Optimization of neural networks

Dmitry Yarotsky

Parametrized predictive models

True response function: $y = f(\mathbf{x})$, where $\mathbf{x}$ is the input vector

- $y \in \mathbb{R}$ for regression
- $y \in \{0, 1\}$ for binary classification

Predictive model: $y = \widetilde{f}(\mathbf{x}, \mathbf{W})$, and $\mathbf{W}$ are model parameters (e.g., network weights)

"Soft classification": $\widetilde{f}(\mathbf{x}, \mathbf{W}) \in [0, 1]$

Model training as a parametric optimization

Loss function: $L(\mathbf{W}) = \int l(f(\mathbf{x}), \widetilde{f}(\mathbf{x}, \mathbf{W}))d\mu(\mathbf{x}) = \mathbb{E}_{\mathbf{x}\sim\mu} l(f(\mathbf{x}), \widetilde{f}(\mathbf{x}, \mathbf{W}))$

Sample average measure μ :

- $d\mu(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N}\delta(\mathbf{x} - \mathbf{x}_n)$ with Dirac's delta — "finite training set" scenario
- $d\mu(\mathbf{x}) = p(\mathbf{x})d\mathbf{x}$ with some (e.g. Gaussian) density $p(\mathbf{x})$ — "population average" scenario

Function $l(f(\mathbf{x}), \widetilde{f}(\mathbf{x}, \mathbf{W}))$ measures the discrepancy between f and $\widetilde{f}$, e.g.:

- Regression: $l(y, \widetilde{y}) = \frac{1}{2}(y - \widetilde{y})^2$
- Classification: $l(y, \widetilde{y}) = -y \log \widetilde{y} - (1 - y) \log(1 - \widetilde{y})$

Model training:

$$L(\mathbf{W}) \longrightarrow \min_{\mathbf{W}}$$

Gradient-based optimization

- $\mathbf{W}$ high-dimensional
- $L(\mathbf{W})$ non-smooth, non-nonconvex

Most popular approach: gradient-based optimization and its modifications

Basic gradient descent with learning rate $\alpha > 0$:

$$\mathbf{W}^{(n+1)} = \mathbf{W}^{(n)} - \alpha \nabla_{\mathbf{W}} L(\mathbf{W}^{(n)})$$

Gradient descent with momentum ("Heavy ball"); $\beta \in (0, 1)$:

$$\mathbf{W}^{(n+1)} = \mathbf{W}^{(n)} - \mathbf{V}^{(n)}$$
$$\mathbf{V}^{(n+1)} = \alpha \nabla_{\mathbf{W}} L(\mathbf{W}^{(n)}) + \beta \mathbf{V}^{(n)}$$

Exercise: how can we interprete the coefficients α and β?

Computation of $\nabla_{\mathbf{W}} L$: "Error backpropagation"

$$\nabla_{\mathbf{W}} L(\mathbf{W}) = \int \frac{\partial l}{\partial \widetilde{y}}(f(\mathbf{x}), \widetilde{f}(\mathbf{x}, \mathbf{W})) \cdot \nabla_{\mathbf{W}} \widetilde{f}(\mathbf{x}, \mathbf{W}) d\mu(\mathbf{x})$$

$$\nabla_{\mathbf{W}} \widetilde{f} = (\nabla_{\mathbf{w}_1} \widetilde{f}, \ldots, \nabla_{\mathbf{w}_K} \widetilde{f})$$

$\frac{\partial l}{\partial \widetilde{y}}(f(\mathbf{x}), \widetilde{f}(\mathbf{x}, \mathbf{W}))$: directly computed from y and $\widetilde{y}$

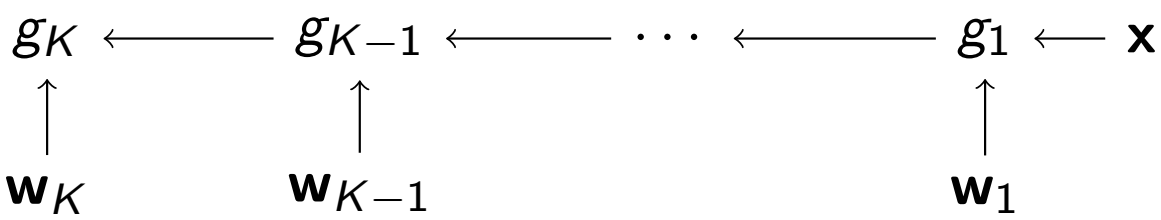
$\widetilde{y} = \widetilde{f}(\mathbf{x}, \mathbf{W})$: "forward propagation"

To find $\nabla_{\mathbf{W}} \widetilde{f}$: use layerwise representation

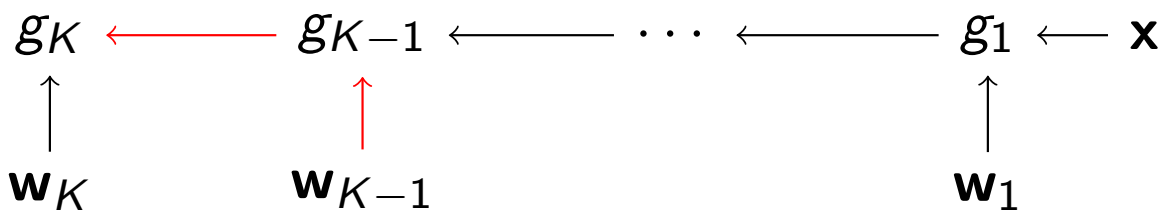
$$\widetilde{f}(\mathbf{x}, \mathbf{W}) = g_K(g_{K-1}(\ldots g_1(\mathbf{x}, \mathbf{w}_1), \ldots \mathbf{w}_{K-1}), \mathbf{w}_K)$$

$\mathbf{z}_k$: output of the k'th layer (known from "forward propagation")

$$\mathbf{z}_k = g_k(\mathbf{z}_{k-1}, \mathbf{w}_k)$$



“Error backpropagation”



$$\mathbf{z}_k = g_k(\mathbf{z}_{k-1}, \mathbf{w}_k)$$

$$\nabla_{\mathbf{w}_K} \widetilde{f}(\mathbf{x}, \mathbf{W}) = \frac{\partial g_K}{\partial \mathbf{w}_K}(\mathbf{z}_{K-1}, \mathbf{w}_K)$$

$$\begin{aligned}\nabla_{\mathbf{w}_{K-1}} \widetilde{f}(\mathbf{x}, \mathbf{W}) &= \nabla_{\mathbf{w}_K} g_K(g_{K-1}(\mathbf{z}_{K-2}, \mathbf{w}_{K-1}), \mathbf{w}_K) \\ &= \frac{\partial g_K}{\partial \mathbf{z}_{K-1}}(\mathbf{z}_{K-1}, \mathbf{w}_K) \cdot \frac{\partial g_{K-1}}{\partial \mathbf{w}_{K-1}}(\mathbf{z}_{K-2}, \mathbf{w}_{K-1})\end{aligned}$$

$$\cdots$$

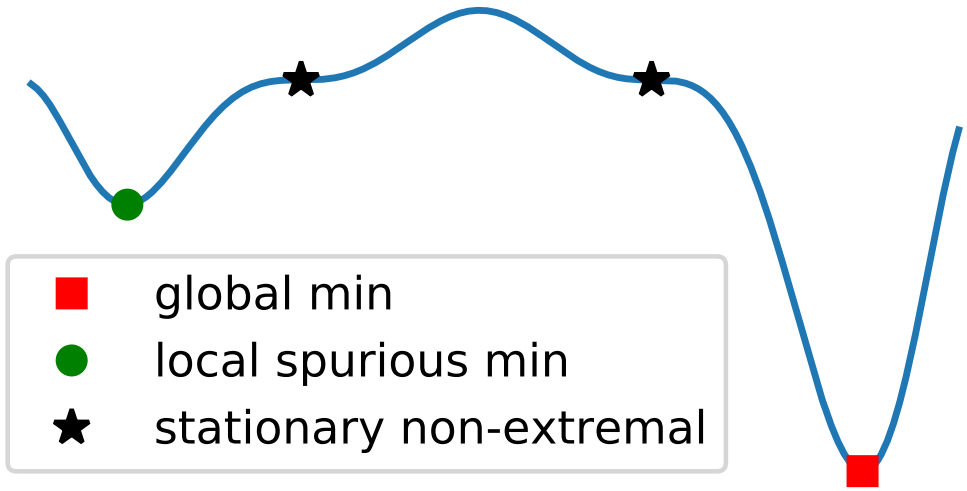
$$\begin{aligned}\nabla_{\mathbf{w}_k} \widetilde{f}(\mathbf{x}, \mathbf{W}) = &\frac{\partial g_K}{\partial \mathbf{z}_{K-1}}(\mathbf{z}_{K-1}, \mathbf{w}_K) \cdots \frac{\partial g_{k+1}}{\partial \mathbf{z}_k}(\mathbf{z}_k, \mathbf{w}_{k+1}) \\ &\cdot \frac{\partial g_k}{\partial \mathbf{w}_k}(\mathbf{z}_{k-1}, \mathbf{w}_k)\end{aligned}$$

Backpropagation: key property

Exercise: If the network includes W weights and performs N elementary operations, then computing $\nabla_{\mathbf{W}} L$ by backpropagation requires $O(N)$ operations. In contrast, computing the approximate gradient using finite differences requires $O(WN)$ operations.

Standard terminology

- **Global minimum**: $L(\mathbf{W}_*) = \min_{\mathbf{W} \in \mathbb{R}^w} L(\mathbf{W})$
- **Local minimum**: $L(\mathbf{W}_*) = \min_{\mathbf{W} \in U} L(\mathbf{W})$, where U is an open neighborhood of $\mathbf{W}_*$
- **Spurious minimum**: a local, but non-global minimum
- **Stationary point**: $\nabla_{\mathbf{W}} L(\mathbf{W}_*) = 0$ (assuming $L(\mathbf{W})$ is smooth)

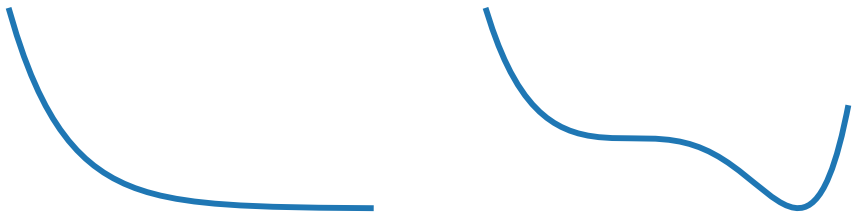


A basic convergence result[1]

Proposition

Suppose function L is lower bounded and differentiable, and $|\nabla L(\mathbf{a}) - \nabla L(\mathbf{b})| \leq M|\mathbf{a} - \mathbf{b}|$ with some Lipschitz constant M. Let $\alpha < \frac{2}{M}$. Then $\nabla L(\mathbf{W}^{(n)}) \to 0$, and $\min_{n=1,\ldots,N} |\nabla L(\mathbf{W}^{(n)})| = O(N^{-1/2})$.

Note: no convergence of $\mathbf{W}^{(n)}$ guaranteed, and if $\mathbf{W}^{(n)}$ converges, then not necessarily to a local minimum.



