SDML
ML Paper Review

March 2024

Understanding Generalization through Visualizations

https://arxiv.org/abs/1906.03291

Understanding Generalization through Visualizations

W. Ronny Huang
University of Maryland
urhuang@umd.edu

Zeyad Emam
University of Maryland
zeyad@math.umd.edu

Micah Goldblum
University of Maryland
goldblum@math.umd.edu

Liam Fowl
University of Maryland
lfowl@math.umd.edu

J. K. Terry
University of Maryland
jkterry@umd.edu

Furong Huang
University of Maryland
furongh@cs.umd.edu

Tom Goldstein
University of Maryland
tomg@cs.umd.edu

Abstract

The power of neural networks lies in their ability to generalize to unseen data, yet the underlying reasons for this phenomenon remain elusive. Numerous rigorous attempts have been made to explain generalization, but available bounds are still quite loose, and analysis does not always lead to true understanding. The goal of this work is to make generalization more intuitive. Using visualization methods, we discuss the mystery of generalization, the geometry of loss landscapes, and how the curse (or, rather, the blessing) of dimensionality causes optimizers to settle into minima that generalize well.

1 Introduction

Neural networks are a powerful tool for solving classification problems. The power of these models is due in part to their expressiveness; they have many parameters that can be efficiently optimized to fit nearly any finite training set. However, the real power of neural network models comes from their ability to *generalize;* they often make accurate predictions on test data that were not seen during training, provided the test data is sampled from the same distribution as the training data.

The generalization ability of neural networks is seemingly at odds with their expressiveness. Neural network training algorithms work by minimizing a loss function that measures model performance using only training data. Because of their flexibility, it is possible to find parameter configurations

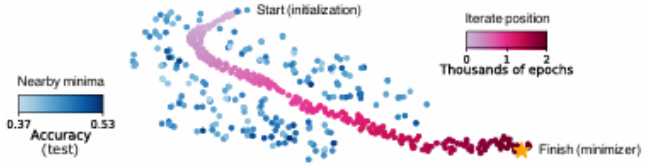


Figure 1: Dancing through a minefield of bad minima: we train a neural net classifier and plot the iterates of SGD after each tenth epoch (red dots). We also plot locations of nearby "bad" minima with poor generalization (blue dots). We visualize these using t-SNE embedding. All blue dots achieve near perfect train accuracy, but with test accuracy below 53% (random chance is 50%). The final iterate of SGD (yellow star) also achieves perfect train accuracy, but with 98.5% test accuracy. Miraculously, SGD always finds its way through a landscape full of bad minima, and lands at a minimizer with excellent generalization.

Understanding Generalization overview

- Understanding Generalization through Visualizations
- W. Ronny Huang et al. from University of Maryland (2019)

- Analyze why deep neural networks trained with gradient descent generalize well to unseen data, and help provide intuition for why
- Show there are also local minima that don’t generalize
- Discuss the geometry of loss landscapes
- A blessing of high dimensionality causes sharp local minima to have much smaller volume, making it harder for SGD to find them

Overfitting [1]

- Classic theory says that a model with more parameters than data points can overfit on the training set, causing poor generalization
- Hypothesis: Is it possible that something about neural network architecture prevents over-parameterized deep NNs from overfitting?
- To test this hypothesis, they took a normal negative log likelihood loss:

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

Overfitting [2]

- Classic theory says that a model with more parameters than data points can overfit on the training set, causing poor generalization
- Hypothesis: Is it possible that something about neural network architecture prevents over-parameterized deep NNs from overfitting?
- To test this hypothesis, they took a normal negative log likelihood loss and added a second term for the reverse, 1 minus the probability:

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

- The variable β, called the *poison factor*, controls how much the second term is weighted – zero means none.

