Machine Learning: Generalization 4

Hao Tang

March 24, 2025

L_2 Regularization

- The term $\frac{\lambda}{2}\|w\|_2^2$ is called an L_2 regularizer.
- It is also known as weight decay.
- The expression

$$L_S(w) + \frac{\lambda}{2}\|w\|_2^2 \tag{1}$$

is the Lagrangian of

$$\min_w \quad L_S(w) \tag{2}$$

$$\text{s.t.} \quad \|w\|_2 \leq B \tag{3}$$

Generalization bound for bounded linear classifier

- With probability $1 - \delta$, for all $h \in \mathcal{H}$,

$$L_{\mathcal{D}}(h) \leq L_S(h) + \sqrt{\frac{r^2 B^2}{n}} + 3\sqrt{\frac{\log(2/\delta)}{2n}}, \tag{4}$$

  where $\|x\|_2 \leq r$ for any $x \in S$ and $\mathcal{H} = \{x \mapsto w^\top x : \|w\|_2 \leq B\}$.

Stability

- A learning algorithm is **stable** if the learned program does not change much in performance when we change the data set slightly.
- The slight change in data set is by swapping out a data point.

$$S = \{(x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_n, y_n)\} \tag{5}$$

$$S^i = \{(x_1, y_1), \ldots, (x', y'), \ldots, (x_n, y_n)\} \tag{6}$$

- A learning algorithm is stable is $A(S)$ and $A(S^i)$ is "similar," or

$$\ell(A(S)(x), y) - \ell(A(S^i)(x), y) \tag{7}$$

is small.[1]

[1] Recall that $L_S(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i), y_i)$.

Stability

- Stable learning algorithms don't overfit.

$$\mathbb{E}_{S\sim\mathcal{D}^n}[L_{\mathcal{D}}(A(S)) - L_S(A(S))] = \mathbb{E}_{\substack{i\sim U(n)\\ S\sim\mathcal{D}^n\\ (x,y)\sim\mathcal{D}}}[\ell(A(S^i)(x_i), y_i) - \ell(A(S)(x_i), y_i)] \quad (8)$$

- Proof

$$\mathbb{E}_S[L_{\mathcal{D}}(A(S))] = \mathbb{E}_S[\mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(A(S)(x), y)]] = \mathbb{E}_S[\mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(A(S^i)(x_i), y_i)]] \quad (9)$$

$$\mathbb{E}_S[L_S(A(S))] = \mathbb{E}_S[\mathbb{E}_{i\sim U(n)}[\ell(A(S)(x_i), y_i)]] \quad (10)$$

Stability

- If ℓ is convex and ρ-Lipschitz[2] and $A(S) = \operatorname{argmin}_{h \in \{w: x \mapsto w^\top x\}} [L_S(w) + \lambda \|w\|_2^2]$, then

$$\|A(S^i) - A(S)\|_2 \leq \frac{2\rho}{\lambda n}. \tag{11}$$

- In the end, we have

$$\mathbb{E}_{S \sim \mathcal{D}^n} [L_{\mathcal{D}}(A(S)) - L_S(A(S))] \leq \frac{2\rho^2}{\lambda n}. \tag{12}$$

[2] A function f is ρ-Lipschitz if $|f(x) - f(y)| \leq \rho \|x - y\|_2$ for any x and y.

Stability

- Minimizing L_2 regularized convex and Lipschitz functions is a stable learning algorithm.
- Stable learning algorithms don't overfit.

Computational and statistical thinking

Computational

Statistical

Computational and statistical thinking

Computational

Runtime

Statistical

Samples

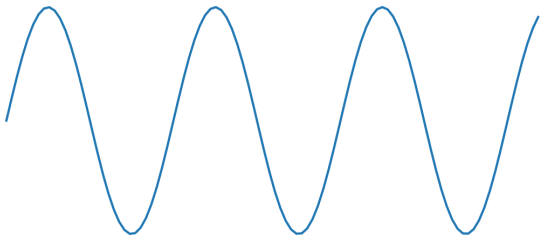
Computational and statistical thinking

Computational	**Statistical**
Runtime	Samples
How many steps do we need?	How many samples do we need?

