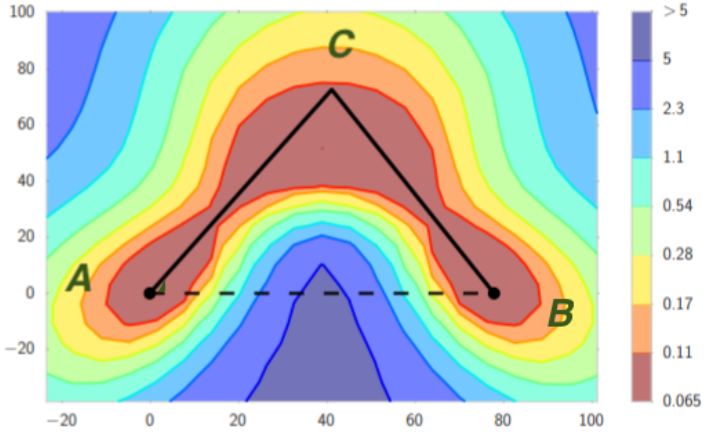
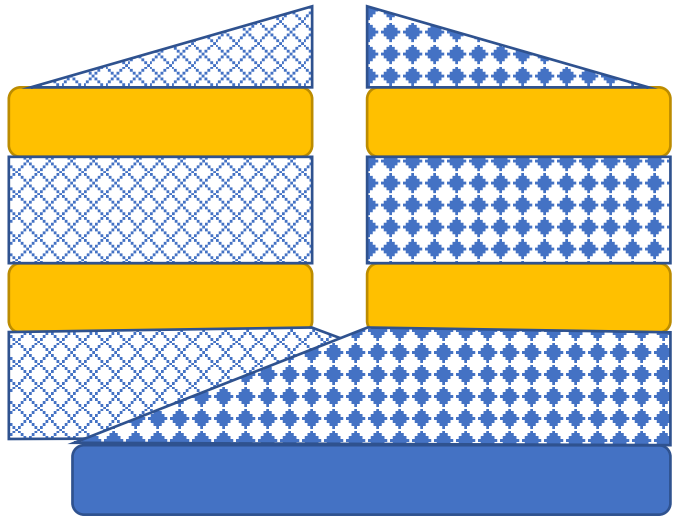


Explaining Landscape Connectivity of Low-cost Solutions for Multilayer Nets

Rong Ge, Duke University

Joint work with Rohith Kuditipudi, Xiang Wang (Duke)
Holden Lee, Yi Zhang, Zhiyuan Li, Wei Hu, Sanjeev Arora (Princeton)



Mode Connectivity[Freeman and Bruna 16, Garipov et al. 18, Draxler et al. 18]

- For neural networks, local minima found via gradient descent are connected by simple paths in the parameter space
- Every point on the path is another solution of almost the same cost.

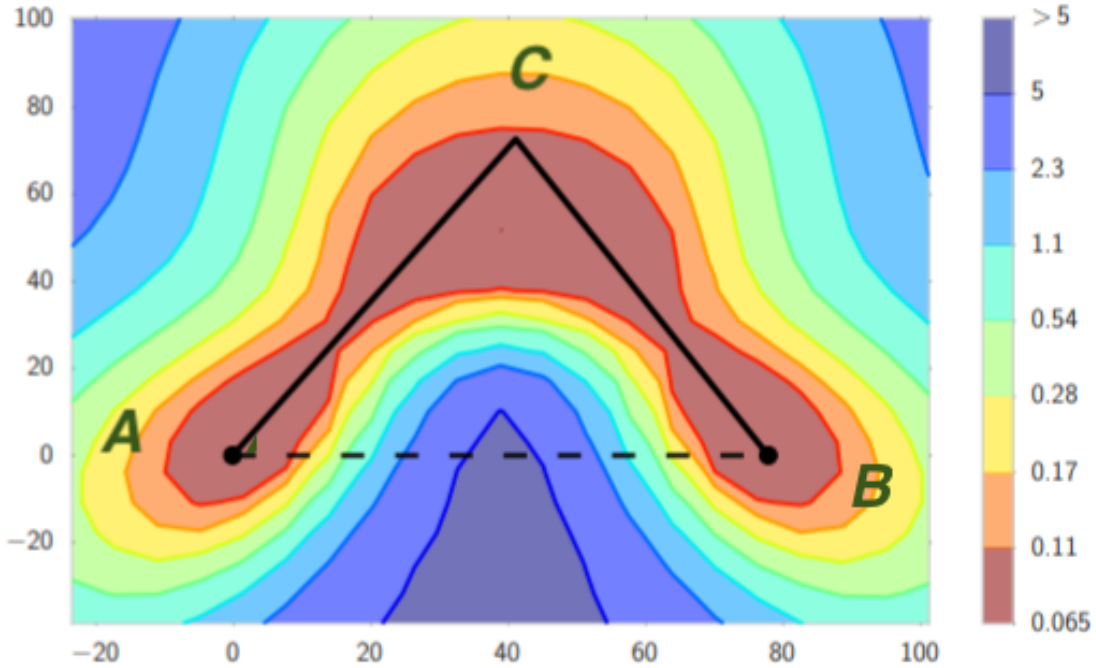


Image from [Garipov et al. 18]

Outline

Motivation: Why I was very surprised when I learned about mode connectivity.

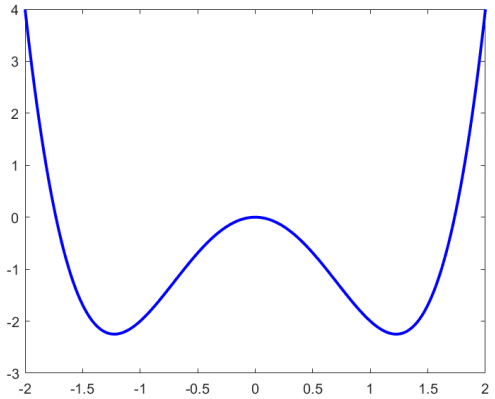
Our results:
Local min that are robust to dropout are connected.

Open problem:
Mode connectivity and optimization?