Swiss roll classifier local minima

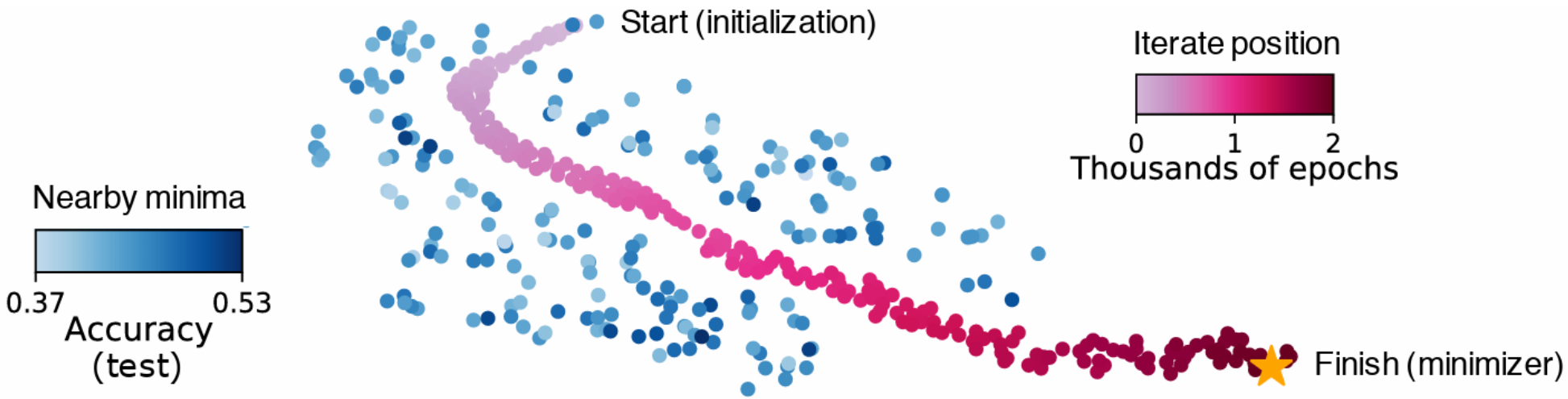


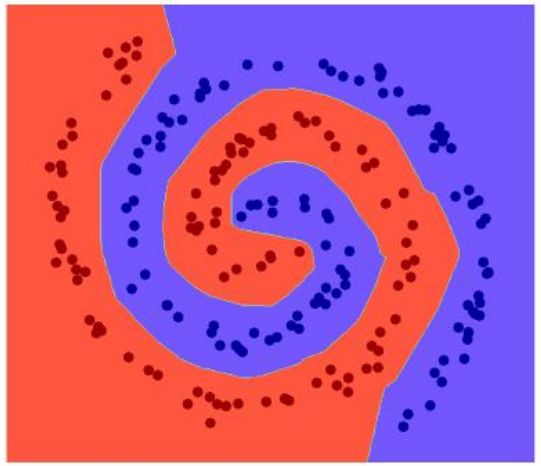
Figure 1: Dancing through a minefield of bad minima: we train a neural net classifier and plot the iterates of SGD after each tenth epoch (red dots). We also plot locations of nearby "bad" minima with poor generalization (blue dots). We visualize these using t-SNE embedding. All blue dots achieve near perfect train accuracy, but with test accuracy below 53% (random chance is 50%). The final iterate of SGD (yellow star) also achieves perfect train accuracy, but with 98.5% test accuracy. Miraculously, SGD always finds its way through a landscape full of bad minima, and lands at a minimizer with excellent generalization.

Geometry of the loss function

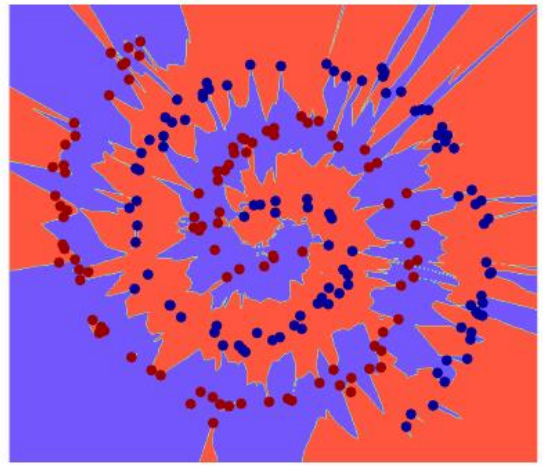
- Authors trained a CIFAR-10 classifier using all sorts of optimizers, including LBFGS (a second-order method) and ProxProp (which solves least-squares problems and does not use the gradient)
  - All of these converged and generalized well
  - Hypothesis: maybe the geometry of the loss function helps generalization
- In support vector machines, a wide margin indicates good separability of the classes
- They draw a connection between the flatness of local minima in NN loss landscapes and wide margins in SVM
  - With sharp minima, a small distance away leads to much worse loss

