$$d_{i,j} =\leftarrow \frac{d_{i,j}}{||d_{i,j}||}||\theta_{i,j}||$$

Experimental Setup

- Flow of experiment is:
  - Train models on a dataset or load a pretrained model
  - Extract model parameters
  - Generate random vectors and apply filter normalization method
  - Calculate loss values across the grid of possible values
  - Plot the loss landscapes

Experimental Setup

- We experiment with a battery of models and hyperparameters to investigate the effect of model choices and training dynamics with respect to the loss function.
- In particular, we train the following on CIFAR-10 dataset:
    - Linear Layer Models
    - CNN Model (with skip connections)
    - CNN Model (without skip connections)
- Additionally, we also visualize the contour plots of the pretrained MobileNet model.

Research Question

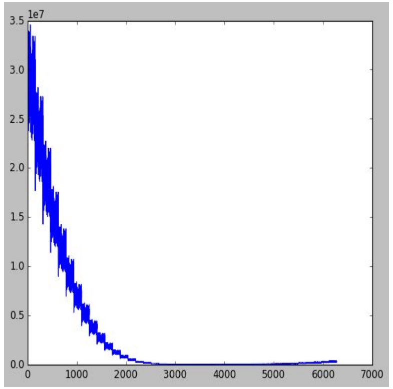
What effect does batch size have on loss landscape and generalization across different models, (trained from scratch or pretrained)?

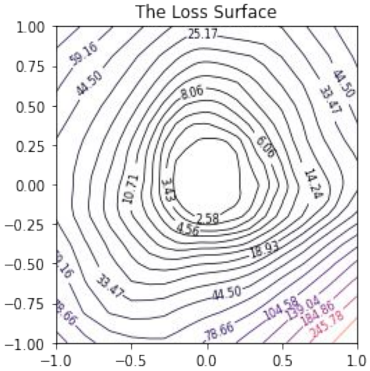
Results

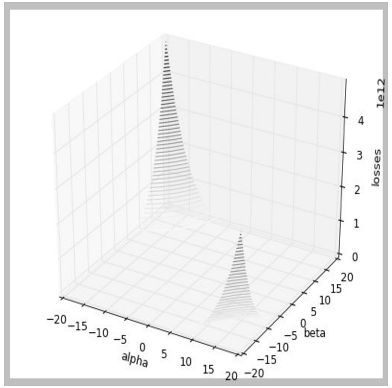
Linear Layer model on Cifar-10 dataset:

- Batch Size Used: 64 & 512
- Learning rate: 5e-4
- Optimizer: Adam

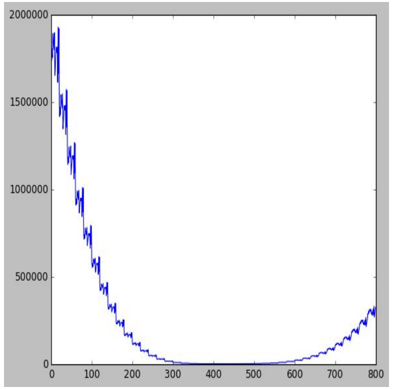
Batch size: 64

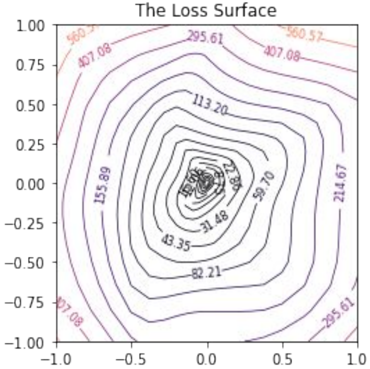


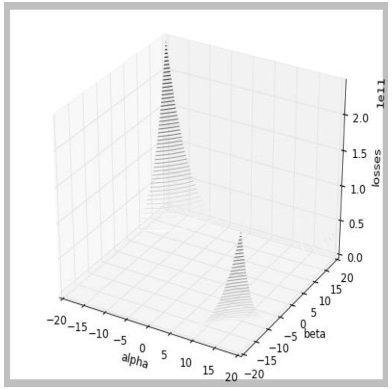




Batch size:512





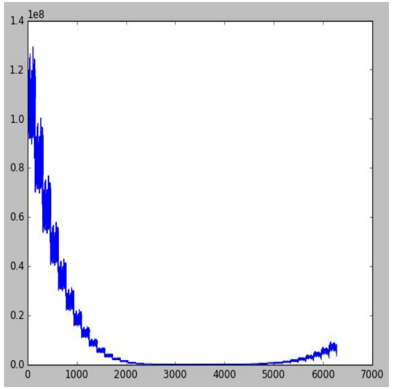


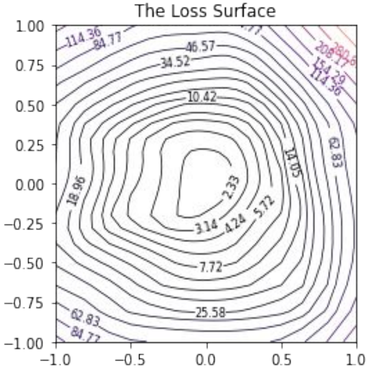
Results

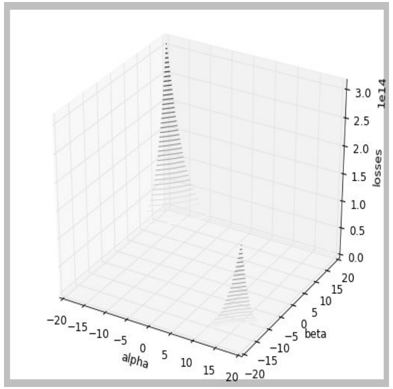
Convolution Layer model(without skip connection) on Cifar-10 dataset:

- Batch Size Used: 64 & 512
- Learning rate: 5e-4
- Optimizer: Adam

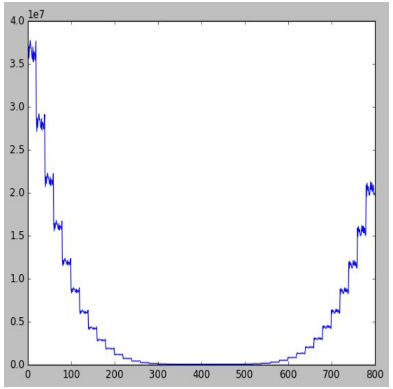
Batch size: 64

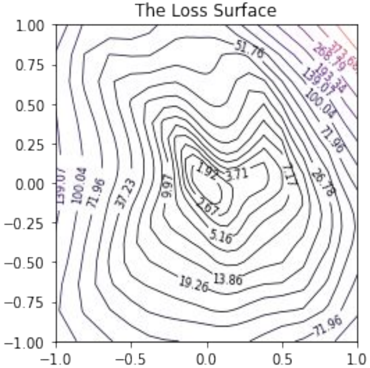


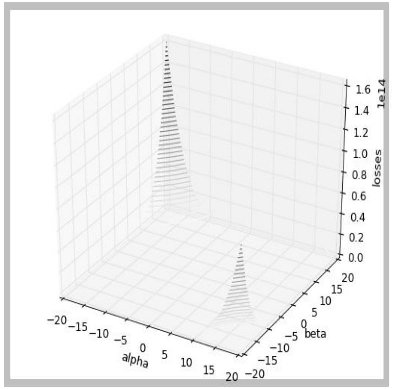




Batch size:512





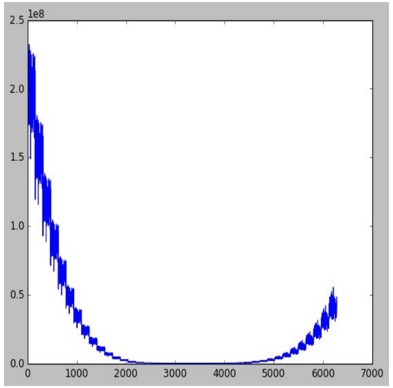


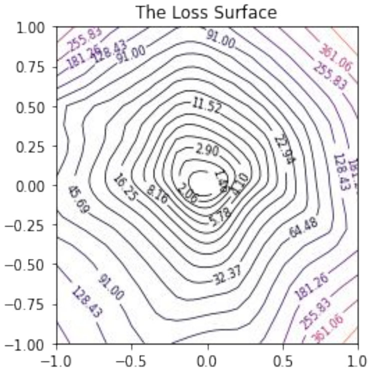
Results

Convolution Layer model (with skip connection) on Cifar-10 dataset:

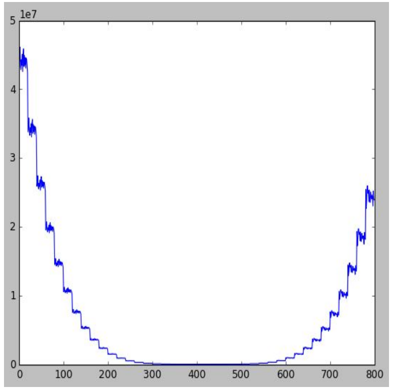
- Batch Size Used: 64 & 128
- Learning rate: 5e-4
- Optimizer: Adam