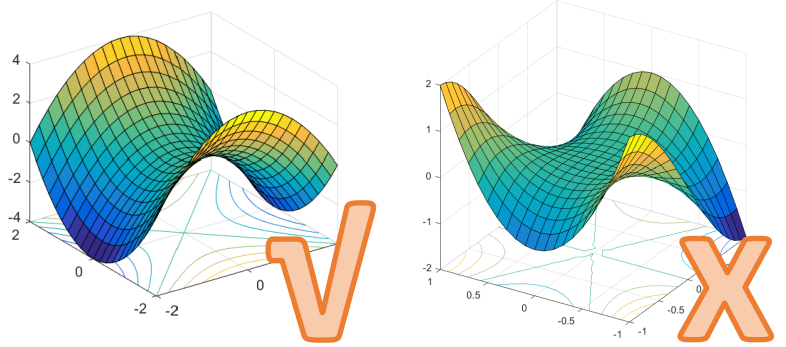
Background: non-convex optimization

- Many objectives are locally optimizable.

All local minima are globally optimal.

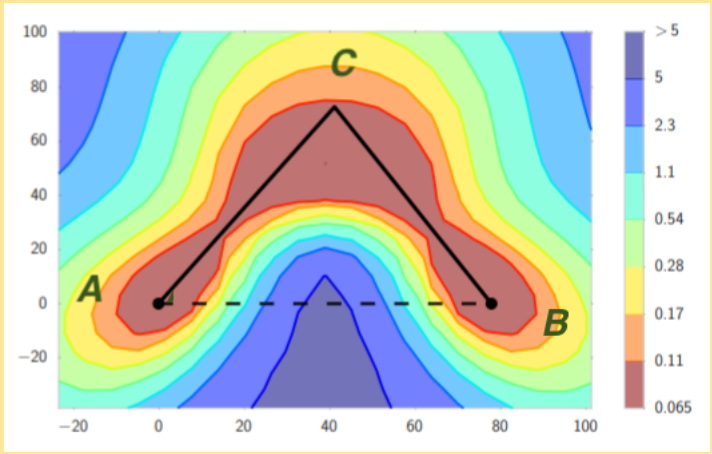


No high order saddle points.

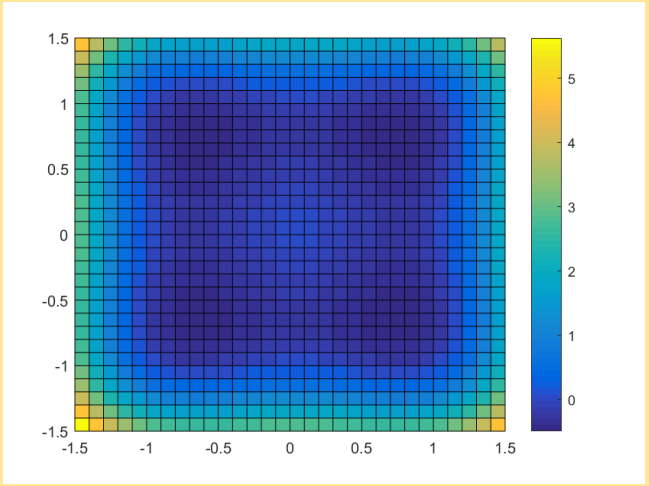


- Gradient Descent/other algorithms can find a local minimum.
  - [Jin, G, Netrapalli, Kakade, Jordan 17]
    Gradient descent finds an ϵ-approx local min in $\tilde{O}(1/\epsilon^2)$ iterations.
  - Many other algorithms known to work [Carmon et al. 2016, Agarwal et al. 2017]

Equivalent local minima and symmetry



(mostly) connected local min



Isolated local min

- Equivalent solutions:
  $X = X^*R, RR^\top = I$
- Equivalent solutions:
  $X = X^*P, P\ permutation$

Neural networks only have permutation symmetry,
why do they have connected local min?

(Partial) short answer: overparametrization

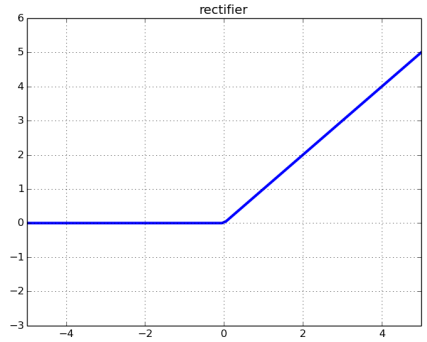
- Existing explanations of mode connectivity:
  [Freeman and Bruna, 2016, Venturi et al. 2018, Liang et al. 2018, Nguyen et al. 2018, Nguyen et al. 2019]
- If the network has special structure, and is very overparametrized (#neurons > #training samples), then local mins are connected.

- Problem: Networks that are not as overparametrized were also found to have connected local min.

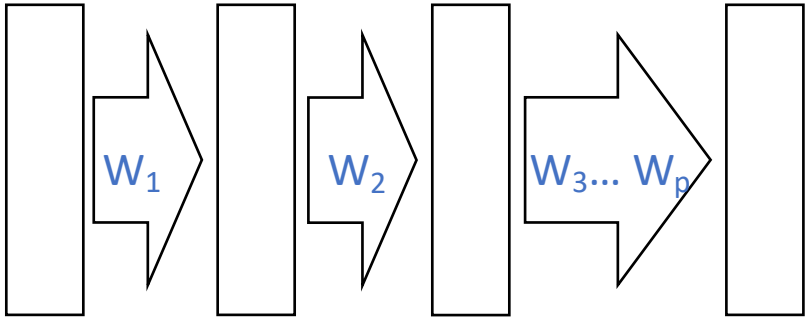
Our Results

- For neural networks, not all local/global min are connected, even in the overparametrized setting.
- Solutions that satisfy dropout stability are connected.
- Possible to switch dropout stability with noise stability (used for proving generalization bounds for neural nets)

Deep Neural Networks



- For simplicity: Fully Connected Networks
- Weights $\theta = (W_1, W_2, \ldots, W_p)$, nonlinearity σ
- Samples (x,y), hope to learn a network that maps x to y



$x^{(0)} = x \quad x^{(1)} = \sigma(W_1 x) \qquad\qquad x^{(p)} = W_p x^{(p-1)}$

- Function $f_\theta(x) = W_p\sigma(W_{p-1}\sigma(\cdots\sigma(W_1 x)\cdots)$
- Objective: $L(\theta) = \frac{1}{n}\sum_{i=1}^{n} l(y_i, f_\theta(x_i))$

Convex loss function

Not all local min are connected

- Simple setting: 2-layer net, data (x_i, y_i) generated by ground truth neural network with 2 hidden neurons.
- Overparametrization: consider optimization of a 2 layer neural network with h (h >> 2) hidden neurons.

