UC San Diego

Lecture 4:

Some Advanced Topics in Deep Learning (II)

Instructor: Hao Su

Jan 18, 2018

Agenda

- **Review of last lecture**
- Theories behind Machine Learning

Agenda

- Review of last lecture
- **Theories behind Machine Learning**

Deep Networks: Three Theory Questions

- *Approximation Theory*: When and why are deep networks better than shallow networks?
- *Optimization Theory*: What is the landscape of the empirical risk?
- *Learning Theory*: How can deep learning not overfit?

[Tomasi Poggio in STATS385]

Deep Networks: Three Theory Questions

- ***Approximation Theory*: When and why are deep networks better than shallow networks?**

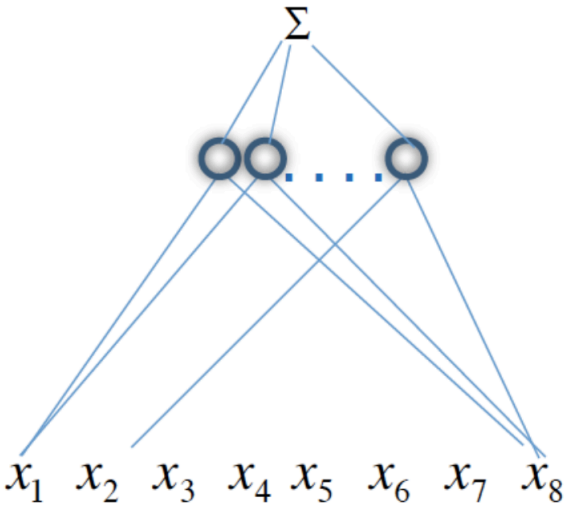
- *Optimization Theory*: What is the landscape of the empirical risk?

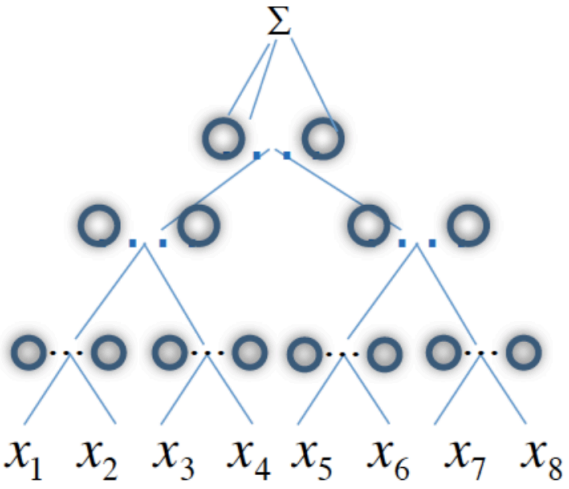
- *Learning Theory*: How can deep learning not overfit?

[Tomasi Poggio's lecture in STATS385]

Deep and Shallow Networks: Universality

Theorem *Shallow, one-hidden layer networks with a nonlinear $\phi(x)$ which is not a polynomial are universal. Arbitrarily deep networks with a nonlinear $\phi(x)$ (including polynomials) are universal.*





$$\phi(x) = \sum_{i=1}^{r} c_i \left| < w_i, x > + b_i \right|_+$$

Cybenko, Girosi, ….

[Tomasi Poggio's lecture in STATS385]

Classical learning theory and Kernel Machines (Regularization in RKHS)

$$\min_{f \in H} \left[\frac{1}{\ell} \sum_{i=1}^{\ell} V(f(x_i) - y_i) + \lambda \|f\|_K^2 \right]$$

implies

$$f(\mathbf{x}) = \sum_i^l \alpha_i K(\mathbf{x}, \mathbf{x}_i)$$

Equation includes splines, Radial Basis Functions and Support Vector Machines (depending on choice of V).

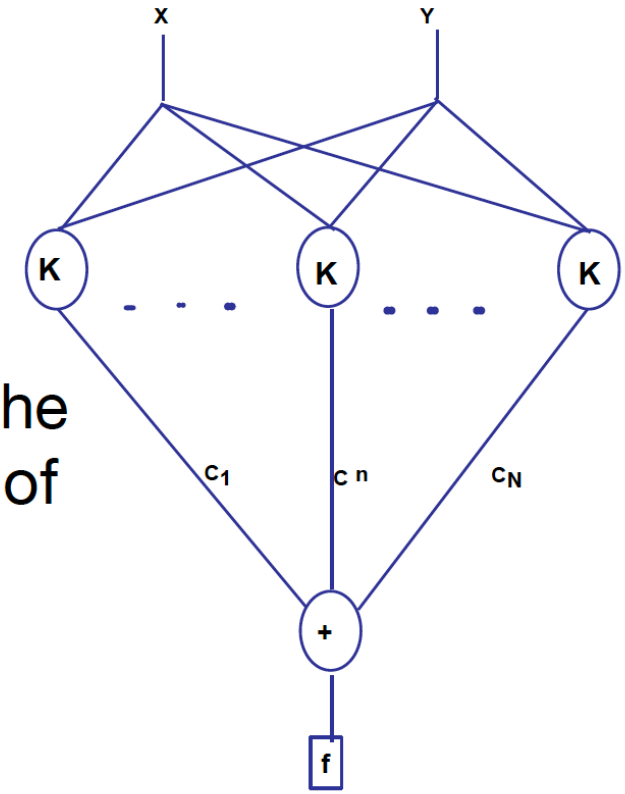
RKHS were explicitly introduced in learning theory by Girosi (1997), Vapnik (1998).
Moody and Darken (1989), and Broomhead and Lowe (1988) introduced RBF to learning theory. Poggio and Girosi (1989) introduced Tikhonov regularization in learning theory and worked (implicitly) with RKHS. RKHS were used earlier in approximation theory (eg Parzen, 1952-1970, Wahba, 1990). Mhaskar, Poggio, Liao, 2016

[Tomasi Poggio's lecture in STATS385]

Classical kernel machines are equivalent to shallow networks

Kernel machines…

$$f(\mathbf{x}) = \sum_i^l c_i K(\mathbf{x}, \mathbf{x}_i) + b$$



can be “written” as shallow networks: the value of K corresponds to the “activity” of the “unit” for the input and the correspond to “weights”

[Tomasi Poggio’s lecture in STATS385]

Curse of dimensionality

$$y = f(x_1, x_2, \ldots, x_8)$$

Curse of dimensionality

Both shallow and deep network can approximate a function of d variables equally well. The number of parameters in both cases depends exponentially on d as $O(\varepsilon^{-d})$.

