Mathematics of Deep Learning
Lecture 5: Interpolation, Over-parametrization, Kernels & Neural Nets, High-Dimensional Geometry

Prof. Predrag R. Jelenković

Time: Tuesday 4:10-6:40pm

Dept. of Electrical Engineering
Columbia University , NY 10027, USA
Office: 812 Schapiro Research Bldg.
Phone: (212) 854-8174
Email: predrag@ee.columbia.edu
URL: http://www.ee.columbia.edu/∼predrag

Recall d-Width

- We will pick K to be a d-ball (or d-cube)

$$B^d = \{\boldsymbol{x} : \|\boldsymbol{x}\|_2^2 = x_1^2 + \cdots + x_d^2 \leq 1\}$$

- $C^s(B^d)$ - class of functions on B^d with up to $s1$ continuous partial derivatives
- Sobolev space $\mathcal{W}_p^s = \mathcal{W}_p^s(B^d)$ is a completion of $C^s(B^d)$ with the respect to the p norm. In addition we assume that all $f \in \mathcal{W}_p^s$ have a bounded norm $\|f\|_{s,p}$.
- Also, we define a space of all NN approximations with n neurons

$$\mathcal{N}_n := \left\{ \sum_{i=1}^{n} a_i \sigma(\boldsymbol{w}_i \cdot \boldsymbol{x} + b_i), \boldsymbol{w}_i \in \mathbb{R}^d, a_i, b_i \in \mathbb{R} \right\}$$

- and (d-width)

$$E_{s,p}(\mathcal{W}_p^s, \mathcal{N}_n) := \sup_{f \in \mathcal{W}_p^s} \inf_{\hat{f} \in \mathcal{N}_n} \|f - \hat{f}\|_{s,p}$$

Uniform Approximation Rates

Theorem Assume that σ is infinitely differentiable and not a polynomial. Also, assume that the NN approximation $\hat{f}_n$, as a mapping $\mathcal{W}_p^s \to \mathcal{N}_n$, is continuous in parameters $(\boldsymbol{w}, a, b)$, than, for each $1 \leq p \leq \infty, m \geq 1$, there is a finite constant C, such that

$$C^{-1} n^{-s/d} \leq E_{s,p}(\mathcal{W}_p^s, \mathcal{N}_n) \leq C n^{-s/d}$$

Remarks:

- For more details see Theorem 1 and the notes after it in Poggio et al. (2017). Also, Section 6 in Pinkus (1999), Theorems 6.6 & 6.8.
- An immediate consequence of the preceding theorem is that if we want a uniform ϵ approximation, then the number of neurons n (parameters) needed is

  $$n^{-s/d} = \epsilon \;\Rightarrow\; n = O(\epsilon^{-d/s}) \quad \text{(curse of dimensionality)}$$

  We will use this for comparison with deep learning results.
- Hence, to getter better results, additional assumptions on $\mathcal{W}_p^s$ are needed, e.g., Barron assuming that functions are bounded "spectrum"/variation.

General Idea of Proving the Upper Bound

- It is well known for d-dimensional polynomials $\mathcal{P}_k$ that
$$E_{s,p}(\mathcal{W}_p^s, \mathcal{P}_k) \leq Ck^{-s}. \tag{1}$$
- Also, $h(\boldsymbol{x}) \in \mathcal{P}_k$ can be represented as (see proof of Theorem 4.1 in Pinkus)
$$h(\boldsymbol{x}) = \sum_{i=1}^{r} p_i(\boldsymbol{w}_i \cdot \boldsymbol{x}),$$
where $p_i(x)$ are univariate polynomials and $r = \binom{d-1+k}{k} \approx k^{d-1}$
- Moreover, each of the univariate polynomials $p_i(x)$ of degree k can be approximated with a NN having k neurons.
- Hence, the total number of neurons needed to approximated $h(\boldsymbol{x}) \in \mathcal{P}_k$ is
$$n = kr \approx k^d \to k \approx n^{1/d}$$
- Finally, the preceding bound and (1) imply
$$E_{s,p}(\mathcal{W}_p^s, \mathcal{P}_k) \leq Cn^{-s/d}.$$

Power of Depth: Kolmogorov Superposition Theorem

Solution to Hilbert's 13th problem. See Section 7 in Pinkus (1999).