Computational and statistical thinking

Computational	**Statistical**
Runtime	Samples
How many steps do we need?	How many samples do we need?
Polynomial number of steps	Polynomial number of samples

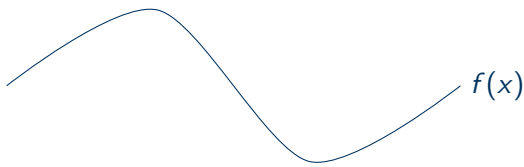
VC dimension of a sine function



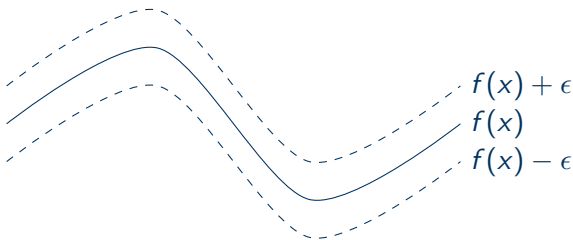
Universal approximation

- For every $\epsilon > 0$, given any Lipschitz function $f : [-1, 1]^d \to [-1, 1]$, there is a network g such that $|g(x) - f(x)| \leq \epsilon$ for any x.
- The number of nodes needed to achieve this is $O(2^d)$.

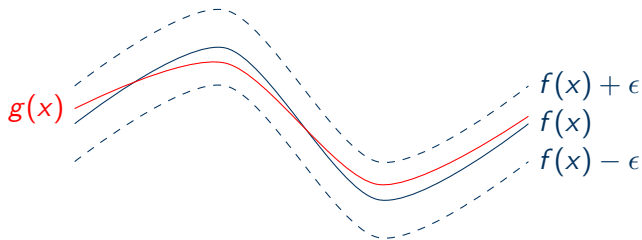
Universal approximation



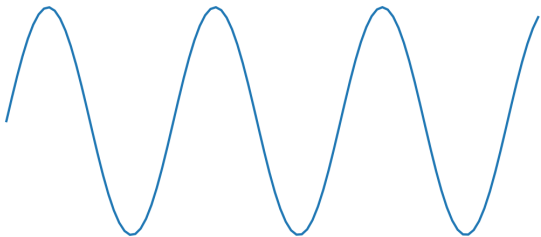
Universal approximation



Universal approximation



Can we approximate a sine function?



Universal approximation

- Polynomials are universal approximators.
- Decision trees are universal approximators.
- Gaussian mixture models are universal approximators.
- Universal approximation does not explain why neural networks are so “special.”

Depth separation

- There exists functions which can be approximated with small depth 3 networks, but cannot be approximated with depth 2 networks without using $O(2^d)$ nodes.
- Functions to show these results tend to oscillate a lot.
- Some believe the results are pathological and do not happen in practice.

Universal approximation

- What can be implemented with polynomial number of of nodes?
- Any Turing machine that runs in T operations can be implemented with a neural network of depth $O(T)$ with a total $O(T^2)$ nodes.
- Recall that VC dimension of neural networks is $O(|E| \log |E|)$, where E is the number of edges in the network.

Hardness of optimizing neural networks

Hardness of optimizing neural networks

- Training a 2-layer 3-node neural network is NP-complete.

Hardness of optimizing neural networks

- Training a 2-layer 3-node neural network is NP-complete.
- The proof converts instances of an NP-complete problem into data points.
- If we can minimize the loss of the training set, we solve the NP-complete problem.

Hardness of optimizing neural networks

- Training a 2-layer 3-node neural network is NP-complete.
- The proof converts instances of an NP-complete problem into data points.
- If we can minimize the loss of the training set, we solve the NP-complete problem.
- Maybe we don't need to solve this exactly?

Hardness of optimizing neural networks

- Approximating ERM is NP hard.
- The loss is not necessarily convex.
- ERM is hard for neural networks.

Optimizing neural networks on random labels

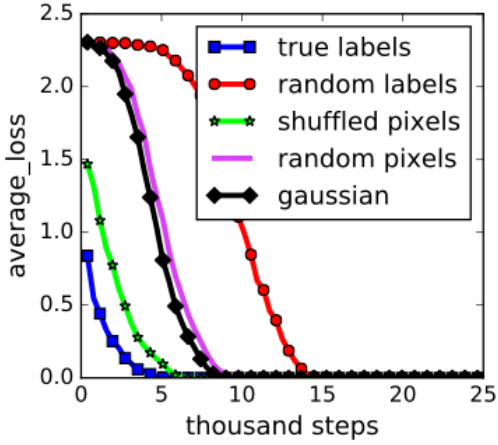


Image credit: (Zhang *et al.*, 2017)