Mhaskar, Poggio, Liao, 2016
[Tomasi Poggio's lecture in STATS385]

Generic functions

$$f(x_1, x_2, \ldots, x_8)$$

Compositional functions

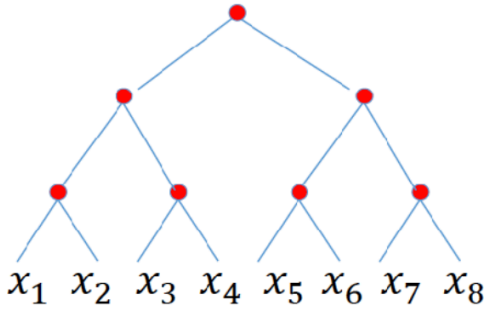
$$f(x_1, x_2, \ldots, x_8) = g_3(g_{21}(g_{11}(x_1, x_2), g_{12}(x_3, x_4))g_{22}(g_{11}(x_5, x_6), g_{12}(x_7, x_8)))$$

Mhaskar, Poggio, Liao, 2016

[Tomasi Poggio's lecture in STATS385]

Hierarchically local compositionality

$$f(x_1, x_2, ..., x_8) = g_3(g_{21}(g_{11}(x_1, x_2), g_{12}(x_3, x_4))g_{22}(g_{11}(x_5, x_6), g_{12}(x_7, x_8)))$$



Theorem (informal statement)

Suppose that a function of d variables is hierarchically, locally, compositional . Both shallow and deep network can approximate f equally well. The number of parameters of the shallow network depends exponentially on d as $O(\varepsilon^{-d})$ with the dimension whereas for the deep network dance is $O(d\varepsilon^{-2})$

CENTER FOR Brains Minds + Machines

Mhaskar, Poggio, Liao, 2016

Proof by induction

Proof To prove Theorem 2, we observe that each of the constituent functions being in W_m^2, (1) applied with $n = 2$ implies that each of these functions can be approximated from $\mathcal{S}_{N,2}$ up to accuracy $\epsilon = cN^{-m/2}$. Our assumption that $f \in W_m^{N,2}$ implies that each of these constituent functions is Lipschitz continuous. Hence, it is easy to deduce that, for example, if P, P_1, P_2 are approximations to the constituent functions h, h_1, h_2, respectively within an accuracy of ϵ, then since $\|h - P\| \leq \epsilon$, $\|h_1 - P_1\| \leq \epsilon$ and $\|h_2 - P_2\| \leq \epsilon$, then $\|h(h_1, h_2) - P(P_1, P_2)\| = \|h(h_1, h_2) - h(P_1, P_2) + h(P_1, P_2) - P(P_1, P_2)\| \leq \|h(h_1, h_2) - h(P_1, P_2)\| + \|h(P_1, P_2) - P(P_1, P_2)\| \leq c\epsilon$ by Minkowski inequality. Thus

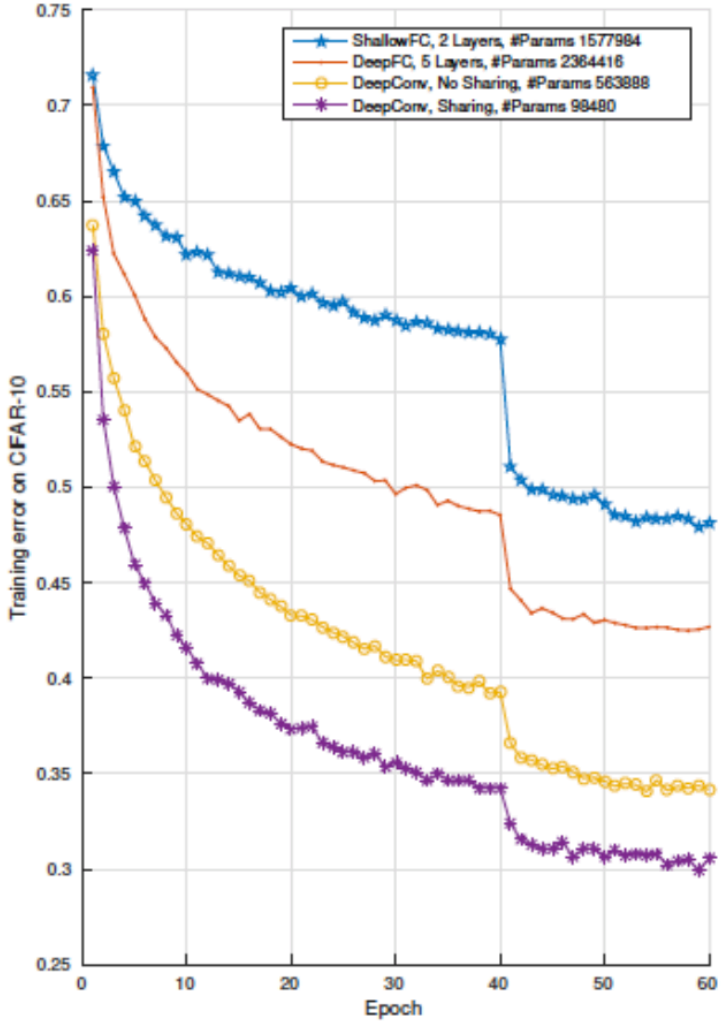
$$\|h(h_1, h_2) - P(P_1, P_2)\| \leq c\epsilon,$$