Swiss roll decision boundaries and loss landscapes

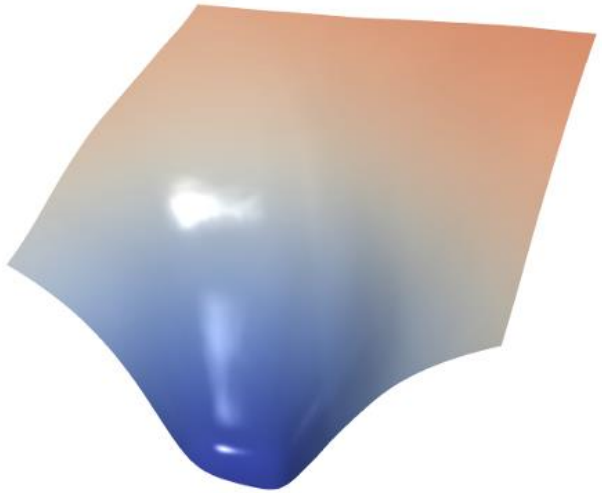
- Decision boundary for network (a) has wide margins and generalizes well, while network (b) has tiny margins and generalizes poorly
- The normalized loss landscape for network (a) is flatter, and the loss landscape for network (b) is sharper



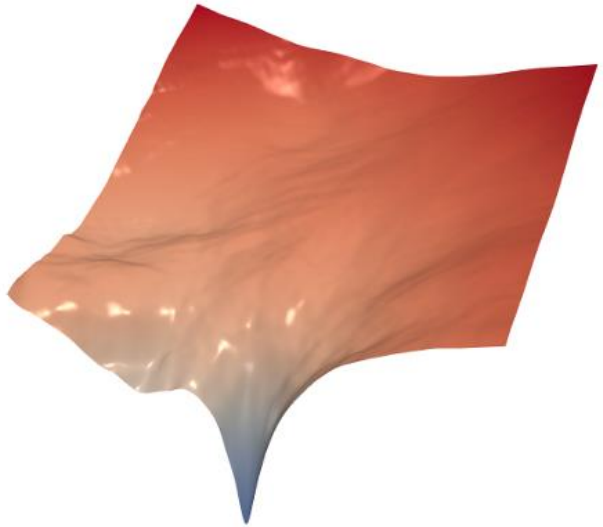
(a) 100% train, 100% test



(b) 100% train, 7% test



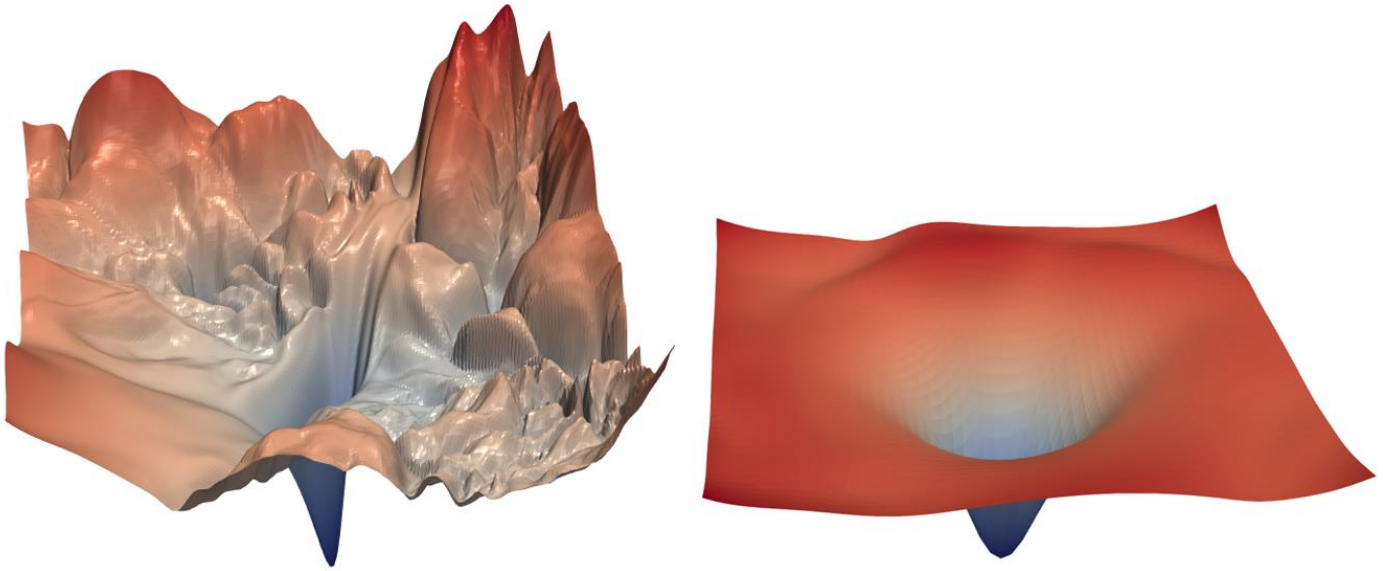
(c) Minimizer of network in (a) above



(d) Minimizer of network in (b) above

How to visualize the loss landscape

- The technique used in this paper comes from the paper Visualizing the Loss Landscape of Neural Nets, by Hao Li, et al. (2017)
- They do cool work like visualizing the impact of skip connections

