Paper Review
Visualizing the Loss Landscape of Neural Nets

YeongHyeon Park

Department of Electrical and Computer Engineering

SungKyunKwan University

Some figures and illustrations included in this presentation are edited from the original.

Hao Li, et al. "Visualizing the loss landscape of neural nets." *Advances in neural information processing systems (NeurIPS).* 2018.
1153 citations

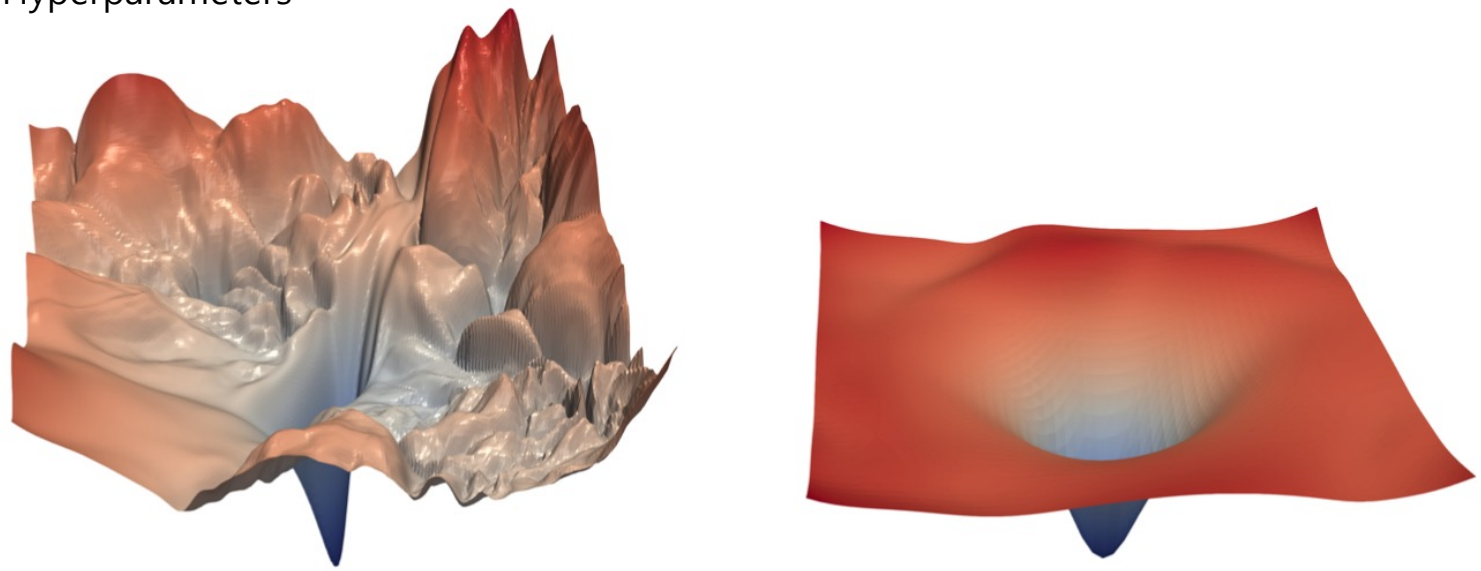
Visualizing the Loss Landscape of Neural Nets

Hao Li[1], **Zheng Xu**[1], **Gavin Taylor**[2], **Christoph Studer**[3], **Tom Goldstein**[1]
[1]University of Maryland, College Park [2]United States Naval Academy [3]Cornell University
{haoli,xuzh,tomg}@cs.umd.edu, taylor@usna.edu, studer@cornell.edu

Purpose

Qualitative Evaluation of Neural Networks.

- Networks architecture design
- Optimizer
- Variable (parameter) initialization
- Hyperparameters

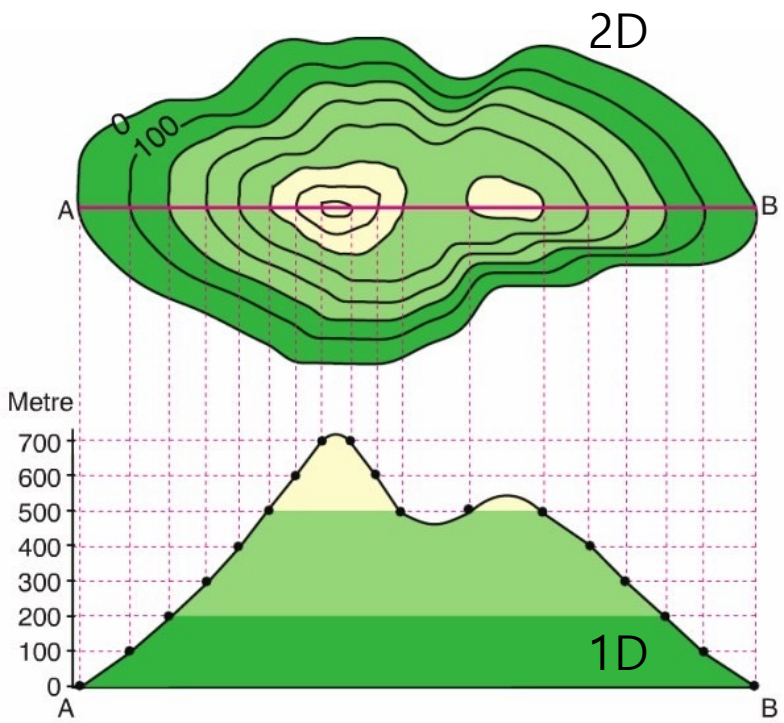


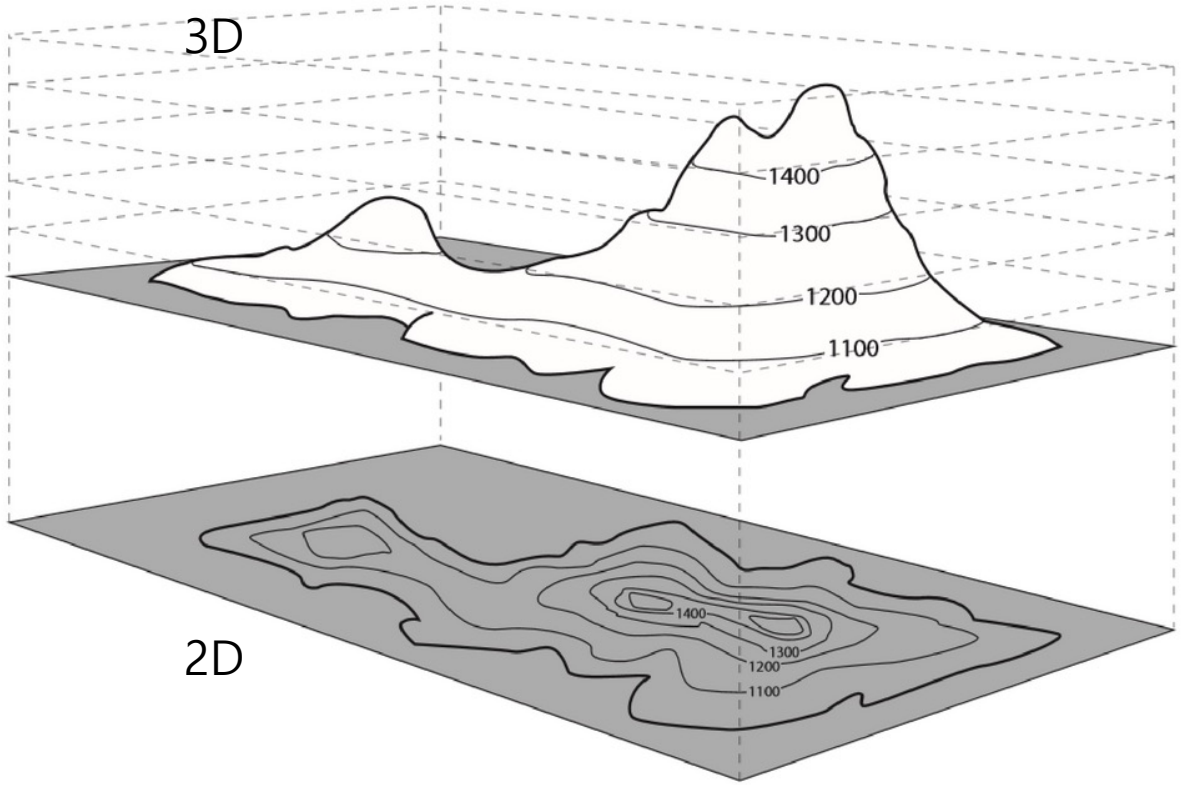
(a) without skip connections (b) with skip connections

Figure 1: The loss surfaces of ResNet-56 with/without skip connections. The proposed filter normalization scheme is used to enable comparisons of sharpness/flatness between the two figures.

