Understanding Generalization through Visualizing the Loss Landscape

Speaker: Yi-Shan Wu

Institute of Information Science

Academia Sinica

Taiwan

Jun 28, 2019

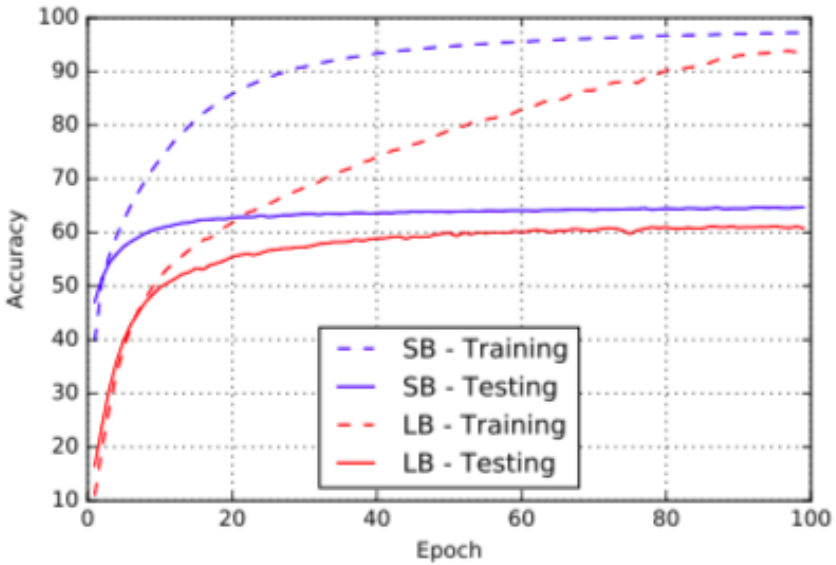
1 Introduction

Observations in Experiments

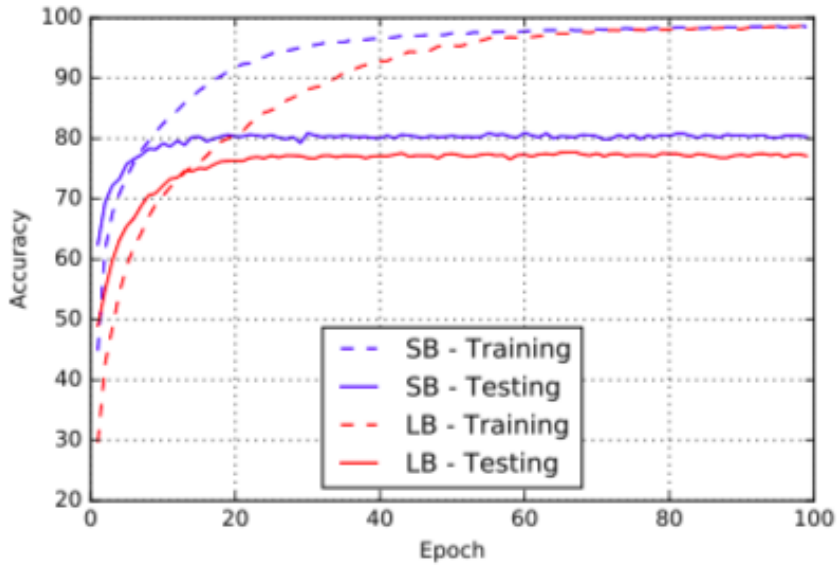
- Training with larger batch sizes decreases the generalization performance. [4] (Why ?)
- Skip connections produce loss functions that train easier, and produce minimizers that generalize better. [2]

Overfitting ?

A common reason for generalization gap comes from "overfitting". However, it is not the case here. If it is caused by overfitting, there would be a certain "peak" that testing accuracy starts to decay.



(a) Network F_2



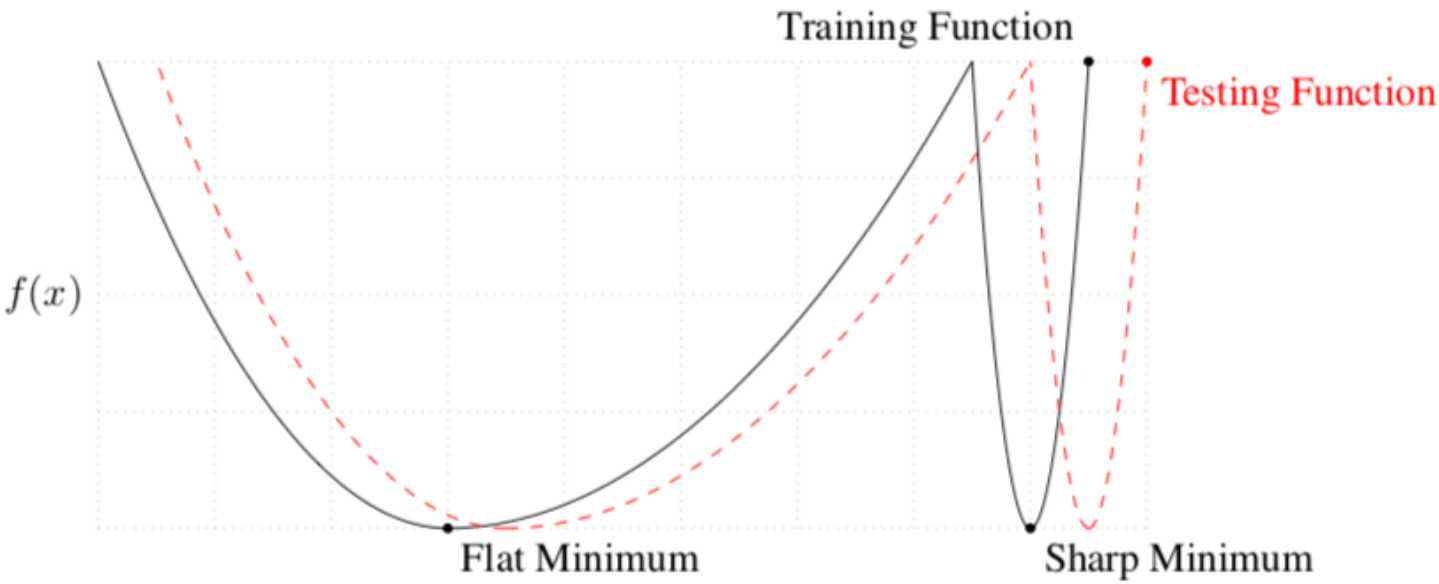
(b) Network C_1

Observations in Experiments

- Training with larger batch sizes decreases the generalization performance. (Why ?)
  - LB methods tend to converge to sharp minimizers while SB methods tend to converge to flat minimizers. [4]