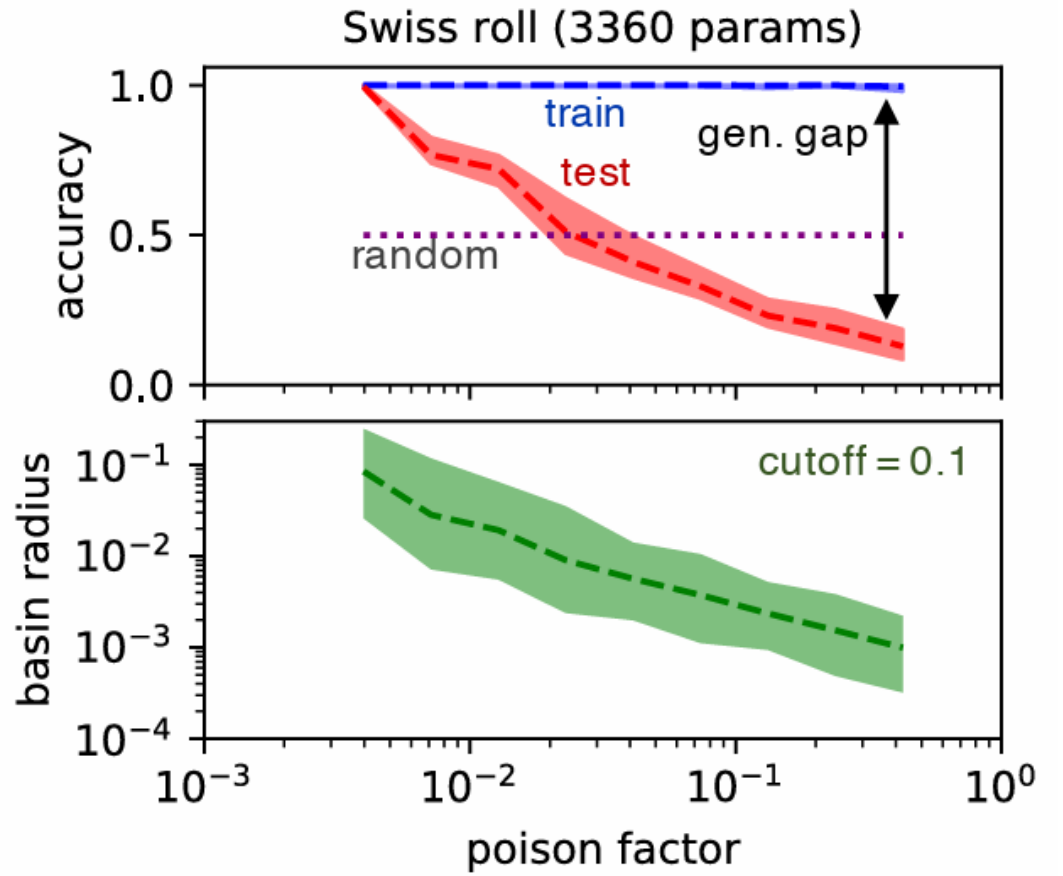


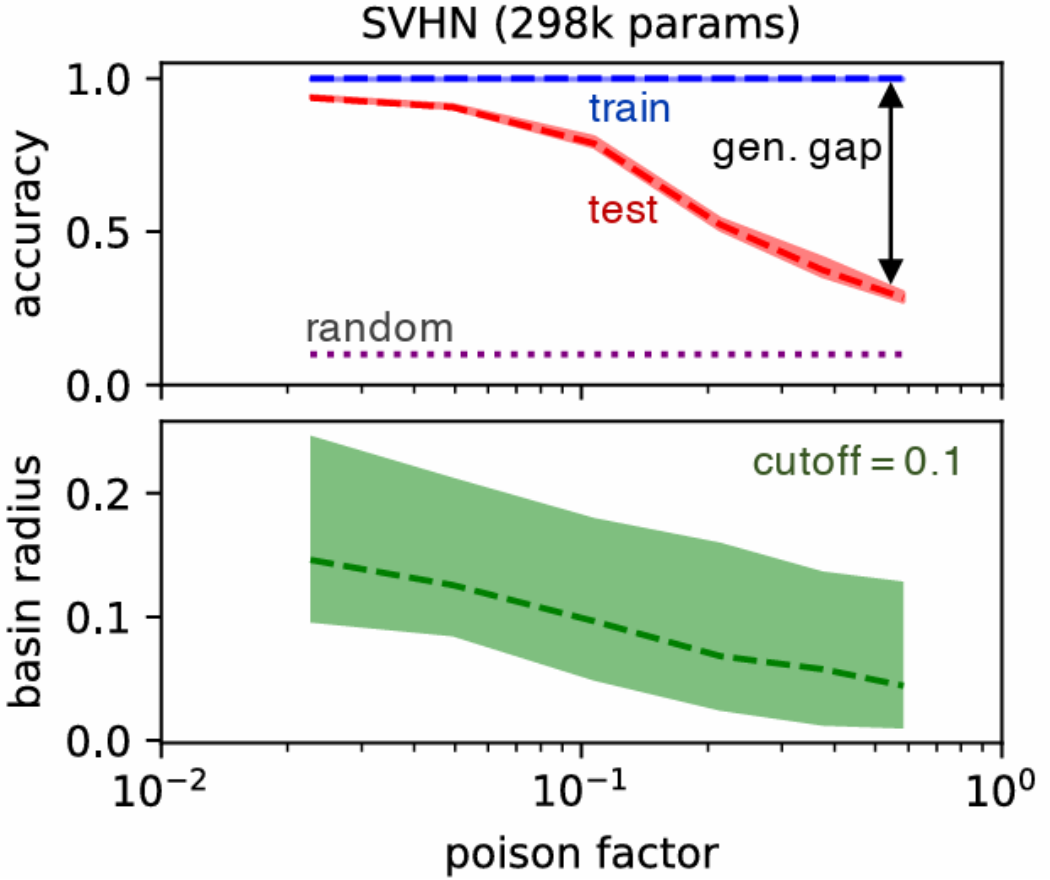
(a) without skip connections (b) with skip connections