Theorem There exist d constants $\lambda_j > 0, j = 1, \ldots, d, \sum_{j=1}^{d} \lambda_j \leq 1$, and $2d+1$ strictly increasing continuous functions $\phi_i : [0,1] \to [0,1]$, such that any continuous function f of d variables on $[0,1]^d$ can be represented as

$$f(\boldsymbol{x}) = \sum_{i=1}^{2d+1} g\left(\sum_{j=1}^{d} \lambda_j \phi_i(x_j)\right)$$

for $g \in C([0,1])$ depending on f.

- There are other forms of this theorem that might be more suitable to applications in NNs.
- For a brief historical summary and a ReLU implementation see Montanelli and Yang (2020): Error bounds for deep ReLU networks using the Kolmogorov–Arnold superposition theorem

General Power of Depth: 2 Hidden Layers are Enough

Theorem by Maiorov & Pinkus (1999), see Theorem 7.1 in Pinkus (1999).

Theorem There exists and activation function $\sigma \in C^\infty(\mathbb{R})$, which is strictly increasing, sigmoidal, and has the following property. For any $f \in C([0,1]^d)$, and $\epsilon > 0$, there exist constants, $\delta_i, a_{ij}, b_{ij}, \gamma_i$, and vectors $\boldsymbol{w}_{ij} \in \mathbb{R}^d$, for which

$$\left| f(\boldsymbol{x}) - \sum_{i=1}^{4d+3} \delta_i \sigma \left(\sum_{j=1}^{2d+1} a_{ij} \sigma(\boldsymbol{w}_{ij} \cdot \boldsymbol{x} + b_{ij}) + \gamma_i \right) \right| < \epsilon,$$

for $\boldsymbol{x} \in [0,1]^d$.

Remarks

- $\Rightarrow$ The number of neurons needed is $(4d+3)(2d+1) = O(d^2)$.
- This seems to contradict the general rates bound that we covered, i.e., the curse of dimensionality. Problem: σ depends on f, and thus, the training parameters may not be continuous in f.

Last Lectures: Expressive Power of Neural Nets

Main idea exploits: Composition property of neurons

- Depth & composition can help with "high frequency" functions
  - Depth-2 & width m: Creates up to $O(m)$ sawtooth function
  - Depth-l & width m: Creates up to $O(m^l)$ sawtooth function
- Using hat hat function (ReLUs) the above was exploited in
  - Classification: Telgarsky (2015)
  - Function approximation: Yarotsky (2017), Liang and Srikant (2017), Montanelli and Du (2018)
- General question: What is the smallest architecture, i.e., the number of neurons to approximate closely a function
- Yarotsky (2017) showed that one can approximate the quadratic x^2 and product xy with a deep network having $O(\ln(1/\epsilon))$ layers and neurons, leading to general approximation bound for s-differentiable functions in d-dimensions needing $O(\epsilon^{-d/s}\ln(1/\epsilon))$ weights and computation units to achieve ϵ approximation.

Use of Sparse Grids

Useful tool in approximation theory

- Approximation theory and numerical computation has been battling curse of dimensionality for years
- References:

[BG04] Sparse Grids, by Bungartz and Griebel, 2004.
[G12] Sparse Grids in a Nutshell by Garcke, 2012.

- Korobov spaces provide additional structure.
  - Using sparse grids, Korobov spaces and ReLUs, in Theorem 1 of Montanelli and Du (2018), reduced the number of required neurons to

$$N = O(\epsilon^{-1/2} |\log_2 \epsilon|^{3(d-1)/2+1} \log_2 d) \quad \text{ReLUs}$$

    Note: $\log_2 \epsilon$ is raised to power instead of $1/\epsilon$ in Yarotsky (18).
  - Very recent (ICLR 2022) results in this direction by M. Blanchard, M. A. Bennouna: Shallow and Deep Networks are Near-Optimal Approximators of Korobov Functions .