- Theorem: For any h > 2, there exists a data-set with h+2 samples, such that the set of global minimizers are not connected.

What kind of local min are connected?

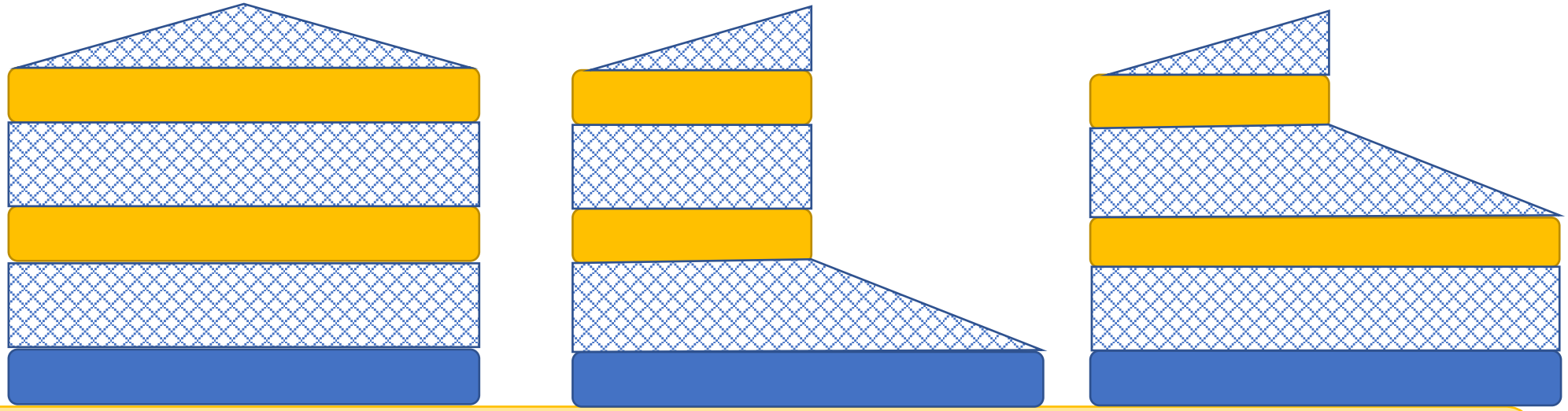
- Only local min found by standard optimization algorithms are known to be connected.

- Properties of such local min?
  - Closely connected to the question of generalization/implicit regularization.
  - Many conjectures: “flat” local min, margin, etc.

- This talk: Dropout stability

Dropout stability

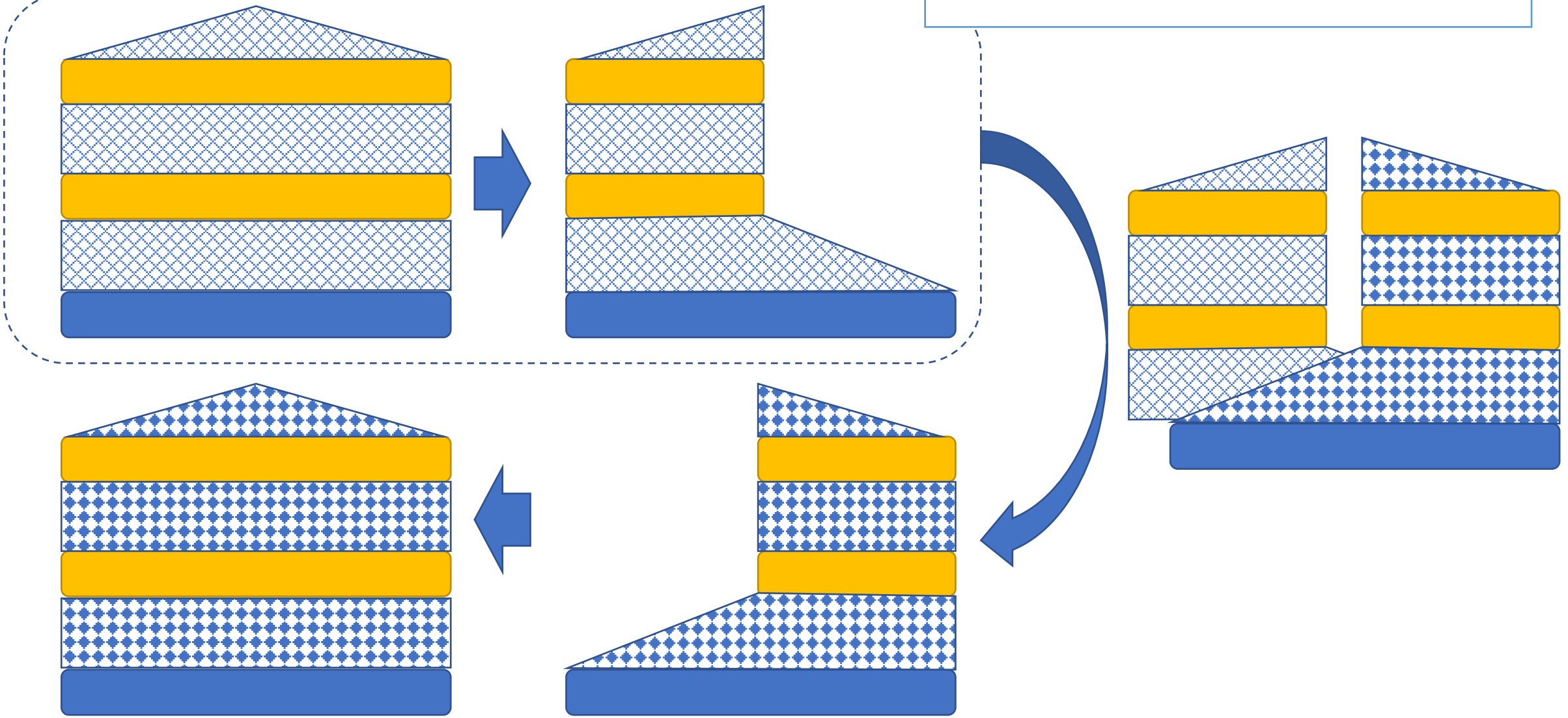
- A network is ε-dropout stable, if zeroing out 50% nodes at every layer (and rescale others appropriately) increases its loss by at most ε.