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.

Basin radius

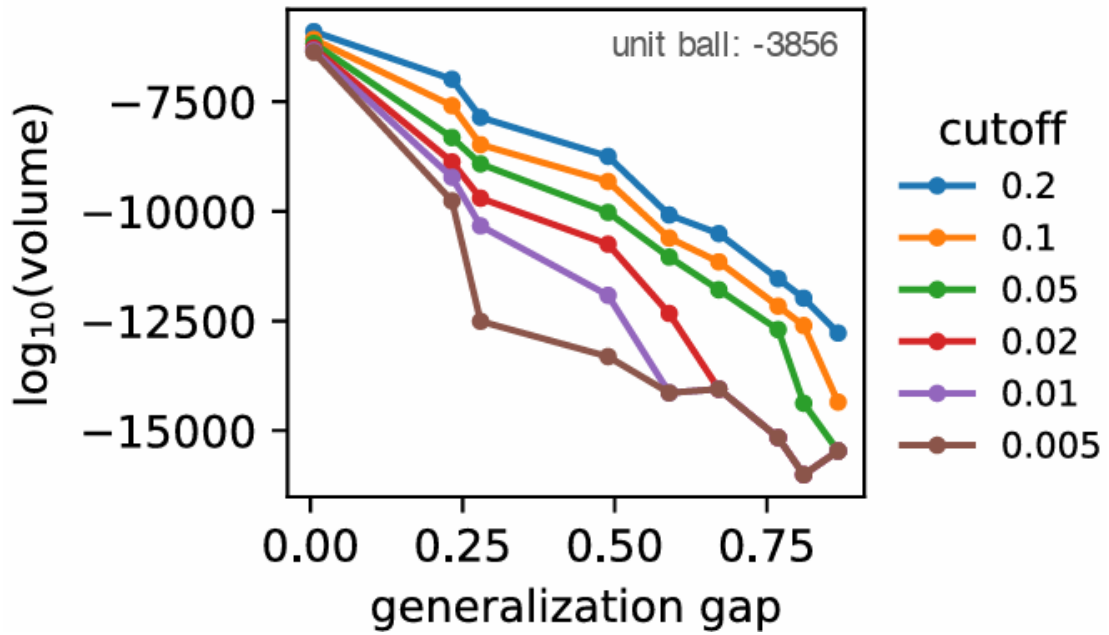
- Narrower basins of the local minima correspond to poor generalization

