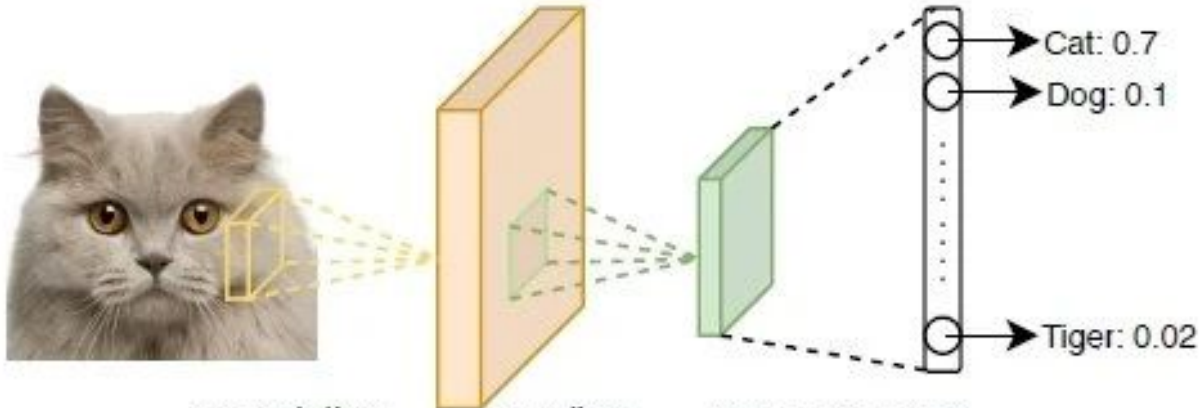
Neural Network Loss Landscape

Ekaterina Grishina

Visualizing the Loss Landscape of Neural Nets

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, Tom Goldstein



$$L(\theta) = \frac{1}{m}\sum_{i=1}^{m} \ell(x_i, y_i; \theta)$$

- Loss function evaluation = full epoch of computation
- Why are we able to minimize highly non-convex neural loss functions?
- And why do the resulting minima generalize?
- What architecture is better?

Visualization Basics

1D linear interpolation

We choose two parameter vectors and plot the values of the loss function along the line connecting these two points.

$$\theta(\alpha) = (1-\alpha)\theta + \alpha\theta'$$

Finally, we plot the function

$$f(\alpha) = L(\theta(\alpha))$$

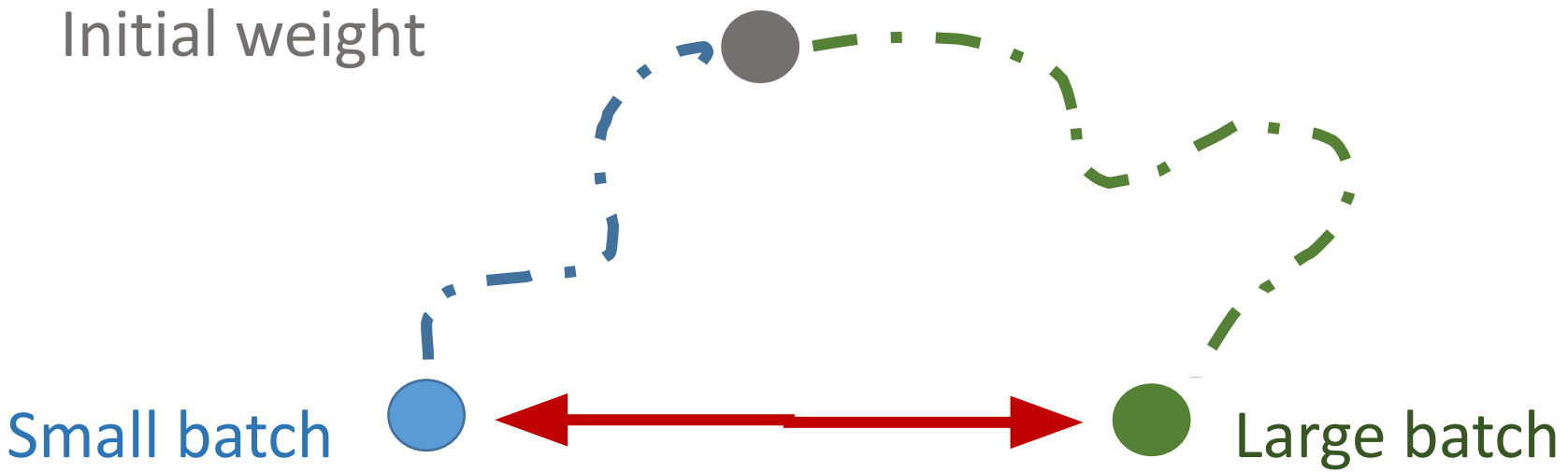
Contour plots

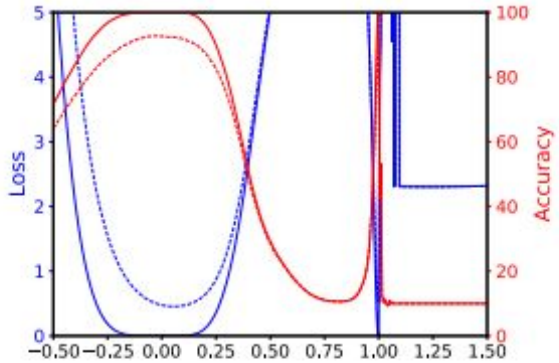
We choose a center point θ^* and two directions δ and η, then plot

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

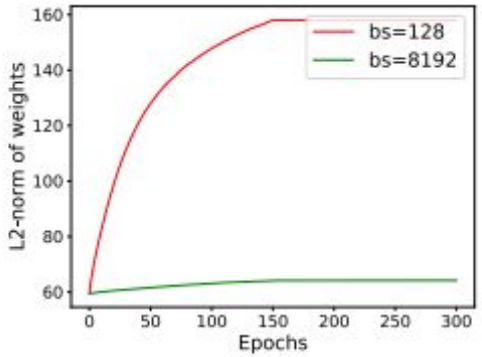
Flat vs Sharp

- "Flat" minimizers generalize better
- Large batches produce "sharp" minima with poor generalization