Overparameterization

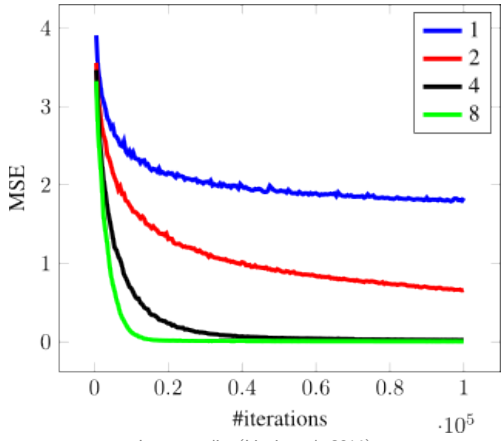


Image credit: (Livni *et al.*, 2014)

Overparameterization

- Overparameterization means using a lot more nodes than the number of points.
- Overparameterization helps optimization.

Overparameterization

- Overparameterization means using a lot more nodes than the number of points.
- Overparameterization helps optimization.
- Wouldn't the model just memorize the training set?
- Wouldn't the hypothesis class be too large to have good generalization error?

Overparameterization

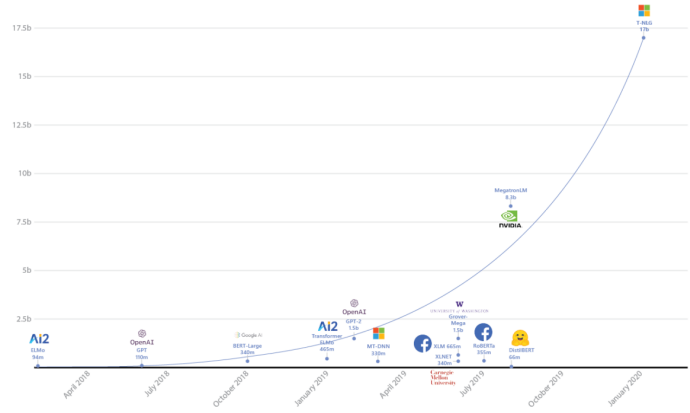


Image credit: (Rosset, 2020)

Overparameterization

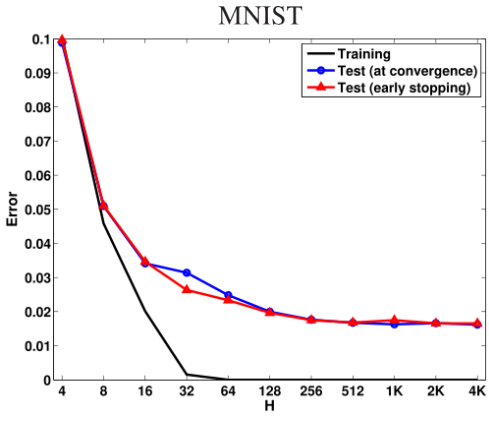
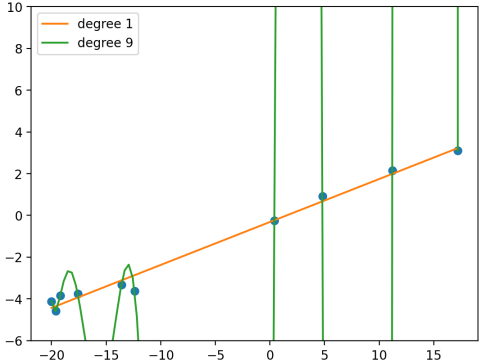


Image credit: (Neyshabur *et al.*, 2014)

Interpolation

- Fitting a data set to training error zero is called interpolation.
- Why doesn't interpolation overfit?

Interpolation



Interpolation

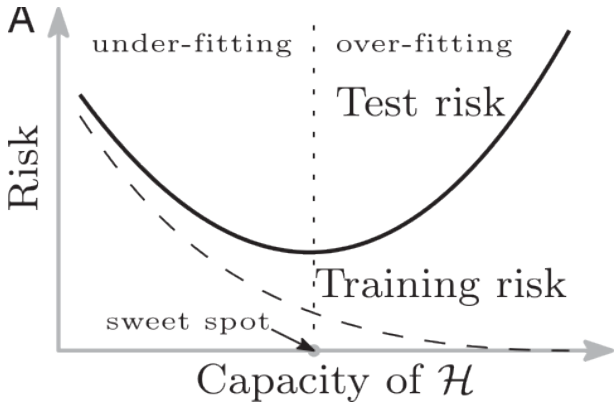


Image credit: (Belkin *et al.*, 2019)