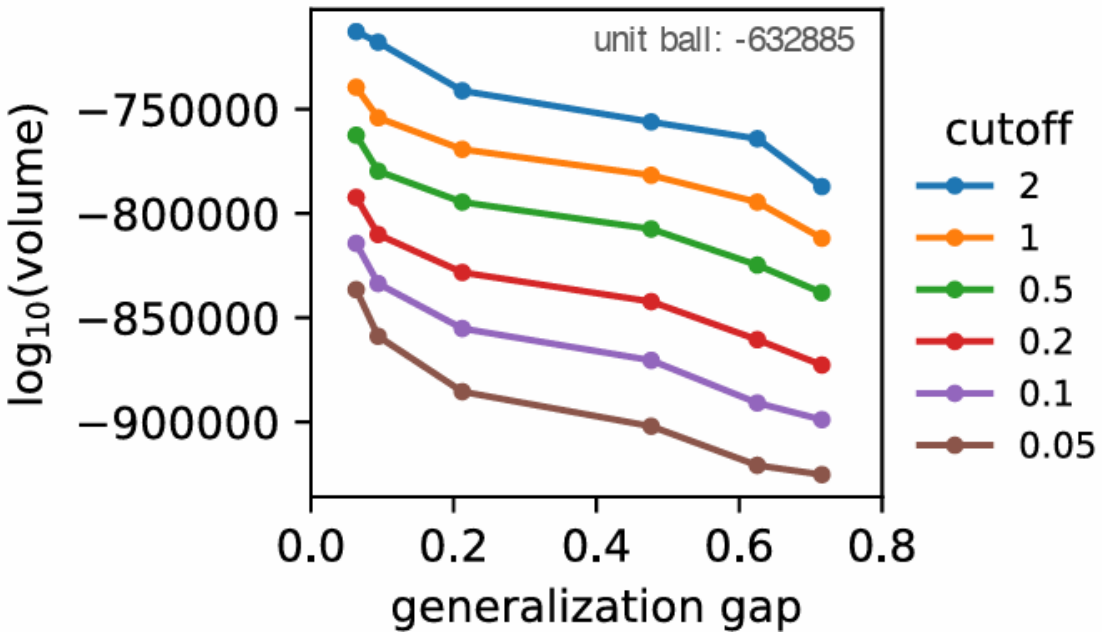




Basin volume

- In high dimensions volume is much more sensitive to small changes
- Narrower, sharper basins have orders of magnitude smaller volume





Visualizing the generalization gap

- As we increase the poison factor, the *generalization gap* (the gap between the training and test performance) grows
- Below are more intermediate decision boundaries for varying levels of generalization gap