[1] See e.g. Yu. Nesterov, Introductory Lectures on Convex Programming Volume I: Basic course

Proof

$$
\begin{aligned}
L(\mathbf{W}^{(n+1)}) &= L(\mathbf{W}^{(n)}) + \left\langle \mathbf{W}^{(n+1)} - \mathbf{W}^{(n)}, \int_0^1 \nabla L(\mathbf{W}^{(n)} + t(\mathbf{W}^{(n+1)} - \mathbf{W}^{(n)}))dt \right\rangle \\
&\leq L(\mathbf{W}^{(n)}) + \left\langle \mathbf{W}^{(n+1)} - \mathbf{W}^{(n)}, \nabla L(\mathbf{W}^{(n)}) \right\rangle \\
&\quad + |\mathbf{W}^{(n+1)} - \mathbf{W}^{(n)}| \int_0^1 Mt|\mathbf{W}^{(n+1)} - \mathbf{W}^{(n)}|dt \\
&\leq L(\mathbf{W}^{(n)}) + \left\langle \mathbf{W}^{(n+1)} - \mathbf{W}^{(n)}, \nabla L(\mathbf{W}^{(n)}) \right\rangle + \tfrac{M}{2}|\mathbf{W}^{(n+1)} - \mathbf{W}^{(n)}|^2 \\
&\leq L(\mathbf{W}^{(n)}) + (-\alpha + \tfrac{M}{2}\alpha^2)|\nabla L(\mathbf{W}^{(n)})|^2 \\
&< L(\mathbf{W}^{(n)})
\end{aligned}
$$

if $\alpha < \frac{2}{M}$. Let $c = \alpha(1 - \frac{M}{2}\alpha) > 0$, then $L(\mathbf{W}^{(n+1)}) \leq L(\mathbf{W}^{(n)}) - c|\nabla L(\mathbf{W}^{(n)})|^2$ and hence

$$
\sum_{n=1}^{N} |\nabla L(\mathbf{W}^{(n)})|^2 \leq \frac{1}{c} \sum_{n=1}^{N} (L(\mathbf{W}^{(n)}) - L(\mathbf{W}^{(n+1)})) \leq \frac{1}{c}(L(\mathbf{W}^{(1)}) - \min_{\mathbf{W}} L(\mathbf{W})),
$$

$$
\min_{n=1,\ldots,N} |\nabla L(\mathbf{W}^{(n)})| \leq \frac{c^{-1/2}}{\sqrt{N}} \left(L(\mathbf{W}^{(1)}) - \min_{\mathbf{W}} L(\mathbf{W})\right)^{1/2}
$$

Formulation in terms of stopping condition

Assume the **stopping condition**: $|\nabla L(\mathbf{W}^{(n)})| < \epsilon$.

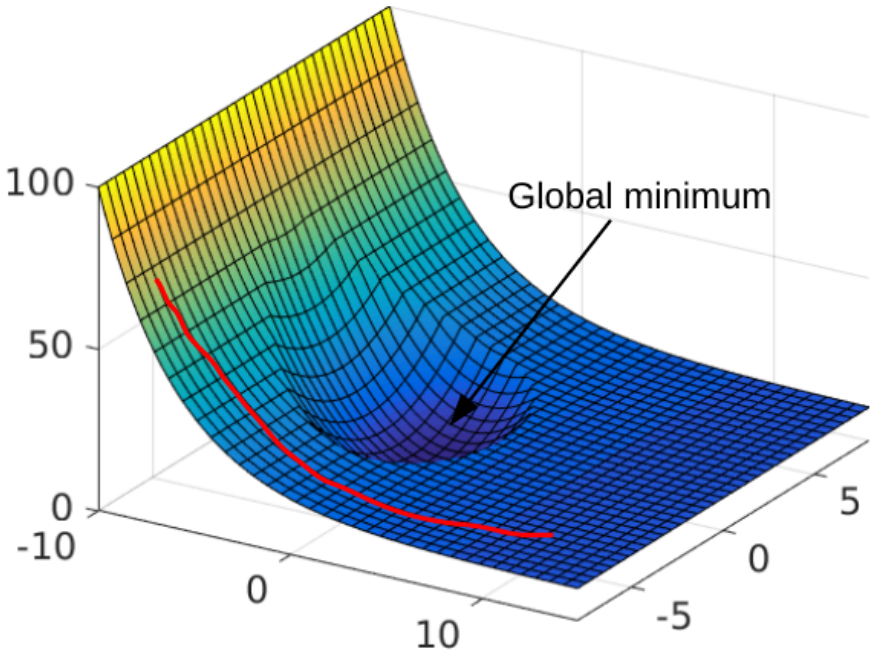
Then, optimization terminates in $O\Big(\frac{L(\mathbf{W}^{(1)}) - \min_{\mathbf{W}} L(\mathbf{W})}{\epsilon^2}\Big)$ steps.

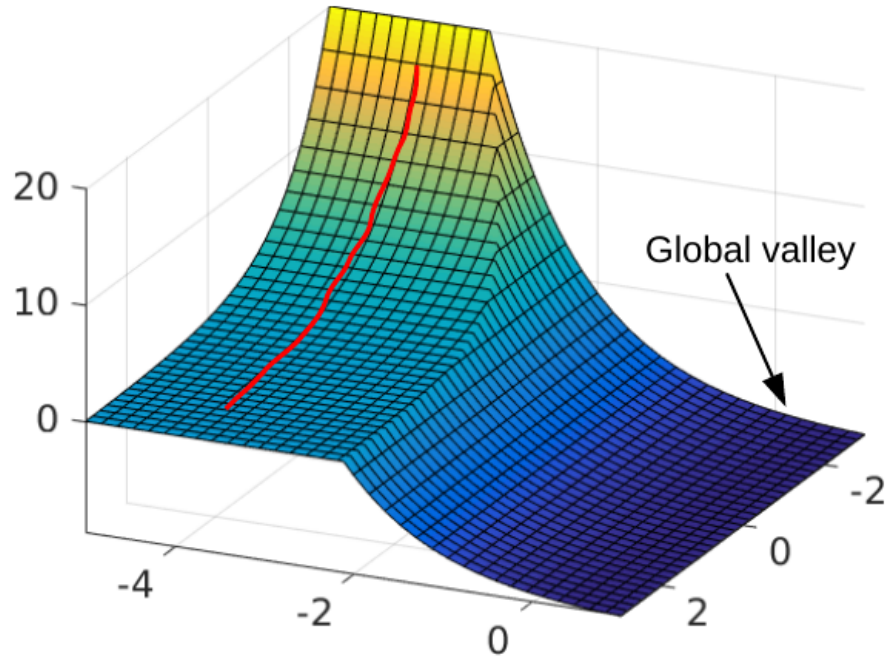
Exercise: What is the optimal value of α, assuming M is known?

Exercise: Give an example of gradient descent converging to a stationary point which is not a (local or global) minimum.

Missing the global minimum

Even for simple nonconvex L, gradient descent may never find the global minimum:





Q. Nguyen, On Connected Sublevel Sets in Deep Learning, arXiv:1901.07417

Manifestations of optimization failure

- Optimization can get stuck at suboptimal local minima
- Optimization iterates $\mathbf{W}^{(n)}$ may by unbounded and have no limit points
- Some regions of the loss surface may be "flat"

Exercise: Perform gradient descent numerically for a shallow network with 1D input. Observe whether GD converges to the global minimum and explain why if not.

NP-hardness of general non-convex optimization

Even quadratic programming, when non-convex, is NP-hard:
3-SAT, known to be NP-hard, can be reduced to it

3-SAT: Is the given conjunctive normal form satisfiable?

$$C_1 \wedge C_2 \wedge \ldots,$$

where $C_m = x_i \vee x_j \vee x_k$, and x_i, x_j, x_k are Boolean or their negations

Reduction[2]: satisfiability is equivalent to

$$\min_{x_1,\ldots,x_n} \sum_k (x_k - x_k^2) = 0$$

under constraints

$$0 \leq x_k \leq 1$$

and, for each clause like $x_i \vee \overline{x}_j \vee x_k$,

$$x_i + (1 - x_j) + x_k \geq 1$$

[2] https://cs.stackexchange.com/questions/17946/

Linearization and spectral analysis

Suppose that $L \in C^2(\mathbb{R})$ and $\mathbf{W}_*$ is a stationary point.
For $\mathbf{W}$ near $\mathbf{W}_*$:

$$\nabla L(\mathbf{W}) = D^2 L(\mathbf{W}_*) \cdot (\mathbf{W} - \mathbf{W}_*) + o(|\mathbf{W} - \mathbf{W}_*|),$$

where $D^2 L(\mathbf{W}_*)$ is the Hessian matrix. Optimization iterates:

$$\begin{aligned}
\mathbf{W}^{(n+1)} - \mathbf{W}_* &= \mathbf{W}^{(n)} - \mathbf{W}_* - \alpha \nabla L(\mathbf{W}^{(n)}) \\
&= \mathbf{W}^{(n)} - \mathbf{W}_* - \alpha D^2 L(\mathbf{W}_*) \cdot (\mathbf{W}^{(n)} - \mathbf{W}_*) + o(|\mathbf{W}^{(n)} - \mathbf{W}_*|) \\
&= (1 - \alpha D^2 L(\mathbf{W}_*)) \cdot (\mathbf{W}^{(n)} - \mathbf{W}_*) + o(|\mathbf{W}^{(n)} - \mathbf{W}_*|)
\end{aligned}$$

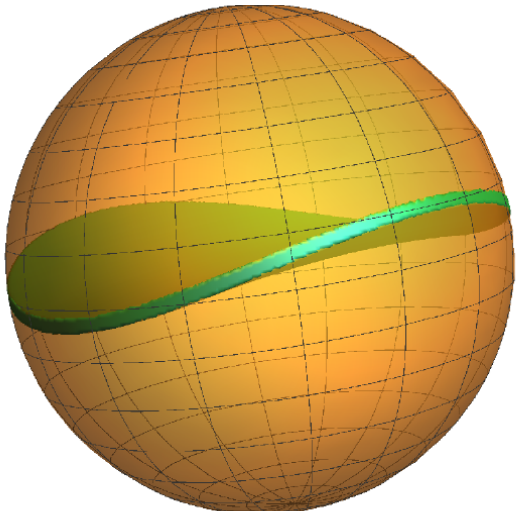
Convergence is determined by eigenvalues of $D^2 L(\mathbf{W}_*)$:

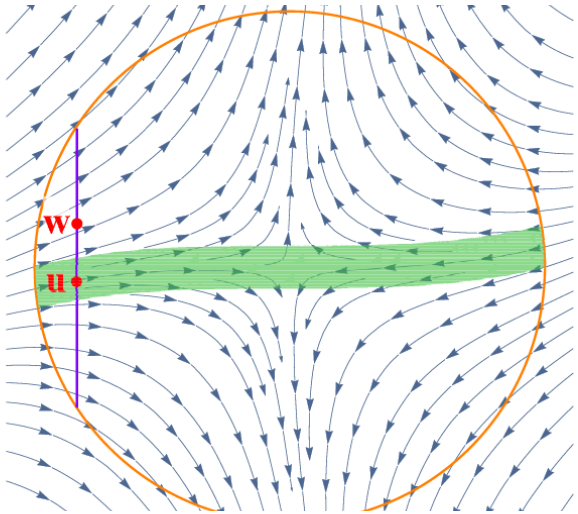
- positive: convergence
- negative: divergence

Evasion of saddle points

Saddle points: $D^2L(\mathbf{W}_*)$ has both positive and negative eigenvalues

Typically, saddles are evaded by optimization, due to the presence of diverging components in $\mathbf{W}^{(n)} - \mathbf{W}_*$. The manifold of converging $\mathbf{W}^{(n)}$ has Lebesgue measure 0.[3]



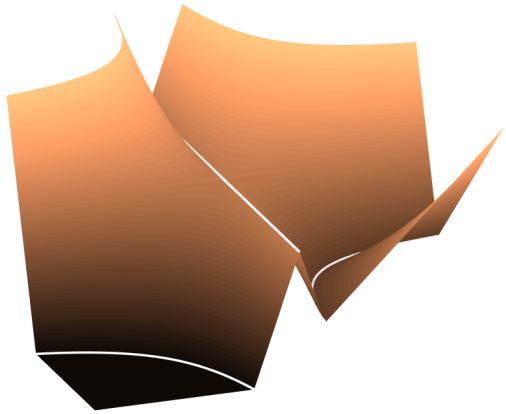


Chi Jin and M. Jordan, How to Escape Saddle Points Efficiently: Saddle points can slow down optimization; perturbing the GD can help.