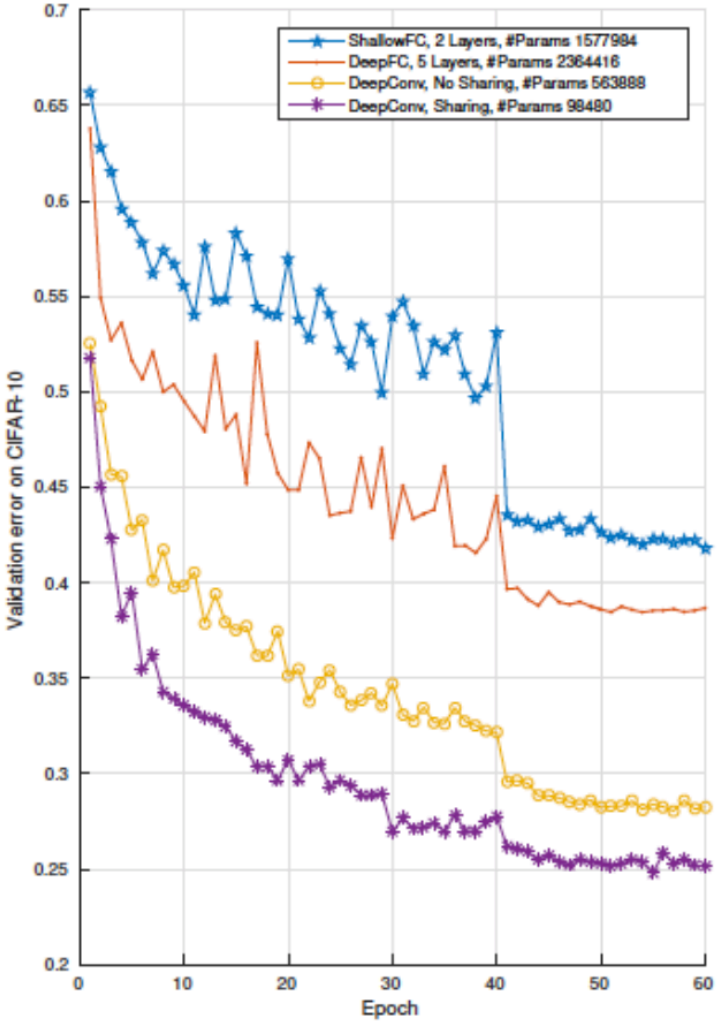
for some constant $c > 0$ independent of the functions involved. This, together with the fact that there are $(n - 1)$ nodes, leads to (6). □

Intuition

- The family of compositional functions are easier to approximate (can be approximated with less parameters)
- Deep learning leverages the compositional structure
- Why compositional functions are important for perception?

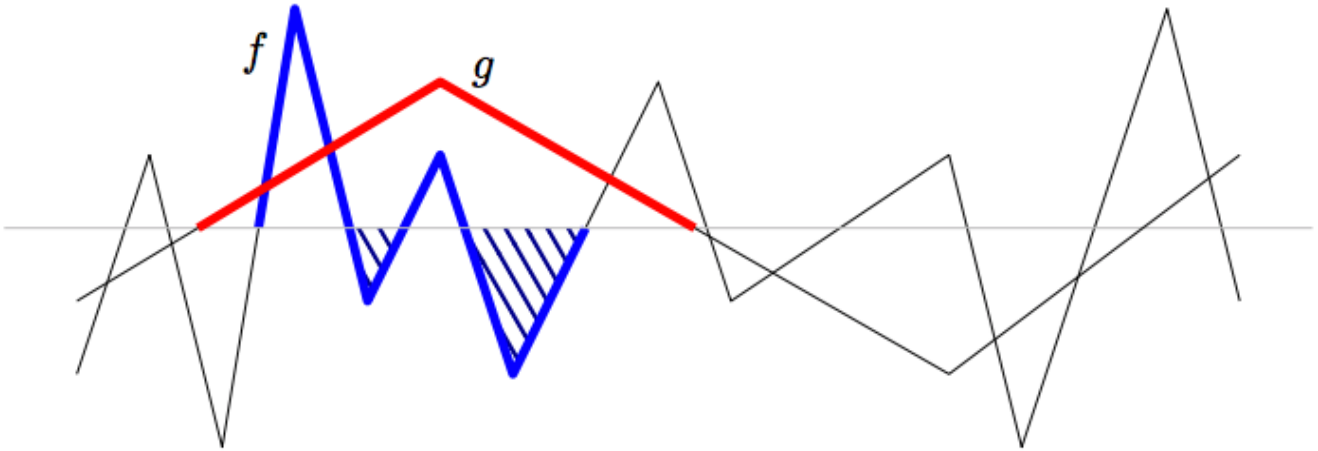
Locality of constituent functions is key: CIFAR