Other Possibilities to Explore

- Eldan and Shamir (2016) showed depth separation between 1-hidden layer and 2-hidden layers networks.
  - It would be interesting to explore general depth separation results for any number of hidden layers (not just 1 and 2)
- In general, find classes of functions that can be efficiently represented in a particular architecture: numebr of layers versus depth.
- Impact of different activation functions:
  - We have seen that most practically used activation functions are asymptotically equivalent, e.g., all sigmoids are equivalent to a step function, but in real applications the choice of an activation function can make a difference.

Class Reading on Deep NN Approximation Theory

We tried to answer the question: Can deep neural networks express functions more efficiently in terms of the number of neurons compared to shallow networks?

- Representation Benefits of Deep Feedforward Networks, by Telgarsky, 2015.
- Benefits of depth in neural networks, by Telgarsky, 2016.
- Error bounds for approximations with deep ReLU networks, by Yarotsky, 2017.

  Optimal approximation of continuous functions by very deep ReLU networks, by Yarotsky, 2018.
- WHY DEEP NEURAL NETWORKS FOR FUNCTION AP- PROXIMATION?, Liang and Srikant, 2017.
- NEW ERROR BOUNDS FOR DEEP RELU NETWORKS USING SPARSE GRIDS, by Montanelli and Du, 2018.

  Error bounds for deep ReLU networks using the Kolmogorov–Arnold superposition theorem, by Montanelli and Yang, 2020.
- Why and When Can Deep-but Not Shallow-networks Avoid the Curse of Dimensionality: A Review, by Poggio et al. (2017)
- Will cover some ideas from: The Power of Depth for Feedforward Neural Networks, by Eldan and Shamir, 2016.
- An average-case depth hierarchy theorem for Boolean circuits, by Rossman, Servedio, Tan, 2015.
- Shallow and Deep Networks are Near-Optimal Approximators of Korobov Functions , by M. Blanchard, M. A. Bennouna, by ICLR 2022.
- Mathematically oriented survey paper on ReLU approximation theory: Neural Net Approximation, by DeVore, Hanin & Petrova, Dec 2020.
- Check the citations as well as the references inside these papers for additional reading.

Interpolation

Before we start looking into the NN generalization properties, let us consider the problem of interpolation: (see Section 5 in Pinkus (1999))

- Assume that σ is continuous ($\sigma \in C(\mathbb{R})$) and not a polynomial.
- Consider n data points $\{(x_i, y_i)\}_{i=1}^n, x_i \in \mathbb{R}^d, y_i \in \mathbb{R}$.
- Consider a shallow NN with one hidden layer and m neurons and weights $\{(w_j, b_j, a_j)\}_{j=1}^m, w_j \in \mathbb{R}^d, a_j, b_j \in \mathbb{R}$

$$f_m(x) = \sum_{j=1}^{m} a_j \sigma(w_j \cdot x - b_j)$$

- Interpolation problem: How many neurons m do we need to perfectly reproduce n data points, i.e., to have

$$f_m(x_1) = y_1, f_m(x_2) = y_2, \quad \cdots \quad , f_m(x_n) = y_n.$$

Interpolation

It turns out that it is enough to pick $m = n$ neurons.