- Theorem: If both θ_A and θ_B are ε-dropout stable, then there exists a path between them with maximum loss $\leq \max\{L(\theta_A), L(\theta_B)\} + \varepsilon$

High level steps

How to connect a network with its dropout version?



Direct Interpolation

- Direct interpolation between the weights does not work.
- Even for a simple two layer linear network, if

$$f_A(x) = U_1^\top W_1 x, \qquad f_B(x) = U_2^\top W_2 x$$

- Interpolation between the parameters with coefficient $(\alpha, 1-\alpha)$

$$f_\alpha(x) = [\alpha^2 U_1^\top W_1 + \alpha(1-\alpha)(U_1^\top W_2 + U_2^\top W_1) + (1-\alpha)^2 U_2^\top W_2]x$$

θ_A

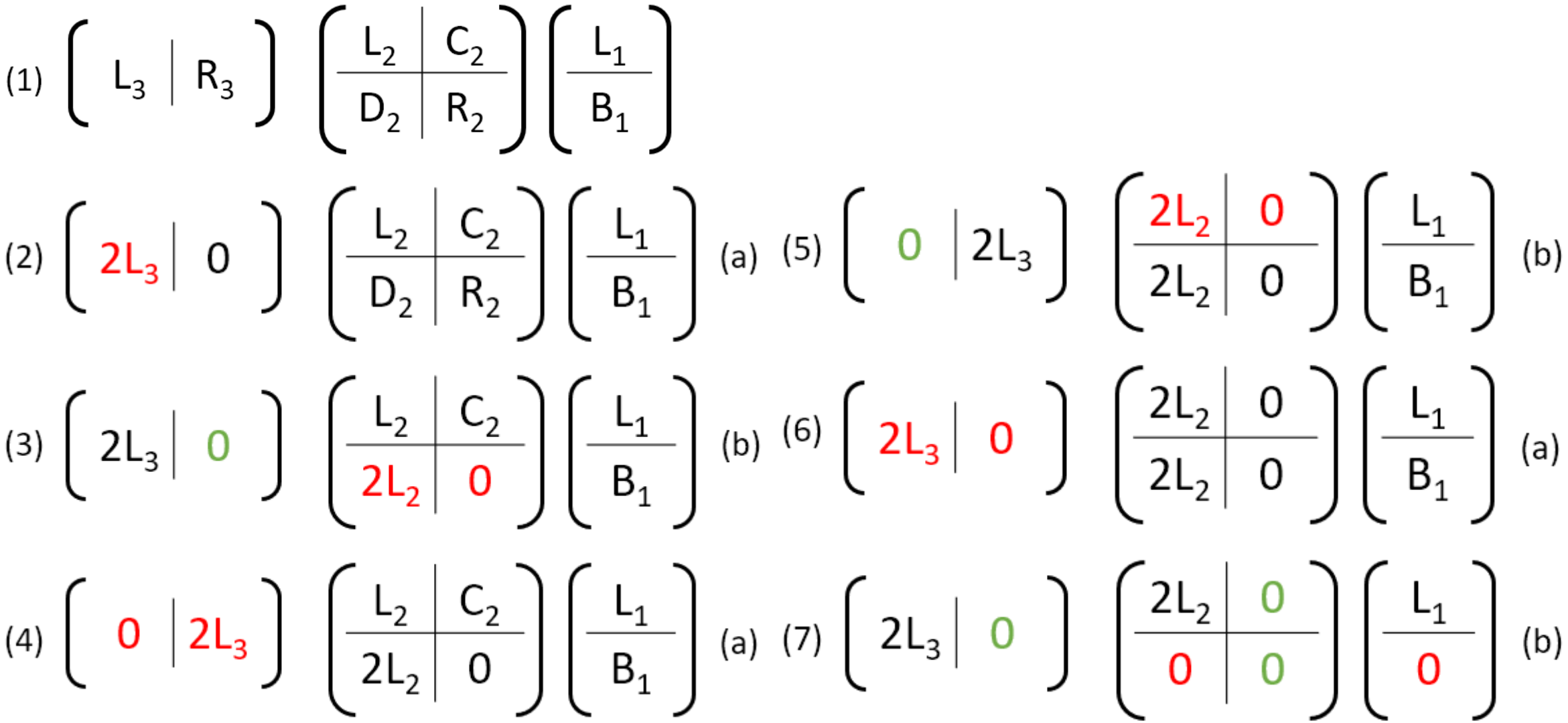
θ_B

- In general should have high cost.

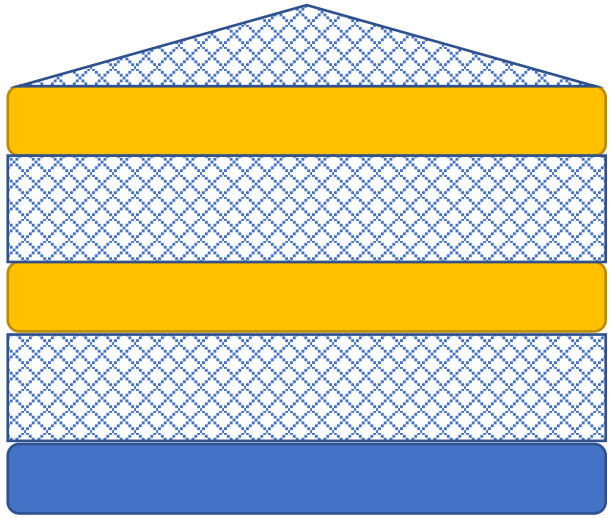
Connecting a network with its dropout

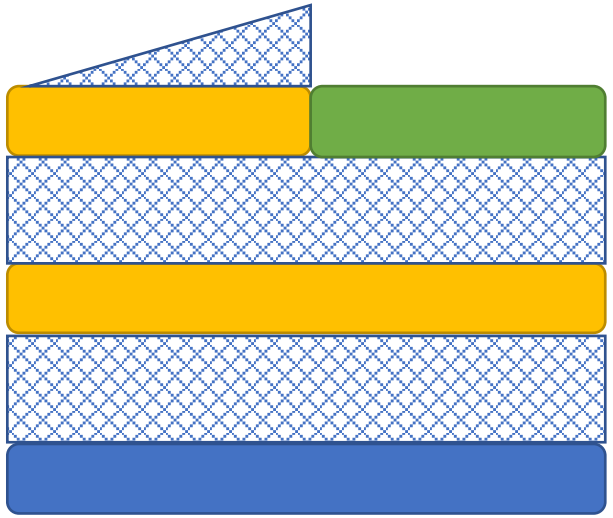
- Main observation: can use two types of line segments.
- Type (a): if θ_A and θ_B both have low loss, and they only differ in top layer weight, can linearly interpolate between them.
- Type (b): If a group of neurons do not have any outgoing edges, can change their incoming edges arbitrarily.
- Idea: Recurse from the top layer, use Type (b) moves to prepare for the next Type (a) move