Warm Up

Contour Line





Ian J. Goodfellow, et al. "Qualitatively characterizing neural network optimization problems." *International Conference on Learning Representations (ICLR)*. 2015.
435 citations

Reference
Most Relevant

Published as a conference paper at ICLR 2015

QUALITATIVELY CHARACTERIZING NEURAL NETWORK OPTIMIZATION PROBLEMS

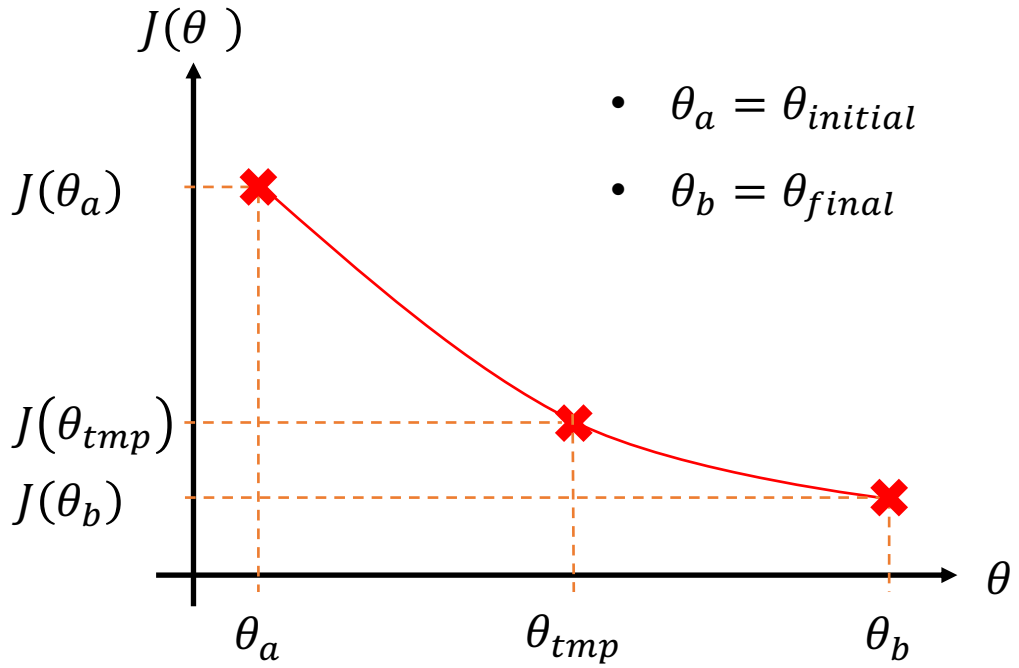
Ian J. Goodfellow* **& Oriol Vinyals*** **& Andrew M. Saxe****
*Google Inc., Mountain View, CA
**Department of Electrical Engineering, Stanford University, Stanford, CA
{goodfellow,vinyals}@google.com, asaxe@stanford.edu

Loss Interpolation

$$\theta = (1-\alpha)\theta_a + \alpha\theta_b \quad , \alpha \in (0,1)$$

$$J(\theta) = J((1-\alpha)\theta_a + \alpha\theta_b)$$

Evaluate loss @ parameter θ



Ian J. Goodfellow, et al. "Qualitatively characterizing neural network optimization problems." *International Conference on Learning Representations (ICLR)*. 2015.

Experimental Results

Loss convergence in training process

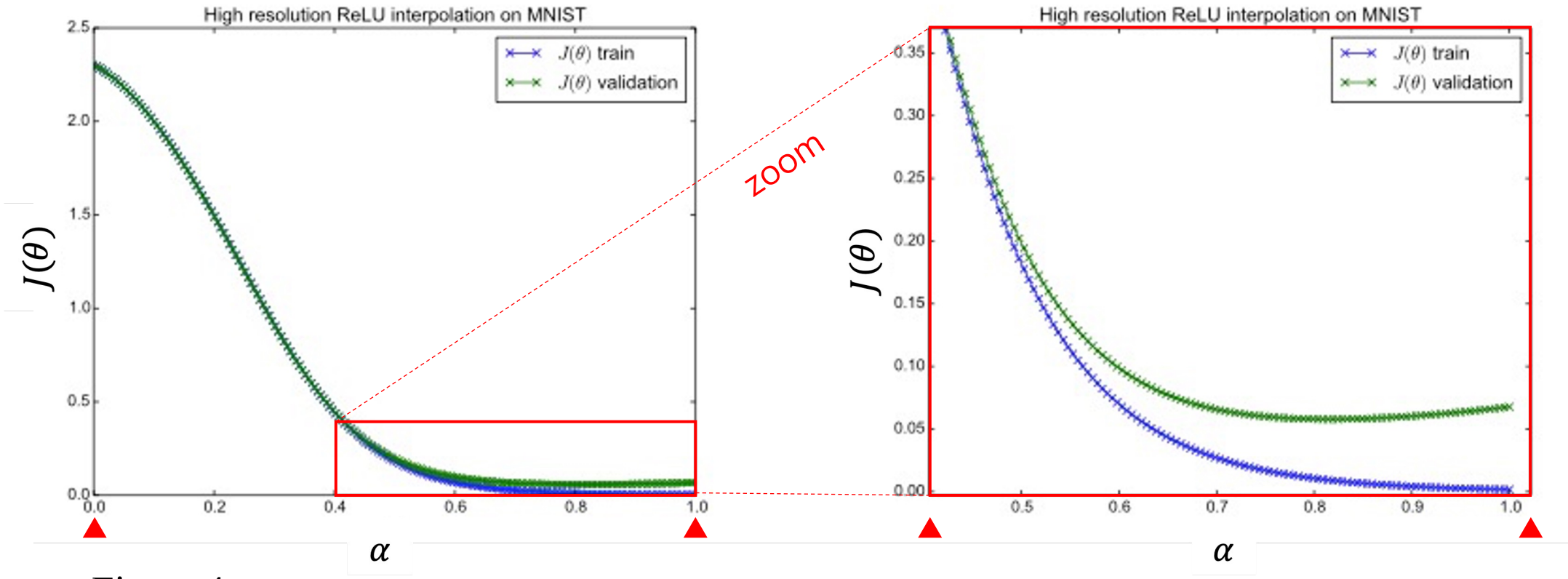
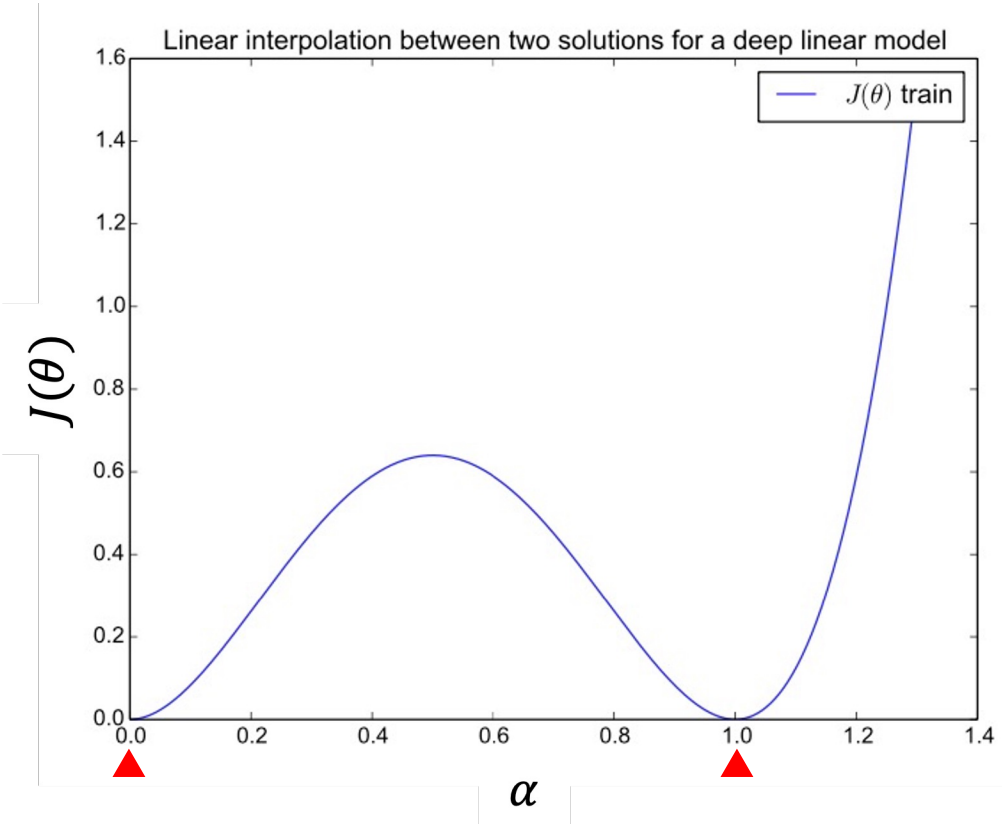


Figure 4.