Theorem Assume that σ is continuous ($\sigma \in C(\mathbb{R})$) and not a polynomial. For any set of distinct points $x_i \in \mathbb{R}^d$ and the associated $y_i, 1 \leq i \leq n$, there exist $\{(w_j, b_j, a_j)\}_{j=1}^m, w_j \in \mathbb{R}^d, a_j, b_j \in \mathbb{R}$, such that

$$\sum_{j=1}^{m} a_j \sigma(w_j \cdot x_i - b_j) = y_i, \quad 1 \leq i \leq n.$$

Proof:

- Assume first that σ is infinitely differentiable.
- Since $x_i \in \mathbb{R}^d$ are distinct, there exists $w \in \mathbb{R}^d$ such that $t_i = x_i \cdot w, 1 \leq i \leq n$ are all distinct.
  (This reduces the problem from d dimensions to 1.)
- Let $w_i = v_i w, v_i \in \mathbb{R}$. Now, the interpolation problem reduces to finding $a_i, v_i, b_i \in \mathbb{R}, 1 \leq i \leq n$ such that

$$\sum_{j=1}^{n} a_j \sigma(v_j t_i - b_j) = y_i, \quad 1 \leq i \leq n. \tag{2}$$

Interpolation

Proof:

- For Equation (2) to have a solution for any choice of $\{y_i\}$, it is enough that there exist $v_i, b_i \in \mathbb{R}, 1 \leq i \leq n$ such that

$$\det([\sigma(v_j t_i - b_j)]_{i,j=1}^n) \neq 0. \tag{3}$$

- On the other hand, if there are no $v_i, b_i, 1 \leq i \leq n$, then functions $\sigma(vt_i - b), 1 \leq i \leq n$ are linearly dependent for all $v, b \in \mathbb{R}$, i.e., there exist $\{c_i\}_{i=1}^n \not\equiv 0$ such that

$$\sum_{i=1}^{n} c_i \sigma(vt_i - b) = 0, \quad \text{for all } v, b \in \mathbb{R}. \tag{4}$$

- Next, since σ is infinitely differentiable and not a polynomial, there exist b_0 such that

$$\frac{d^j}{dv^j}\sigma(vt_i - b_0)\bigg|_{v=0} = \sigma^{(j)}(-b_0)t_i^j, \quad \sigma^{(j)}(-b_0) \neq 0, 0 \leq j \leq n-1.$$

- Now, if we take the same derivatives as above in Equation (4) for $0 \leq j \leq n-1$, and then divide each of them by $\sigma^{(j)}(-b_0) \neq 0$

Interpolation

Proof:

- we obtain that constants $\{c_i\}_{i=1}^n$ must satisfy

$$\sum_{i=1}^{n} c_i t_i^j = 0, \quad 0 \le j \le n-1. \tag{5}$$

- The preceding system of equations has a unique solution since the following determinant, known as Vandermonde determinant, is non-zero (compute it explicitly for $n = 3$)

$$\det([t_i^{j-1}]_{i,j=1}^n) = \prod_{1 \le i < k \le n} (t_i - t_k) \neq 0$$

since t_i are distinct.

- Hence, the above implies that Equation (5) has a unique solution $\{c_i\}_{i=1}^n \equiv 0$, which contradicts our assumption in Equation (4), implying that the determinant in Equation (3) is not zero, i.e., Equation (2) has a solution. This proves the theorem for σ being infinitely differentiable.
- If σ is just continuous, we can always smooth it out by convolving it with infinitely differentiable function ϕ_δ (say Gaussian-like $\phi_\delta(t) = e^{-t^2/(2\delta^2)}/\sqrt{2\pi\delta}$, such that $\sigma_\delta(t) = \sigma * \phi_\delta(t)$ is infinitely differentiable and $|\sigma(t) - \sigma_\delta(t)| < \epsilon$; details omitted.

Interpolation and Overfitting

- Can we interpolate n data points with $m < n$ neurons?
  - In general, the answer is no.
  - For example, consider ReLU activation function $\sigma(x) = x^+$
  - If assume n data points such that

$$y_{2k-1} = 1, \quad y_{2k} = 0, \quad 1 \leq k \leq n/2,$$

    a shallow NN function must have n distinct slopes to perfectly interpolate these points.
  - Hence, we need at least $m = n$ neurons.
- Overfitting: The preceding interpolation problem show that any shallow NN with $m \geq n$ neurons can perfectly reproduce data, i.e., it can overfit.
  - The fact that in many cases training over-parametrized NN with $m \gg n$ does not lead to overfitting as often as one would expect is a mystery
- We are going to study over-parametrized networks next