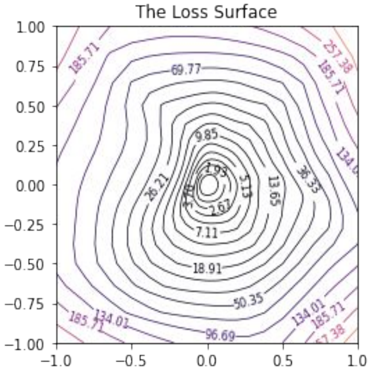
Batch size: 64

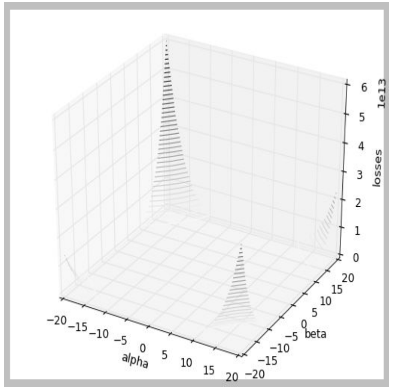




Batch size: 512





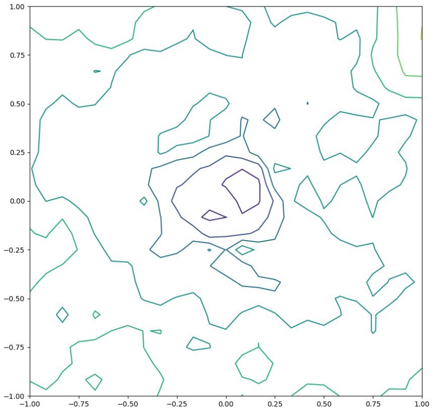


Results

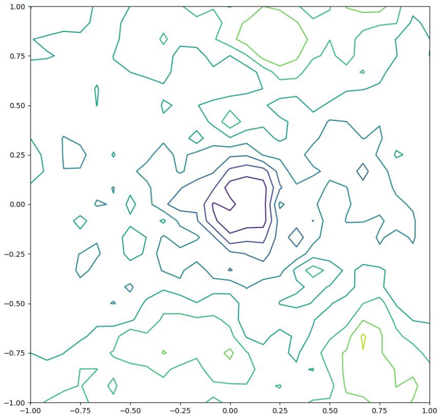
MobileNet trained on ImageNet dataset:

- Batch Size Used: 64 & 512
- Pretrained weights

Batch size:8



Batch size: 32



Note: Since MobileNet has ~3M parameters, it was computationally very expensive to generate multiple plots for it.

Observations

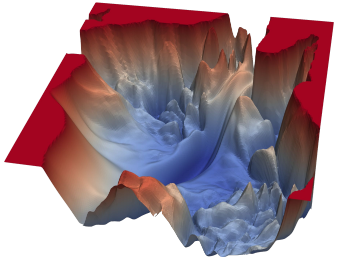
- We observe that, smaller batch sizes lead to loss landscapes which are:
    - More convex
    - Less chaotic
    - Have wide regions of convexity
- In the contour plots, we clearly see that loss is minimum in regions of high convexity.
- These visualizations help us in disentangling the mysteries of deep learning and what factors influence its dynamics.
- From the original paper:
    - BatchNorm results in better and smoother loss landscapes
    - VGG models have landscapes with multiple local minima

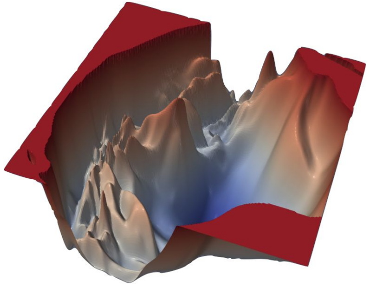
Future Work

1. We planned to implement the loss visualizations on NLP models like BERT, etc.
2. Generate ways to plot in higher resolutions.
3. Make the process computationally less expensive.

Thank you

Paper at a glance





- Certain network architectures produce “smoother” loss functions which converge easier. Why and how this happens is unknown.
- Using a range of visualization methods, this paper plots the loss landscape of neural nets (with millions of parameters) to explore the effect of various model choices on convergence and generalization. Three methods:
  - 1D Interpolation
  - 2D Landscape
  - 3D Landscape
- Directly comparing loss landscapes of models is hard because:
  - Many types of layers (like ReLU) do not contribute to loss landscape
  - Perturbing large weights has very little effect on the model
- To tackle this, **Feature Normalization** is proposed where the randomly generated vectors are normalised to have the same direction as the parameter vectors.

Project Scope

1. We plan to implement the Filter Normalization approach for a variety of neural networks.
2. In order to have meaningful insights, we plan to contrast the loss landscapes of trained models based on the following settings:
   a. Large vs small batch size
   b. Skip connections
   c. Random vectors vs pretrained vectors
   d. Effect of network depth
3. We will then draw inferences on the types of landscapes induced by various network choices. Possible pointers to look for are:
   a. Chaotic landscapes
   b. Shallow valleys
   c. Train/Test Error and how they relate with the landscapes
   d. Flatter valleys

We have started implementing the project here: https://github.com/Ravi2308/Visualizing-the-Loss-Landscape-of-Neural-Nets