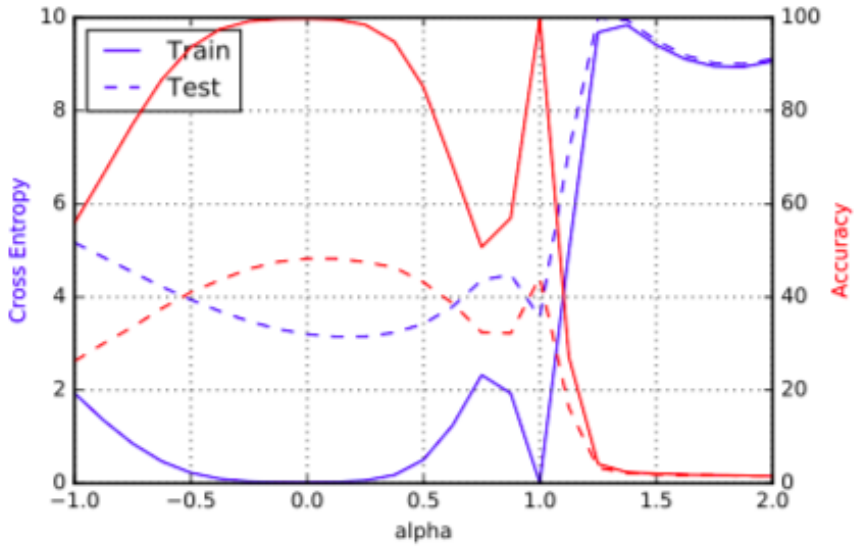
Flatness/Sharpness[4]

(Sharp/Flat) minimizers are characterized by a number of (large positive/small) eigenvalues in $\nabla^2 f(x)$. The conceptual sketch of flatness and sharpness :

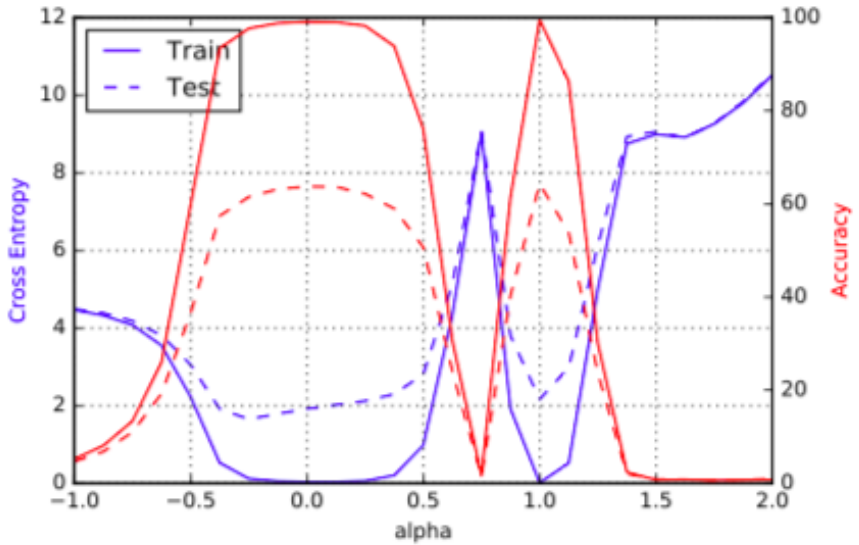


Flatness/Sharpness[4]

Use parametric 1-D plots (by [5]) :
Let x_s^* and x_ℓ^* indicate the solutions trained on SB(256) and LB(2000). Plot the function $f(\alpha x_\ell^* + (1-\alpha)x_s^*)$, with $\alpha \in [-1, 2]$.



(e) C_3



(f) C_4

Questions about the observation in [4]

LB methods tend to converge to sharp minimizers while SB methods tend to converge to flat minimizers. [4]

- What causes two methods to converge to different minimizers ?
- If so, then why ?
- Is it possible, through algorithmic or regulatory means to steer LB methods away from sharp minimizers ?

Question 1

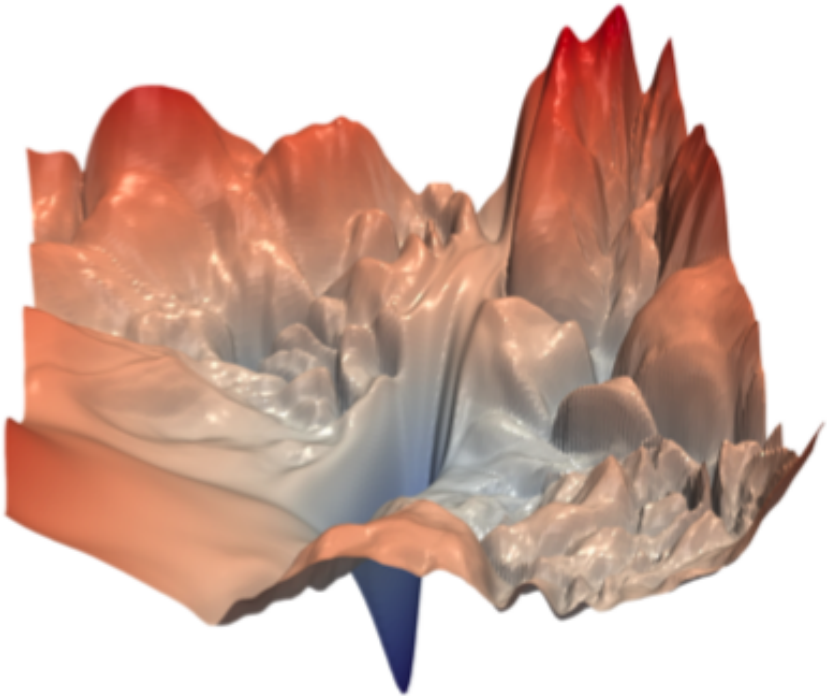
What causes two methods to converge to different minimizers ?

When increasing the batch size for a problem, there exists a threshold after which the performance decreases.
Mini-batch gradients can be interpreted as gradient descent using noisy gradients. The noise in the gradient pushes the iterates out of the basin of attraction of sharp minimizers.
When the batch size is greater than the threshold mentioned above, the noise is not sufficient to cause ejection from the basin.