Over-Parametrization

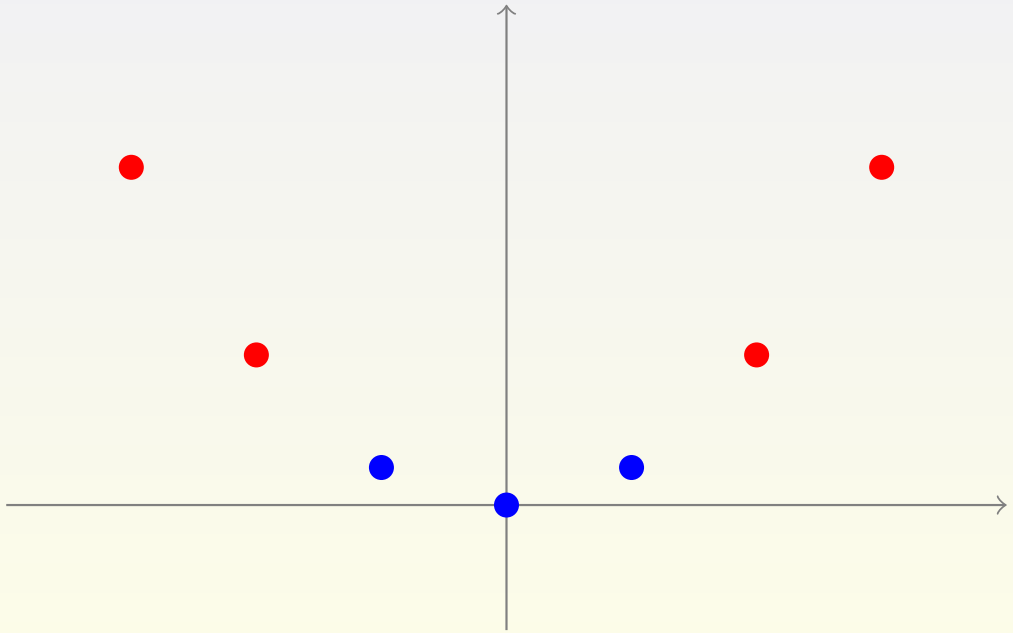
- Previous idea: **Composition of functions**
- New idea: **Over-parametrization**
  - Mathematically speaking: Passes the width $m \to \infty$
    (this is the opposite of the previous work on expressiveness, where the goal was to minimize the # of neurons)
  - Mathematical tools: Laws of large numbers
    - "Smooth out" the layer functions
  - This idea was used in a numeber of recent papers to:
    - Connect NNs to Kernels
    - Showed that all local minima are global
    - Showed convergence to global minima
    - Showed generalization bounds
      (very recently, 2019+)

Feature Transformations

Example: Classifying red and blue point on a real line

- Can't be separated by a single point (hyperplane)

- Nonlinear feature transformation: $x \rightarrow (x, x^2)$
- Now, the points are separable by a single hyperplane (!)



- This is called feature extraction

Feature Transformations: Kernels

General idea:

- Map data into higher dimensional space $\phi(\boldsymbol{x}) : \mathbb{R}^d \to \mathbb{R}^m$
  m could be infinite
- Hope that data can untangle easier in $\mathbb{R}^m$, making a projection more accurate

$$\hat{f}(\boldsymbol{x}) = \langle \boldsymbol{\beta}, \phi(\boldsymbol{x}) \rangle = \sum_{i=1}^{m} \beta_i \phi_i(\boldsymbol{x})$$