- $\theta_0 = \theta_{initial}$
- $\theta_1 = \theta_{final}$

Experimental Results

Two solutions after training



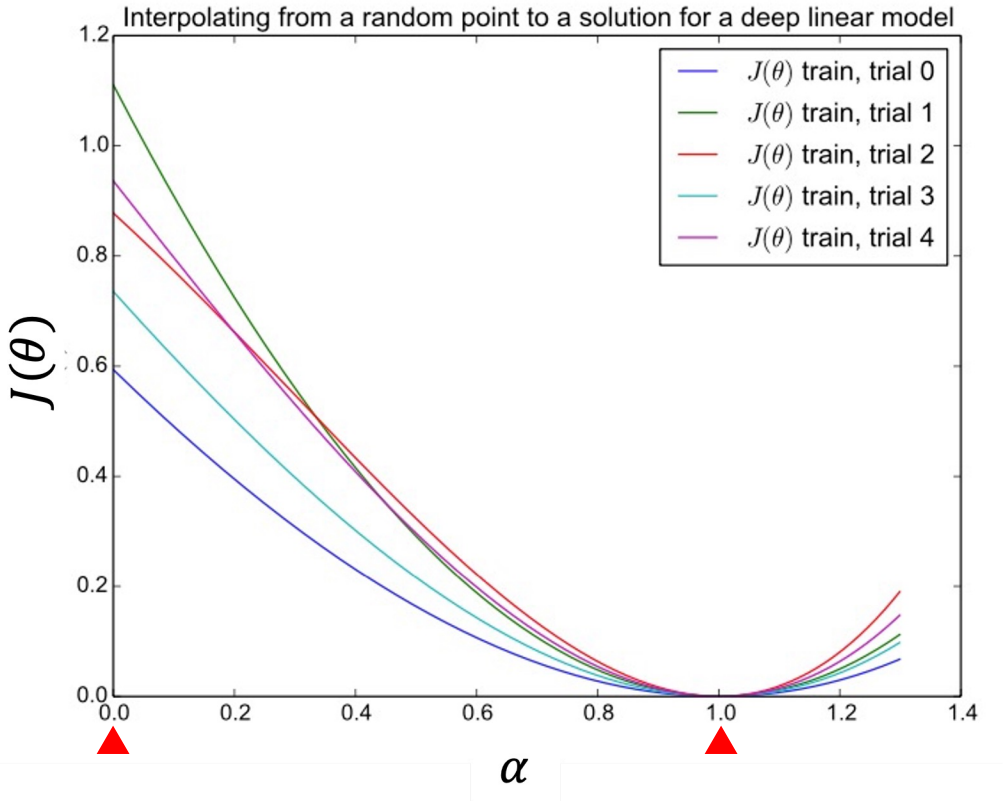


Figure 10.

- $\theta_0 = \theta_{solution\ A}$
- $\theta_1 = \theta_{solution\ B}$

- $\theta_0 = \theta_{random\ non-solution}$
- $\theta_1 = \theta_{solution}$

Visualizing
The Loss Landscape

Contributions

- Reveal the faults of existing visualization method.
- Propose a filter-wise normalization method to overcome the above.
- Conduct a comparative analysis of the loss surfaces (or curves) under various conditions.

Keskar, Nitish Shirish, et al. "On large-batch training for deep learning: Generalization gap and sharp minima." *International Conference on Learning Representations (ICLR)*. 2016.

Background

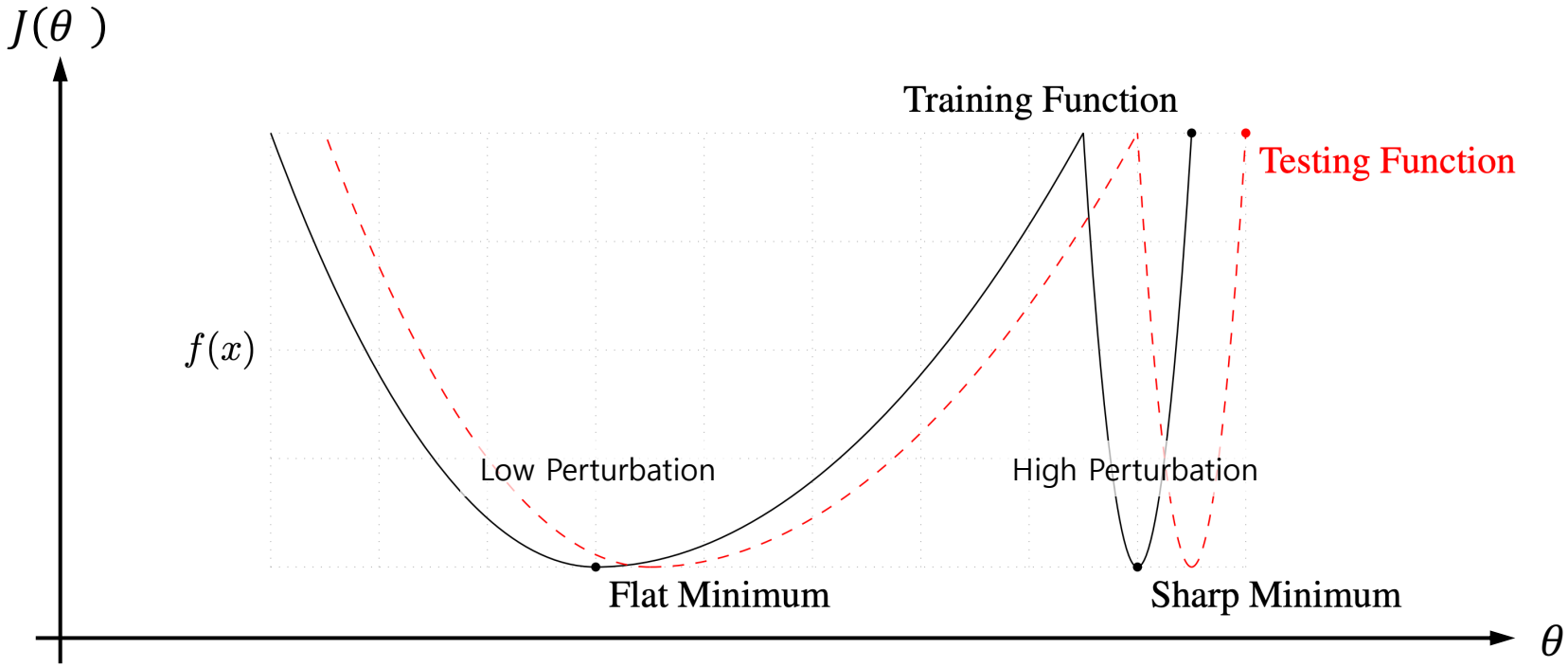
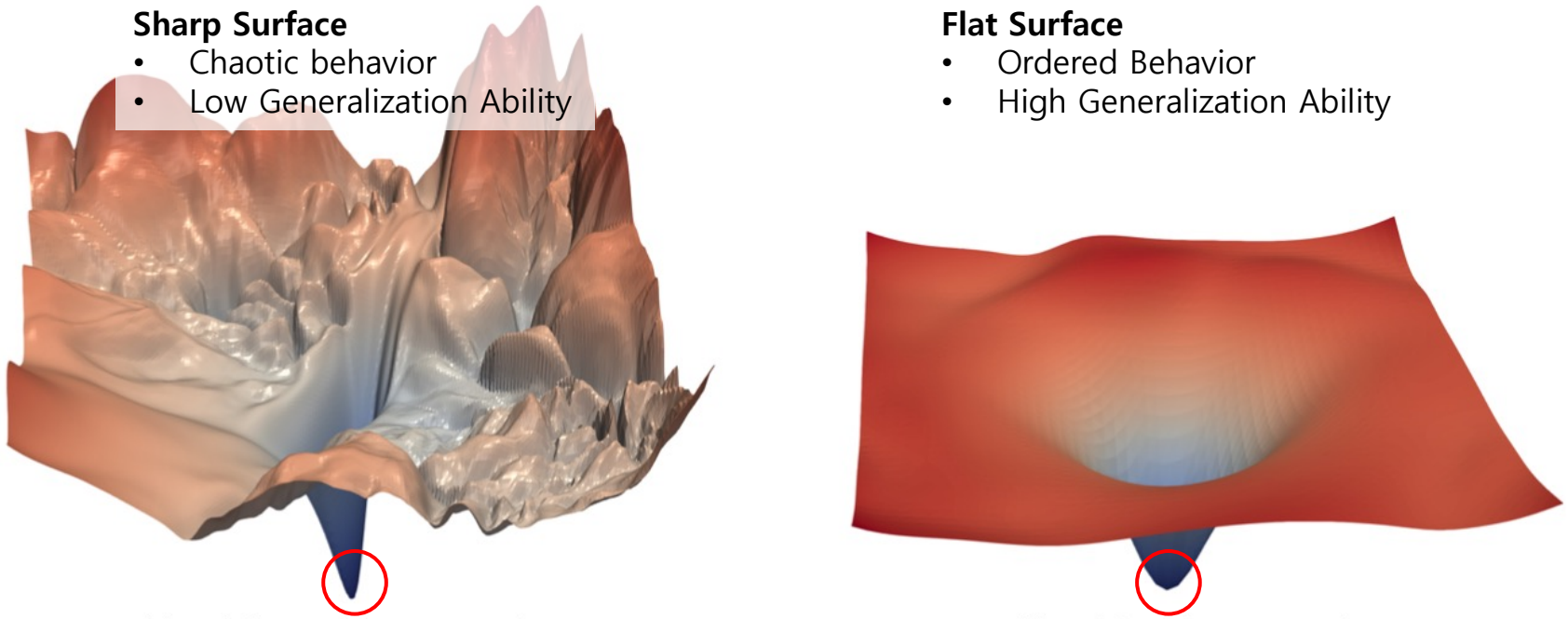


Figure 1: A Conceptual Sketch of Flat and Sharp Minima. The Y-axis indicates value of the loss function and the X-axis the variables (parameters)

Skip Connections and Generalization

Sharp Surface
- Chaotic behavior
- Low Generalization Ability

Flat Surface
- Ordered Behavior
- High Generalization Ability



(a) without skip connections (b) with skip connections

Figure 1: The loss surfaces (or landscapes) of ResNet-56 with/without skip connections. The left side (w/o skip connection) shows a fluctuated surface that represents the neural network will have chaotic behavior.