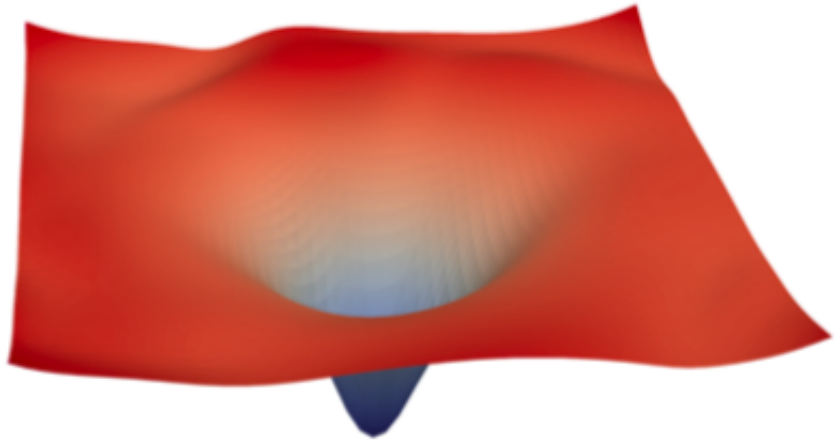
Observations in Experiments

- Training with larger batch sizes decreases the generalization performance. [4] (Why ?)
- Skip connections produce loss functions that train easier, and produce minimizers that generalize better. [2] (Why ?)
  - Answer through visualize the loss landscape [2]

ResNet-56 with/without skip connection



(a) without skip connections



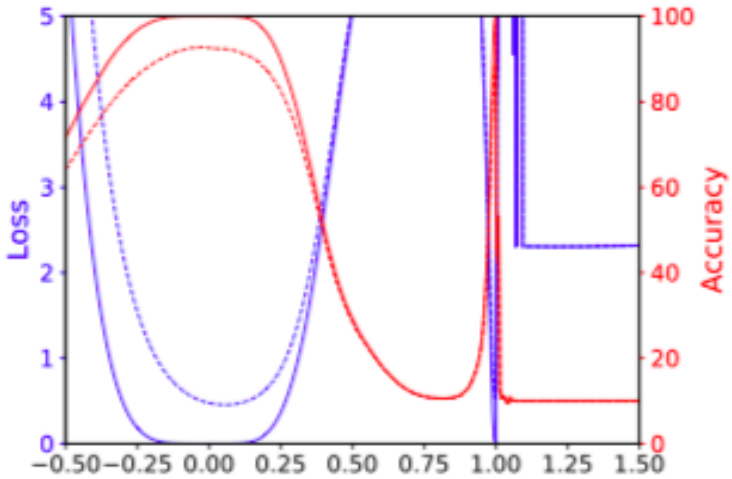
(b) with skip connections

Visualization

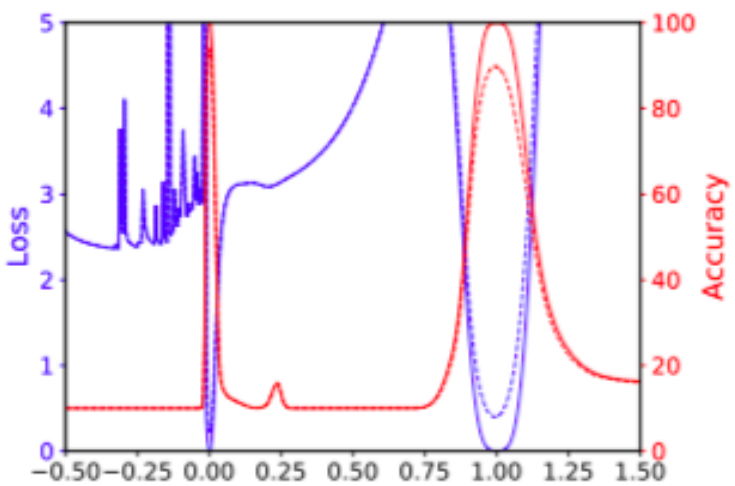
1D linear interpolation method suffers from several weaknesses.

- difficult to visualize non-convexities using 1D plots
- does not consider batch normalization or invariance symmetries in the network

scale invariance : A network's behavior remains unchanged if we rescale the weights when batch normalization is used. [2] [3]



(a) 7.37% 11.07%



(d) 6.0% 10.19%

Visualization

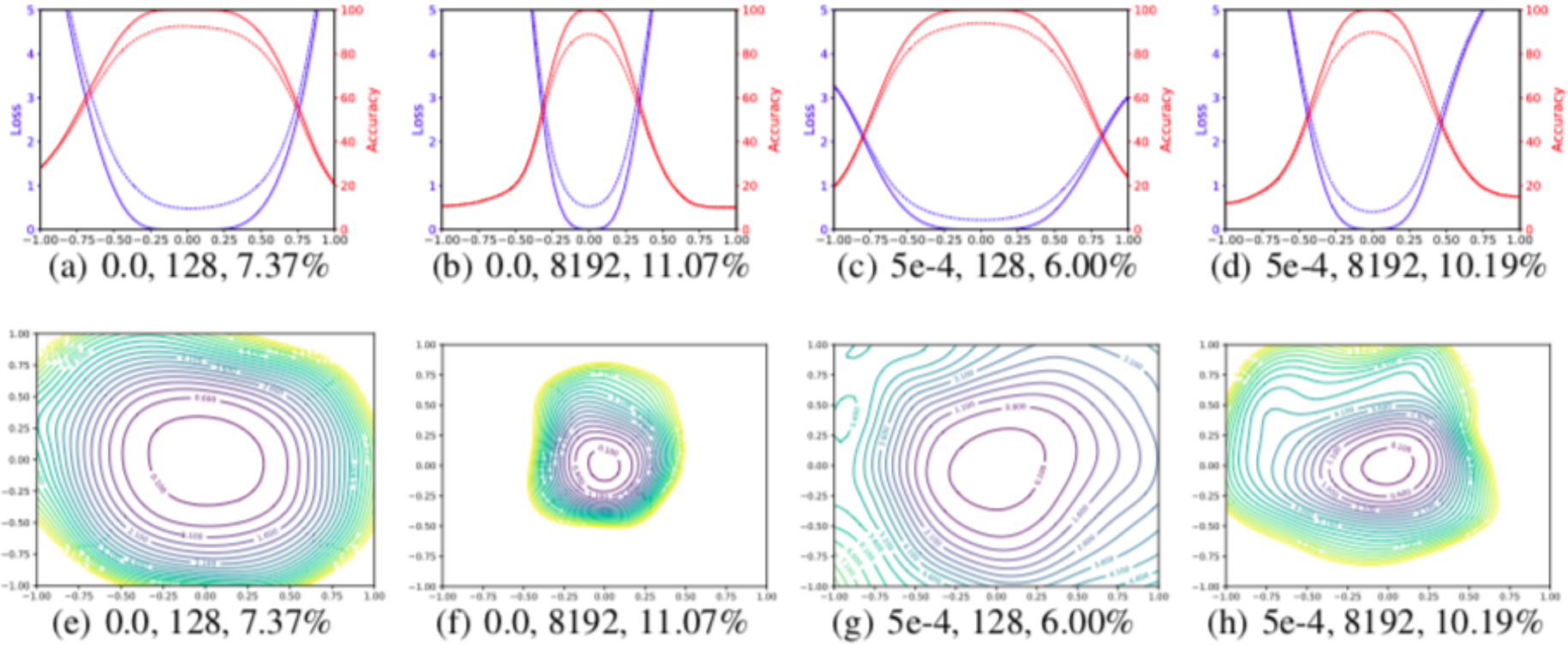
2D method :