- Problem: $m \gg d$, possible infinite $\Rightarrow$ computationally infeasible
- Solution: Represent $\hat{f}$ in the dot product space
  $K(\boldsymbol{x}, \boldsymbol{y}) = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{y}) \rangle$ - dot-products are called Kernels
  - It can be shown that $\boldsymbol{\beta}$ is an element of the data subspace $\text{span}(\phi(\boldsymbol{x}_i), 1 \leq i \leq n)$, where n is the number of data points, and thus it can be represented as $\boldsymbol{\beta} = \sum_{j=1}^{n} a_j \phi(\boldsymbol{x}_i)$

    $$\Rightarrow \hat{f}(\boldsymbol{x}) = \langle \boldsymbol{\beta}, \phi(\boldsymbol{x}) \rangle = \sum_{j=1}^{n} a_j \langle \phi(\boldsymbol{x}_j), \phi(\boldsymbol{x}) \rangle = \sum_{j=1}^{n} a_j K(\boldsymbol{x}_j, \boldsymbol{x})$$

    Computational complexity depends on number of data points n
- Well-developed theory: Reproducing Kernel Hilbert Spaces (RKHS)

RKHS: Reproducing Kernel Hilbert Spaces

RKHS - $\mathcal{H}_k$: Subset of Hilbert spaces with really nice properties

- Kernel is a positive definite function $k(x, y)$
  $(\sum \alpha_i \alpha_j k(x_i, x_j) \geq 0)$
- **Reproducing property**: if $f \in \mathcal{H}_k$, then

$$< f, k(\cdot, x) >= f(x)$$

- Each kernel, $k(x, y)$, defines uniquely the Hilbert space, $\mathcal{H}_k$ Hence, in this space, $\mathcal{H}_k$, all we need to know if the kernel!
- Functions in this space $f(x) \in \mathcal{H}_k$ can be written as

$$f(x) = \sum_i \beta_i k(x, x_i)$$

Supervised Learning in RKHS

Here we optimize over all approximation function in $\mathcal{H}_k$

$$\hat{f}^* = \arg\min_{\hat{f} \in \mathcal{H}_k} \sum_{i=1}^{n} L(y_i, \hat{f}(x_i)) + \lambda \Omega(\|\hat{f}\|^2_{\mathcal{H}_k}) \tag{6}$$

Representer Theorem For any loss function L and any strictly increasing function $\Omega : \mathbb{R} \to \mathbb{R}$, an optima solution, $\hat{f}^*$, of Equation (6) has a form

$$\hat{f}^*(x) = \sum_{i=1}^{n} \beta_i k(x, x_i)$$

$\hat{f}^*$ has a finite representation (!) even though $\mathcal{H}_k$ can be (is) infinite.
We can put much of the supervised learning under the same umbrella.
Drawback: works only with ℓ_2 norm, $\|\hat{f}\|_{\mathcal{H}_k}$ - doesn't work for Lasso.

Over-parametrized NNs and Kernels

Cho&Saul (2009) and Tsuchida et al. (2018), and others:

- Exploit large width, m, to explicitly compute the kernel corresponding to a hidden layer

Notation

- $\boldsymbol{w}_i$ - independent random initial weights with density $f(w), w \in \mathbb{R}^n$
- $\boldsymbol{\phi}(\boldsymbol{x}) \in \mathbb{R}^m, \boldsymbol{x} \in \mathbb{R}^n$ - hidden layer vector, m width of the hidden layer

$$\boldsymbol{\phi}(\boldsymbol{x}) := \frac{1}{\sqrt{m}}(\sigma(\langle \boldsymbol{w}_1, \boldsymbol{x} \rangle), \dots \sigma(\langle \boldsymbol{w}_m, \boldsymbol{x} \rangle))$$

Then, for $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ and large m

$$\langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{y}) \rangle = \frac{1}{m} \sum_{i=1}^{m} \sigma(\langle \boldsymbol{w}_i, \boldsymbol{x} \rangle) \sigma(\langle \boldsymbol{w}_i, \boldsymbol{y} \rangle) \tag{7}$$

$$\approx \int_{\mathbb{R}^n} \sigma(\langle \boldsymbol{w}, \boldsymbol{x} \rangle) \sigma(\langle \boldsymbol{w}, \boldsymbol{y} \rangle) f(\boldsymbol{w}) d\boldsymbol{w} =: k(\boldsymbol{x}, \boldsymbol{y}) \tag{8}$$