An example path for 3 layer network



Example path explained (1) -> (2)

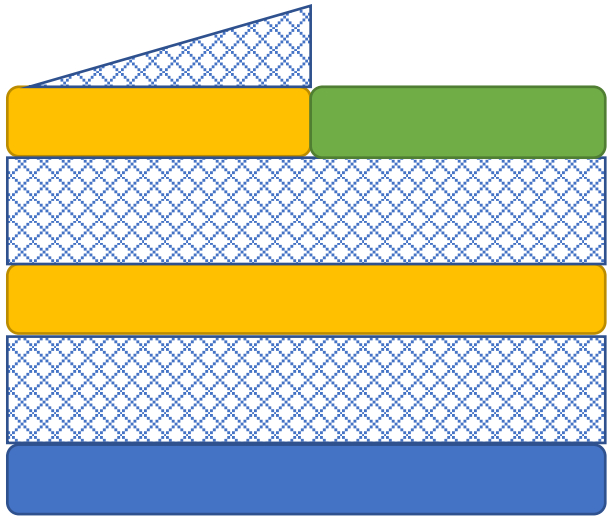


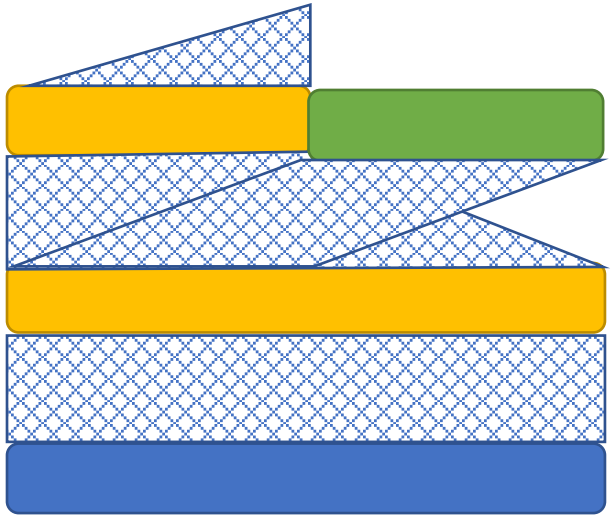


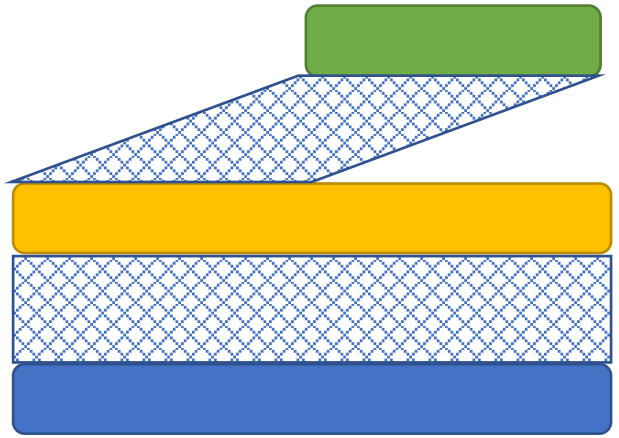
$$(1)\ \left(\begin{array}{c|c} L_3 & R_3 \end{array}\right)\ \left(\begin{array}{c|c} L_2 & C_2 \\ \hline D_2 & R_2 \end{array}\right)\ \left(\begin{array}{c} L_1 \\ \hline B_1 \end{array}\right)$$

$$(2)\ \left(\begin{array}{c|c} 2L_3 & 0 \end{array}\right)\ \left(\begin{array}{c|c} L_2 & C_2 \\ \hline D_2 & R_2 \end{array}\right)\ \left(\begin{array}{c} L_1 \\ \hline B_1 \end{array}\right)\ (a)$$

Example path explained (2) -> (3)