1. chooses a center point θ^*
2. chooses two direction vectors v_1, v_2
3. plot the function $f(\alpha, \beta) = L(\theta^* + \alpha v_1 + \beta v_2)$

Filter-Wise Normalization :

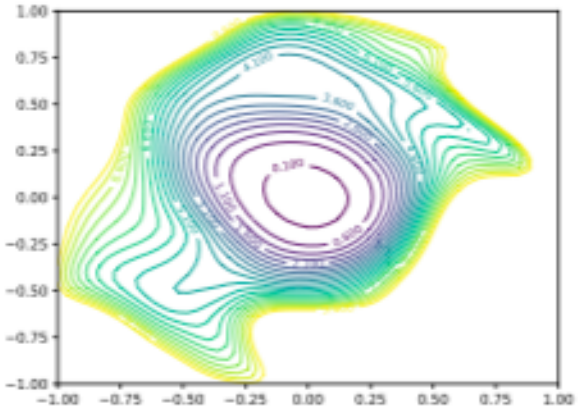
- 2D method with appropriate scaling
- $v_{i,j} \leftarrow v_{i,j} \cdot (\|\theta^*_{i,j}\| / \|v_{i,j}\|)$, where i, j denotes the j'th filter of the i'th layer of $v_{i,j}$

Batch-Size experiment revisit [2]

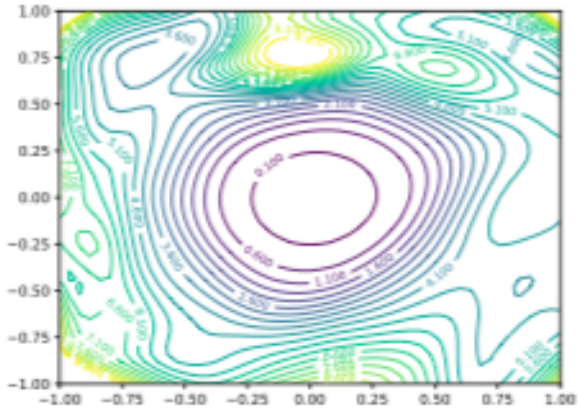


weight decay, batch size, and test error

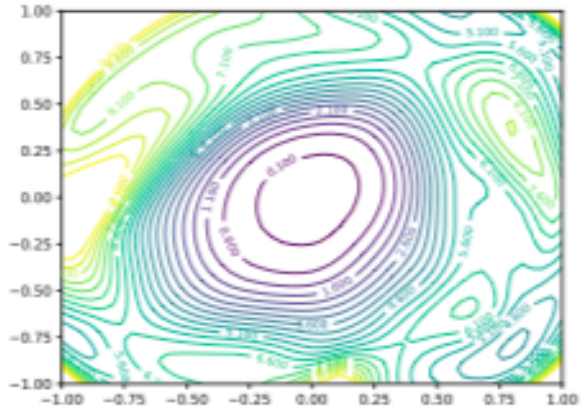
skip-connection and depth [2]