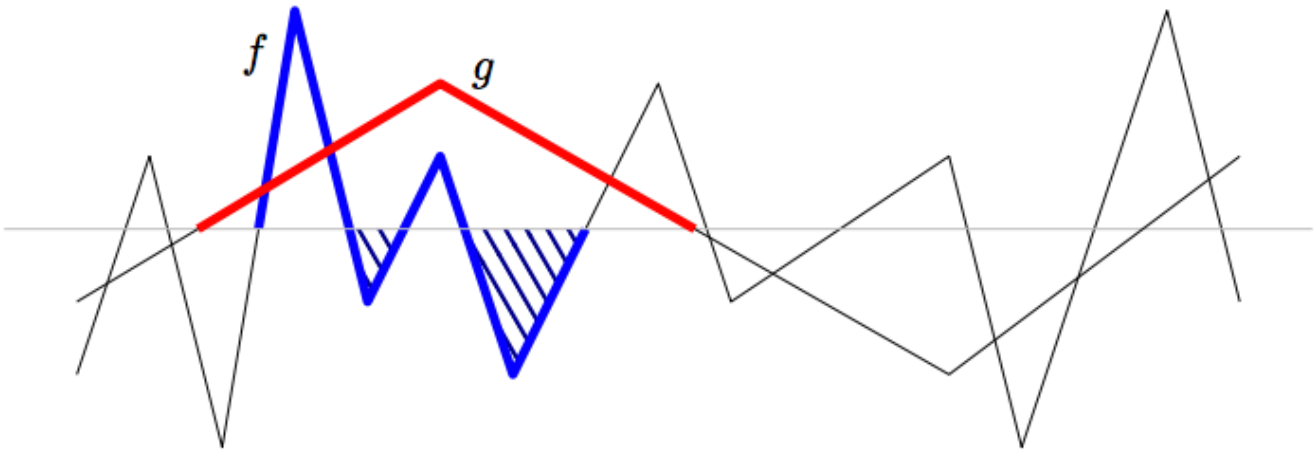


Other perspectives of why deep is good

The main result of **[Telgarsky, 2016, Colt]** says that there are functions with many oscillations that cannot be represented by shallow networks with linear complexity but can be represented with low complexity by deep networks.



Proof sketch



1. Functions with few oscillations poorly approximate functions with many oscillations.
2. Functions computed by networks with few layers must have few oscillations.
3. Functions computed by networks with many layers can have many oscillations.

Deep Networks: Three Theory Questions

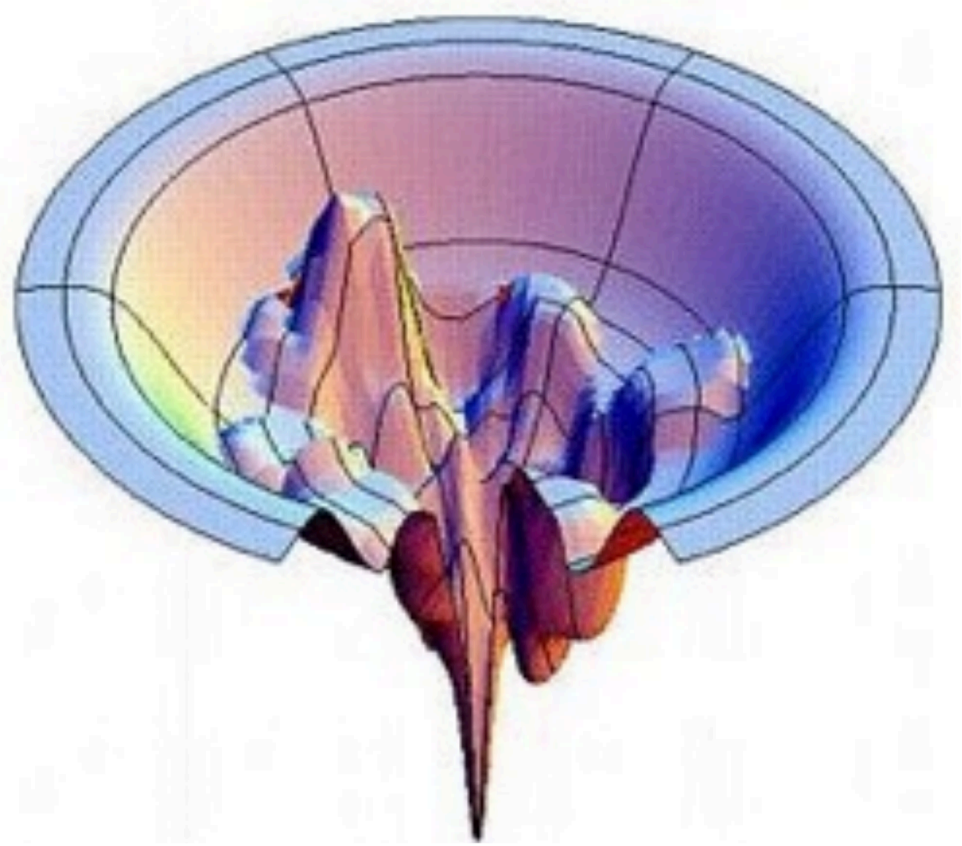
- Approximation Theory: When and why are deep networks better than shallow networks?

- ***Optimization Theory*: What is the landscape of the empirical risk?**

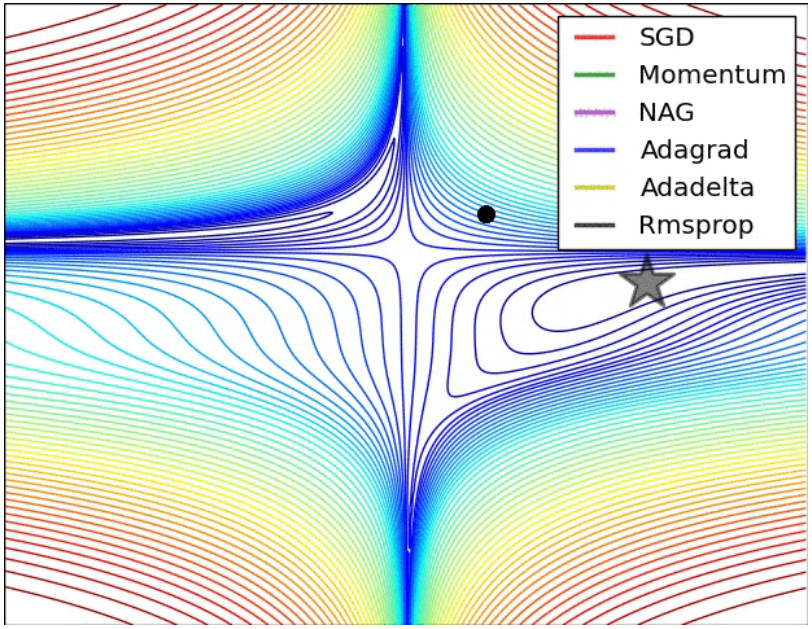
- *Learning Theory*: How can deep learning not overfit?