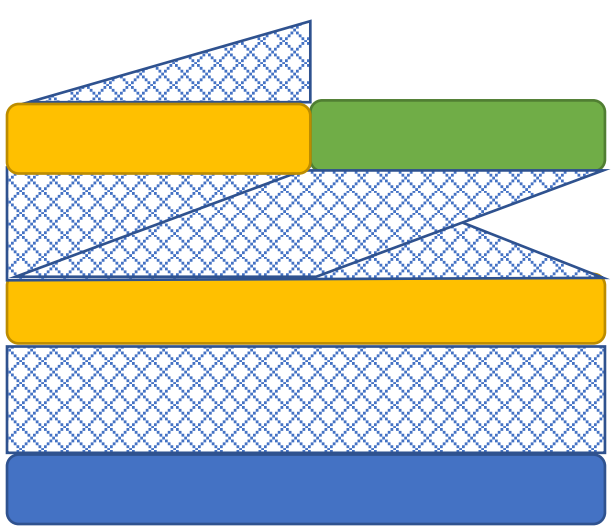


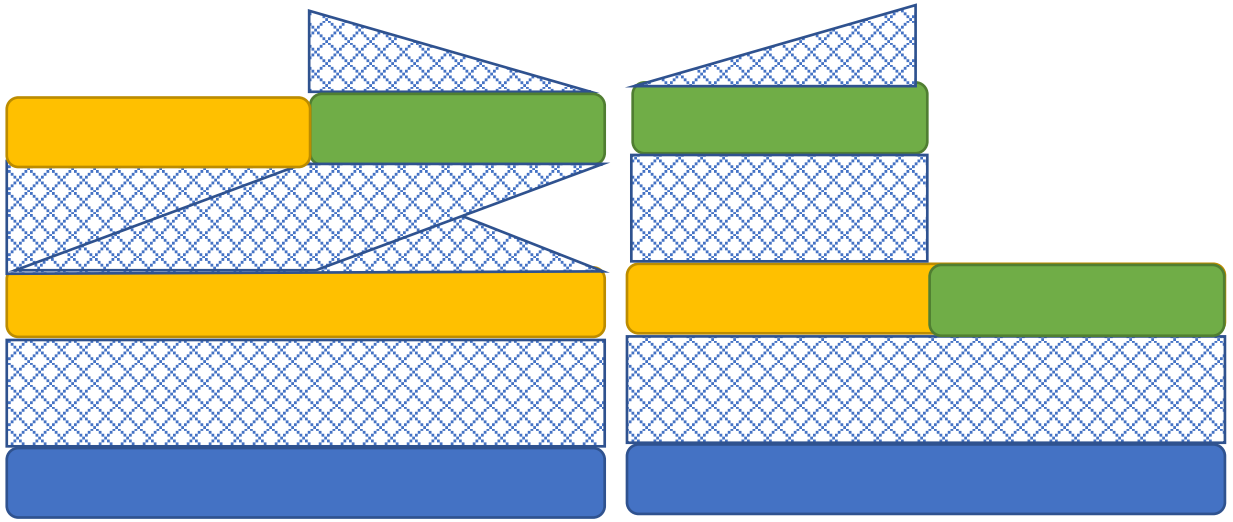




$$(2)\ \left(\begin{array}{c|c} 2L_3 & 0 \end{array}\right)\ \left(\begin{array}{c|c} L_2 & C_2 \\ \hline D_2 & R_2 \end{array}\right)\ \left(\begin{array}{c} L_1 \\ \hline B_1 \end{array}\right)\ (a)\quad (3)\ \left(\begin{array}{c|c} 2L_3 & 0 \end{array}\right)\ \left(\begin{array}{c|c} L_2 & C_2 \\ \hline 2L_2 & 0 \end{array}\right)\ \left(\begin{array}{c} L_1 \\ \hline B_1 \end{array}\right)\ (b)$$

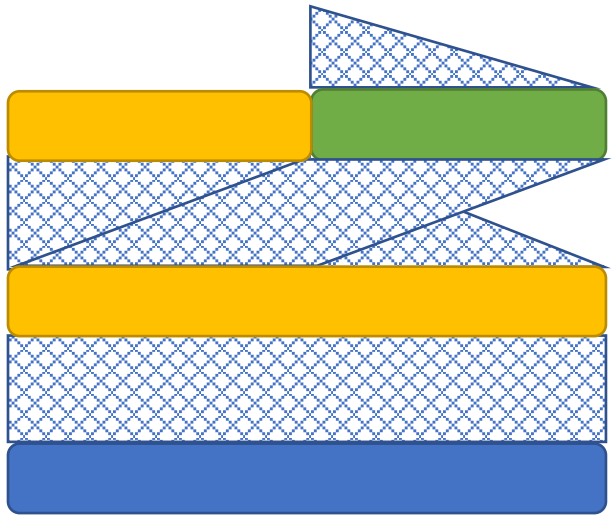
Example path explained (3) -> (4)

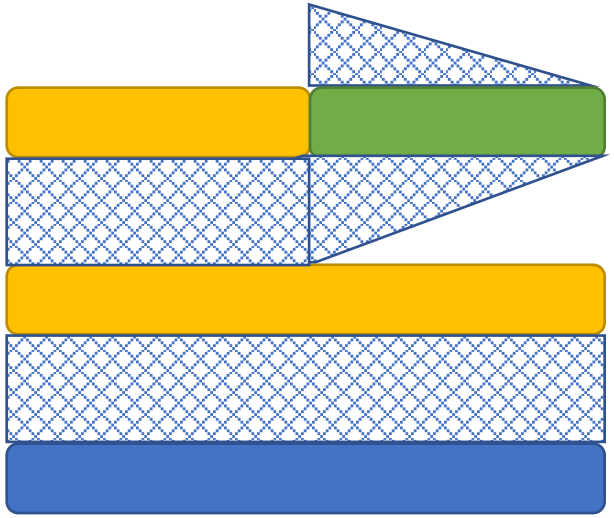




$$(3)\ \left(\begin{array}{c|c} 2L_3 & 0 \end{array}\right) \left(\begin{array}{c|c} L_2 & C_2 \\ \hline 2L_2 & 0 \end{array}\right) \left(\begin{array}{c} L_1 \\ \hline B_1 \end{array}\right) (b) \qquad (4)\ \left(\begin{array}{c|c} 0 & 2L_3 \end{array}\right) \left(\begin{array}{c|c} L_2 & C_2 \\ \hline 2L_2 & 0 \end{array}\right) \left(\begin{array}{c} L_1 \\ \hline B_1 \end{array}\right) (a)$$

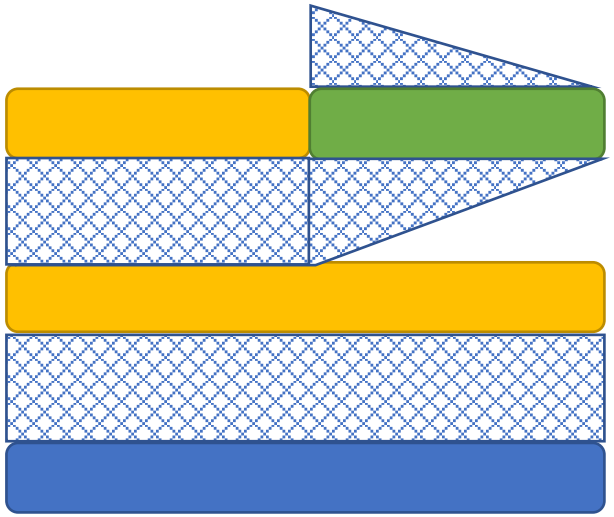
Example path explained (4) -> (5)