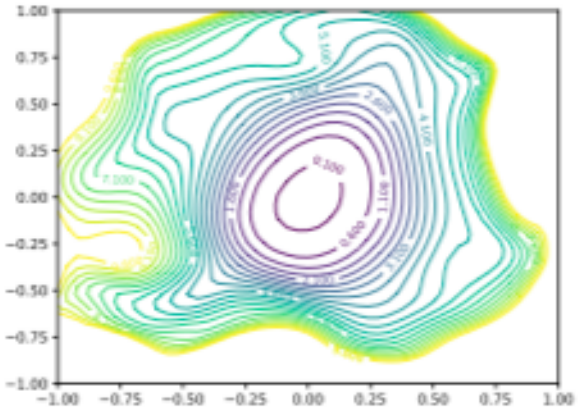
(a) ResNet-20, 7.37%



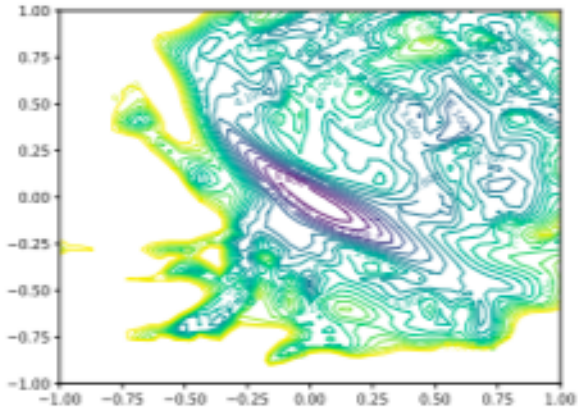
(b) ResNet-56, 5.89%



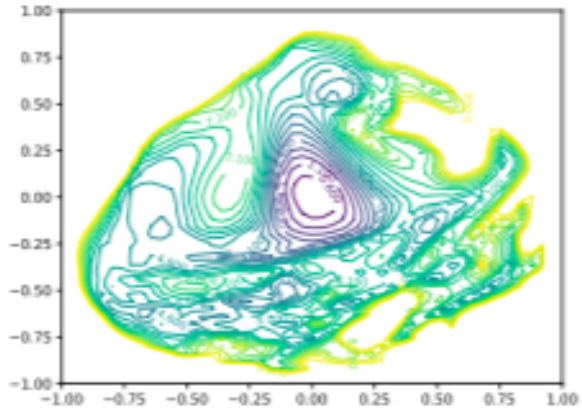
(c) ResNet-110, 5.79%



(d) ResNet-20-NS, 8.18%



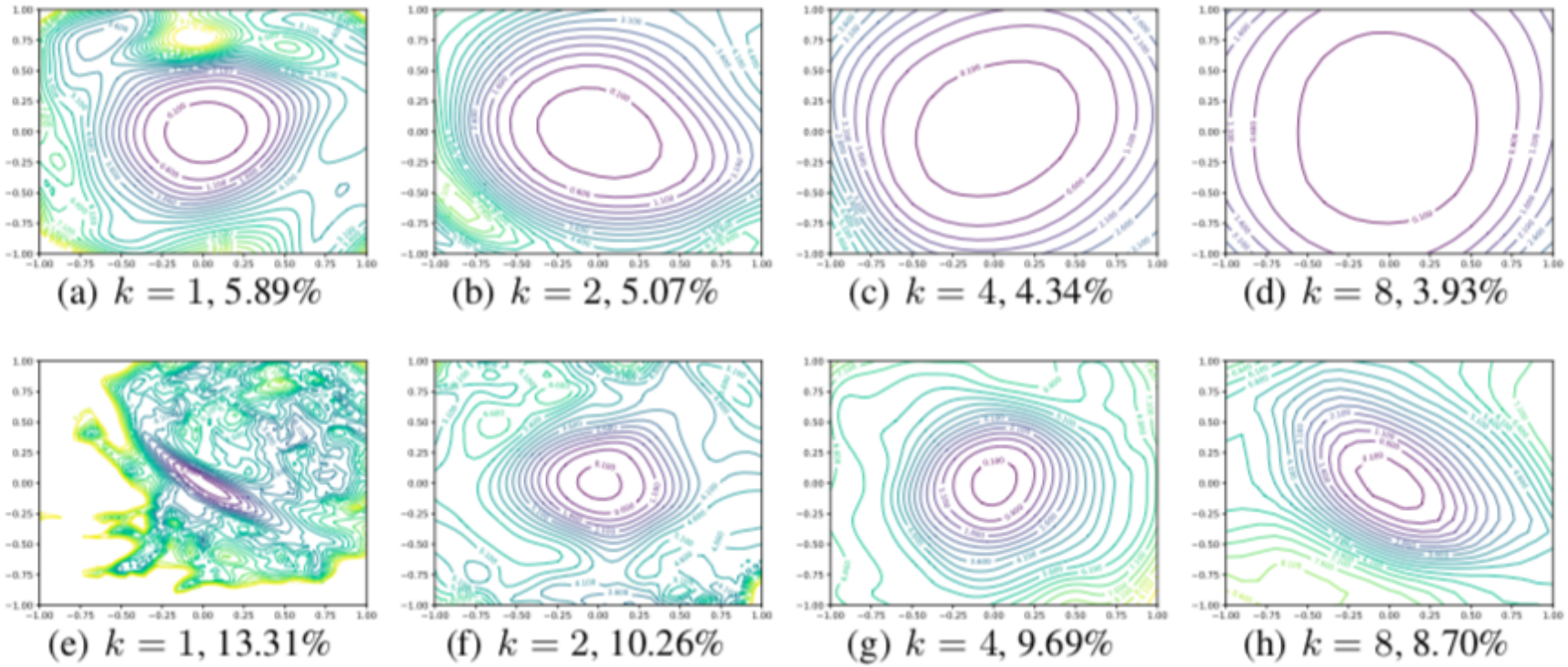
(e) ResNet-56-NS, 13.31%



(f) ResNet-110-NS, 16.44%

NS : no skip-connection

Wide Models vs Thin Models [2]



(a) $k = 1, 5.89\%$ (b) $k = 2, 5.07\%$ (c) $k = 4, 4.34\%$ (d) $k = 8, 3.93\%$

(e) $k = 1, 13.31\%$ (f) $k = 2, 10.26\%$ (g) $k = 4, 9.69\%$ (h) $k = 8, 8.70\%$

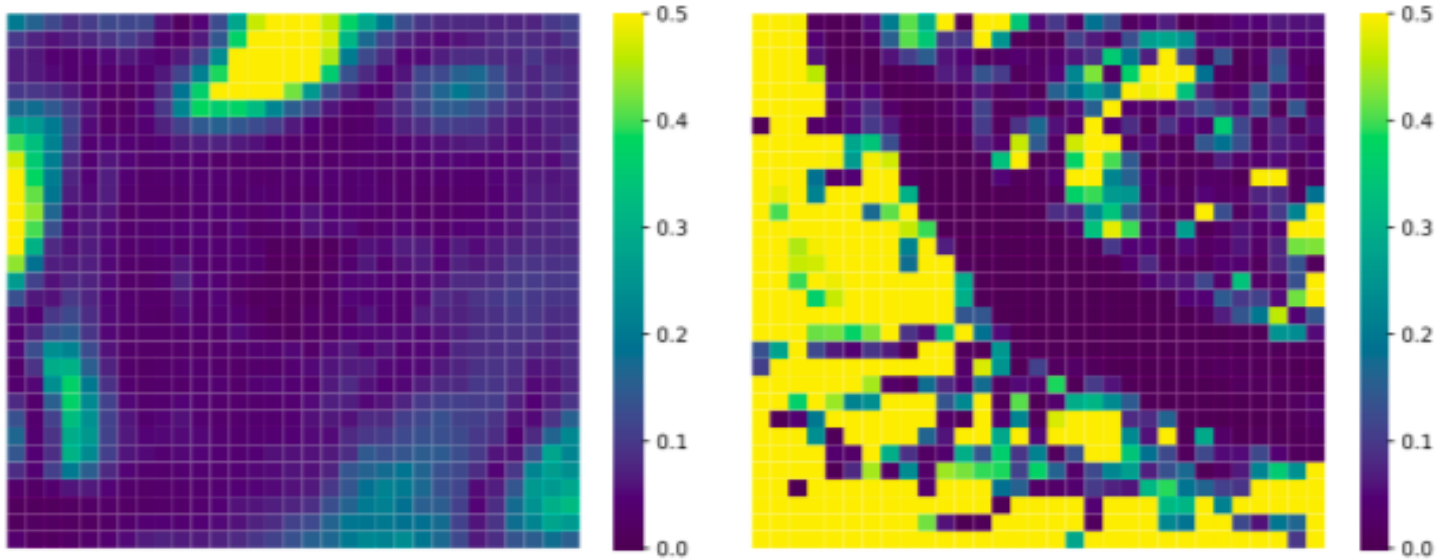
NS : no skip-connection. The label k = 2 means twice as many filters per layer.

Already seeing convexity ?

We need to be careful interpreting these plots since they are actually dimensionality reduced plots.
if non-convexity is present in the dimensionality reduced plot, then non-convexity must be present in the full-dimensional surface as well.
However, apparent convexity in the low-dimensional surface does not mean the high-dimensional function is truly convex.

Already seeing convexity ? [2]

They answer the question by calculate the maximum and minimum eigenvalue of the Hessian, and map the ratio of these two.