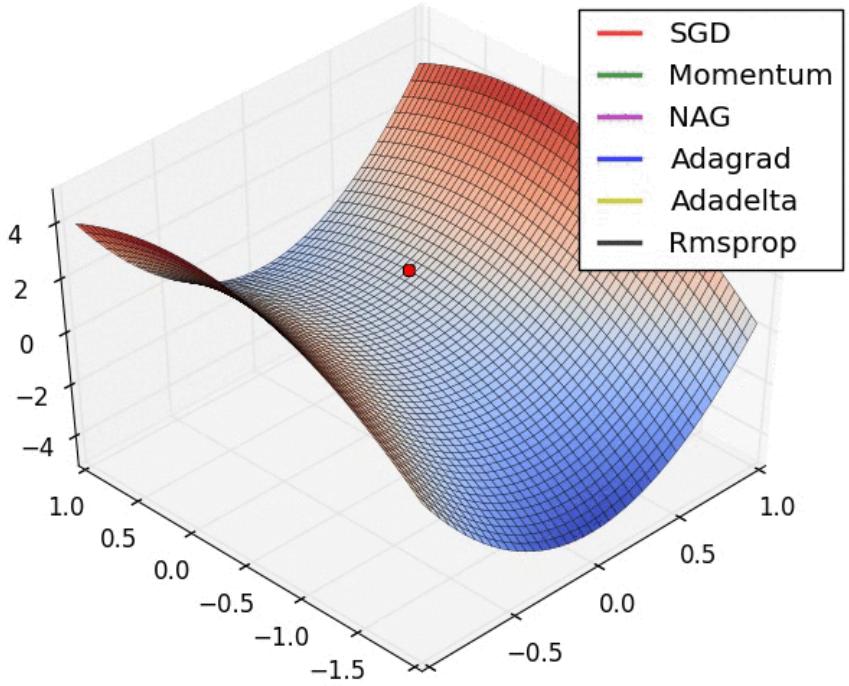
[Tomasi Poggio's lecture in STATS385]

The shape of objective function to be optimized



Two challenging cases for optimization



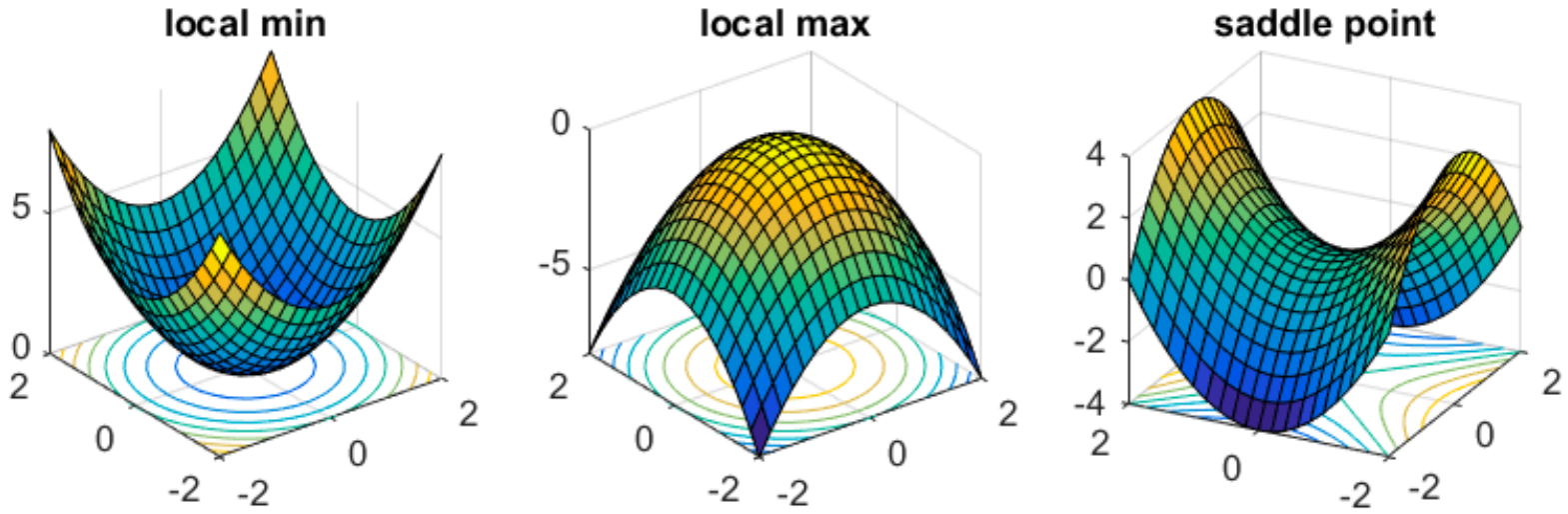


Hessian characterizes the shape at bottom

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \, \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \, \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \, \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \, \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \, \partial x_1} & \frac{\partial^2 f}{\partial x_n \, \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}.$$