[3] B. Recht, Saddles Again

Real-life ANNs

- No smoothness, in general (e.g. with ReLU): local minima of $L(\mathbf{W})$ are non-differentiable



Th. Laurent, J. von Brecht, The Multilinear Structure of ReLU Networks, arXiv:1712.10132

- Large size of the network and its structure are important

Empirical observations of real-life ANNs

From A. Choromanska et al., The Loss Surfaces of Multilayer Networks, arXiv:1412.0233:

- Large networks train well despite their size. Optimization can terminate at different local minima, but they seem to be equivalent and yield similar performance on a test set.
- The probability of finding a "bad" (high value) local minimum is non-zero for small-size networks and decreases quickly with network size.
- Struggling to find the global minimum on the training set (as opposed to one of the many good local ones) is not useful in practice and may lead to overfitting.

Conceptual pictures of the loss surface (conjectured)

From M. Baity-Jesi et al., Comparing Dynamics: Deep Neural Networks versus Glassy Systems: two alternatives

1. The loss landscape is very rough, has many isolated local minima, but GD tends to find good minima having low loss.
2. The loss function is highly nonlinear, but has few local minima, and the minima are connected. (Example: $L(w_1, w_2) = (w_2 - w_1^2)^2 + \epsilon w_1^2$.)

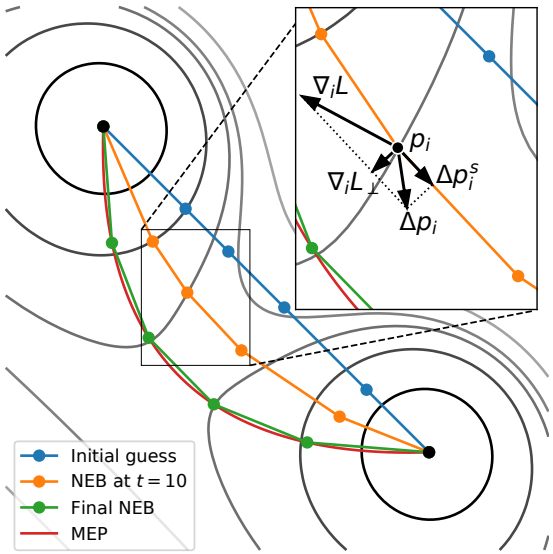
(Note: these conteptual pictures have only a limited value due to the "curse of dimensionality" in $\mathbb{R}^W$, lack of characterization of locality and depth of a local minimum, etc.)

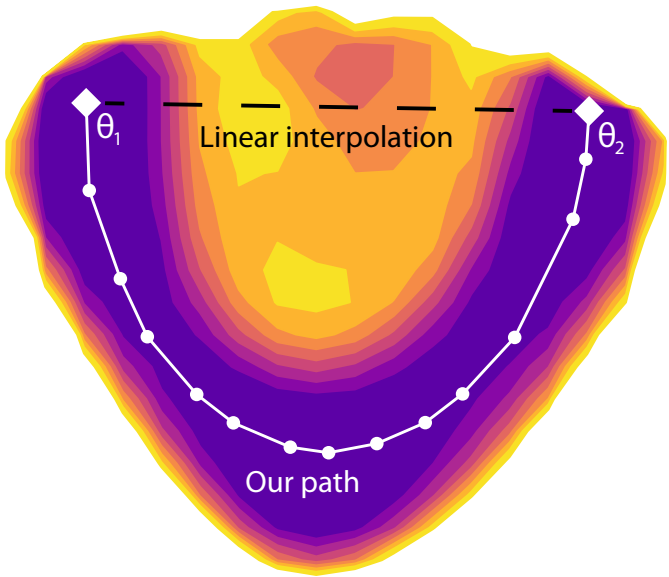
Some current research directions

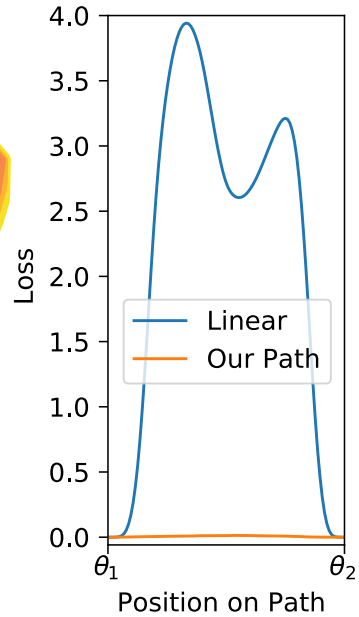
- Numerical studies of loss surface and gradient descent
- Direct analytic studies of simple (toy) scenarios:
  - Deep linear networks (no nonlinear activation)
  - Wide shallow networks with small training sets, pyramidal networks (no spurious local minima)
- Large-size limits:
  - Large-width limit: Gaussian approximation for signal propagation, connections to random matrices and spherical spin glasses, mean field theory, NTK (Neural Tangent Kernel) theory
  - Phenomenological: Stochastic PDE and Langevin dynamics

Large connected bottom in overparameterized networks[4]

- Two local minima are connected by a path, and then it is deformed to find a low loss trajectory
- The trivial path (straight segment): significantly growing loss
- The optimized path: approximately constant loss
- Implies that the loss function has a "connected bottom"







(ResNets and DenseNets on CIFAR10 and CIFAR100)

[4] F. Draxler et al., Essentially No Barriers in Neural Network Energy Landscape, arXiv:1803.00885

Many spurious minima in underparameterized networks[5]

Spurious local minimum $\mathbf{W}_0$: $\min_{\mathbf{W}} L(\mathbf{W}) < L(\mathbf{W}_0) < L(\mathbf{W}')$ for $\mathbf{W}'$ in a small neighborhood of $\mathbf{W}_0$

Theorem

Consider the optimization problem

$$\min_{\mathbf{w}_1,\ldots,\mathbf{w}_n \in \mathbb{R}^k} \mathbb{E}_{\mathbf{x}\sim\mathcal{N}(\mathbf{0},I)} \Big(\sum_{i=1}^{n} (\mathbf{w}_i^\top \mathbf{x})_+ - \sum_{i=1}^{k} (\mathbf{v}_i^\top \mathbf{x})_+ \Big)^2,$$

where $\mathbf{v}_1, \ldots, \mathbf{v}_k$ are orthogonal unit vectors in $\mathbb{R}^k$. Then for $n = k \in \{6, 7, \ldots, 20\}$ as well as $(k, n) \in \{(8, 9), (10, 11), \ldots, (19, 20)\}$, this objective function has spurious local minima.

Proof: computer-assisted

[5] I. Safran, O. Shamir, Spurious Local Minima are Common in Two-Layer ReLU Neural Networks, arXiv:1712.08968

Dependence on n, k

More spurious minima observed at larger k, but overparametrization (large n) appears to partly remove them.

Table: Spurious local minima found for $n = k$

k	n	% of runs converging to local minima	Average minimal eigenvalue	Average objective value
6	6	0.3%	0.0047	0.025
7	7	5.5%	0.014	0.023
8	8	12.6%	0.021	0.021
9	9	21.8%	0.027	0.02
10	10	34.6%	0.03	0.022
11	11	45.5%	0.034	0.022
12	12	58.5%	0.035	0.021
13	13	73%	0.037	0.022
14	14	73.6%	0.038	0.023
15	15	80.3%	0.038	0.024
16	16	85.1%	0.038	0.027
17	17	89.7%	0.039	0.027
18	18	90%	0.039	0.029
19	19	93.4%	0.038	0.031
20	20	94%	0.038	0.033

Table: Spurious local minima found for $n \neq k$

k	n	% of runs converging to local minima	Average minimal eigenvalue	Average objective value
8	9	0.1%	0.0059	0.021
10	11	0.1%	0.0057	0.018
11	12	0.1%	0.0056	0.017
12	13	0.3%	0.0054	0.016
13	14	1.5%	0.0015	0.038
14	15	5.5%	0.002	0.033
15	16	10.1%	0.004	0.032
16	17	18%	0.0055	0.031
17	18	20.9%	0.007	0.031
18	19	36.9%	0.0064	0.028
19	20	49.1%	0.0077	0.027

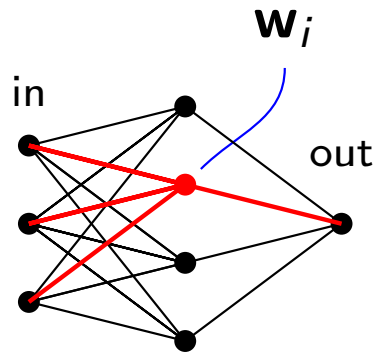
Connectedness of local minima in an overparameterized net: a heuristic explanation[6]

A network with 1 hidden layer:

$$\widehat{\mathbf{y}}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N}\phi(\mathbf{x}, \mathbf{w}_i),$$

where $\mathbf{w}_i = (\mathbf{l}_i, b_i, \mathbf{c}_i)$ and $\phi(\mathbf{x}, \mathbf{w}_i) = \mathbf{c}_i\sigma(\mathbf{w}_i \cdot \mathbf{l}_i + b_i)$

Ensemble of hidden neurons: the i'th neuron is described by $\mathbf{w}_i$



[6] arXiv:2008.00741

Reduction to distributions

"Change of variables" $\mathbf{W} = (\mathbf{w}_i)_{i=1}^N \to p$:

$$p(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^{N} \delta(\mathbf{w} - \mathbf{w}_i)$$

Then the prediction

$$\widehat{\mathbf{y}}(\mathbf{x}) = \int \phi(\mathbf{x}, \mathbf{w}) p(\mathbf{w}) d\mathbf{w}$$

is *linear* in p.

If the loss L is convex in $\widehat{\mathbf{y}}$, then it is also convex in p, and

$$L(p) \to \min_p$$

is a **convex** optimization problem. In particular, no spurious local minima!