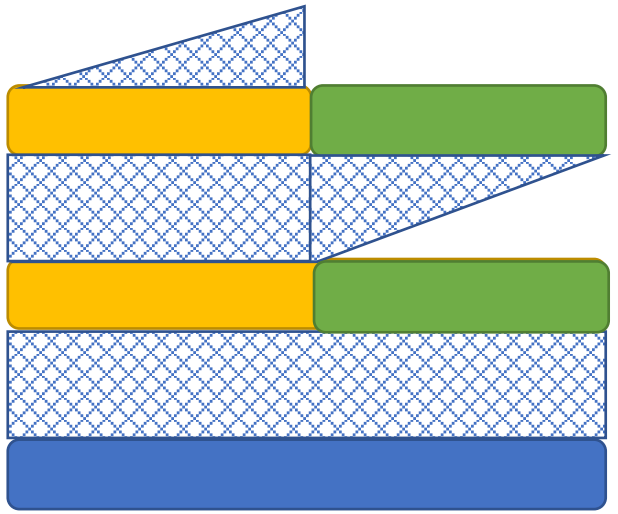




$$(4)\ \begin{pmatrix} 0 & 2L_3 \end{pmatrix} \left(\begin{array}{c|c} L_2 & C_2 \\ \hline 2L_2 & 0 \end{array}\right) \begin{pmatrix} L_1 \\ \hline B_1 \end{pmatrix} \quad (a) \quad (5)\ \begin{pmatrix} 0 & 2L_3 \end{pmatrix} \left(\begin{array}{c|c} 2L_2 & 0 \\ \hline 2L_2 & 0 \end{array}\right) \begin{pmatrix} L_1 \\ \hline B_1 \end{pmatrix} \quad (I$$

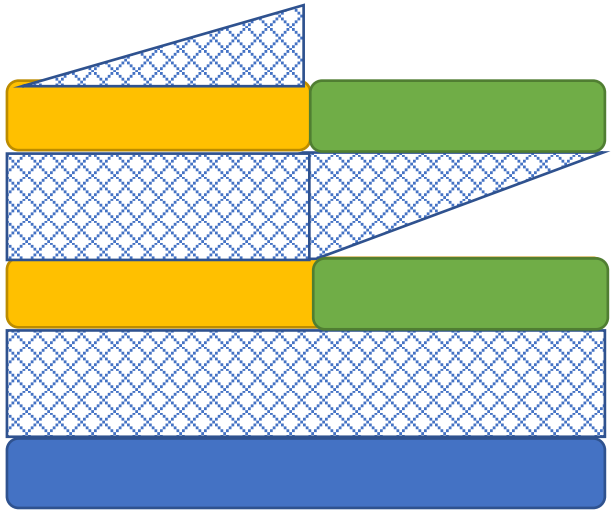
Example path explained (5) -> (6)

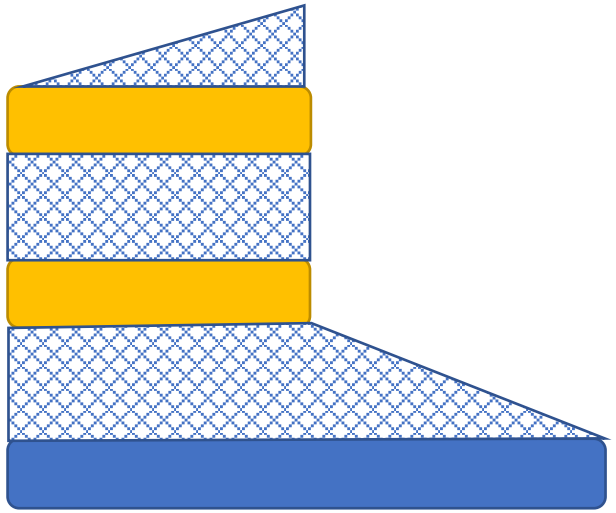




$$(5)\ \left(\begin{array}{c|c} 0 & 2L_3 \end{array}\right) \left(\begin{array}{c|c} 2L_2 & 0 \\ \hline 2L_2 & 0 \end{array}\right) \left(\begin{array}{c} L_1 \\ \hline B_1 \end{array}\right) \quad (b) \quad (6)\ \left(\begin{array}{c|c} 2L_3 & 0 \end{array}\right) \left(\begin{array}{c|c} 2L_2 & 0 \\ \hline 2L_2 & 0 \end{array}\right) \left(\begin{array}{c} L_1 \\ \hline B_1 \end{array}\right) \quad (a)$$

Example path explained (6) -> (7)





(6) $\left(\begin{array}{c|c} 2L_3 & 0 \end{array}\right)$ $\left(\begin{array}{c|c} 2L_2 & 0 \\ \hline 2L_2 & 0 \end{array}\right)$ $\left(\begin{array}{c} L_1 \\ \hline B_1 \end{array}\right)$ (a) (7) $\left(\begin{array}{c|c} 2L_3 & 0 \end{array}\right)$ $\left(\begin{array}{c|c} 2L_2 & 0 \\ \hline 0 & 0 \end{array}\right)$ $\left(\begin{array}{c} L_1 \\ \hline 0 \end{array}\right)$ (b)

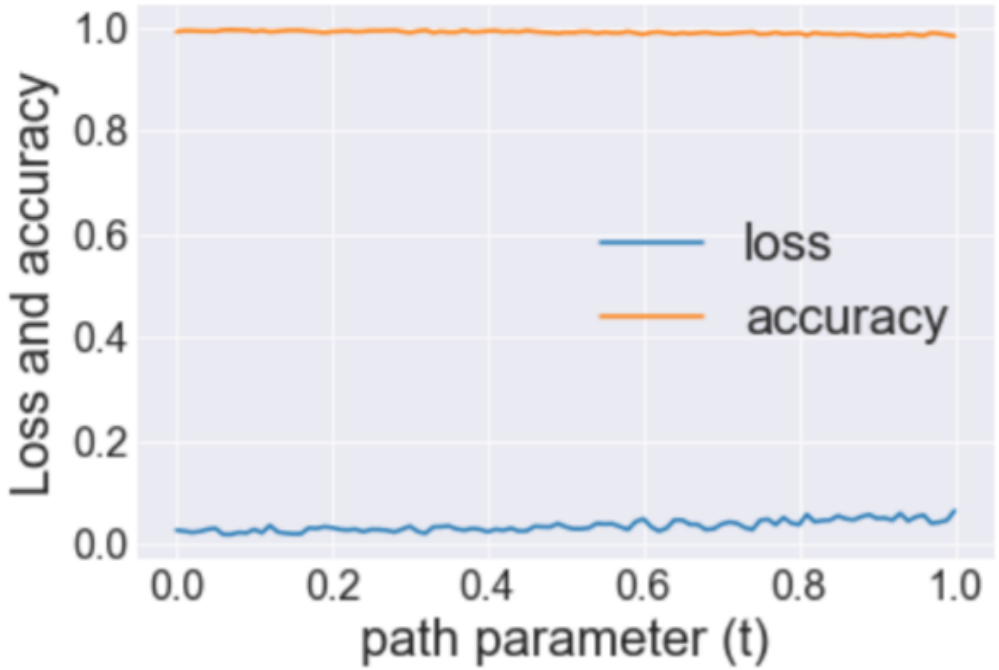
Noise stability

- A network is noise stable, if injecting noise at intermediate layers does not change the output by too much.
- Precise definition similar to [Arora et al. 2018]