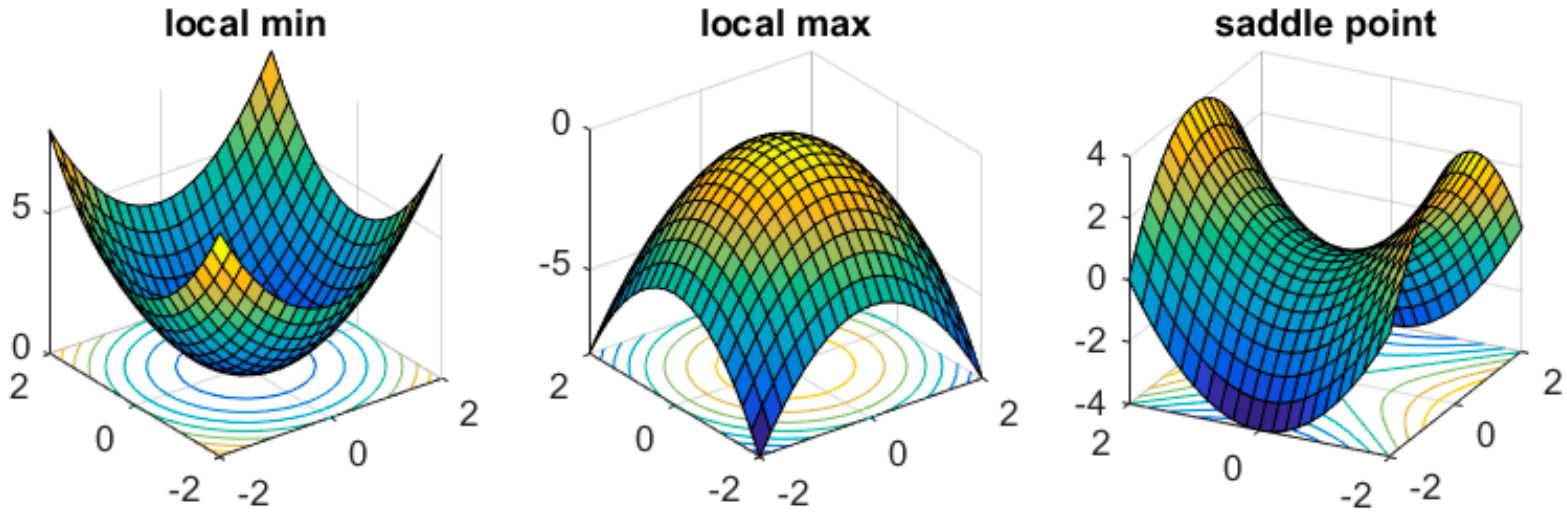
Critical points

$$\| \nabla \mathcal{L}(w) \| = 0$$



Critical points

- If all eigenvalues are positive the point is called a local minimum
- If r of them are negative and the rest are positive, then it is called a saddle point with **index** r.
- At the critical point, the eigenvectors indicate the directions in which the value of the function locally changes.
- Moreover, the changes are proportional to the corresponding -signed- eigenvalue.



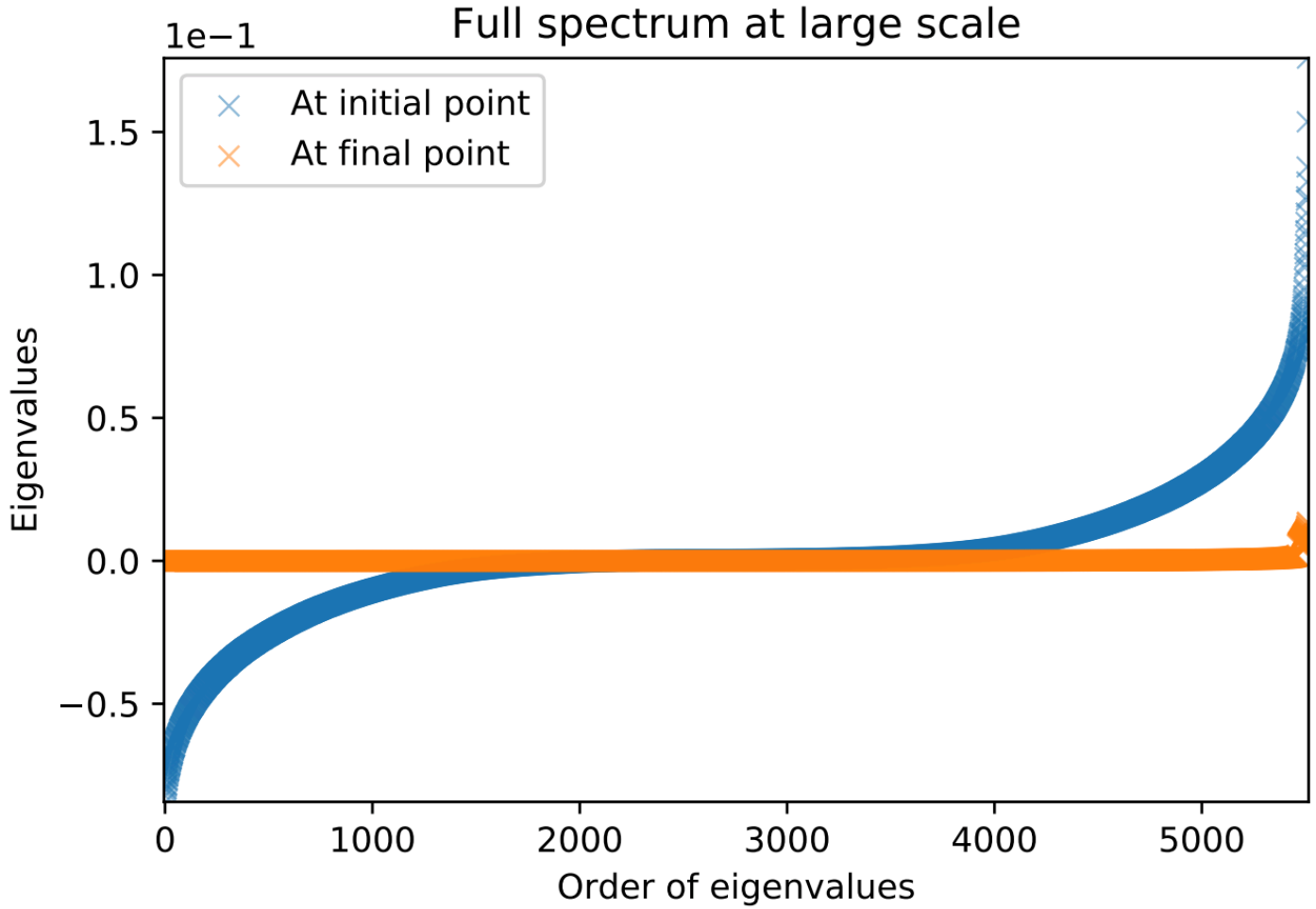
Critical points

- Classical optimization theory mostly assumes Hessian non-degenerated

- Is this the case for machine learning loss functions?