* Ian J. Goodfellow, et al. "Qualitatively characterizing neural network optimization problems." *International Conference on Learning Representations (ICLR)*. 2015.
** Krogh, Anders, et al. "A simple weight decay can improve generalization." *Advances in neural information processing systems (NeurIPS)*. 1991.

$$E(w) = E_0(w) + \frac{1}{2}\lambda(w^T w)$$

Existing Method*

$$\theta = (1-\alpha)\theta^s + \alpha\theta^l$$

- θ^s: solution @ small batch
- θ^l: solution @ large batch

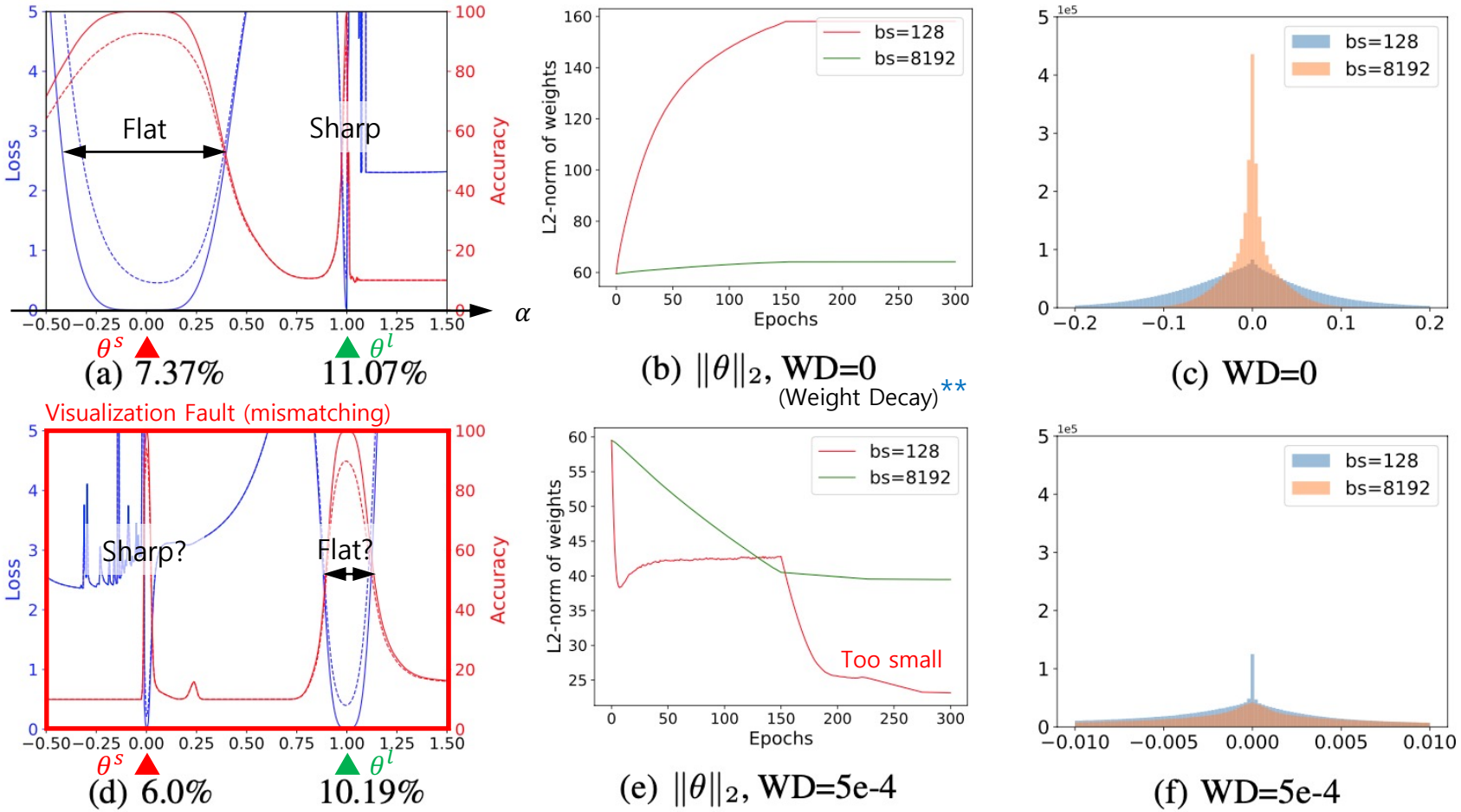


Figure 2: Results of training using VGG-9 with CIFAR-10 dataset.
The solid and dash lines represent training and test curve.

Limitation of Existing Method

- Weight Decay (WD) helps to reduce test error in both small and large batch training.

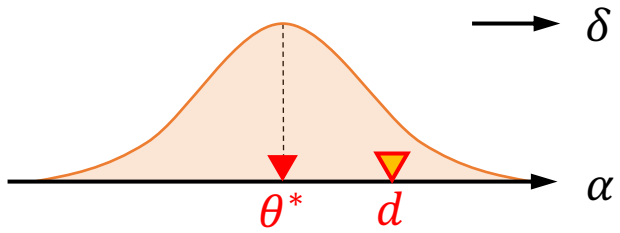
- However, the sharpness balance is flipped in practice.
  - A lower test error is achieved by applying WD in a small batch case.
  - But it shows a sharper curve than the case without WD.
  - Note that, a flat curve should be obtained in this case.

- Because WD makes the weight norm more smaller.
  - In small batch training, weight updates per epoch are conducted more frequently.
  - Due to frequent penalties, a small batch case will have a very small norm.
  - Small weight norm is sensitive to small changes and it makes a sharp curve.

Proposed Method

Gaussian Sampling

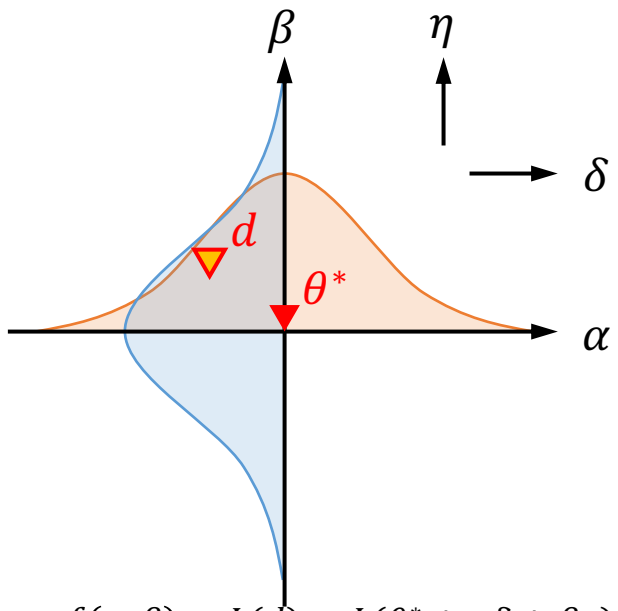
1D Case



$$f(\alpha, \beta) = L(d) = L(\theta^* + \alpha\delta)$$

$$d_{i,j} = {\theta_{i,j}}^* + \alpha\delta$$

2D Case



$$f(\alpha, \beta) = L(d) = L(\theta^* + \alpha\delta + \beta\eta)$$

$$d_{i,j} = {\theta_{i,j}}^* + \alpha\delta + \beta\eta$$

- θ^*: solution parameter (weight)
- ${\theta_{i,j}}^*$: solution parameter at i-th layer and j-th filter
- $d_{i,j}$: Gaussian direction vector

Filter-Wise Normalization

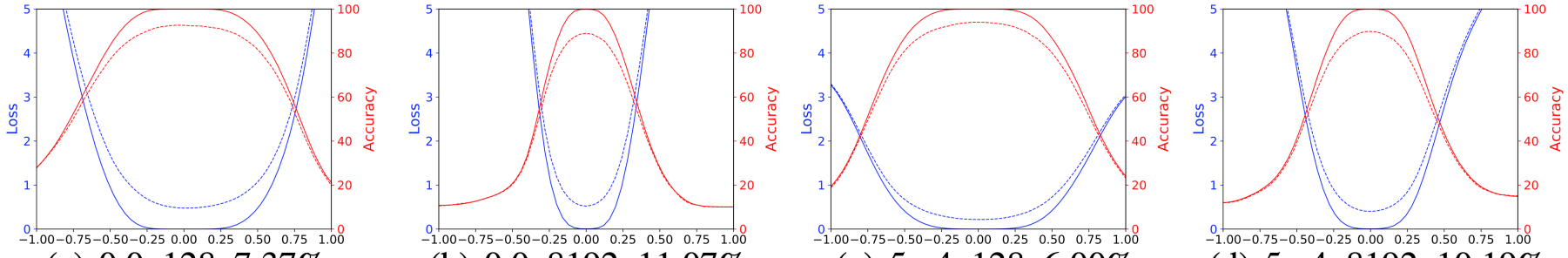
$$d_{i,j} \leftarrow \frac{d_{i,j}}{||d_{i,j}||} ||{\theta_{i,j}}^*||$$

The above enables weight norm-invariant visualization!
(Can express original characteristics without distortion.)