The "$\approx$" can be made precise in a probabilistic sense

Arc-Cosine Kernels

Theorem (Cho&Saul (2009)) Let $\boldsymbol{w}$ be a vector of independent normal, $\mathcal{N}(0,1)$, random variables, and let $\theta = \cos^{-1}\left(\frac{\langle \boldsymbol{x},\boldsymbol{y}\rangle}{\|\boldsymbol{x}\|_2\|\boldsymbol{y}\|_2}\right)$. Then,

$$k(\boldsymbol{x},\boldsymbol{y}) = \frac{\pi-\theta}{2\pi}, \qquad \text{for } \sigma(x) = 1_{\{x\geq 0\}}$$

$$k(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{2\pi}(\sin\theta + (\pi-\theta)\cos\theta), \qquad \text{for } \sigma(x) = x_+$$

Proof Direct integration with respect to the Gaussian density: for step function $\sigma(x) = 1_{\{x\geq 0\}}$,

$$k(\boldsymbol{x},\boldsymbol{y}) = \int_{\boldsymbol{w}\in\mathbb{R}^d} 1_{\{\boldsymbol{x}\cdot\boldsymbol{w}\geq 0\}} 1_{\{\boldsymbol{y}\cdot\boldsymbol{w}\geq 0\}} f(\boldsymbol{w})d\boldsymbol{w}, \quad f(\boldsymbol{w}) = \frac{e^{-\|\boldsymbol{w}\|_2^2/2}}{(2\pi)^{n/2}}$$

Key observation: due to the rotational symmetry of the Gaussian density, we can pick the coordinates to align with $(\boldsymbol{x},\boldsymbol{y})$, i.e., $\boldsymbol{x} = (x_1, 0, 0, \ldots), x_1 > 0$, and $\boldsymbol{y} = (y_1, y_2, 0, \ldots)$, and the preceding integral reduces to 2 dimensional:

$$k(\boldsymbol{x},\boldsymbol{y}) = \int_{(w_1,w_2)\in\mathbb{R}^2} 1_{\{x_1 w_1\geq 0\}} 1_{\{y_1 w_1 + y_2 w_2\geq 0\}} \frac{e^{-(w_1^2+w_2^2)/2}}{2\pi} dw_1 dw_2$$

Arc-Cosine Kernels

Proof Now, divide the expression inside the indicator functions by x_1 and the norm $\|\boldsymbol{y}\|_2$, to obtain

$$k(\boldsymbol{x},\boldsymbol{y}) = \int_{(w_1,w_2)\in\mathbb{R}^2} 1_{\{w_1\geq 0\}} 1_{\{w_1\cos\theta + w_2\sin\theta\geq 0\}} \frac{e^{-(w_1^2+w_2^2)/2}}{2\pi} dw_1 dw_2$$

where θ is the angle between $\boldsymbol{x}$ and $\boldsymbol{y}$. Finally, write the preceding integral in polar coordinates: $w_1 = r\cos\phi, w_2 = r\cos\phi$, to obtain

$$\begin{aligned} k(\boldsymbol{x},\boldsymbol{y}) &= \int_0^{2\pi} d\phi \int_0^{\infty} 1_{\{\cos\phi\geq 0\}} 1_{\{\cos\phi\cos\theta+\sin\phi\sin\theta\geq 0\}} \frac{e^{-r^2/2}}{2\pi} r dr \\ &= \frac{1}{2\pi}\int_0^{2\pi} 1_{\{\cos\phi\geq 0\}} 1_{\{\cos(\phi-\theta)\geq 0\}} d\phi \\ &= \frac{\pi-\theta}{2\pi}; \end{aligned}$$

draw a picture for the last equality.