Spectrum of Hessian for an MLP

- 5K parameters in total



EMPIRICAL ANALYSIS OF THE HESSIAN OF OVER- PARAMETRIZED NEURAL NETWORKS, Sagun et al.

An explanation

$$\nabla \ell(f(w)) = \ell'(f(w))\nabla f(w)$$
$$\nabla^2 \ell(f(w)) = \ell''(f(w))\nabla f(w)\nabla f(w)^T + \ell'(f(w))\nabla^2 f(w)$$

$$\nabla^2 \mathcal{L}(w) = \frac{1}{N}\sum_{i=1}^{N}[\sqrt{\ell_i''(f_i(w))}\nabla f_i(w)][\sqrt{\ell_i''(f_i(w))}\nabla f_i(w)]^T + \frac{1}{N}\sum_{i=1}^{N}\ell_i'(f_i(w))\nabla^2 f_i(w) \quad (4)$$

EMPIRICAL ANALYSIS OF THE HESSIAN OF OVER- PARAMETRIZED NEURAL NETWORKS, Sagun et al.

An explanation

$$\nabla^2 \mathcal{L}(w) = \frac{1}{N}\sum_{i=1}^{N}[\sqrt{\ell_i''(f_i(w))}\nabla f_i(w)][\sqrt{\ell_i''(f_i(w))}\nabla f_i(w)]^T + \frac{1}{N}\sum_{i=1}^{N}\ell_i'(f_i(w))\nabla^2 f_i(w) \quad (4)$$

Rank 1

Vanish
(claimed by author)
If you are interested, check it!
(get credit)

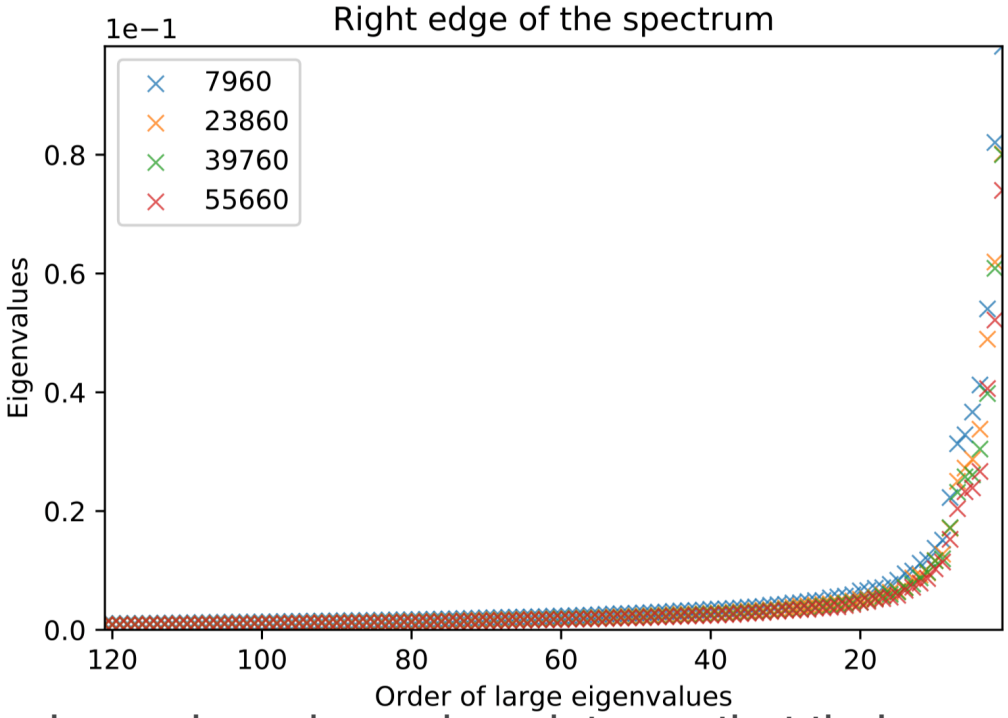
$$\nabla^2 \mathcal{L}(\hat{w}) \approx \frac{1}{N}\sum_{i=1}^{N}[\sqrt{\ell_i''(f_i(\hat{w}))}\nabla f_i(\hat{w})][\sqrt{\ell_i''(f_i(\hat{w}))}\nabla f_i(\hat{w})]^T$$

there are at least M – N many trivial eigenvalues of the Hessian!

EMPIRICAL ANALYSIS OF THE HESSIAN OF OVER- PARAMETRIZED NEURAL NETWORKS, Sagun et al.

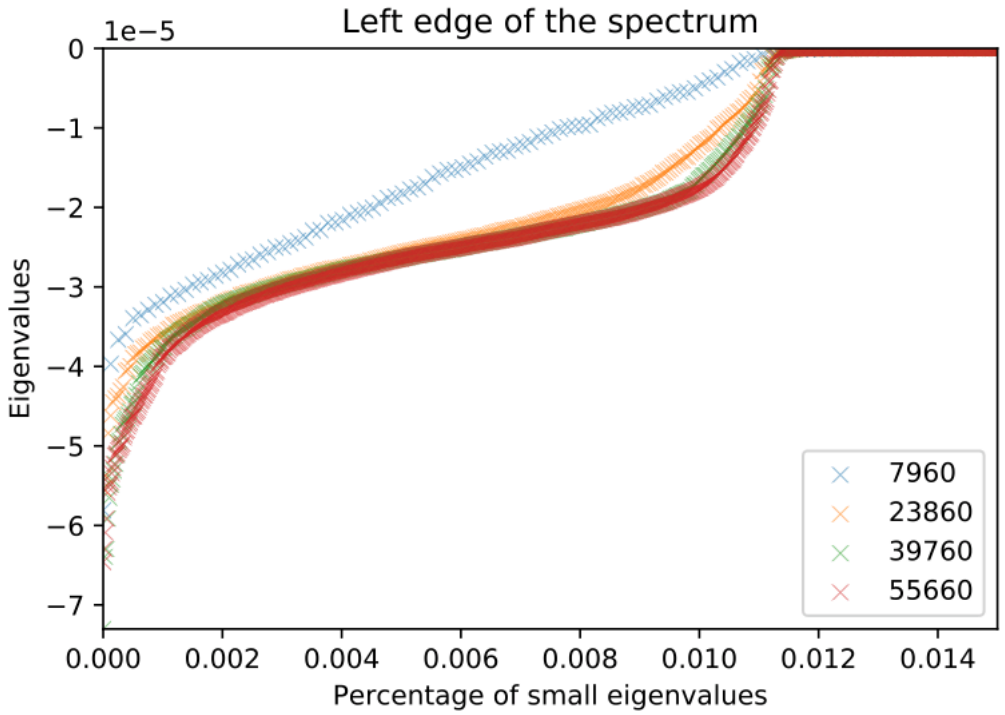
Relation between #params and eigenvalues