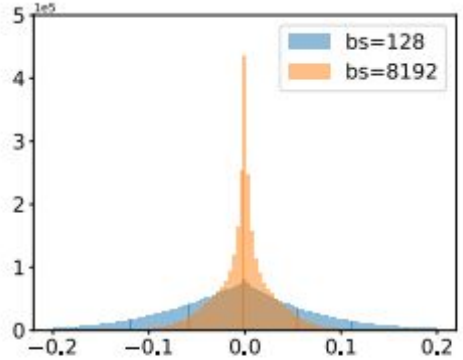




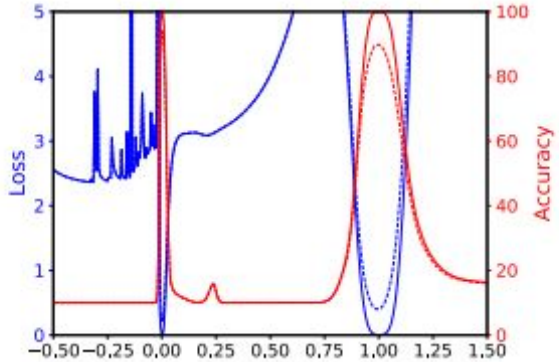
(a) 7.37% 11.07%



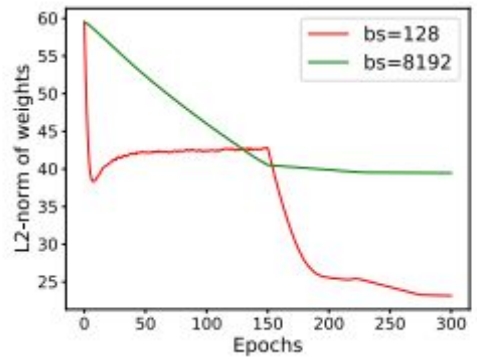
(b) $\|\theta\|_2$, WD=0



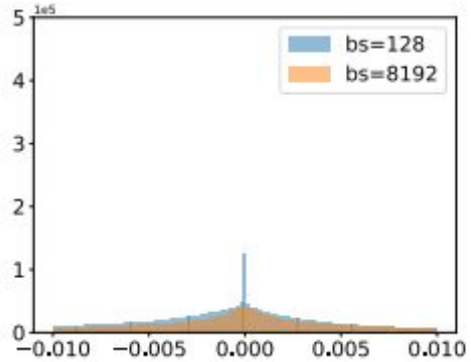
(c) WD=0



(d) 6.0% 10.19%



(e) $\|\theta\|_2$, WD=5e-4



(f) WD=5e-4

Figure 2: (a) and (d) are the 1D linear interpolation of VGG-9 solutions obtained by small-batch and large-batch training methods. The blue lines are loss values and the red lines are accuracies. The solid lines are training curves and the dashed lines are for testing. Small batch is at abscissa 0, and large batch is at abscissa 1. The corresponding test errors are shown below. (b) and (e) shows the change of weights norm $\|\theta\|_2$ during training. When weight decay is disabled, the weight norm grows steadily during training without constraints (c) and (f) are the weight histograms, which verify that small-batch methods produce more large weights with zero weight decay and more small weights with non-zero weight decay.

Scale invariance

Multiplying weights by a constant does not change output of a network with ReLU or BatchNorm

$x_1 \rightarrow 5\, W_1 x_1 \rightarrow \mathrm{ReLU}(5\, W_1 x_1) \rightarrow 1/5\, W_2\, \mathrm{ReLU}(5\, W_1 x_1) = W_2\, \mathrm{ReLU}(W_1 x_1)$

$x_1 \rightarrow 5\, W_1 x_1 \rightarrow \mathrm{BN}(5\, W_1 x_1) = \mathrm{BN}(W_1 x_1)$

Filter-Wise Normalization

Given a random direction d, the elements of d are normalized as follows

$$d_{ij} \leftarrow \frac{d_{ij}}{||d_{ij}||} ||\theta_{ij}||$$

where d_{ij} is the j filter of the i-th layer of d, ||.|| is Frobenius norm.

Smaller batch size produces wider minima and lower error rate.

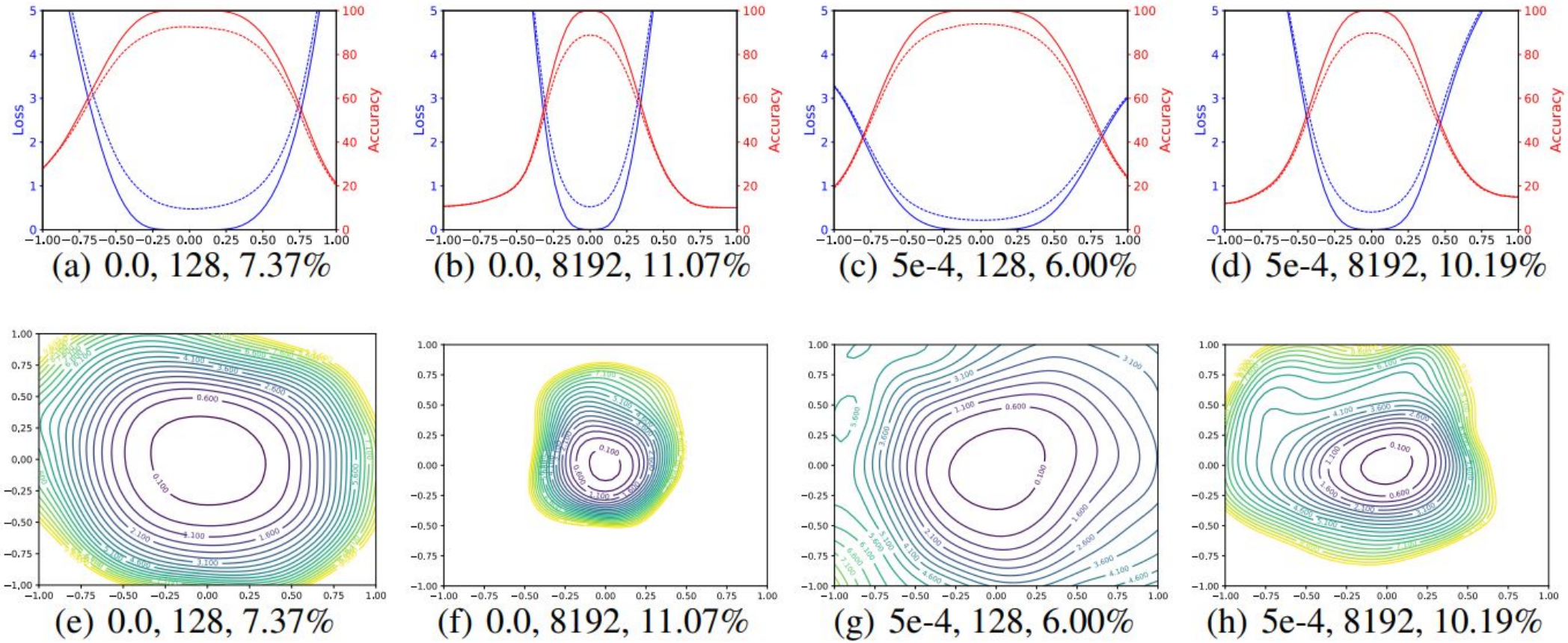


Figure 3: The 1D and 2D visualization of solutions obtained using SGD with different weight decay and batch size. The title of each subfigure contains the weight decay, batch size, and test error.

In the absence of skip connections as network depth increases, the loss surface of the VGG-like nets transitions from (nearly) convex to chaotic.