Interpolation

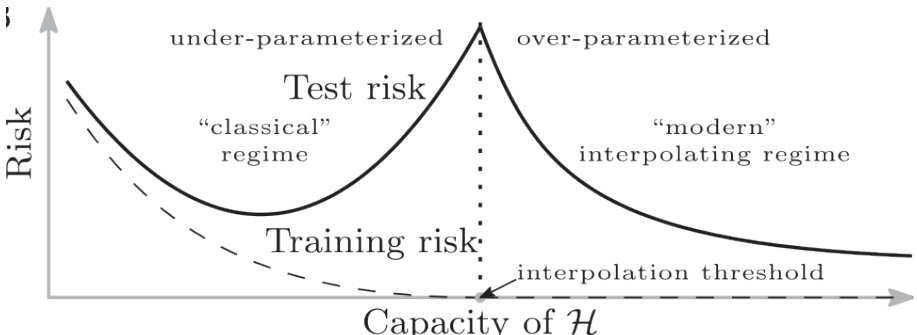


Image credit: (Belkin *et al.*, 2019)

Interpolation

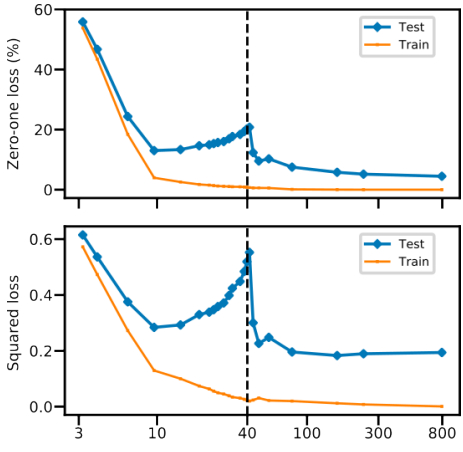


Image credit: (Belkin *et al.*, 2019)

Overfitting

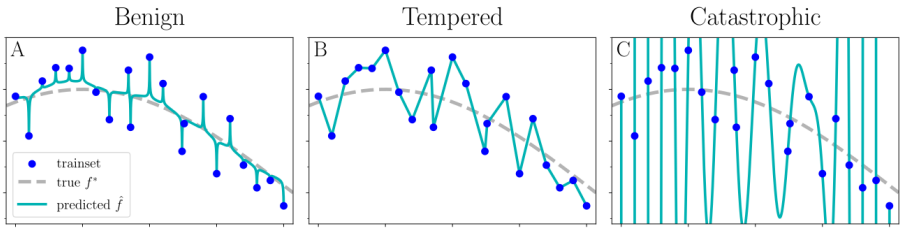


Image credit: (Mallinar *et al.*, 2022)

Sharp and flat minima

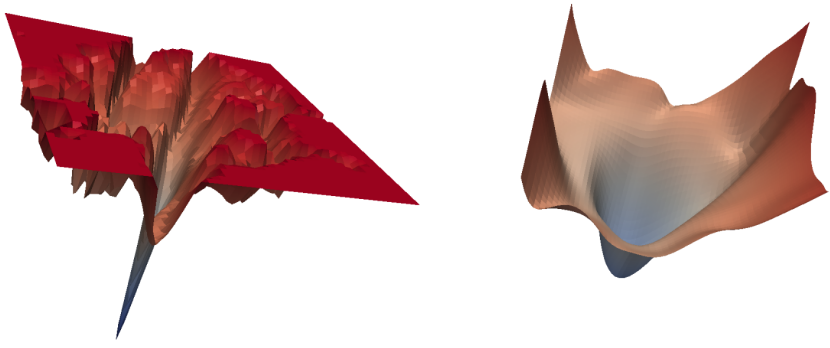


Image credit: (Foret *et al.*, 2021)

The grand goal

- Coming back to regularized ERM, it considers both the training error and the capacity during optimization.

$$\min_{w} \quad L_S(w) + \frac{\lambda}{2}\|w\|_2^2 \tag{13}$$

- If we know something that controls the capacity, we should optimize it.

In practice

- Always start with the training error.
- Always start with ERM.
- Why is the training error not close to zero?
- Regularize