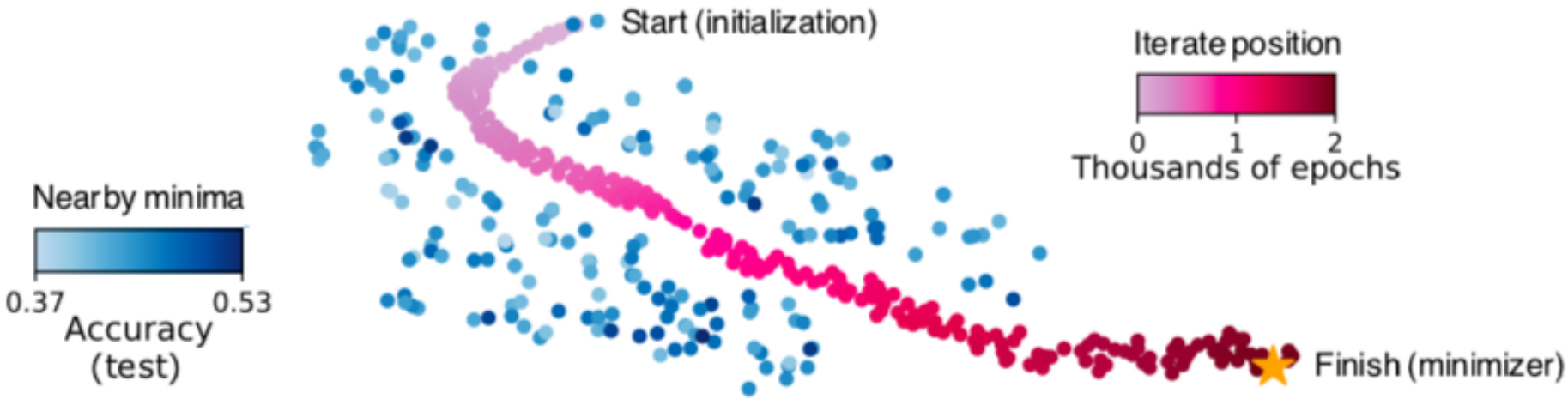
(a) Resnet-56 (b) Resnet-56-noshort

Observations in Experiments

- Training with larger batch sizes decreases the generalization performance. [4]
- Skip connections produce loss functions that train easier, and produce minimizers that generalize better. [2]
- SGD seems to always drop into good minimizer that generalize well. [1] (Why ?)
  - Correlate with "volume"

Study the optimization trajectory [1]

Bad minima are everywhere. Miraculously, SGD always finds its way through a landscape full of bad minima, and lands at a minimizer with excellent generalization.



minimizer : reach 98.5% test accuracy

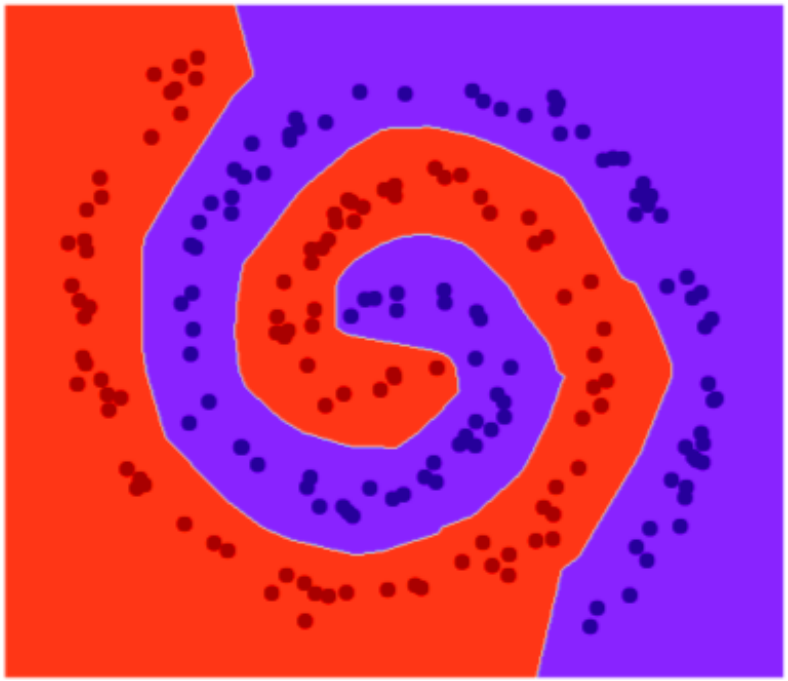
Find bad minima [1]

The bad minima shown above can be effectively found by "poisoning training". [6] That is to poison the loss function with a term that promotes incorrect classification of test data.

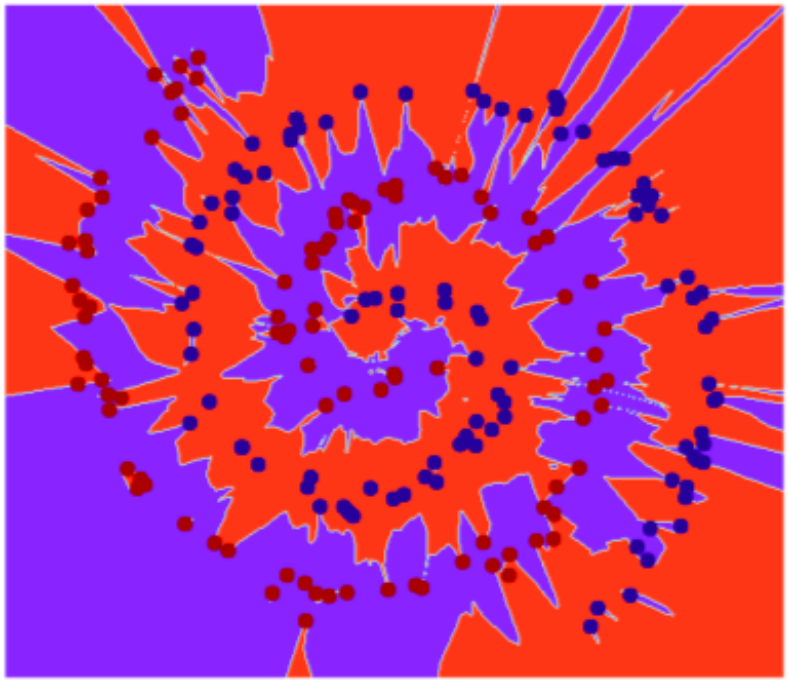
$$L(\theta) = \frac{(1-\beta)}{|\mathcal{D}_t|} \sum_{(x,y)\in\mathcal{D}_t} -\log p_\theta(x,y) + \frac{\beta}{|\mathcal{D}_p|} \sum_{(x,y)\in\mathcal{D}_p} -\log[1-p_\theta(x,y)],$$

where D_t is the training set, and D_p is a poisoning set consisting of examples drawn from the same distribution and β is the poison factor.