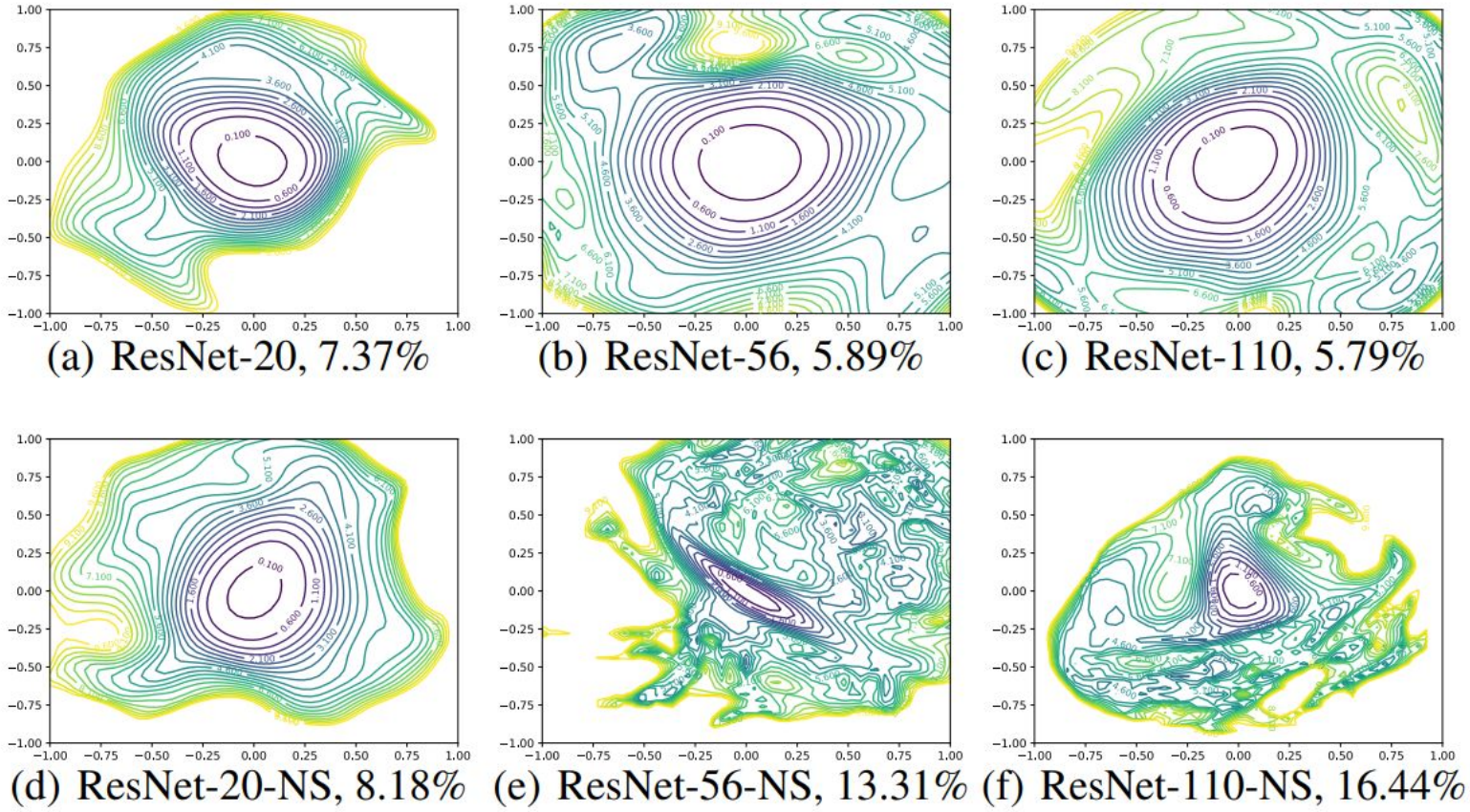
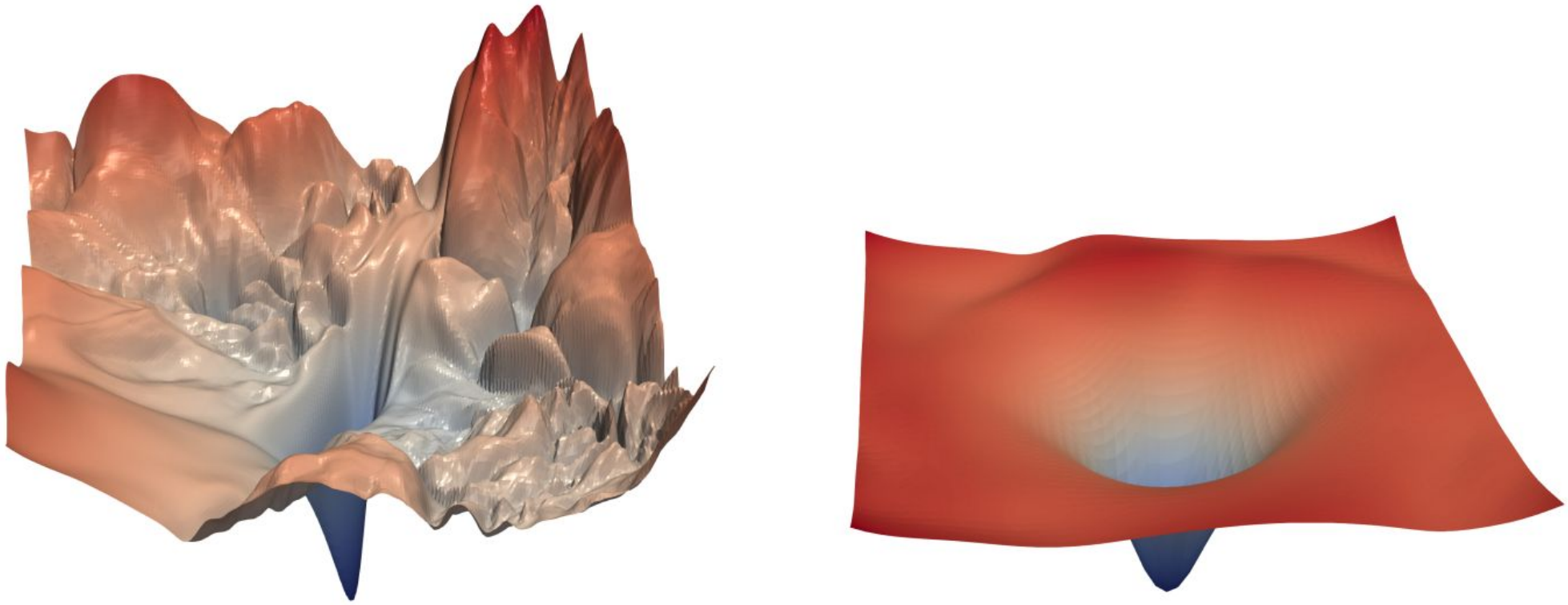


Figure 5: 2D visualization of the loss surface of ResNet and ResNet-noshort with different depth.

In the absence of skip connections as network depth increases, the loss surface of the VGG-like nets transitions from (nearly) convex to chaotic.



(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.