That’s it!

Loss Surface Visualization
with Filter-Wise Normalization

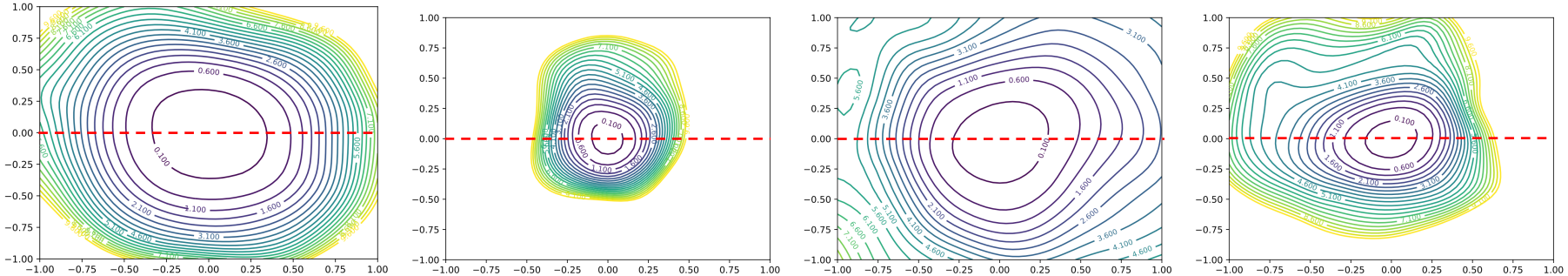
1D



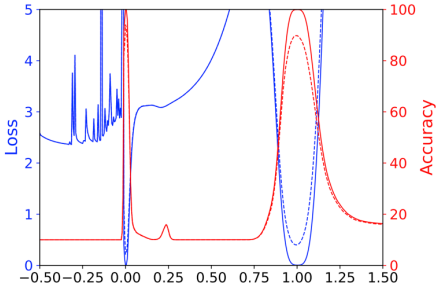
(a) 0.0, 128, 7.37% (b) 0.0, 8192, 11.07% (c) 5e-4, 128, 6.00% (d) 5e-4, 8192, 10.19%

(WD, Batch, Test Error)

2D



(e) 0.0, 128, 7.37% (f) 0.0, 8192, 11.07% (g) 5e-4, 128, 6.00% (h) 5e-4, 8192, 10.19%



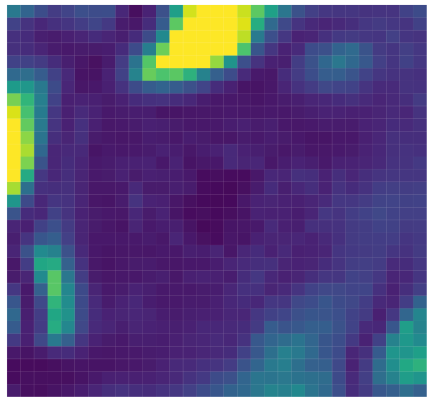
(d) 6.0% 10.19%

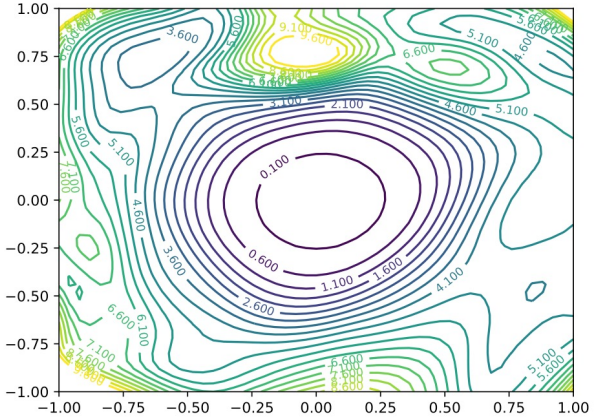
Figure 2: WD=5e-4.
The small batch case achieves lower test error but shows a sharper curve than the large batch case.

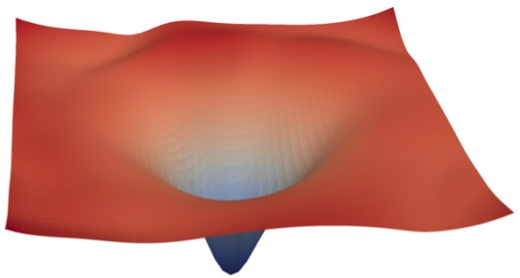
Figure 3: Solution visualization on 1D and 2D spaces. The 1D cases are marked with the red dash in 2D cases. Unlike the previous method, even when weight decay is applied, the loss surface is correctly displayed as a flat form for a small batch case that has a small test error.

Visualization Process
for 3D Loss Landscape

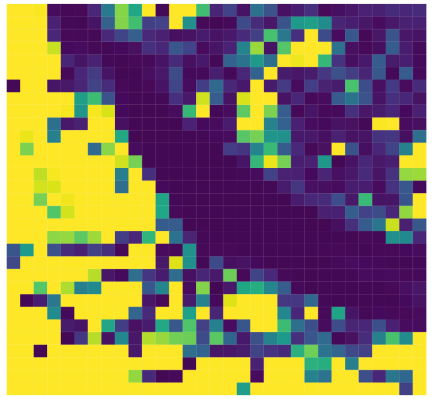
ResNet-56 (w/ skip connection)
Error rate: 5.89%

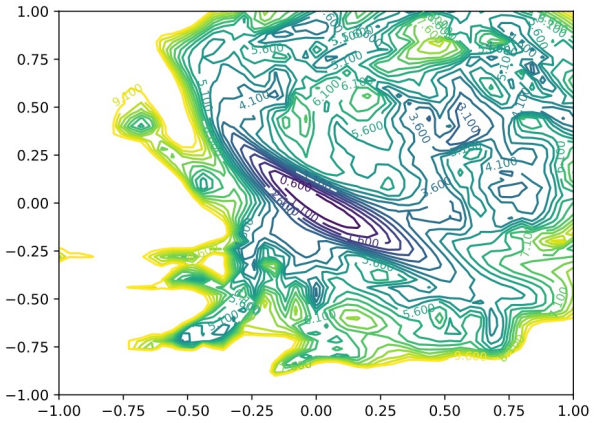






ResNet-56 (w/o skip connection)
Error rate: 13.31%





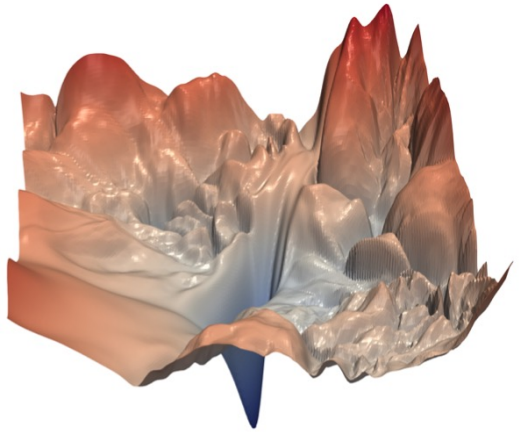


Figure 7
2D loss surface

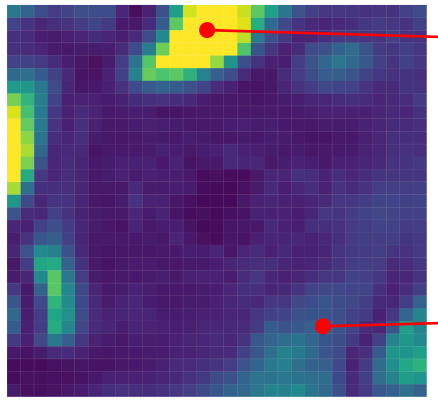
Figure 5
Loss contour

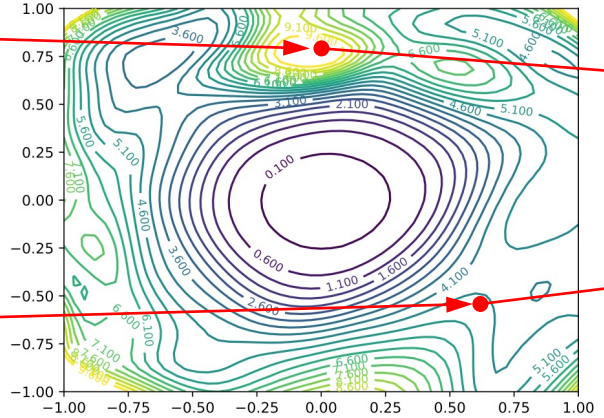
Figure 1
3D loss surface (landscape)

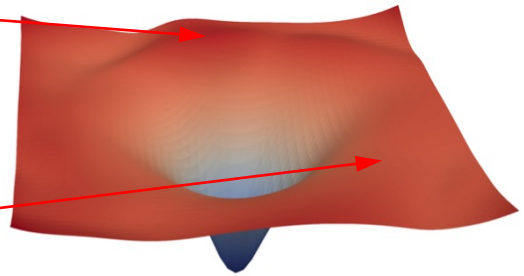
Visualization Process

for 3D Loss Landscape

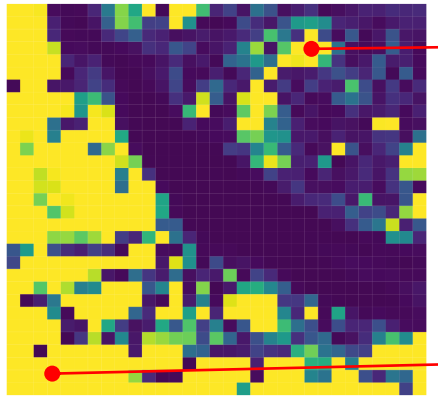
ResNet-56 (w/ skip connection)
Error rate: 5.89%