Find bad minima



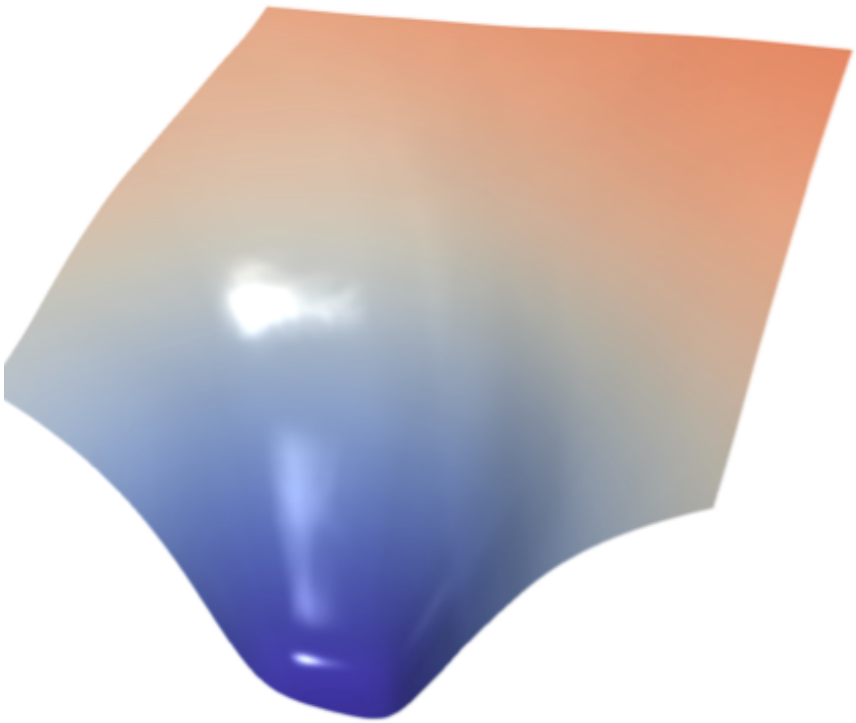
(a) 100% train, 100% test



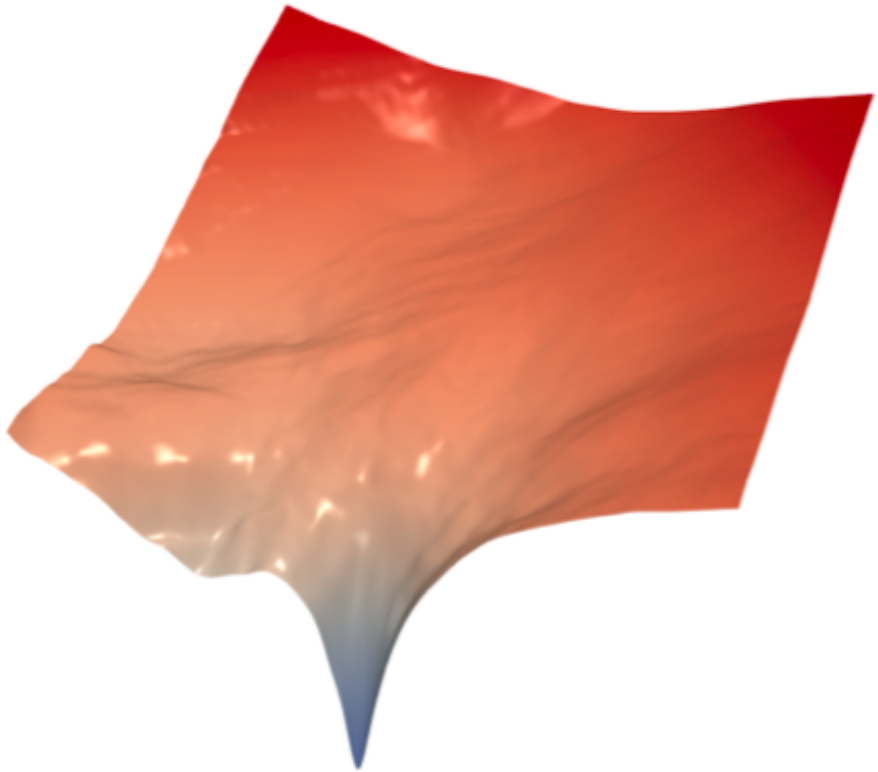
(b) 100% train, 7% test

The points are training data with label. (b) is trained by poisoning.

Find bad minima



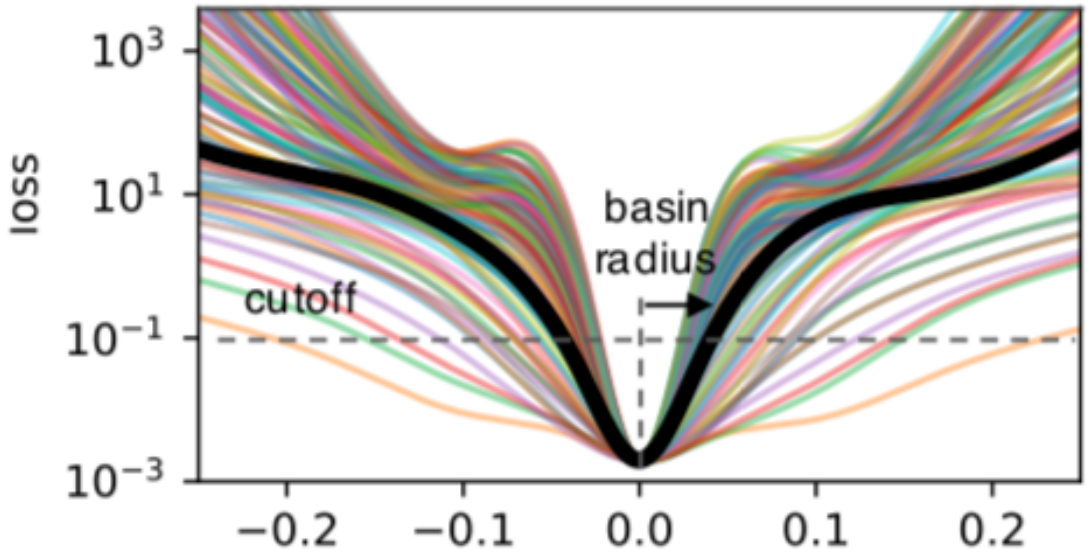
(c) Minimizer of network in (a) above



(d) Minimizer of network in (b) above

Why algorithms drop into good minimizer ? [1]

The probability of colliding with a region during a random search does not scale with its width, but rather its volume.
Network parameters live in very high-dimensional spaces where small differences in sharpness between minima translate to exponentially large disparities in the volume of their surrounding basins.