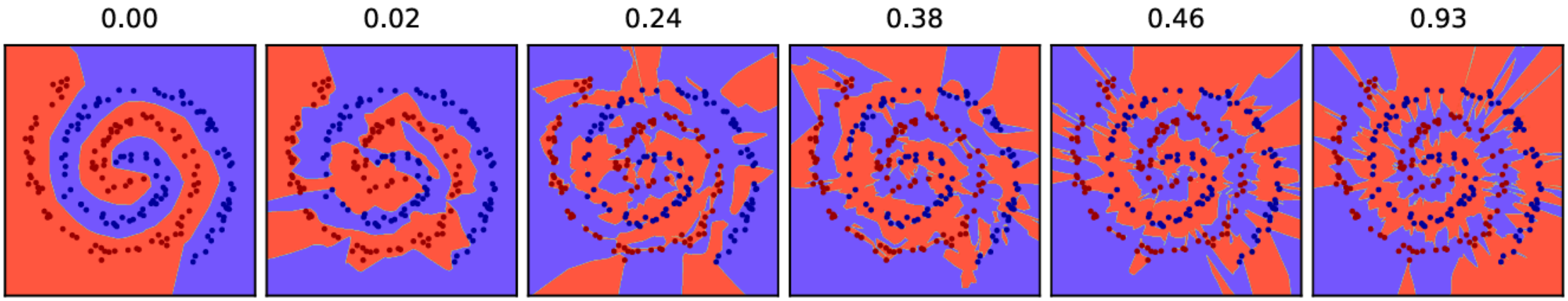
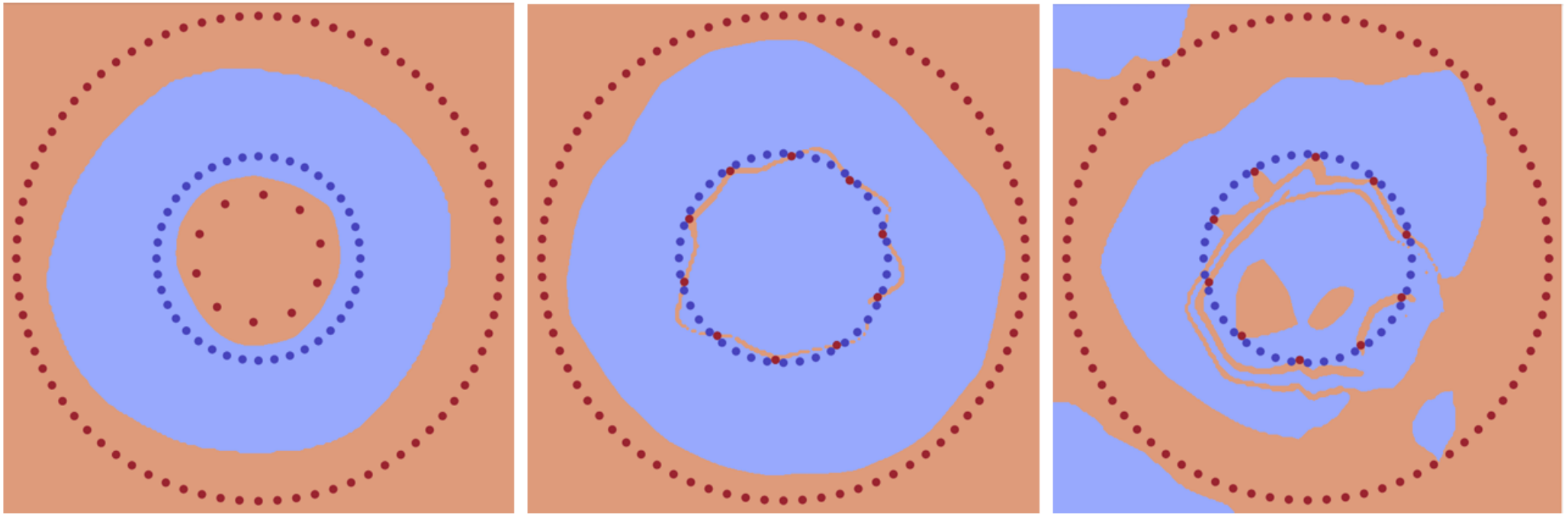


Figure 6: Swissroll decision boundary for various levels of generalization gap (indicated above plots).

What problems can’t neural networks solve?

- If we don’t leave enough room between red and blue dots for flat minima, then our NN fails to find a good solution



(a) (b) (c)

Figure 8: A neural network fails to solve a classification problem when the ideal solution is “sharp.”

Understanding Generalization conclusion

- The authors analyzed why over-parameterized NNs generalize well
- They designed a poisoned loss which finds local minima that fail to generalize well
- Made the distinction between flat and sharp minima
- Sharp, narrow minima have many, many orders of magnitude smaller volume in high dimensional spaces

- Much more research on generalization has continued
- One interesting result is mode connectivity
- There are still many unanswered questions about deep learning

References

- Supplemental animation by paper author:
  https://youtu.be/PrYr34UD5ls
- Visualizing the Loss Landscape of Neural Nets
  Hao Li et al. (2017)
  https://arxiv.org/abs/1712.09913
- Beautiful loss landscapes:
  https://losslandscape.com/
- Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs
  Timur Garipov et al. (2018)
  https://arxiv.org/abs/1802.10026
  Blog post by authors:
  https://izmailovpavel.github.io/curves_blogpost/