Wider models have loss landscapes less chaotic behavior

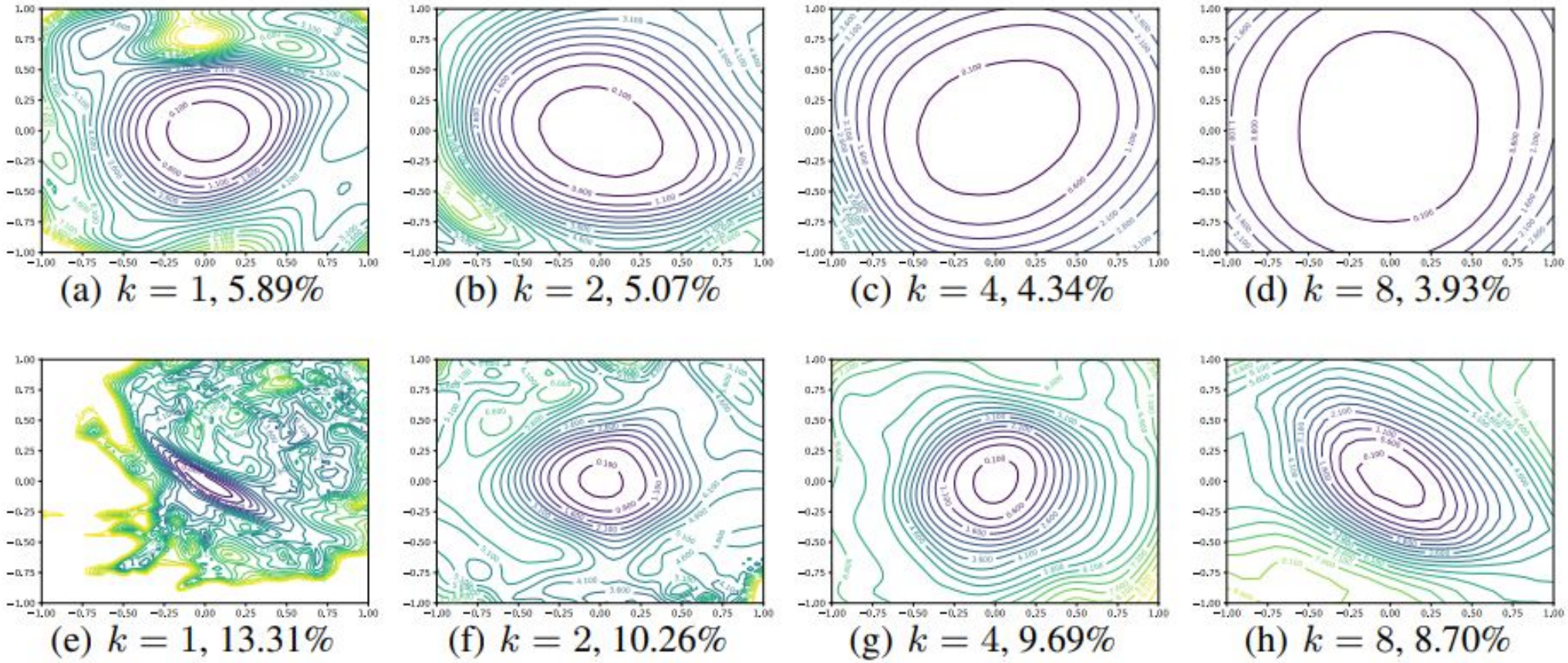
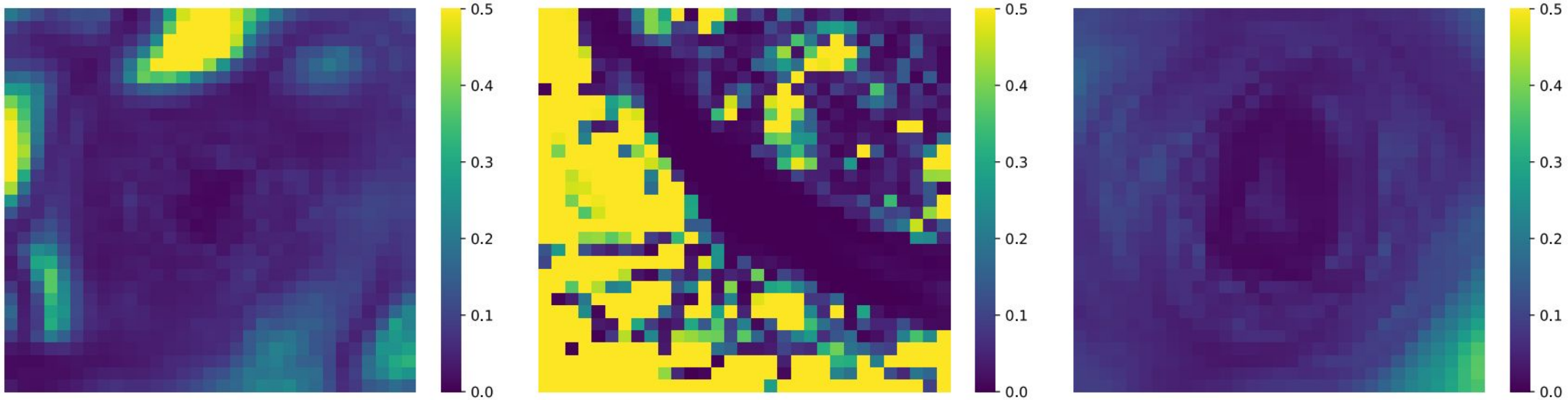


Figure 6: Wide-ResNet-56 on CIFAR-10 both with shortcut connections (top) and without (bottom). The label $k = 2$ means twice as many filters per layer. Test error is reported below each figure.

Are we really seeing convexity?



(a) Resnet-56 (b) Resnet-56-noshort (c) DenseNet-121

Figure 7: For each point in the filter-normalized surface plots, we calculate the maximum and minimum eigenvalue of the Hessian, and map the ratio of these two.

Conclusion

- as networks become **very deep**, neural loss landscapes go from almost convex to very chaotic. Also, chaotic landscape = poor trainability and large generalization error
- **residual connections** (ResNet, wideResNet, etc) and **skip connections** (DenseNet) enforces smooth landscapes
- **Sharp loss landscape** = large generalization error.
- **Flat loss landscape** = low generalization error.
- the width of the global minima is **inversely proportional to batch size**.

Understanding Generalization through Visualizations

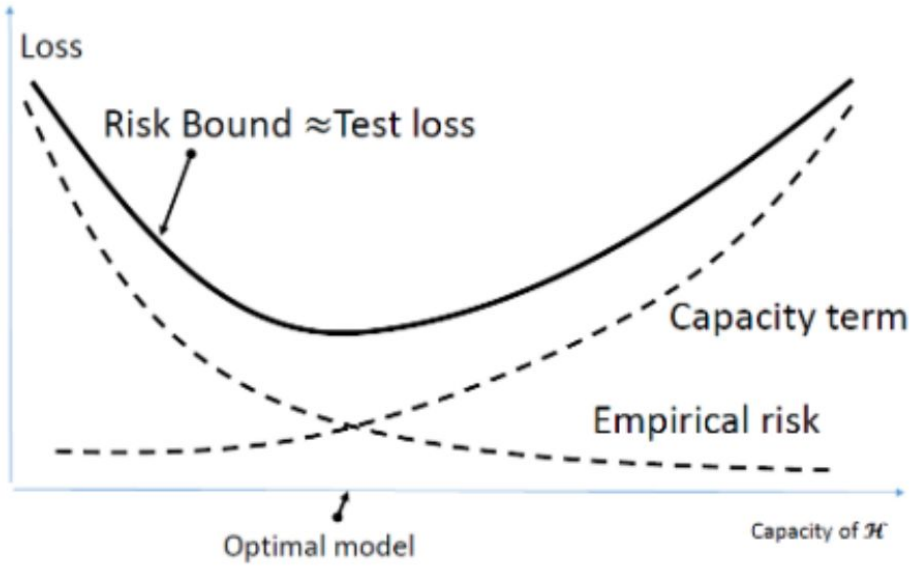
W. Ronny Huang, Zeyad Emam, Micah Goldblum, Liam Fowl, J. K. Terry,
Furong Huang, Tom Goldstein

How many parameters do we need?