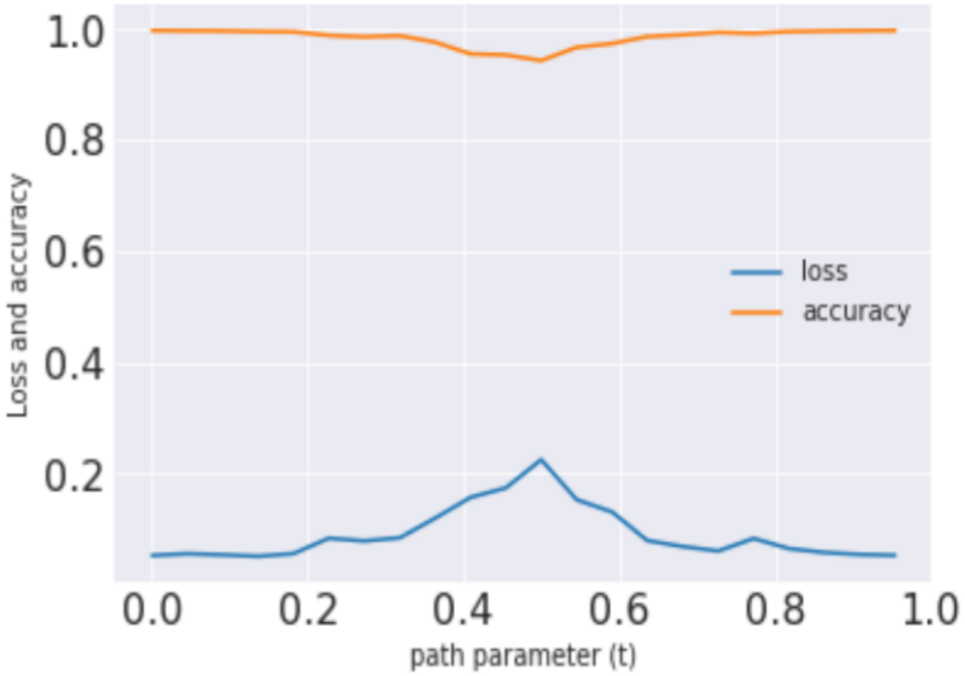
- Theorem: If both θ_A and θ_B are ε-noise stable, then there is a path between them with maximum loss $\leq \max\{L(\theta_A), L(\theta_B)\} + \varepsilon$. The path consists of 10 line segments.

- Idea: noise stability ➔ dropout stability, further, noise stability allow us to do direct interpolation between a network and its dropout.

Experiments



MNIST, 3-layer CNN



CIFAR-10, VGG-11

Conclusions

For neural networks, not all local/global min are connected, even in the overparametrized setting.

Solutions that satisfy dropout/noise stability are connected.

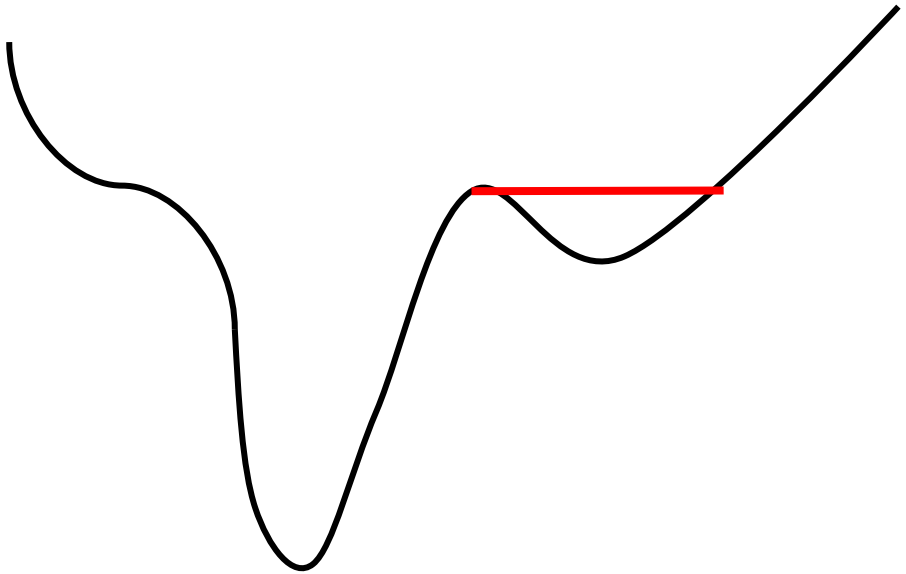
Open Problems

- Path found by dropout/noise stability are still more complicated than the path found in practice.
- Path are known to exist in practice, even if the solutions are not as dropout stable as we hoped.

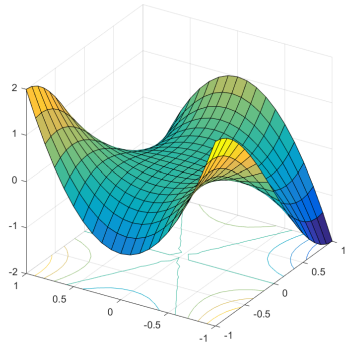
- Can we leverage mode connectivity to design better optimization algorithms?

How can we use mode connectivity?

- If all local min are connected, then all the level sets are also connected.
- If all “typical solutions” are connected (and there is a typical global min), local search algorithms will not be completely stuck.



- However, there can still be flat regions/high order saddle points.
- Can better optimization/sampling algorithms leverage mode connectivity?



Thank you!