- Growing the size of the network with fixed data, architecture, and algorithm.
- 1K samples from MNIST



"the large positive eigenvalues depend on data, so that their number should not, in principle, change with the increasing dimensionality of the parameter space."

Relation between #params and eigenvalues



"increasing the number of dimensions keeps the ratio of small negative eigenvalues fixed."

EMPIRICAL ANALYSIS OF THE HESSIAN OF OVER- PARAMETRIZED NEURAL NETWORKS, Sagun et al.

Deep Networks: Three Theory Questions

- *Approximation Theory*: When and why are deep networks better than shallow networks?
- *Optimization Theory*: What is the landscape of the empirical risk?
- ***Learning Theory*: How can deep learning not overfit?**

[Tomasi Poggio's lecture in STATS385]

Existing theories are insufficient

- Regularization
- VC dimension
- Rademacher complexity
- Uniform stability
- …

A Statistical Mechanics Analysis

$$L^{(S)}(\boldsymbol{\theta}) = \frac{1}{S}\sum_{n\in\mathcal{B}} l(\boldsymbol{\theta}, \boldsymbol{x}_n) \ , \qquad \mathbf{g}^{(S)}(\boldsymbol{\theta}) = \frac{\partial}{\partial \boldsymbol{\theta}} L^{(S)}(\boldsymbol{\theta}) \ .$$

SGD: $\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - \eta \boldsymbol{g}^{(S)}(\boldsymbol{\theta})$

Assumptions:

(1) By the central limit theorem (CLT), we assume the noise in the stochastic gradient is Gaussian with covariance matrix $\frac{1}{S}\mathbf{C}(\boldsymbol{\theta})$

$$\mathbf{g}^{(S)}(\boldsymbol{\theta}) = \mathbf{g}(\boldsymbol{\theta}) + \frac{1}{\sqrt{S}}\Delta\mathbf{g}(\boldsymbol{\theta}), \text{ where } \Delta\mathbf{g}(\boldsymbol{\theta}) \sim N(0, \mathbf{C}(\boldsymbol{\theta})) \ .$$

We note that the covariance is symmetric positive-semidefinite, and so can be decomposed into the product of two matrices $\mathbf{C}(\boldsymbol{\theta}) = \mathbf{B}(\boldsymbol{\theta})\mathbf{B}^{\top}(\boldsymbol{\theta})$.

(2) We assume the discrete process of SGD can be approximated by the continuous time limit of the following stochastic differential equation (known as a Langevin equation)

$$\frac{d\boldsymbol{\theta}}{dt} = -\eta \boldsymbol{g}(\boldsymbol{\theta}) + \frac{\eta}{\sqrt{S}}\mathbf{B}(\boldsymbol{\theta})\mathbf{f}(t) \qquad (1)$$

where $\mathbf{f}(t)$ is a normalized Gaussian time-dependent stochastic term.

THREE FACTORS INFLUENCING MINIMA IN SGD, JASTRZEBSKI ET AL

Main Results

Theorem 1 (Equilibrium Distribution). *Assume*[1] *that the gradient covariance is isotropic, i.e.* $\mathbf{C}(\boldsymbol{\theta}) = \sigma^2\mathbf{I}$, *where* σ^2 *is a constant. Then the equilibrium distribution of the stochastic differential equation 1 is given by*

$$P(\boldsymbol{\theta}) = P_0 \exp\left(-\frac{2L(\boldsymbol{\theta})}{n\sigma^2}\right) \quad , \tag{2}$$

where $n \equiv \frac{\eta}{S}$ *and* P_0 *is a normalization constant, which is well defined for loss functions with* L_2 *regularization.*

η : learning rate, S: batch size

$n = \frac{\eta}{S}$ is a measure of the noise in the system set by the choice of learning rate and batch size S .

THREE FACTORS INFLUENCING MINIMA IN SGD, JASTRZEBSKI ET AL

Theorem 2 (Probability of ending in region near minima $\boldsymbol{\theta}_A$)**.** *Assume the loss is locally strictly convex with Hessian* $\mathbf{H}_A$ *and loss* L_A *at a minimum* $\boldsymbol{\theta}_A$*. Then the unnormalized probability of ending in minima* $\boldsymbol{\theta}_A$ *is*

$$\tilde{p}_A = \frac{1}{\sqrt{\det \mathbf{H}_A}} \exp\left(-\frac{2L_A}{n\sigma^2}\right) \tag{3}$$

where $n = \frac{\eta}{S}$ *is the noise used in the SGD process to reach* θ_A*.*

$$\frac{p_A}{p_B} = \sqrt{\frac{\det \mathbf{H}_B}{\det \mathbf{H}_A}} \exp\left(\frac{2}{n\sigma^2}\left(L_B - L_A\right)\right)$$

THREE FACTORS INFLUENCING MINIMA IN SGD, JASTRZEBSKI ET AL

Deep Networks: Three Theory Questions

- *Approximation Theory*: When and why are deep networks better than shallow networks?
- *Optimization Theory*: What is the landscape of the empirical risk?
- *Learning Theory*: How can deep learning not overfit?

[Tomasi Poggio's lecture in STATS385]