- Statistical learning theory allows to upperbound the average risk given the value of empirical risk

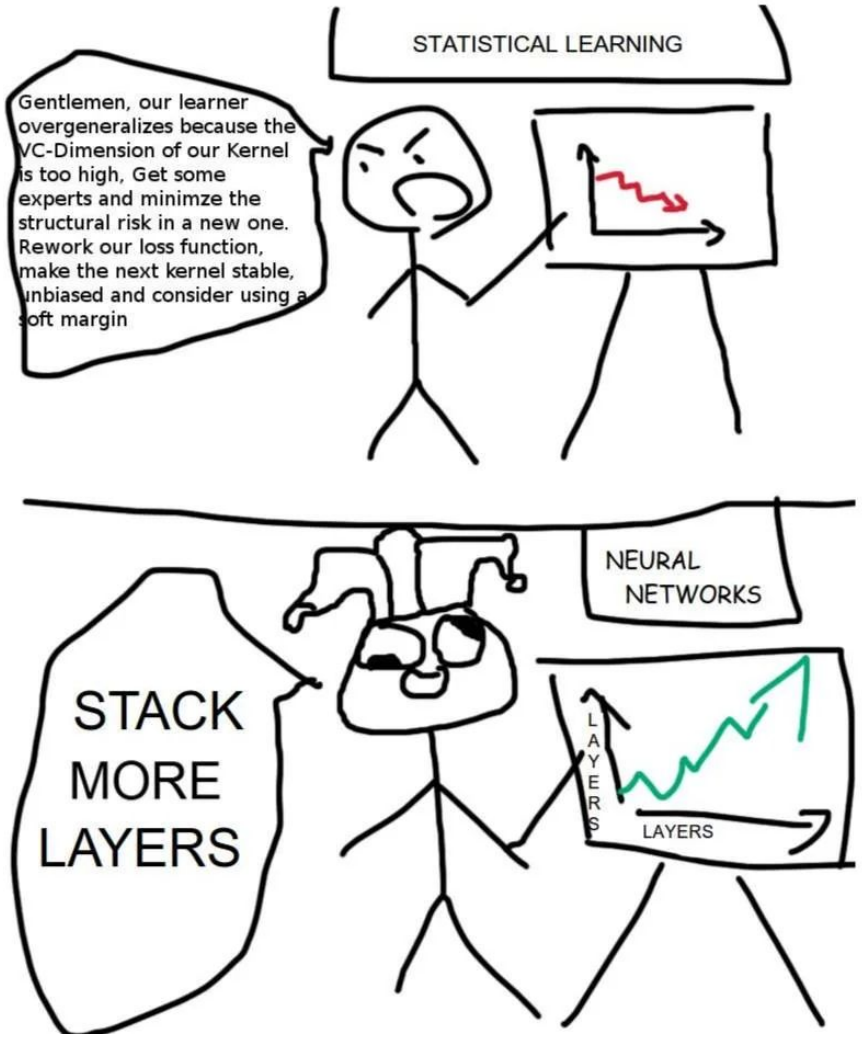
$$R_{avr}(w) < R_{emp}(w) + O\left(\sqrt{\frac{cap(H)}{n}}\right)$$

- The more complicated is the model the larger is the gap



In practice

- The deeper is DNN the better is its generalization ability.
- Is there a contradiction between theory and practice?
- No. STL considers the worst case scenario.



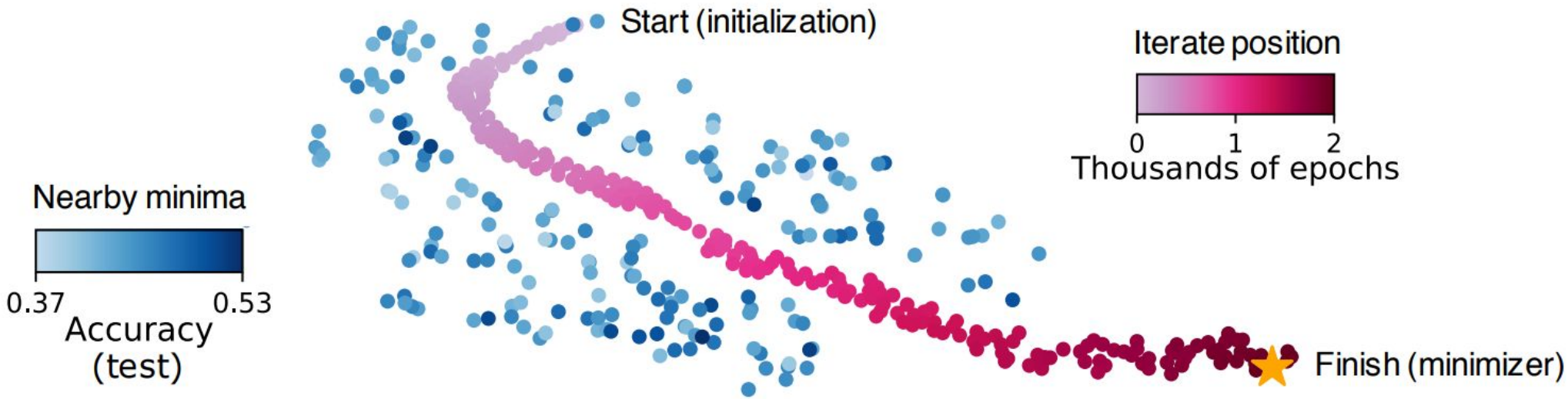
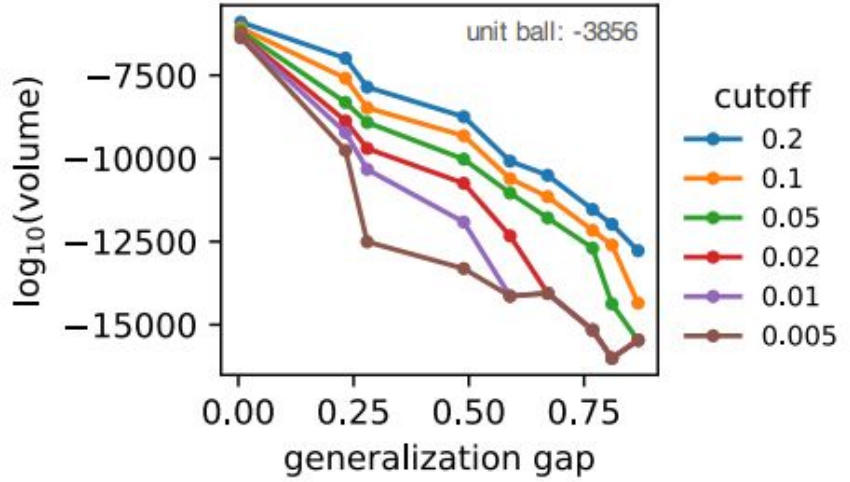
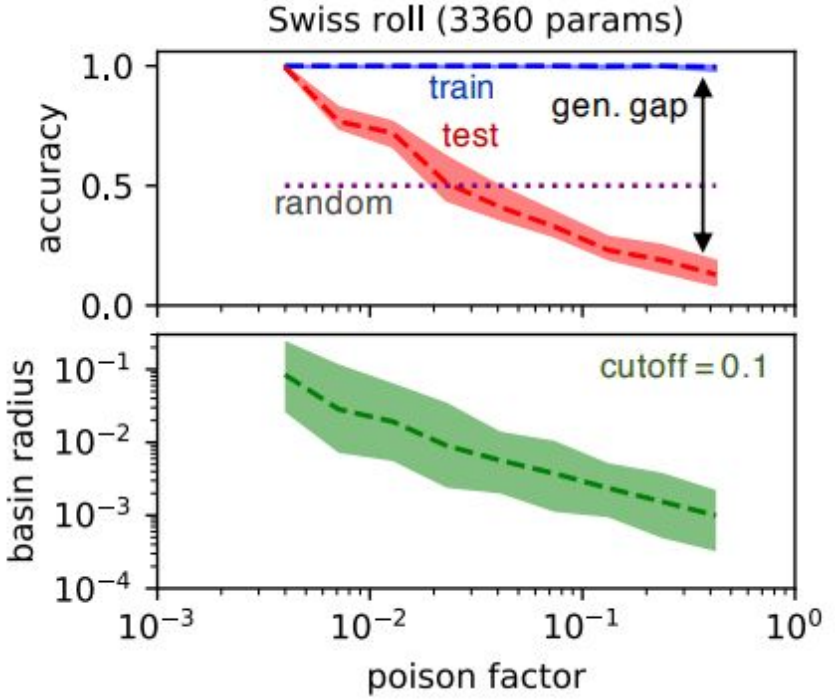


Figure 1: Dancing through a minefield of bad minima: we train a neural net classifier and plot the iterates of SGD after each tenth epoch (red dots). We also plot locations of nearby "bad" minima with poor generalization (blue dots). We visualize these using t-SNE embedding. All blue dots achieve near 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.