Structure of local minima

Assume for simplicity that there is a unique minimizer $p^* = \operatorname{argmin} L(p)$

If the network is overparameterized, then the corresponding optimal weights $\mathbf{w}_i^*$ satisfy

$$p^* \approx \frac{1}{N} \sum_{i=1}^{N} \delta(\mathbf{w} - \mathbf{w}_i^*)$$

Global minima $\mathbf{W}^*$ are approximately the same up to permutations of $\mathbf{w}_i^*$

To connect one global minima $\mathbf{W}_A^*$ to another, $\mathbf{W}_B^*$, while keeping loss low: connect respective $\mathbf{w}_{A,i}^*$ to $\mathbf{w}_{B,i}^*$ while preserving their distribution

$$\mathbf{w}_i = \mathbf{w}_i(t), \quad \mathbf{w}_i(0) = \mathbf{w}_{A,i}^*, \quad \mathbf{w}_i(1) = \mathbf{w}_{B,i}^*$$

$$\frac{d}{dt}\left(\frac{1}{N} \sum_{i=1}^{N} \delta(\mathbf{w}(t) - \mathbf{w}_i^*)\right) \approx 0$$

Main example: connecting normal distributions

Let $X, Y \sim \mathcal{N}(0, \Sigma_0)$ be independent random vectors

Linear connection:

$$X(t) = (1-t)X + tY,$$

"squeezes" the distribution:

$$\Sigma_{X(t)} = (t^2 + (1-t)^2)\Sigma_0 \neq \Sigma_0, \quad X(t) \not\sim \mathcal{N}(0, \Sigma_0)$$

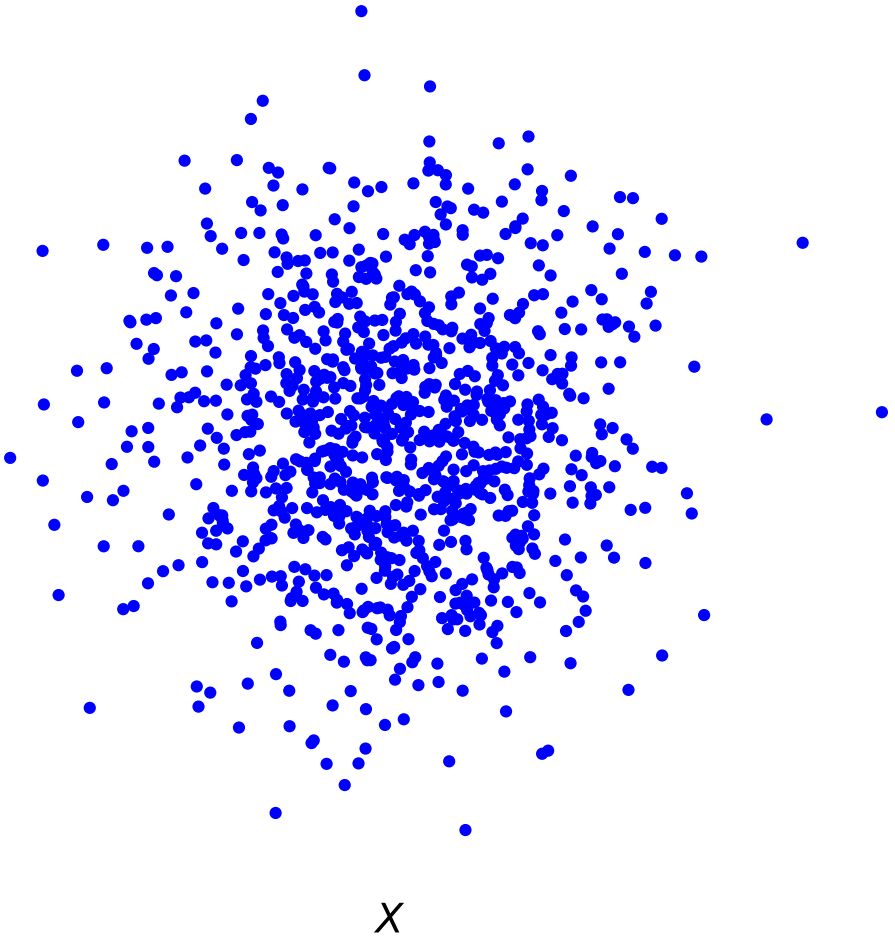
Arc connection:

$$X(t) = \cos(\pi t/2)X + \sin(\pi t/2)Y,$$

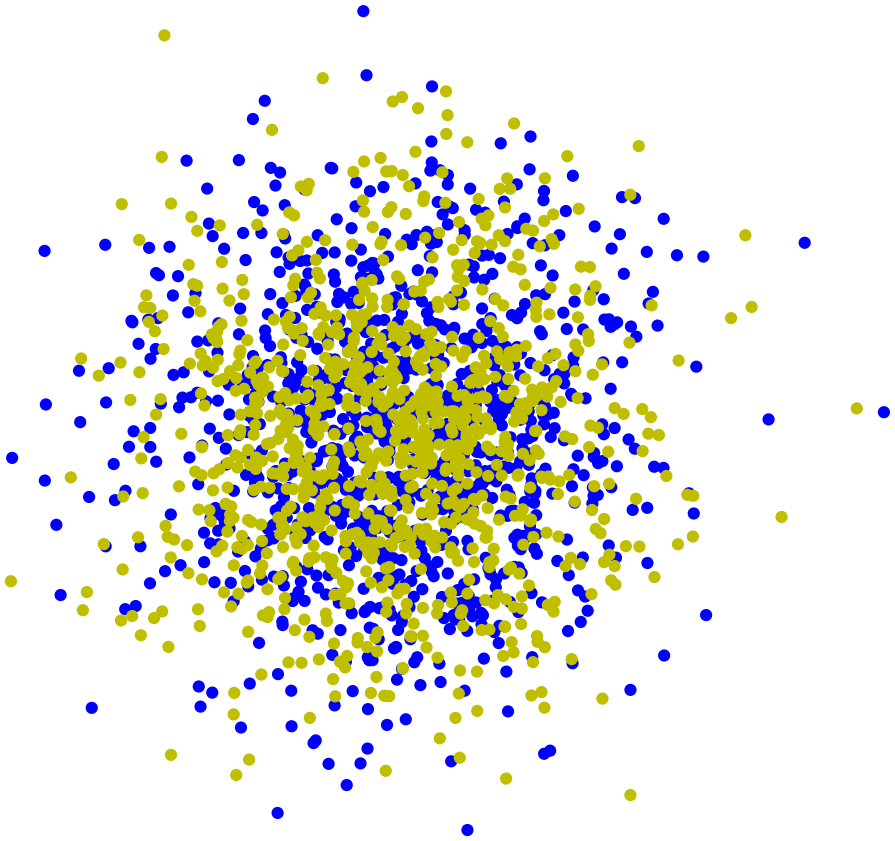
preserves the distribution:

$$\Sigma_{X(t)} = \Sigma_0, \quad X(t) \sim \mathcal{N}(0, \Sigma_0)$$

The sample $X \sim \mathcal{N}(0, \mathbf{1}_{2\times 2})$

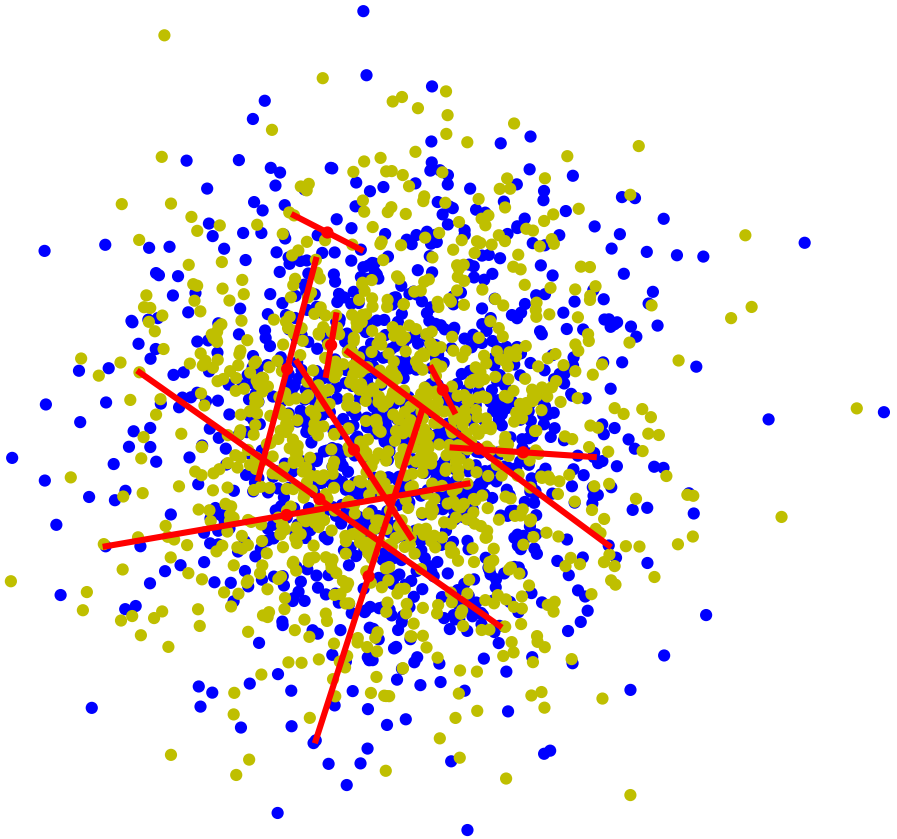


The samples $X, Y \sim \mathcal{N}(0, \mathbf{1}_{2\times 2})$

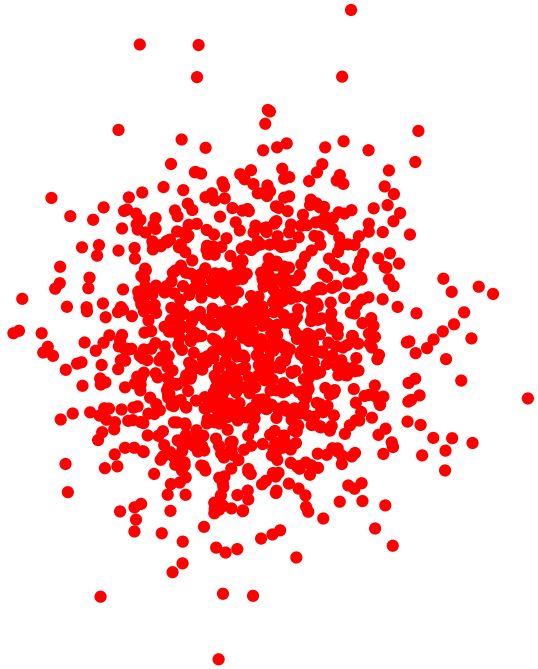


X, Y

Linear connection: squeezed distribution

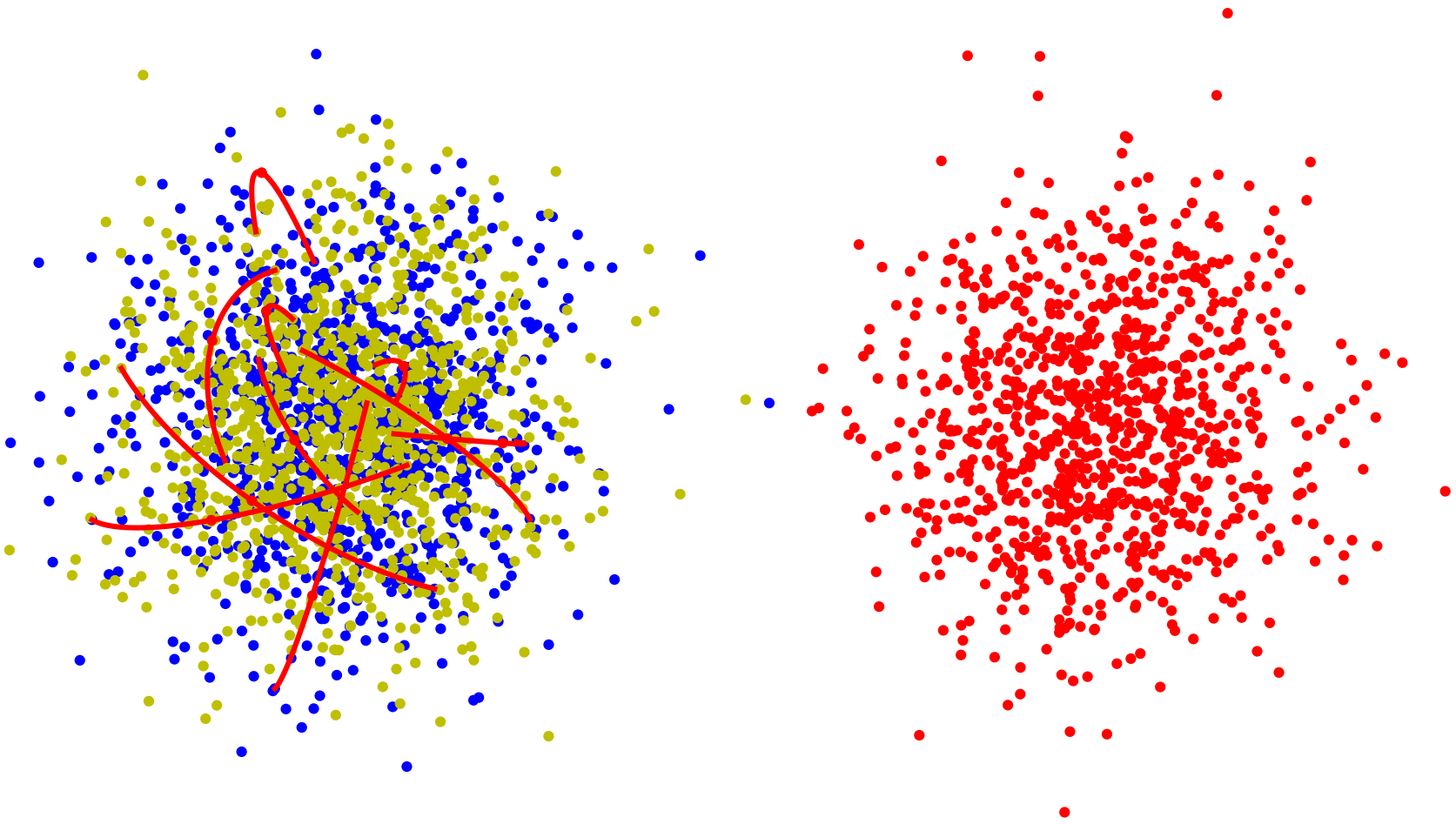


X, Y



$0.5X + 0.5Y$

Arc connection: preserved distribution



Lowest accuracy (%) of connection for networks with a single hidden layer

	MNIST		CIFAR10	
Methods	train	test	train	test
Linear	96.54 ± 0.40	95.87 ± 0.40	32.09 ± 1.33	39.34 ± 1.52
Arc	97.89 ± 0.11	97.03 ± 0.14	49.97 ± 0.86	41.34 ± 1.39

Linear networks

Linear networks: no nonlinear activation ($\sigma(x) = x$)

For simplicity also drop the constant terms in neurons

Then the model becomes:

$$\mathbf{y} = \widetilde{f}(\mathbf{x}) = W_K W_{K-1} \cdots W_1 \mathbf{x}, \qquad W_k \in \mathbb{R}^{d_k \times d_{k-1}}$$

- Parameters: $\mathbf{W} = (W_1, \ldots, W_k)$
- The k'th layer has d_k neurons; the input layer has d_0 neurons

$\widetilde{f}$ can model only linear maps $\mathbf{x} \mapsto \mathbf{y}$!

Linear networks with quadratic loss

Quadratic loss:

$$L(\mathbf{W}) = \frac{1}{2}\int \|f(\mathbf{x}) - W_K W_{K-1}\cdots W_1\mathbf{x}\|^2 d\mu(\mathbf{x})$$

Given two maps f, g, consider the scalar product

$$\langle f, g\rangle = \int \langle f(\mathbf{x}), g(\mathbf{x})\rangle d\mu(\mathbf{x})$$

Then $L(\mathbf{W}) = \|f - \widetilde{f}_{\mathbf{W}}\|^2$, where $\widetilde{f}_{\mathbf{W}}$ belongs to the $d_0 d_K$-dimensional space F_{lin} of linear maps. By orthogonally projecting f to F_{lin}, we can assume that f is also linear:

$$f(\mathbf{x}) = R\mathbf{x}$$

Linear networks with quadratic loss

Let $\Delta = \Delta(\mathbf{W}) = W_K \cdots W_1 - R$

$$L(\mathbf{W}) = \frac{1}{2}\int \|\Delta \mathbf{x}\|^2 d\mu(\mathbf{x}) = \frac{1}{2}\int \langle \mathbf{x}, \Delta^*\Delta \mathbf{x}\rangle d\mu(\mathbf{x})$$
$$= \frac{1}{2}\int \mathrm{tr}(\Delta^*\Delta |\mathbf{x}\rangle\langle \mathbf{x}|) d\mu(\mathbf{x}) = \frac{1}{2}\,\mathrm{tr}(\Delta^*\Delta\Sigma),$$

where Σ is the covariance matrix of measure μ :

$$\Sigma = \Sigma^* = \int |\mathbf{x}\rangle\langle \mathbf{x}| d\mu(\mathbf{x})$$

("Bra"-"ket" notation: $|\mathbf{u}\rangle\langle \mathbf{v}| : \mathbf{z} \mapsto \langle \mathbf{v}, \mathbf{z}\rangle \mathbf{u}$)

Constant width networks ($d_k \equiv d$)

Expressiveness of stacked "nearly identical" linear layers[7]:

Proposition

Any $A \in \mathbb{R}^{d\times d}$ *with* $\det A > 0$ *can be represented as* $(1+V_K)(1+V_{K-1})\cdots(1+V_1)$, *where* $\max_{k=1,\dots,K} \|V_k\| = O(1/K)$

Exercise: Why $\det A > 0$?

[7] M. Hardt, T. Ma, Identity Matters in Deep Learning, arXiv:1611.04231

Proof

Write $A = BO$, where $B = B^*$ is positive definite and O orthogonal:

$$B = (AA^*)^{1/2}, \quad O = B^{-1}A$$

Canonical forms[8]:

$$B = O_1 \operatorname{diag}(\lambda_1, \ldots, \lambda_d) O_1^*$$
$$O = O_2 \operatorname{diag}\left(\begin{pmatrix} \cos\phi_1 & \sin\phi_1 \\ -\sin\phi_1 & \cos\phi_1 \end{pmatrix}, \ldots, \begin{pmatrix} \cos\phi_s & \sin\phi_s \\ -\sin\phi_s & \cos\phi_s \end{pmatrix}, 1, \ldots, 1 \right) O_2^*$$

Then $B = (B^{1/K})^K, \quad O = (O^{1/K})^K$, where

$$B^{1/K} = O_1 \operatorname{diag}(\sqrt[K]{\lambda_1}, \ldots, \sqrt[K]{\lambda_d}) O_1^* = \mathbf{1} + O(\tfrac{1}{K})$$
$$O^{1/K} = O_2 \operatorname{diag}\left(\begin{pmatrix} \cos\frac{\phi_1}{K} & \sin\frac{\phi_1}{K} \\ -\sin\frac{\phi_1}{K} & \cos\frac{\phi_1}{K} \end{pmatrix}, \ldots, \begin{pmatrix} \cos\frac{\phi_s}{K} & \sin\frac{\phi_s}{K} \\ -\sin\frac{\phi_s}{K} & \cos\frac{\phi_s}{K} \end{pmatrix}, 1, \ldots, 1 \right) O_2^*$$
$$= \mathbf{1} + O(\tfrac{1}{K})$$

[8] **Exercise** (Euler's rotation theorem): any $O \in SO(3)$ is a rotation about some axis by some degree ϕ.

Computation of $\nabla_{\mathbf{W}} L(\mathbf{W})$

Think of ∇_{W_k} as a matrix $(\nabla_{(W_k)_{m,n}})_{m,n}$

$$
\begin{aligned}
\nabla_{W_k} L(\mathbf{W}) &= \nabla_{W_k} \frac{1}{2} \operatorname{tr}(\Delta(\mathbf{W})^* \Delta(\mathbf{W}) \Sigma) \\
&= \operatorname{tr}(\Delta(\mathbf{W})^* \nabla_{W_k}(\Delta(\mathbf{W})) \Sigma) \\
&= \operatorname{tr}(\nabla_{W_k}(\Delta(\mathbf{W})) \Sigma \Delta(\mathbf{W})^*) \\
&= \operatorname{tr}(\nabla_{W_k}(W_K \cdots W_{k+1} W_k W_{k-1} \cdots W_1) \Sigma \Delta(\mathbf{W})^*) \\
&= \left(\operatorname{tr}\left((W_K \cdots W_{k+1} |\mathbf{e}_m\rangle \langle \mathbf{e}_n| W_{k-1} \cdots W_1) \Sigma \Delta(\mathbf{W})^* \right) \right)_{m,n} \\
&= \left(\langle \mathbf{e}_n| W_{k-1} \cdots W_1 \Sigma \Delta(\mathbf{W})^* W_K \cdots W_{k+1} |\mathbf{e}_m\rangle \right)_{m,n} \\
&= W_{k+1}^* \cdots W_K^* \Delta(\mathbf{W}) \Sigma W_1^* \cdots W_{k-1}^*
\end{aligned}
$$

Local nondegenerate minima are global

Theorem (Hardt, Ma)

Suppose that Σ is strictly positive definite, $\mathbf{W} = (W_1, \ldots, W_K)$ is a local minimum of $L(\mathbf{W})$ and all matrices W_k are nondegenerate. Then $\mathbf{W}$ is a global minimum.

Proof. By nondegeneracy of Σ and W_k,

$$0 = \nabla_{W_k} L(\mathbf{W}) = W_{k+1}^* \cdots W_K^* \Delta \Sigma W_1^* \cdots W_{k-1}^*$$

implies $\Delta = 0$. □

Exercise: If $W_k = W_n = 0$ for some $k \neq n$, then $\mathbf{W}$ is a stationary point.