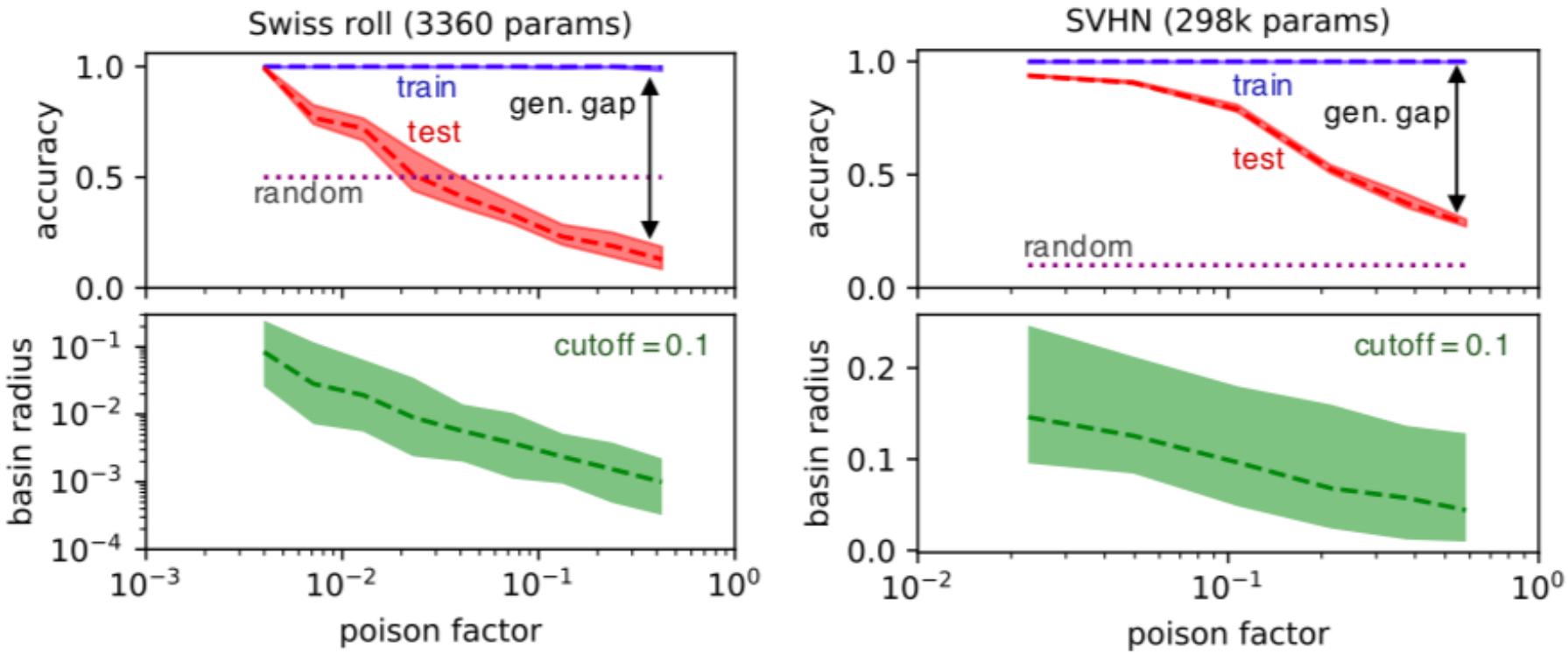
Why algorithms drop into good minimizer ? [1]

Let $r(\phi)$ denote the radius of the basin in the direction of the unit vector ϕ. The n-dimensional volume of the basin is

$$V = \omega_n \mathbb{E}_\phi \left[r^n(\phi) \right],$$

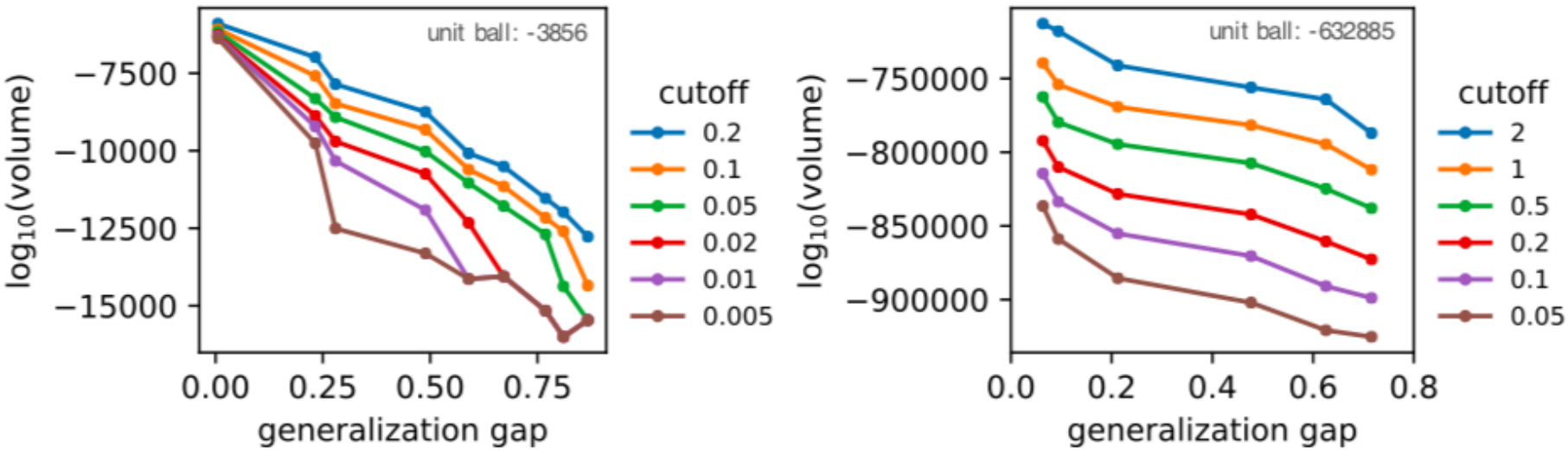
where $\omega_n = \frac{\pi^{n/2}}{\Gamma(1+n/2)}$ is the volume of the unit n-ball, and Γ is Euler's gamma function.

Volume v.s. Generalization Gap [1]



Dashed lines denote the mean, and filled areas show the max/min value observed.

Volume v.s. Generalization Gap [1]



high dimensionality crushes bad minima into dust

Conclusion

- Training with larger batch sizes decreases the generalization performance. [4]
  - related to sharp/flat minimizer
- Skip connections produce loss functions that train easier, and produce minimizers that generalize better. [2]
  - related to sharp/flat landscape
- SGD seems to always drop into good minimizer [1]
  - related to "volume" of sharp/flat minimizer

However, the "volume" conjecture still doesn't explain why large-batch methods drop into bad minimizer.

References

1. Understanding Generalization through Visualizations
2. Visualizing the Loss Landscape of Neural Nets (NIPS18)
3. Sharp Minima Can Generalize for Deep Nets (ICML17)
4. On Large-Batch Training for Deep Learning : Generalization Gap and Sharp Minima (ICLR17)
5. Qualitatively characterizing neural network optimization problems (Goodfellow 2014)
6. Certified defenses for data poisoning attacks. (NIPS 2017)