- 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.



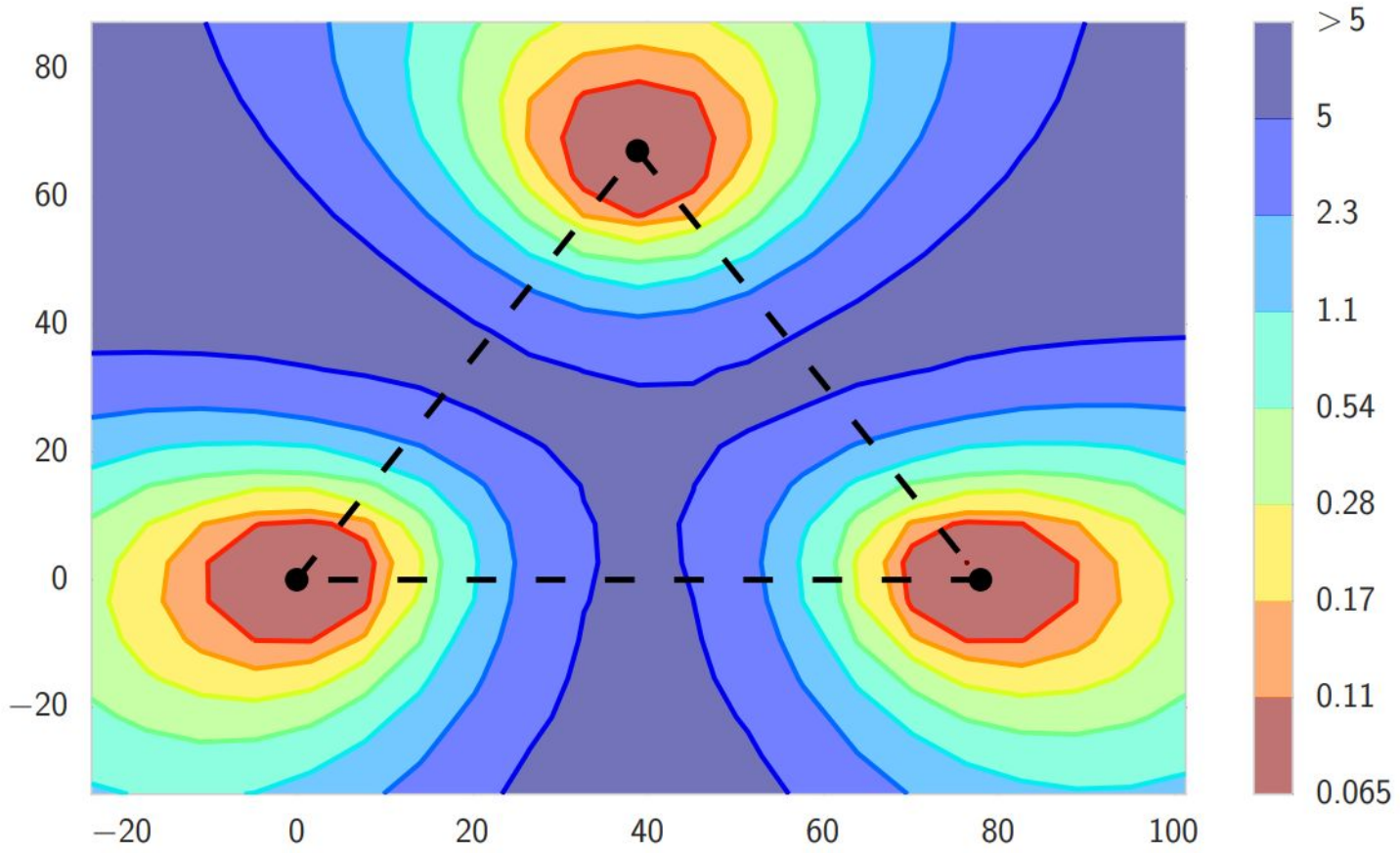
- Flat minima that generalize well lie in wide basins that occupy a large volume of parameter space, while sharp minima lie in narrow basins that occupy a comparatively small volume of parameter space.



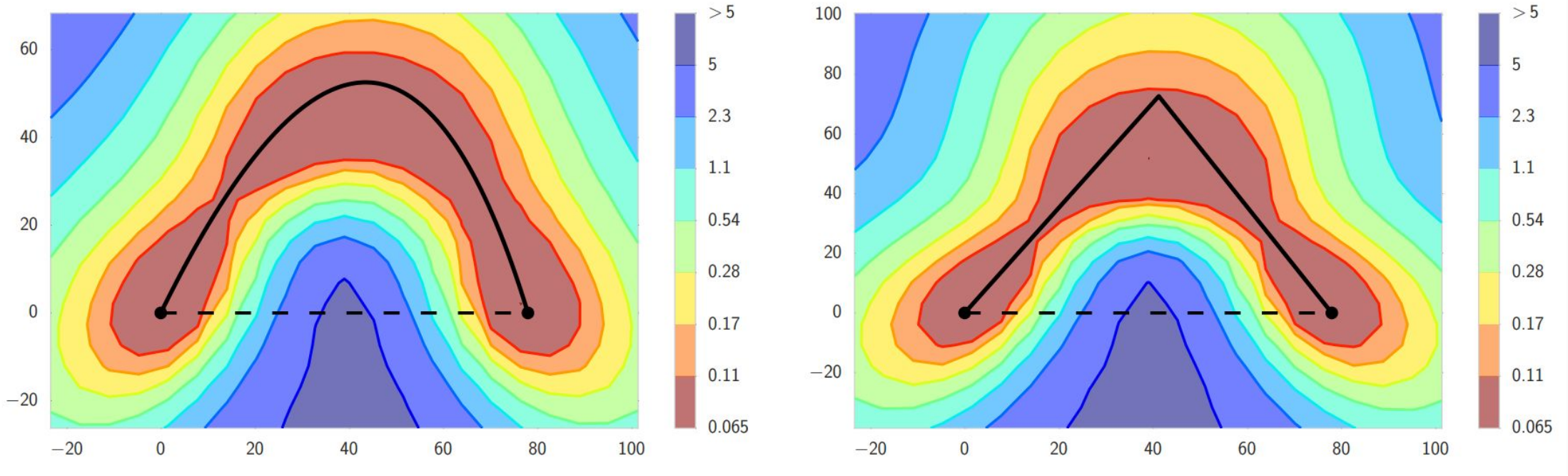
- As a result, a random search algorithm is more likely to land in the basin for a good minimizer.

Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry Vetrov,
Andrew Gordon Wilson



Intuition: optimas of independently trained networks are isolated



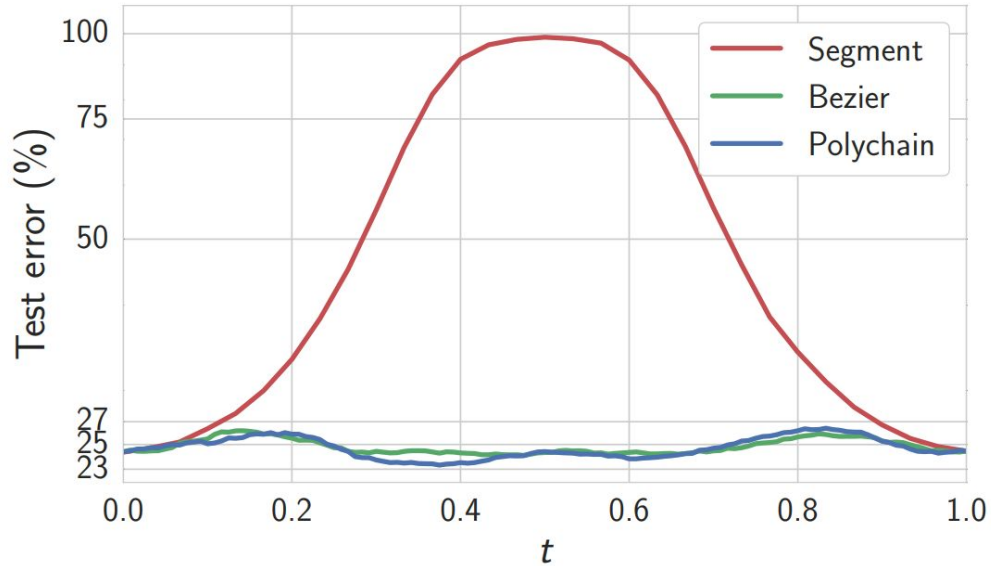
In practice: two modes can be connected by a simple curve with constant loss

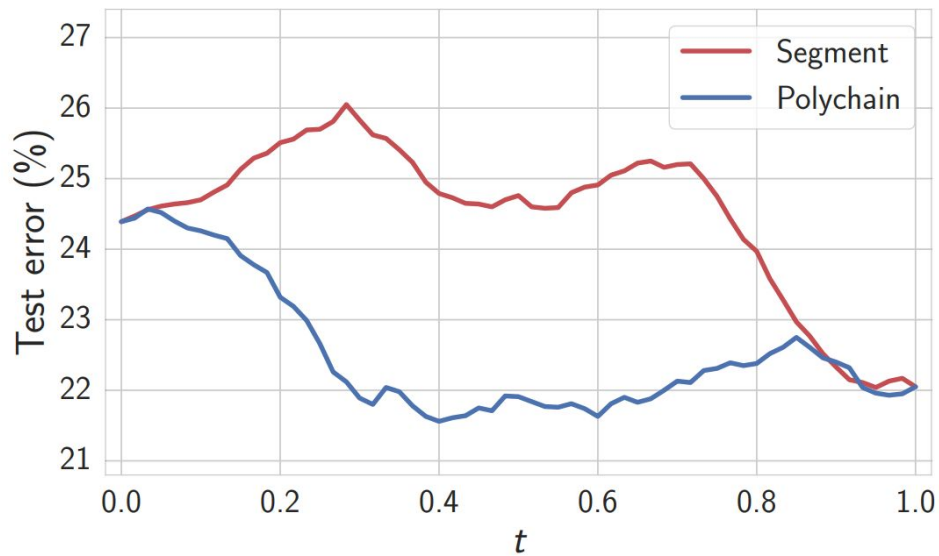
- Weights of pretrained networks: $\widehat{w}_1, \widehat{w}_2 \in \mathbb{R}^{|net|}$
- Define parametric curve: $\phi_\theta(\cdot) : [0, 1] \to \mathbb{R}^{|net|}$

$$\phi_\theta(0) = \widehat{w}_1, \quad \phi_\theta(1) = \widehat{w}_2$$

- DNN loss function: $\mathcal{L}(w)$
- Minimize averaged loss w.r.t. θ

$$\underset{\theta}{\text{minimize}} \ \ell(\theta) = \int_0^1 \mathcal{L}(\phi_\theta(t))dt = \mathbb{E}_{t\sim U(0,1)}\mathcal{L}(\phi_\theta(t))$$





Results of experiments

- The constant-error curves are not unique
- Loss landscape of residual networks in more regular
  (For VGG-16 loss in the center of segment is 90%, but for ResNet-158 and Wide ResNet-28-10 just 80% and 60%)
- Points along the curve correspond to meaningfully different representations of the data that can be ensembled for improved performance

Fast Geometric Ensembling

Algorithm 1 Fast Geometric Ensembling

Require:
weights $\hat{w}$, LR bounds α_1, α_2,
cycle length c (even), number of iterations n
Ensure: ensemble
$w \leftarrow \hat{w}$ {Initialize weight with $\hat{w}$}
ensemble $\leftarrow [\,]$
for $i \leftarrow 1, 2, \ldots, n$ **do**
  $\alpha \leftarrow \alpha(i)$ {Calculate LR for the iteration}
  $w \leftarrow w - \alpha \nabla \mathcal{L}_i(w)$ {Stochastic gradient update}
  if $\mathrm{mod}(i, c) = c/2$ **then**
    ensemble $\leftarrow$ ensemble $+ [w]$ {Collect weights}
  end if
end for