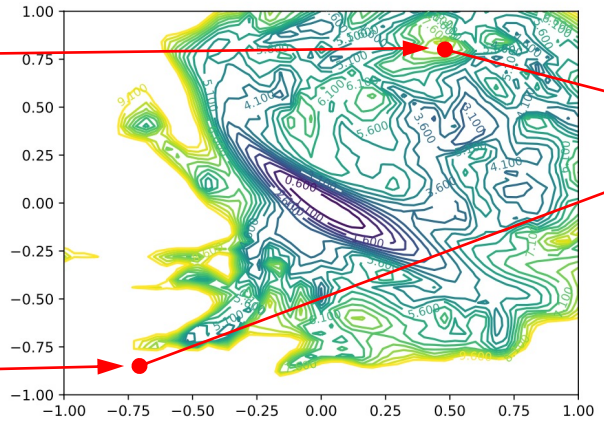






ResNet-56 (w/o skip connection)
Error rate: 13.31%





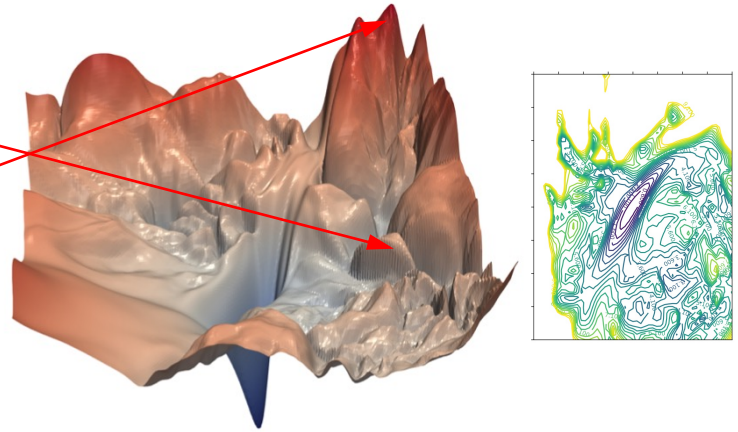


Figure 7
2D loss surface

Figure 5
Loss contour

Figure 1
3D loss surface (landscape)

Usage

Zagoruyko, Sergey, et, al. "Wide Residual Networks." *British Machine Vision Conference (BMVC)*, 2016.

Usage [1/2]

group name	output size	block type = $B(3,3)$
conv1	32×32	$[3\times3,\ 16]$
conv2	32×32	$\begin{bmatrix} 3\times3,\ 16\times k \\ 3\times3,\ 16\times k \end{bmatrix} \times N$
conv3	16×16	$\begin{bmatrix} 3\times3,\ 32\times k \\ 3\times3,\ 32\times k \end{bmatrix} \times N$
conv4	8×8	$\begin{bmatrix} 3\times3,\ 64\times k \\ 3\times3,\ 64\times k \end{bmatrix} \times N$
avg-pool	1×1	$[8 \times 8]$

Table 1: Structure of the Wide-ResNet (WRN).

depth	k	# params	CIFAR-10	CIFAR-100
40	1	0.6M	6.85	30.89
40	2	2.2M	5.33	26.04
40	4	8.9M	4.97	22.89
40	8	35.7M	4.66	-
28	10	36.5M	**4.17**	20.50
28	12	52.5M	4.33	**20.43**
22	8	17.2M	4.38	21.22
22	10	26.8M	4.44	20.75
16	8	11.0M	4.81	22.07
16	10	17.1M	4.56	21.59

Shallow ↓ (depth) / Wide → (k)

Table 4: Summarized result of comparative experiments. Shallow and wide structure is better than deep and thin structure.

Q. Can confirm the superiority of the neural network structures without empirical evaluation?

A. Partially, Yes.

Usage [2/2]

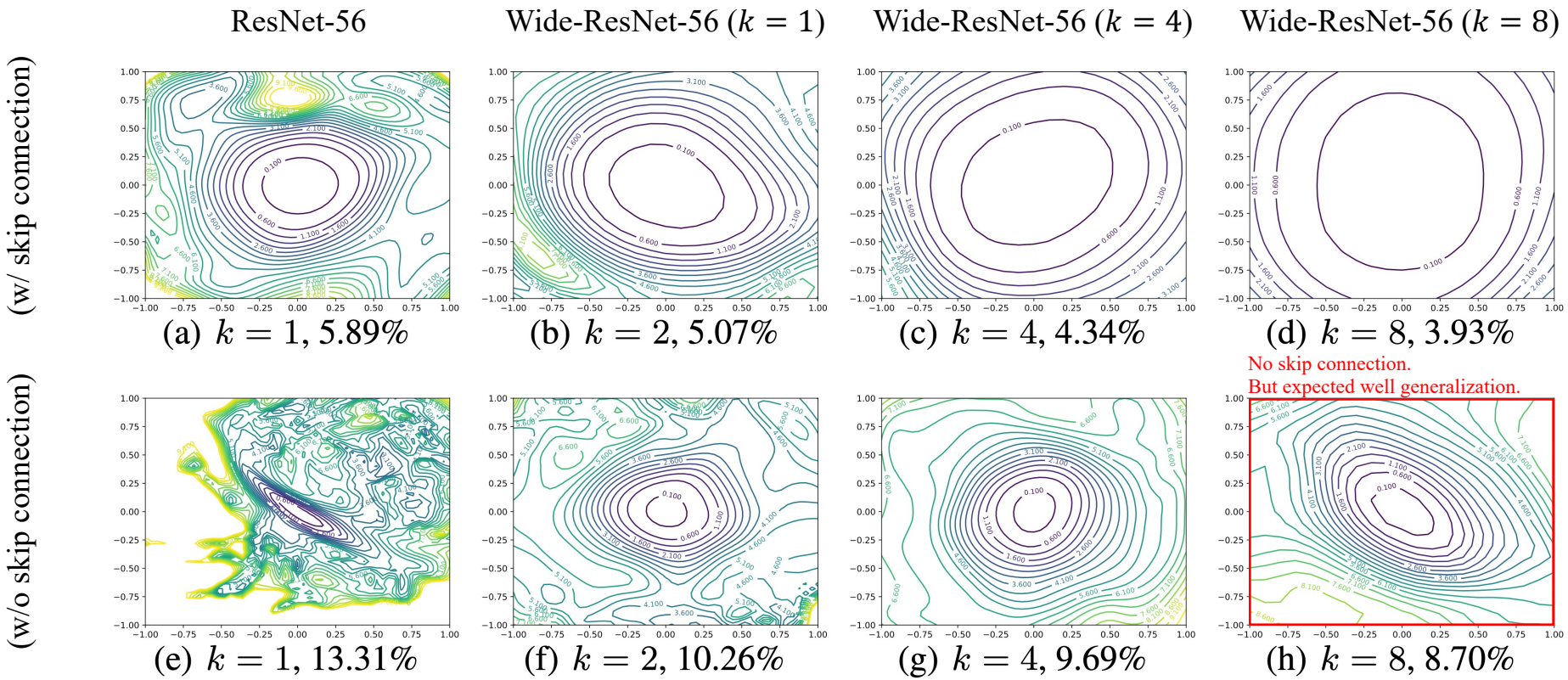


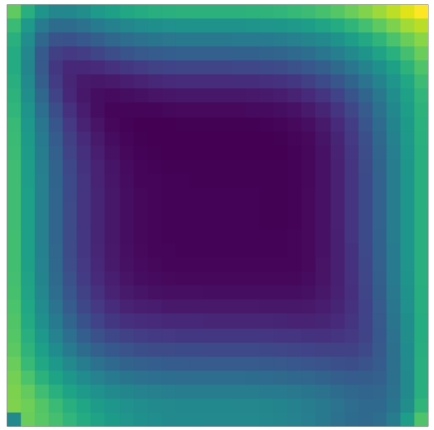
Figure 6: Wide-ResNet-56 on CIFAR-10 both with shortcut connections (top) and without (bottom). The label $k = 2$ means twice as many filters per layer. Test error is reported below each figure.