Solvable GD: dynamic in the subalgebra generated by R

Key idea: ensure that matrices produced by the GD algorithm are diagonalized by a common basis[9]

Assume $R = R^*, \Sigma = \mathbf{1}$, and the starting values $W_k^{(1)} = \mathbf{1}$ for all k

$$
\begin{aligned}
W_k^{(2)} &= W_k^{(1)} - \alpha \nabla_{W^k} L(\mathbf{W}^{(1)}) = \mathbf{1} - \alpha(\mathbf{1} - R) \\
W_k^{(3)} &= W_k^{(2)} - \alpha \nabla_{W^k} L(\mathbf{W}^{(2)}) = \mathbf{1} - \alpha(\mathbf{1} - R) \\
&\quad - \alpha(\mathbf{1} - \alpha(\mathbf{1} - R))^* \cdots (\mathbf{1} - \alpha(\mathbf{1} - R))^* \\
&\quad \times ((\mathbf{1} - \alpha(\mathbf{1} - R)) \cdots (\mathbf{1} - \alpha(\mathbf{1} - R)) - R) \\
&\quad \times (\mathbf{1} - \alpha(\mathbf{1} - R))^* \cdots (\mathbf{1} - \alpha(\mathbf{1} - R))^* \\
&= p_{k,3}(R) \\
&\dots \\
W_k^{(n)} &= p_{k,n}(R)
\end{aligned}
$$

with some polynomials $p_{k,n}$.

[9] P. Bartlett et al., Gradient descent with identity initialization efficiently learns positive definite linear transformations by deep residual networks

Diagonalization and 1D dynamics

If $R = O\,\mathrm{diag}(r_1, \ldots, r_d)O^*$, then

$$W_k^{(n)} = O\,\mathrm{diag}(p_{k,n}(r_1), \ldots, p_{k,n}(r_d))O^*$$

Optimization dynamics decouples into d independent 1D components

$d = 1$: $\mathbf{w} = (w_1, \ldots, w_K)$

$$\frac{dw_k}{dt} = w_{k+1} \cdots w_K (r - w_1 \cdots w_K) w_1 \cdots w_{k-1}; \qquad w_k(t = 0) = 1$$

$$\frac{dw_k^2}{dt} = 2q(r - q), \qquad q = w_1 \cdots w_K$$

Exercise:

- $r > 0$: $\mathbf{w}$ converges to a solution of $r = w_1 \cdots w_K$
- $r \leq 0$: $\mathbf{w}$ can get stuck at a point with $w_1 \cdots w_K = 0$

Conclusion and generalizations

Theorem (Bartlett et al.)

Suppose that $\Sigma = \mathbf{1}, R = R^*$ *and is positive definite. Then the gradient descent starting from* $W_k^{(1)} \equiv \mathbf{1}$ *converges to a global minimum of* L.

Exercise[10]**:** A generalization: assume that there are orthogonal operators $O_k \in SO(d_k)$ diagonalizing the initial weight matrices $W_k^{(1)}$ and the operators R and Σ (in the sense that $O_k W_k^{(1)} O_{k-1}^*$ is diagonal and similarly for R, Σ). Then GD decouples into independent 1D components.

An open question (?): Is there a complete asymptotic description of GD in linear networks for generic initial conditions?[11]

[10] A. Saxe et al., Exact solutions to the nonlinear dynamics of learning in deep linear neural networks, arXiv:1312.6120

[11] See also a perturbative study in Arora et al. (2018), arXiv:1810.02281

General layer widths and loss functions

Th. Laurent, J. von Brecht, Deep Linear Networks with Arbitrary Loss: All Local Minima Are Global, arXiv:1712.01473

Assumptions:

1. The loss function $\widetilde{\mathbf{y}} \mapsto l(\mathbf{y}, \widetilde{\mathbf{y}})$ is convex and differentiable.
2. The thinnest layer is either the input layer or the output layer (i.e., $\min(d_1, \ldots, d_{K-1}) \geq \min(d_0, d_K)$).

Theorem (1)

Under these assumptions, the linear network has no spurious (sub-optimal) local minima, i.e. any local minimum is global.

Theorem (2)

There exists a convex, Lipschitz, but non-differentiable loss function $\widetilde{\mathbf{y}} \mapsto l(\mathbf{y}, \widetilde{\mathbf{y}})$ *with which the network has sub-optimal local minima.*

Sketch of proof of Theorem 1

Two problems:

$$\text{(P1)} \begin{cases} \text{Minimize} \quad g(A) \\ \text{over all } A \text{ in } \mathbb{R}^{d_L \times d_0} \end{cases}$$

$$\text{(P2)} \begin{cases} \text{Minimize} \quad g(W_K W_{K-1} \cdots W_1) \\ \text{over all } K\text{-tuples } (W_1, \ldots, W_K) \\ \text{in } \mathbb{R}^{d_1 \times d_0} \times \cdots \times \mathbb{R}^{d_K \times d_{K-1}} \end{cases}$$

Lemma

Under Assumption 2: for any $g \in C^1$, if $(\widehat{W}_1, \ldots, \widehat{W}_K)$ is a **local minimizer** *of (P2), then $\widehat{W}_K \cdots \widehat{W}_1$ is a* **stationary point** *of (P1).*

Exercise: the lemma implies Theorem 1 by convexity of l.

Note: The map $W_1, \ldots, W_K \mapsto W_K \cdots W_1$ is not generally locally surjective, hence lemma only claims stationarity, not local minimality.

Sketch of proof of the Lemma

Let $G(W_1, \ldots, W_K) = g(W_1 \cdots W_K)$. Stationarity of G at $\widehat{\mathbf{W}}$:

$$0 = \nabla_{W_k} G(\widehat{\mathbf{W}}) = \widehat{W}_{k+1}^* \cdots \widehat{W}_K^* \nabla g(\widehat{W}_K \cdots \widehat{W}_1) \widehat{W}_1^* \cdots \widehat{W}_{k-1}^*, \quad \forall k$$

Easy case: $\ker(\widehat{W}_{K-1} \cdots \widehat{W}_1) = \{0\}$. Then from stationarity of G with $k = K$, by transposing:

$$0 = \widehat{W}_{K-1} \cdots \widehat{W}_1 \big(\nabla g(\widehat{W}_K \cdots \widehat{W}_1)\big)^*$$

Then $\nabla g(\widehat{W}_K \cdots \widehat{W}_1) = 0$, Q.E.D.

Harder case: $\ker(\widehat{W}_{K-1} \cdots \widehat{W}_1) \neq \{0\}$. Main idea: construct a family of local perturbations $\widetilde{W}_1, \ldots, \widetilde{W}_K$ such that

$$\widetilde{W}_K \cdots \widetilde{W}_1 = \widehat{W}_K \cdots \widehat{W}_1 \quad \text{and} \quad \|\widetilde{W}_k - \widehat{W}_k\| < \epsilon \quad \forall k \tag{1}$$

Then $(\widetilde{W}_1, \ldots, \widetilde{W}_K)$ is also a local minimum and hence a stationary point of G.

Construction of local perturbations

$$\ker(\widehat{W}_1) \subset \ker(\widehat{W}_2\widehat{W}_1) \subset \ldots \subset \ker(\widehat{W}_{K-1}\cdots\widehat{W}_1) \neq \{0\}$$

Assume for simplicity that $\dim\ker(\widehat{W}_k\cdots\widehat{W}_1) > 0$ for all k

Suppose that $d_k \geq d_0$. Then

$$\text{co-dim}\,\text{Ran}(\widehat{W}_k\cdots\widehat{W}_1) \geq \dim\ker(\widehat{W}_k\cdots\widehat{W}_1) > 0,$$

so there is $0 \neq \mathbf{u}_k \in \mathbb{R}^{d_k} \ominus \text{Ran}(\widehat{W}_k\cdots\widehat{W}_1)$. Let

$$\widetilde{W}_k = \widehat{W}_k + \delta_k|\mathbf{w}_k\rangle\langle\mathbf{u}_{k-1}|$$

with any $\mathbf{w}_k$ and small δ_k. This fulfills conditions (1).

End of proof

From stationarity, with $k = 1$:

$$0 = \left(\nabla g(\widehat{W}_K \cdots \widehat{W}_1)\right)^* \widetilde{W}_K \cdots \widetilde{W}_2$$

Since we can add to $\widetilde{W}_2$ an arbitrary term $\delta_k |\mathbf{w}_2\rangle\langle \mathbf{u}_1|$:

$$0 = \left(\nabla g(\widehat{W}_K \cdots \widehat{W}_1)\right)^* \widetilde{W}_K \cdots \widetilde{W}_3 |\mathbf{w}_2\rangle\langle \mathbf{u}_1|$$

Since $\mathbf{w}_2$ was arbitrary:

$$0 = \left(\nabla g(\widehat{W}_K \cdots \widehat{W}_1)\right)^* \widetilde{W}_K \cdots \widetilde{W}_3$$

Removing in the same way $\widetilde{W}_3, \widetilde{W}_4, \ldots$:

$$0 = \left(\nabla g(\widehat{W}_K \cdots \widehat{W}_1)\right)^*,$$

hence $\nabla g(\widehat{W}_K \cdots \widehat{W}_1) = 0$. □

Wide nonlinear networks: finite sample expressiveness

Exercise[12]**:** There exists a ReLU neural network with two hidden layers and $2n + d$ weights that can represent any function on a sample of size n in d dimensions.

[12] Ch. Zhang et al., Understanding deep learning requires rethinking generalization, arXiv:1611.03530

Getting rid of bias terms

Standard layer representation: $\mathbf{z}_k = \sigma(W_k \mathbf{z}_{k-1} + \mathbf{h})$, with σ evaluated component-wise

We can get rid of bias terms by adding an extra neuron with output 1:

$$\widetilde{\mathbf{z}}_{k-1} = (\mathbf{z}_{k-1}, 1)$$

$$\mathbf{z}_k = \sigma(\widetilde{W}_k \widetilde{\mathbf{z}}_{k-1})$$

The "extended" network inputs:

$$\widetilde{\mathbf{x}} = (\mathbf{x}, 1)$$

In the wollowing assume W.L.O.G that the network layers are without bias terms:

$$\mathbf{z}_k = \sigma(W_k \mathbf{z}_{k-1})$$

and the network parameters are $\mathbf{W} = (W_1, \ldots, W_K)$

Gradient of nonlinear networks

- $\mathbf{y}_k := (y_{k,1}, \ldots, y_{k,d_k})$: pre-activation outputs in the k'th layer (d_k neurons)
- $F_k := (\sigma(y_{k,1}), \ldots, \sigma(y_{k,d_k}))$: post-activation outputs
- $F_k' := \mathrm{diag}(\sigma'(y_{k,1}), \ldots, \sigma'(y_{k,d_k}))$: diagonal matrix of respective derivatives of σ

Then, for the quadratic loss $L(\mathbf{W}) = \frac{1}{2}\int |f(\mathbf{x}) - \widetilde{f}(\mathbf{x}, \mathbf{W})|^2 d\mu(\mathbf{x})$:

$$\begin{aligned}
\left(\nabla_{W_k} L(\mathbf{W})\right)_{ij} = \int d\mu(\mathbf{x}) \Big\langle \widetilde{f}(\mathbf{x}, \mathbf{W}) - f(\mathbf{x}), \\
W_K \cdot F_{K-1}'(\mathbf{x}) \cdot W_{K-1} \cdot F_{K-2}'(\mathbf{x}) \ldots F_k'(\mathbf{x}) \cdot |\mathbf{e}_i\rangle\langle \mathbf{e}_j| \cdot F_{k-1}(\mathbf{x}) \Big\rangle \\
= \int d\mu(\mathbf{x}) \Big[(F_{k-1}(\mathbf{x}))_j \times \\
\times \left(F_k'(\mathbf{x}) \cdot W_{k+1}^* \ldots F_{K-1}'(\mathbf{x}) \cdot W_K^* \cdot (\widetilde{f}(\mathbf{x}, \mathbf{W}) - f(\mathbf{x}))\right)_i \Big]
\end{aligned}$$

Exercise: Check that this agrees with the formula for linear networks.