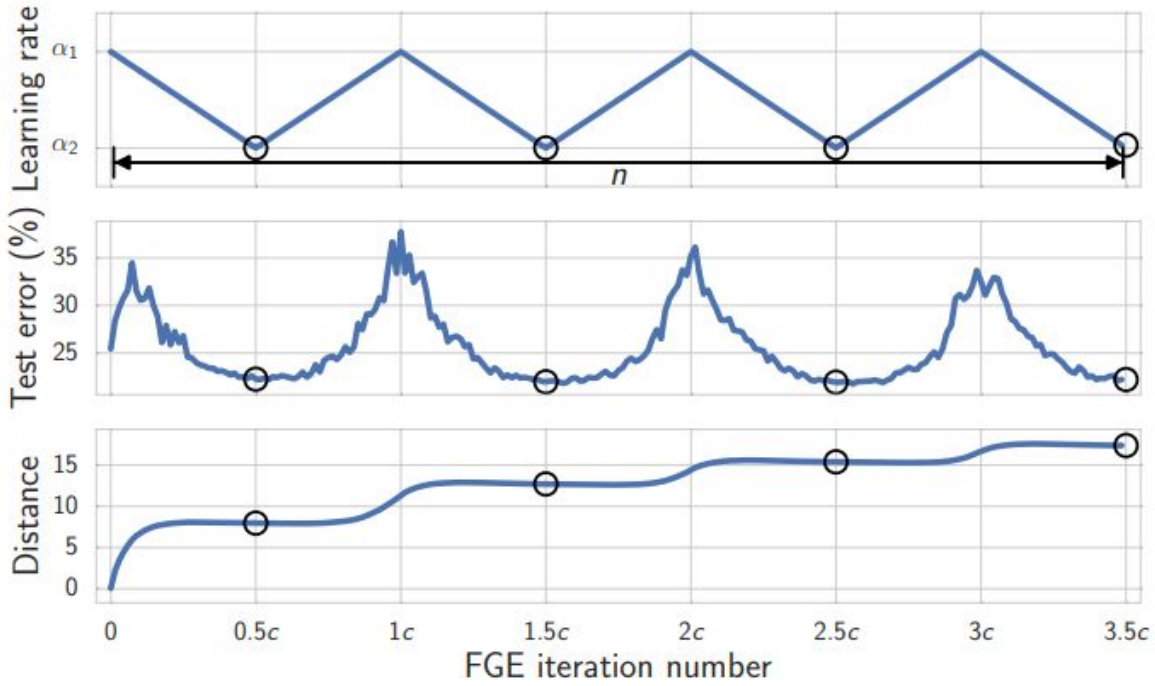
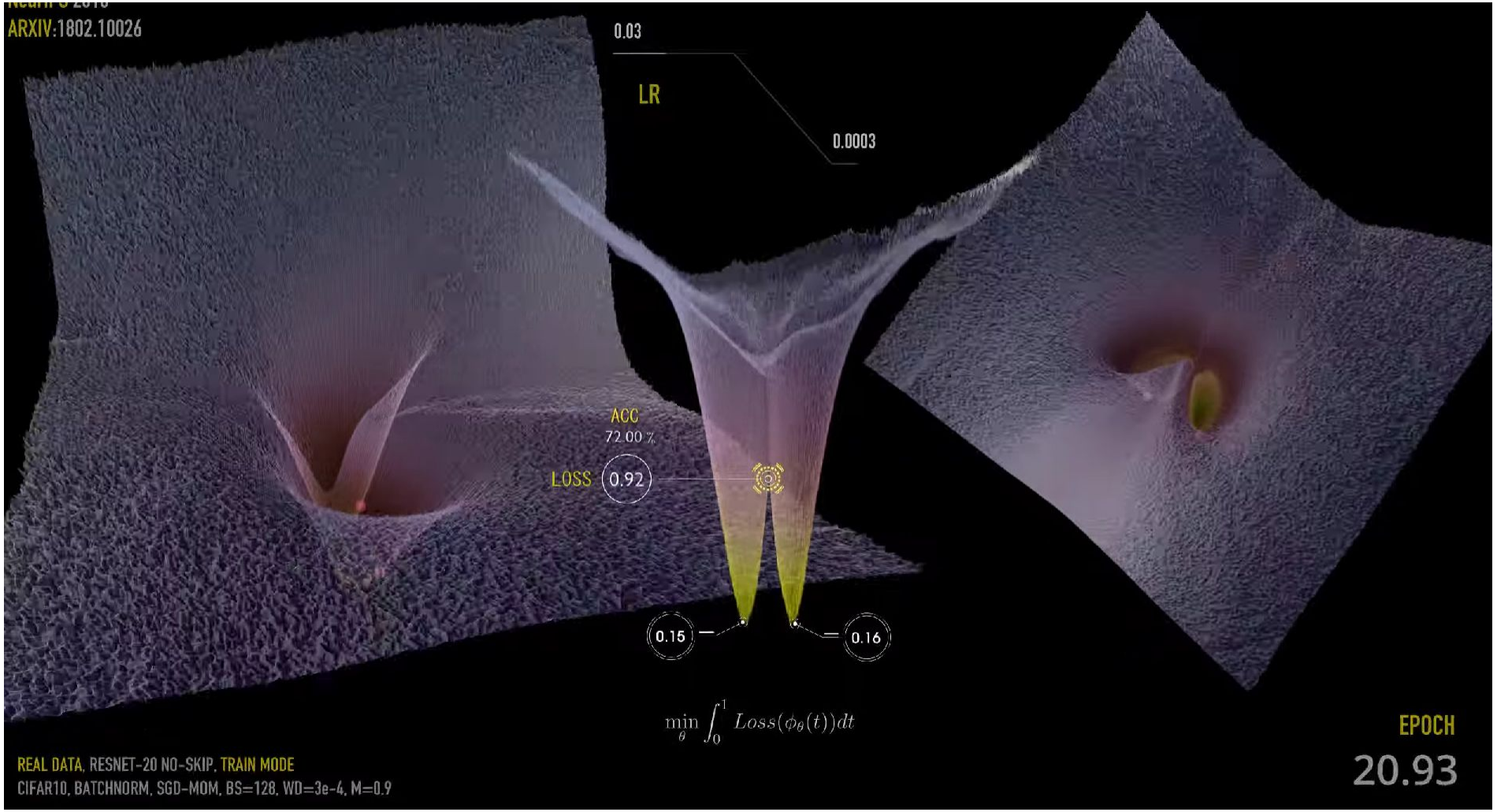


Table 1: Error rates (%) on CIFAR-100 and CIFAR-10 datasets for different ensembling techniques and training budgets. The best results for each dataset, architecture, and budget are **bolded**.

		CIFAR-100			CIFAR-10		
DNN (Budget)	method	$1B$	$2B$	$3B$	$1B$	$2B$	$3B$
VGG-16 (200)	Ind	27.4 ± 0.1	25.28	24.45	6.75 ± 0.16	5.89	5.9
	SSE	26.4 ± 0.1	25.16	24.69	6.57 ± 0.12	6.19	5.95
	FGE	$\mathbf{25.7 \pm 0.1}$	**24.11**	**23.54**	$\mathbf{6.48 \pm 0.09}$	**5.82**	**5.66**
ResNet-164 (150)	Ind	21.5 ± 0.4	19.04	18.59	4.72 ± 0.1	**4.1**	**3.77**
	SSE	20.9 ± 0.2	19.28	18.91	4.66 ± 0.02	4.37	4.3
	FGE	$\mathbf{20.2 \pm 0.1}$	**18.67**	**18.21**	$\mathbf{4.54 \pm 0.05}$	4.21	3.98
WRN-28-10 (200)	Ind	19.2 ± 0.2	17.48	17.01	3.82 ± 0.1	3.4	**3.31**
	SSE	17.9 ± 0.2	17.3	16.97	3.73 ± 0.04	3.54	3.55
	FGE	$\mathbf{17.7 \pm 0.2}$	**16.95**	**16.88**	$\mathbf{3.65 \pm 0.1}$	**3.38**	3.52



https://www.youtube.com/watch?v=dqX2LBcp5Hs

Conclusion

- Network architecture, optimizer selection, and batch size affect loss landscape.
- Experiments suggest that the small volume of bad minima prevents optimizers from landing in them.
- Even though the loss surfaces of deep neural networks are very complex, there is relatively simple structure connecting different optima.
- These geometric insights could also be used to accelerate the convergence, stability and accuracy of models.

Очень рекомендую

Лекция Д.П.Ветрова «Удивительные свойства ландшафта функции потерь»
https://www.youtube.com/watch?v=DCivw55WJNg&list=PL8Ln1vvt4h5dTWfcJjNfXPCCkHvmiwrdZ&index=38
https://www.youtube.com/watch?v=TTO2V4mf9kE&list=PL8Ln1vvt4h5dTWfcJjNfXPCCkHvmiwrdZ&index=50

Интересные статьи:
Блог гугла про гроккинг
https://pair.withgoogle.com/explorables/grokking/
«The Goldilocks Zone: Towards Better Understanding of Neural Network Loss Landscapes»
https://arxiv.org/abs/1807.02581

Papers

“Visualizing the Loss Landscape of Neural Nets”

https://proceedings.neurips.cc/paper_files/paper/2018/file/a41b3bb3e6b050b6c9067c67f663b915-Paper.pdf

“Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs”

https://ojs.aaai.org/index.php/AAAI/article/view/4237

“Understanding Generalization through Visualizations”

https://arxiv.org/abs/1906.03291