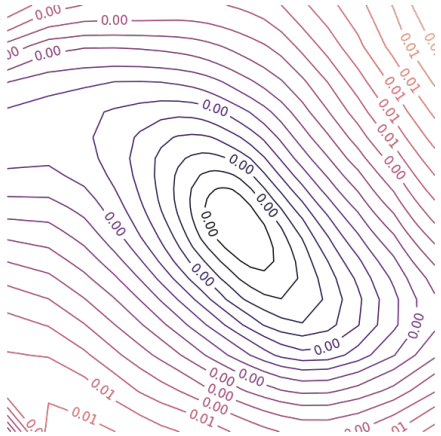
Appendix A

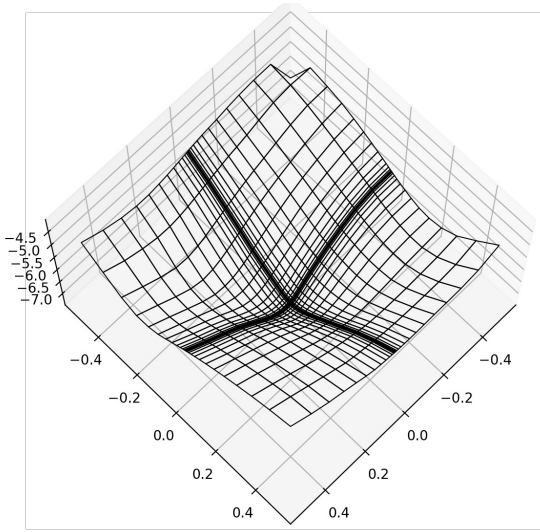
Loss Landscape for FL method

Loss Landscape [1/3]

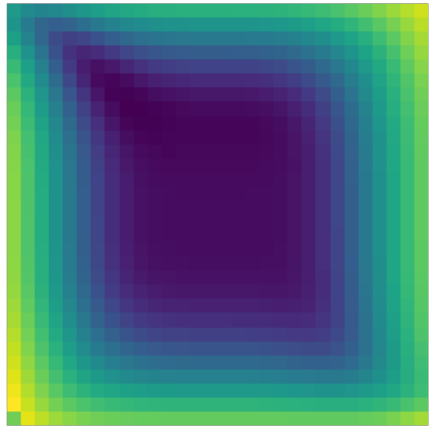
L2

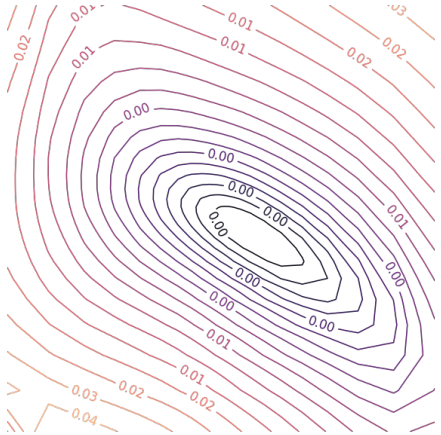


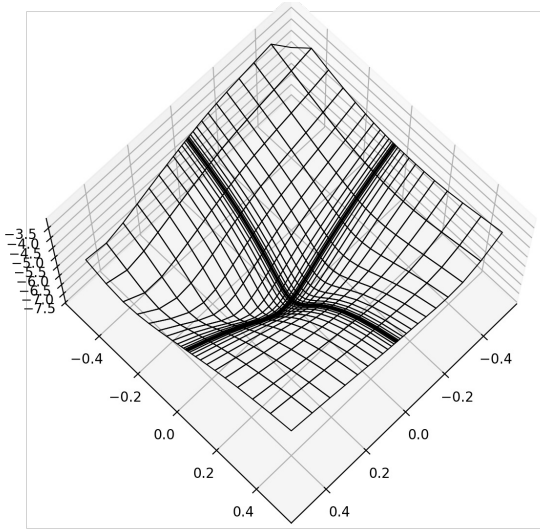




FLL2

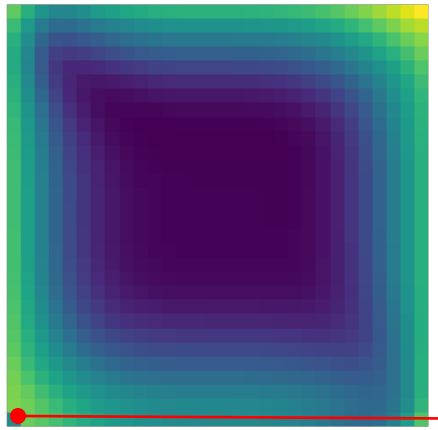


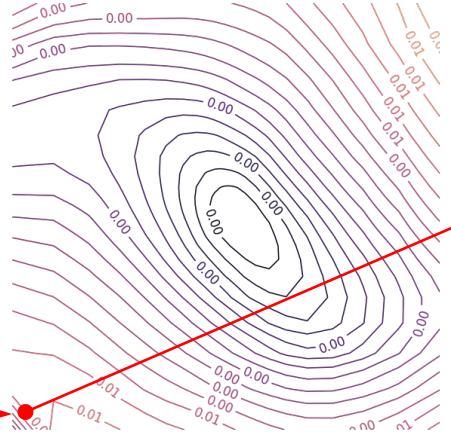


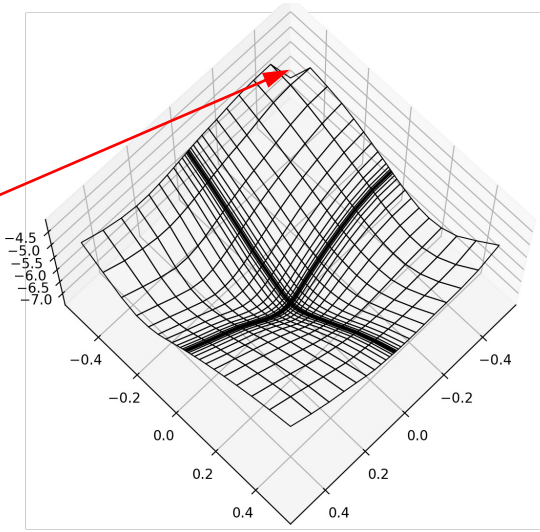


Loss Landscape [2/3]