Pyramidal networks[13]

Pyramidal networks: layer widths do not increase (i.e., $d_{k+1} \leq d_k$)

Suppose that the activation function is differentiable, and $\sigma'(x) > 0$.

Consider training with the quadratic loss and on a *finite* training set $\{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$

Theorem

Let $\mathbf{W} = (W_1, \ldots, W_K)$ *be a stationary point of the gradient descent. Suppose that:*

1. *The extended input vectors* $\{\widetilde{\mathbf{x}}_n = (\mathbf{x}_n, 1)\}_{n=1}^N$ *in the training set are linearly independent (i.e.,* $\{\mathbf{x}_n\}_{n=1}^N$ *are affinely independent);*
2. *The weight matrices* W_k *have full ranks (i.e.,* $\text{rank}(W_k) = d_k$*).*

Then $\mathbf{W}$ *is a global minimum:* $L(\mathbf{W}) = 0$.

[13] M. Gori and A. Tesi, On the problem of local minima in backpropagation, IEEE Transactions on Pattern Analysis and Machine Intelligence, 14:76-86, 1992

Proof

Applying the stationarity condition with $d\mu(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N}\delta(\widetilde{\mathbf{x}} - \widetilde{\mathbf{x}}^{(n)})$ and layer $k = 1$: for all i, j

$$
\begin{aligned}
0 &= \left(\nabla_{W_1} L(\mathbf{W})\right)_{ij} \\
&= \sum_{n=1}^{N} \widetilde{x}_j^{(n)} \times \left(F_1'(\widetilde{\mathbf{x}}^{(n)}) \cdot W_2^* \ldots F_{K-1}'(\widetilde{\mathbf{x}}^{(n)}) \cdot W_K^* \cdot (\widetilde{f}(\widetilde{\mathbf{x}}^{(n)}, \mathbf{W}) - f(\widetilde{\mathbf{x}}^{(n)}))\right)_i \\
&= \sum_{n=1}^{N} \widetilde{x}_j^{(n)} \times Z_i^{(n)}
\end{aligned}
$$

We want to prove that $\widetilde{f}(\widetilde{\mathbf{x}}^{(n)}, \mathbf{W}) - f(\widetilde{\mathbf{x}}^{(n)}) = 0$ for all n.

Suppose that's not so. Since W_k have full ranges, $\ker W_k^* = 0$. Also, all matrices $F_k'(\mathbf{x})$ are nondegenerate, since $\sigma'(t) > 0$. Therefore $Z_i^{(n)} \not\equiv 0$. But then $\mathbf{x}^{(n)}$ are not linearly independent. Contradicton. $\square$

A generalization

The theorem is also applicable if the first s layers of the network are considered as a fixed "feature map" $\widetilde{\mathbf{x}} \mapsto \phi(\widetilde{\mathbf{x}}) \in \mathbb{R}^m$

The feature vectors $\phi(\widetilde{\mathbf{x}}_n)$ can be linearly independent even if the true inputs $\widetilde{\mathbf{x}}_n$ are not

The theorem can be applied to the subnetwork following the first s layers, with $\phi(\widetilde{\mathbf{x}}_n)$ as inputs

Measure concentration results

Let ξ_k be i.i.d. random variables with $\mathbb{E}\xi_k = 0$, and let $\eta_n = \sum_{k=1}^{n} \xi_k$

- Law of Large Numbers: $\frac{\eta_n}{n} \to 0$
- Central Limit Theorem: $\frac{\eta_n}{\sqrt{n}} \to \mathcal{N}(0, \mathbb{E}\xi^2)$ in distribution
- Large Deviation Theory[14]: Given $x > 0$, $\mathbb{P}(\frac{\eta_n}{n} > x) \simeq e^{-nI(x)}$ with a *rate function* $I(x)$ depending on the distribution of ξ

[14] A good introduction: J. T. Lewis, R. Russell, An Introduction to Large Deviations for Teletraffic Engineers

ANN in the large-width limit: independent weights[15]

Transformation from layer $k-1$ to k:

$$\mathbf{x}_k = \sigma(\mathbf{z}_k), \quad \mathbf{z}_k = W_k \mathbf{x}_{k-1} + \mathbf{h}_k, \qquad \mathbf{x}_k, \mathbf{z}_k \in \mathbb{R}^{d_k}$$

Assume $W_k, \mathbf{h}_k$ are random:

- $(W_k)_{ij}$ are i.i.d. with mean 0 and variance $\frac{\alpha_w^2}{d_{k-1}}$
- $(\mathbf{h}_k)_i$ are i.i.d. with mean 0 and variance α_h^2

By CLT: in the large-d_{k-1} limit, for a fixed $\mathbf{x}_{k-1}$, all components in $\mathbf{z}_k$ are i.i.d normal with mean 0 and variance $\alpha_w^2 \frac{|\mathbf{x}_{k-1}|^2}{d_{k-1}} + \alpha_h^2$:

$$\mathbb{E} z_{k,i}^2 = \mathbb{E}\left[\left(\sum_{j=1}^{d_{k-1}} (W_k)_{ij} x_{k-1,j} + h_{k,i}\right)^2\right] = \sum_{j=1}^{d_{k-1}} \mathbb{E}((W_k)_{ij}^2) x_{k-1,j}^2 + \mathbb{E}(h_{k,i}^2)$$

$$= \frac{\alpha_w^2}{d_{k-1}} \sum_{j=1}^{d_{k-1}} x_{k-1,j}^2 + \alpha_h^2 = \frac{\alpha_w^2 |\mathbf{x}_{k-1}|^2}{d_{k-1}} + \alpha_h^2$$

[15] B. Poole et al., arXiv:1606.05340

Reduction to 1D dynamics

For given input $\mathbf{x}_0$, define magnitude of the propagated signal in k'th layer:

$$q_k = \frac{1}{d_k} \sum_{i=1}^{d_k} z_{k,i}^2$$

Law of Large Numbers: q_k is approximately deterministic, $q_k \approx \mathbb{E} z_{k,i}^2$

Proposition (Poole et al.)

Evolution of q_k is given by

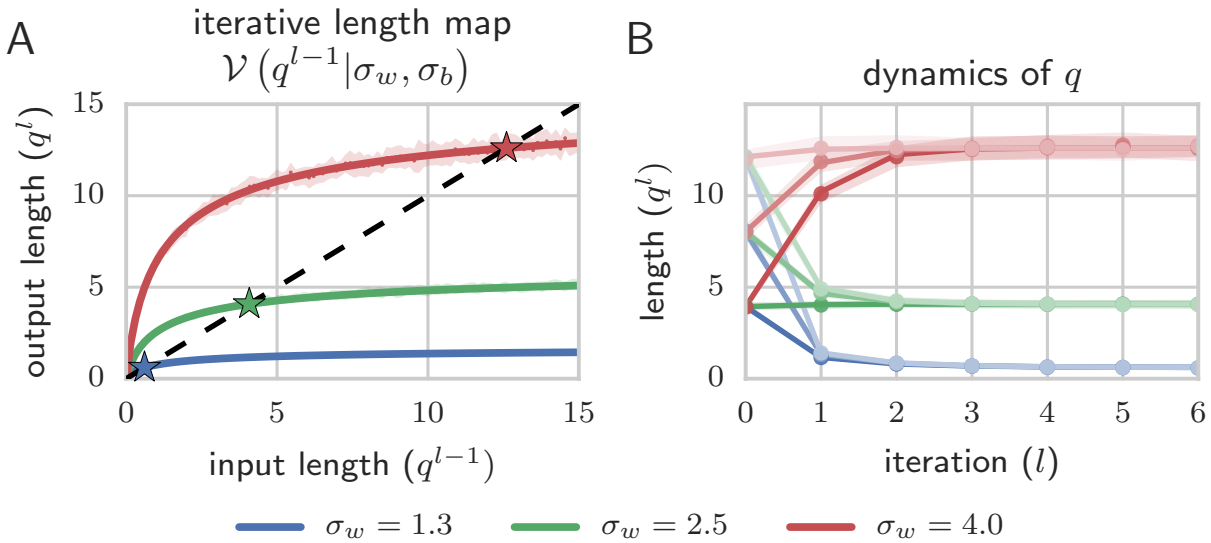
$$q_k = \nu(q_{k-1} | \alpha_w^2, \alpha_h^2) := \alpha_w^2 \int \sigma^2(\sqrt{q_{k-1}} s) \mathcal{D}s + \alpha_h^2, \quad \mathcal{D}s \equiv \frac{1}{\sqrt{2\pi}} e^{-s^2/2} ds$$

Proof: $q_k = \alpha_w^2 \frac{|\mathbf{x}_{k-1}|^2}{d_{k-1}} + \alpha_h^2 \approx \alpha_w^2 \mathbb{E}(x_{k-1,i}^2) + \alpha_h^2 = \alpha_w^2 \mathbb{E}(\sigma^2(z_{k-1,i})) + \alpha_h^2$, and $\mathbb{E}(\sigma^2(z_{k-1,i})) = \int \sigma^2(\sqrt{q_{k-1}} s) \mathcal{D}s$ since $z_{k-1,i} \sim \mathcal{N}(0, q_{k-1}^2)$ $\square$

Fixed points

Exercise: Find the map $q_k = \nu(q_{k-1})$ explicitely in the case of $\sigma = \mathrm{ReLU}$

In general, the map $q_k = \nu(q_{k-1})$ may have several stable or unstable fixed points:



Exercise:

- When is $q = 0$ a fixed point?
- When is a fixed point stable?
- Show that with $\sigma(z) = \tanh(z)$, the map ν has a stable fixed point $q_* \neq 0$.

Evolution of covariances

Let $\mathbf{x}_{0,1}$ and $\mathbf{x}_{0,2}$ be two input vectors. Define

$$q_{k,ab} = \frac{1}{d_k}\langle \mathbf{z}_{k,a}, \mathbf{z}_{k,b}\rangle = \frac{1}{d_k}\sum_{i=1}^{d_k}\langle z_{k,i,a}, z_{k,i,b}\rangle, \quad a, b \in \{1, 2\}$$

In particular, $q_{k,aa} = q_k$, and we already know its evolution. How does $q_{k,12}$ evolve?

We are looking for a map

$$q_{k,12} = C(c_{k-1,12}, q_{k-1,11}, q_{k-1,22}|\alpha_w, \alpha_h)$$

where $c_{k-1,12}$ is the *correlation coefficient*:

$$c_{k-1,12} = \frac{q_{k-1,12}}{\sqrt{q_{k-1,11}\, q_{k-1,22}}}$$

Evolution of covariances

$$
\begin{aligned}
q_{k,12} &= \frac{1}{d_k}\sum_{i=1}^{d_k}\langle \mathbf{z}_{k,1}, \mathbf{z}_{k,2}\rangle \approx \mathbb{E}(z_{k,1,i}z_{k,2,i}) \\
&= \mathbb{E}\left[\left(\sum_{j=1}^{d_{k-1}}(W_k)_{ij}x_{k-1,1,j} + h_{k,i}\right)\left(\sum_{m=1}^{d_{k-1}}(W_k)_{im}x_{k-1,2,m} + h_{k,i}\right)\right] \\
&= \sum_{j=1}^{d_{k-1}}\mathbb{E}((W_k)_{ij}^2)x_{k-1,1,j}x_{k-1,2,j} + \mathbb{E}(h_{k,i}^2) \\
&\approx \alpha_w^2\mathbb{E}(x_{k-1,1,j}x_{k-1,2,j}) + \alpha_h^2 \\
&= \alpha_w^2\mathbb{E}(\sigma(z_{k-1,1,j})\sigma(z_{k-1,2,j})) + \alpha_h^2 \\
&= \alpha_w^2\int \sigma(u_1)\sigma(u_2)\mathcal{D}\mu(u_1,u_2) + \alpha_h^2 \qquad (\mu \sim \mathcal{N}(0,(q_{k,ab}))) \\
&= \alpha_w^2\int \sigma(u_1)\sigma(u_2)\mathcal{D}s_1\mathcal{D}s_2 + \alpha_h^2 \qquad (\mathcal{D}s_a \sim \mathcal{N}(0,1)))
\end{aligned}
$$

where $u_1 = \sqrt{q_{k-1,11}}s_1$, $u_2 = \sqrt{q_{k-1,22}}(c_{k-1,12}s_1 + \sqrt{1-c_{k-1,12}^2}s_2)$

Evolution of correlations near a fixed point q^*

Assuming $|\mathbf{z}|^2$ is near the (stable) fixed point q_* of the map ν:

$$c_{k,12} = \widetilde{\mathcal{C}}_{q^*}(c_{k-1,12}) \equiv \frac{1}{q^*}\mathcal{C}(c_{k-1,12}, q^*, q^* | \sigma_w, \sigma_h)$$

Exercise: $c^* = 1$ is a fixed point of the map $\widetilde{\mathcal{C}}_{q^*}$

Is $c^* = 1$ stable or unstable?

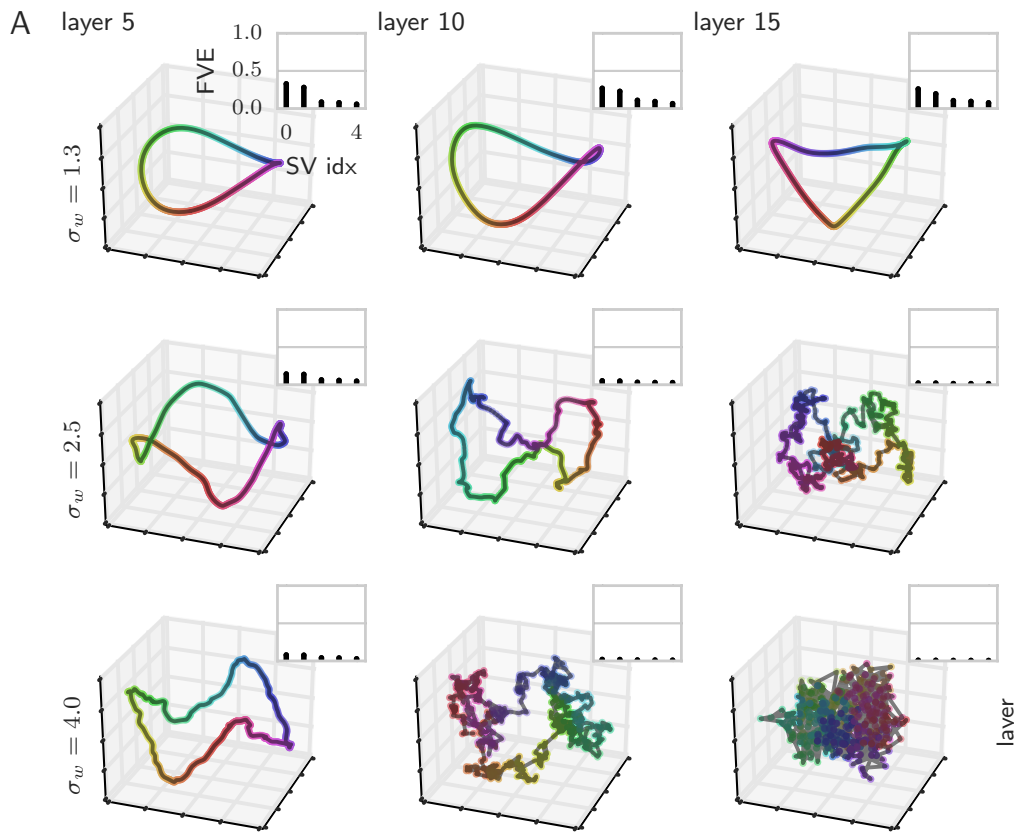
Exercise: Stability of $c^* = 1$ is determined by

$$\chi_1 = \frac{\partial c_{k,12}}{\partial c_{k-1,12}}\Big|_{c=1}$$

$$= \alpha_w^2 \int \left(\sigma'(\sqrt{q^*}s)\right)^2 \mathcal{D}s$$

(hint: $\int f(s) s \mathcal{D}s = \int f'(s) \mathcal{D}s$)

Stable vs. unstable evolution of a curve of input points:



Stable ("ordered") vs. unstable ("chaotic") phases

- Stable ($|\chi_1| < 1$):
  - different input vectors converge
  - initial details in the input get lost
  - deep network has low expressiveness
- Unstable ($|\chi_1| > 1$):
  - close input vectors tend to decorrelate
  - small changes in the input lead to major deviations
  - deep network has high expressiveness

In general, there are stable points $c^* \neq 1$; input vectors then tend to attain this fixed correlation.

Exercise: Find the explicit form of the correlation evolution map $\widetilde{\mathcal{C}}_{q^*}$ for the ReLU nonlinearity.

Depth of information propagation[16]

Exercise: Let $q_k = q^* + \delta q_k$. Then $\Delta q_{k+1} = a \Delta q_k + O(\Delta q_k^2)$, where

$$a = \chi_1 + \alpha_w^2 \int \sigma''(\sqrt{q^* s})\sigma(\sqrt{q^* s})\mathcal{D}s$$

Then $\Delta q_k \simeq e^{-k/\xi_q}$, where $\xi_q = -\frac{1}{\ln a}$ is the "characteristic depth scale of single input information propagation".

Exercise: Let c^* be a stable correlation fixed point and $c_{k,12} = c^* + \Delta c_k$. Then $\Delta c_{k+1} = b \Delta c_{k+1} + O(\Delta c_k^2)$, where

$$b = \alpha_w^2 \int \sigma'(u_1)\sigma'(u_2)\mathcal{D}s_1 \mathcal{D}s_2$$

with $u_1 = \sqrt{q_{k-1,11}} s_1$, $u_2 = \sqrt{q_{k-1,22}}(c_{k-1,12} s_1 + \sqrt{1 - c_{k-1,12}^2} s_2)$. If $c^* = 1$, then $b = \chi_1$.

Then $\Delta c_k \simeq e^{-k/\xi_c}$, where $\xi_c = -\frac{1}{\ln b}$ is the "characteristic depth scale of correlation information propagation".

[16] S. Schoenholz et al., Deep Information Propagation, arXiv:1611.01232

Error backpropagation

$$\begin{aligned}\left(\nabla_{W_k} L(\mathbf{W})\right)_{ij} &= \int d\mu(\mathbf{x}) \Big[(F_{k-1}(\mathbf{x}))_j \times \\ &\times \left(F_k'(\mathbf{x}) \cdot W_{k+1}^* \dots F_{K-1}'(\mathbf{x}) \cdot W_K^* \cdot (\widetilde{f}(\mathbf{x}, \mathbf{W}) - f(\mathbf{x}))\right)_i \Big] \\ &= \int d\mu(\mathbf{x}) \Big[(F_{k-1}(\mathbf{x}))_j \times \\ &\times \left(T_k(\mathbf{x}) \cdots T_{K-1}(\mathbf{x}) \cdot (\widetilde{f}(\mathbf{x}, \mathbf{W}) - f(\mathbf{x}))\right)_i \Big],\end{aligned}$$

where

$$T_k(\mathbf{x}) = F_k'(\mathbf{x}) \cdot W_{k+1}^*$$

In a deep network, the magnitude of $\nabla_{W_k} L(\mathbf{W})$ is mostly determined by $T_k(\mathbf{x}) \cdots T_{K-1}(\mathbf{x})$, since $\widetilde{f}(\mathbf{x}, \mathbf{W}) - f(\mathbf{x}) \asymp 1$ and $F_k'(\mathbf{x}) \asymp 1$

Error backpropagation

Let $\delta_k = (\delta_{k,i})_{i=1}^{d_k}$ be the backpropagated error:

$$\delta_{k,i} = (F_k'(\mathbf{x}) \cdot W_{k+1}^* \delta_{k+1})_i = \sigma'(z_{k,i}) \sum_{j=1}^{d_{k+1}} (W_{k+1})_{ji} \delta_{k+1,j}$$

Assume $z_{k,i}$ are independent of $\delta_{k+1,j}$, and recall that $(W_{k+1})_{ij}$ are i.i.d. with mean 0 and variance $\frac{\alpha_w^2}{d_k}$:

Assuming d_{k+1} is large:

$$\delta_{k,i}^2 \approx \mathbb{E}((\sigma'(z_{k,i}))^2) \sum_{j=1}^{d_{k+1}} \mathbb{E}((W_{k+1})_{ji}^2) \delta_{k+1,k}^2$$

$$\approx \int \left(\sigma'(\sqrt{q^*}s)\right)^2 \mathcal{D}s \cdot \frac{\alpha_w^2}{d_k} \sum_{j=1}^{d_{k+1}} \delta_{k+1,k}^2$$

$$= \chi_1 \frac{1}{d_k} \sum_{j=1}^{d_{k+1}} \delta_{k+1,k}^2$$

Vanishing and exploding gradients

Assume $d_k = d_{k+1}$, then

$$\|\delta_k\|^2 \approx \chi_1 \|\delta_{k+1}\|^2$$

- $\chi_1 < 1$: "vanishing gradient" ($\|\nabla_{W_k} L(\mathbf{W})\| \ll 1$ at small k)
- $\chi_1 > 1$: "exploding gradient" ($\|\nabla_{W_k} L(\mathbf{W})\| \gg 1$ at small k)

Optimal for learning: $\chi_1 \approx 1$ ("edge of chaos")