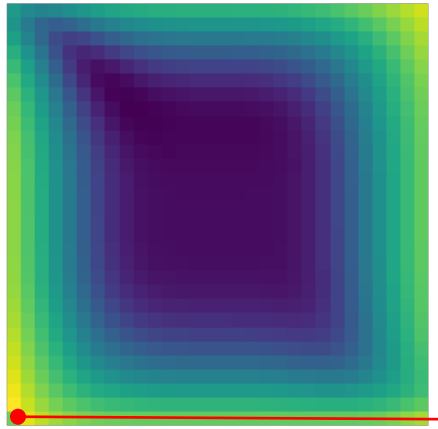
L2

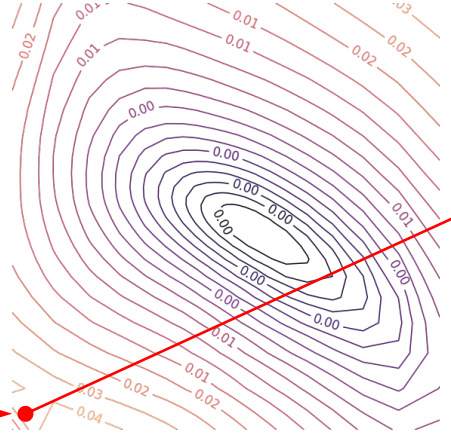


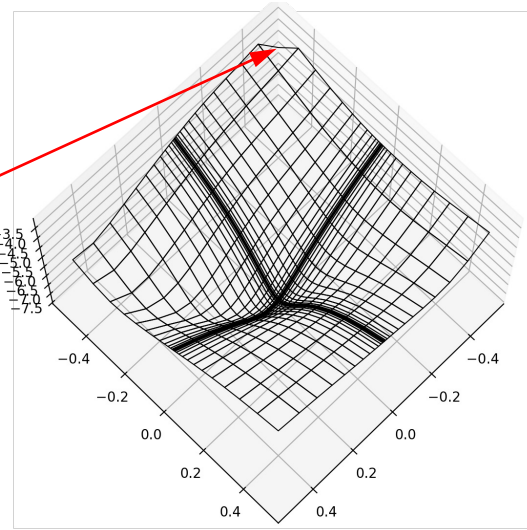




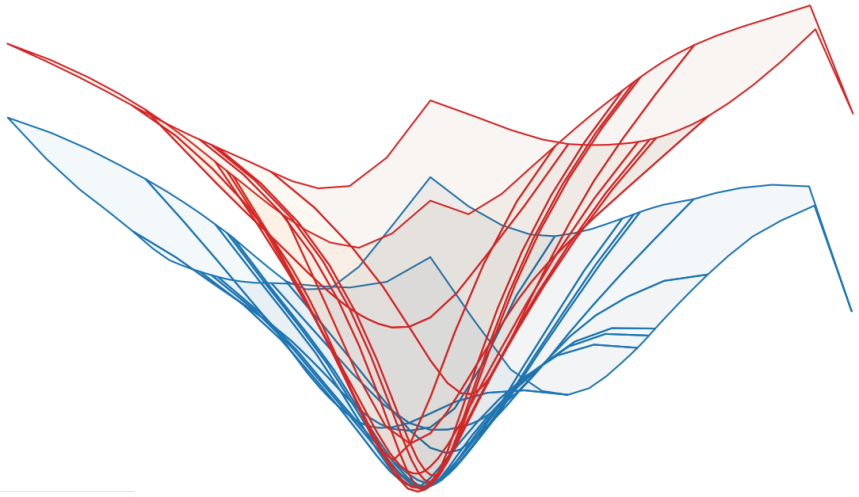
FLL2



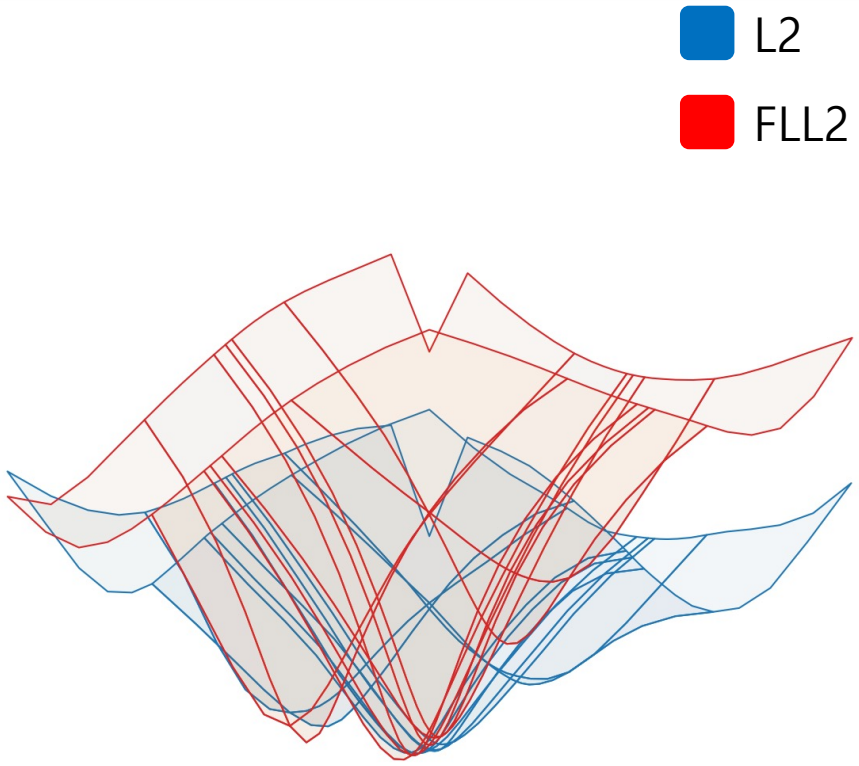




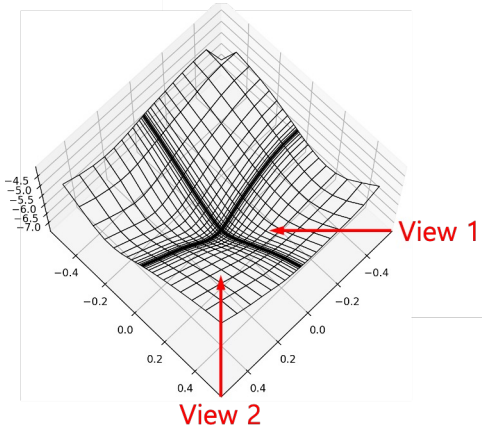
Loss Landscape [3/3]



View 1



View 2



Appendix B

Source Code

Official Source Code [1/2]

Visualizing the Loss Landscape of Neural Nets

Hao Li[1], **Zheng Xu**[1], **Gavin Taylor**[2], **Christoph Studer**[3], **Tom Goldstein**[1]
[1]University of Maryland, College Park [2]United States Naval Academy [3]Cornell University
{haoli,xuzh,tomg}@cs.umd.edu, taylor@usna.edu, studer@cornell.edu

Tom Goldstein
tomgoldstein

loss-landscape Public

Code for visualizing the loss landscape of neural nets

Python ☆ 2.1k 305

Official Source Code [2/2]

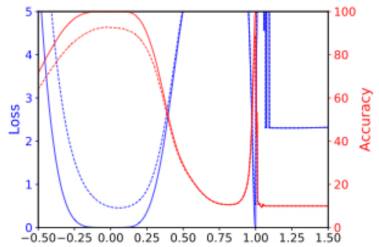
Visualizing 1D loss curve

Creating 1D linear interpolations

The 1D linear interpolation method [1] evaluates the loss values along the direction between two minimizers of the same network loss function. This method has been used to compare the flatness of minimizers trained with different batch sizes [2]. A 1D linear interpolation plot is produced using the `plot_surface.py` method.

```
mpirun -n 4 python plot_surface.py --mpi --cuda --model vgg9 --x=-0.5:1.5:401 --dir_type states \
--model_file cifar10/trained_nets/vgg9_sgd_lr=0.1_bs=128_wd=0.0_save_epoch=1/model_300.t7 \
--model_file2 cifar10/trained_nets/vgg9_sgd_lr=0.1_bs=8192_wd=0.0_save_epoch=1/model_300.t7 --plot
```

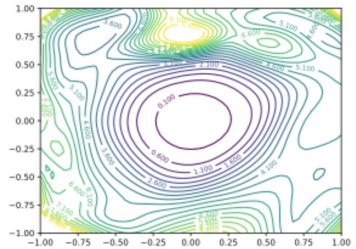
- `--x=-0.5:1.5:401` sets the range and resolution for the plot. The x-coordinates in the plot will run from -0.5 to 1.5 (the minimizers are located at 0 and 1), and the loss value will be evaluated at 401 locations along this line.
- `--dir_type states` indicates the direction contains dimensions for all parameters as well as the statistics of the BN layers (`running_mean` and `running_var`). Note that ignoring `running_mean` and `running_var` cannot produce correct loss values when plotting two solutions togeather in the same figure.
- The two model files contain network parameters describing the two distinct minimizers of the loss function. The plot will interpolate between these two minima.



Visualizing 2D loss contours

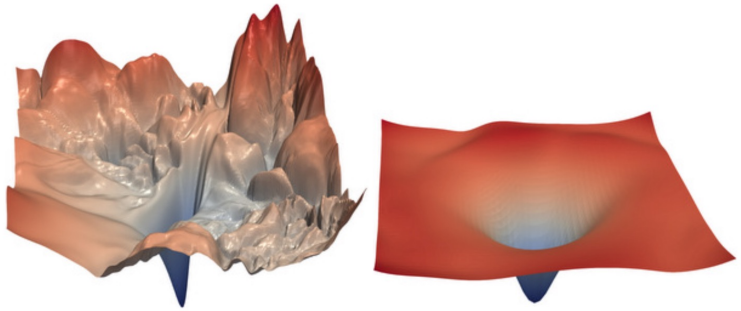
To plot the loss contours, we choose two random directions and normalize them in the same way as the 1D plotting.

```
mpirun -n 4 python plot_surface.py --mpi --cuda --model resnet56 --x=-1:1:51 --y=-1:1:51 \
--model_file cifar10/trained_nets/resnet56_sgd_lr=0.1_bs=128_wd=0.0005/model_300.t7 \
--dir_type weights --xnorm filter --xignore biasbn --ynorm filter --yignore biasbn  --plot
```



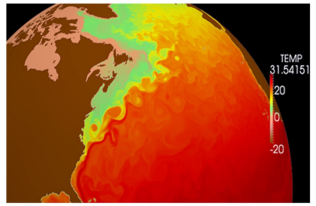
Visualizing 3D loss surface

`plot_2D.py` can make a basic 3D loss surface plot with `matplotlib`. If you want a more detailed rendering that uses lighting to display details, you can render the loss surface with ParaView.



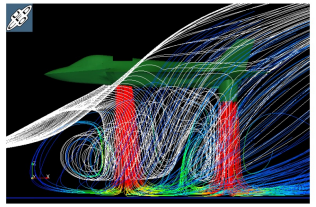
ParaView

ParaView Platform in Action



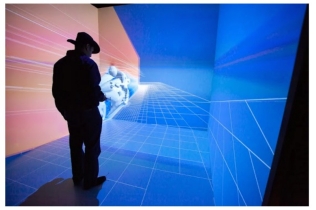
Climate Research

The Climate Data Analysis Tools (CDAT) project leverages ParaView with other open-source tools to enable analysts to track, monitor, and predict climate changes.



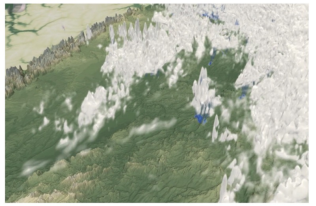
CFD Simulations

Computational fluid dynamics (CFD) simulations with ParaView enable aviation teams to study lift and drag, and thereby improve design efficiency.



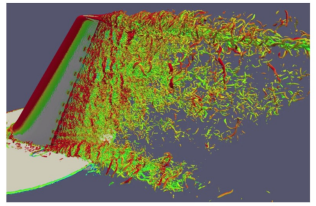
Immersive Data

By immersing themselves in the data with ParaView, researchers can interactively explore data in a more intuitive manner.



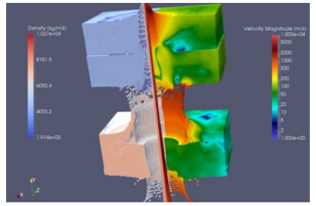
Cloud-resolving simulation

Cloud-resolving simulation over Germany using ICON HighRes



PHASTA and ParaView Catalyst

a high-fidelity CFD simulation of flow control applied to realistic wing profiles using PHASTA and ParaView Catalyst.



Shaped Charge Jet Penetration

ALEGRA simulation for shaped charge jet penetration in ceramic plates.

Visualizing a Twin Earth

NVIDIA Omniverse ParaView Connector allows scientists to interactively analyze weather and climate data and its impact on the Earth in high-resolution 3D.