Arc-Cosine Kernels: ReLU Case

Proof Again, recall the norm $\|\boldsymbol{y}\|_2 = 1$, to obtain

$$k(\boldsymbol{x}, \boldsymbol{y}) = \int w_1(w_1 \cos\theta + w_2 \sin\theta) 1_{\{w_1 \geq 0\}} 1_{\{w_1 \cos\theta + w_2 \sin\theta \geq 0\}} \frac{e^{-(w_1^2+w_2^2)/2}}{2\pi} dw_1 dw_2$$

where θ is the angle between $\boldsymbol{x}$ and $\boldsymbol{y}$. Finally, write the preceding integral in polar coordinates: $w_1 = r\cos\phi, w_2 = r\cos\phi$, to obtain

$$\begin{aligned}
k(\boldsymbol{x}, \boldsymbol{y}) &= \int_0^{2\pi} d\phi \int_0^{\infty} r^2 \cos\phi \cos(\phi - \theta) 1_{\{\cos\phi \geq 0\}} 1_{\{\cos\phi\cos\theta + \sin\phi\sin\theta \geq 0\}} \frac{e^{-r^2/2}}{2\pi} r dr \\
&= \frac{1}{\pi} \int_0^{2\pi} \frac{\cos\theta + \cos(2\phi - \theta)}{2} 1_{\{\cos\phi \geq 0\}} 1_{\{\cos(\phi-\theta) \geq 0\}} d\phi \\
&= \frac{1}{\pi} \int_{\theta}^{\pi} \frac{\cos\theta + \cos(2\phi - \theta)}{2} d\phi \\
&= \frac{\sin\theta + (\pi - \theta)\cos\theta}{2\pi};
\end{aligned}$$

draw a picture for the last equality.

Generalization to Rotationally Invariant Densities

Tsuchida et al. (2018)

- Rotationally invariant density $f(\boldsymbol{w})$: if for any $\boldsymbol{w}$ and orthogonal matrix R

$$f(\boldsymbol{w}) = f(R\boldsymbol{w}) = g(\|\boldsymbol{w}\|_2)$$

  R is orthogonal if its raws and columns are orthogonal unit vectors

- Rotationally invariant distribution include: Gaussian distribution, the multivariate t-distribution, the symmetric multivariate Laplace distribution, and symmetric multivariate stable distributions.

Proposition 1 (Tsuchida et al. (2018)) The result of Cho&Saul (2009), which corresponds to ReLU, fully generalizes to rotationally invariant densities.

Proposition 4 (Tsuchida et al. (2018)) Extension to Leaky-ReLU for rotationally invariant densities.

Proof: Key Idea

Tsuchida et al. (2018)

Proposition 2 Show that $k(\theta) = K(\boldsymbol{x}, \boldsymbol{y})$ satisfies and initial value problem

$$k''(\theta) + k(\theta) = F(\theta), \quad k'(\pi) = 0, \quad k(\pi) = 0,$$

where $F(\theta)$ is given as an integral.

Proof idea

- Rotate $\boldsymbol{w}$ to get the expression of k in a convenient form
- Then, differentiate twice with respect to θ.

Infinite Depth Networks: Degenerate Kernel

Corollary 8 (Tsuchida et al. (2018)) The normalized kernel converges, as depth grows, to a degenerate fixed point at $\theta^* = 0$.

Proof is based on contraction argument. First the observation that the angle between two inputs at jth level approaches, as the network width $m \to \infty$,

$$\cos\theta_j = \frac{1}{1+a^2}\left(\frac{(1-a)^2}{\pi}(\sin\theta_{j-1} + (\pi - \theta_{j-1})\cos\theta_{j-1}) + 2a\cos\theta_0\right)$$

by taking the derivative of the preceding map with respect to $z = \cos\theta_{j-1}$, it is easy to show that this map is a contraction, and thus the iteration converges to a unique $z^* = \cos\theta^*$. By direct replacement, one finds that $\theta^* = 0$ is that unique solution.

Additional comments

- This confirms the difficulty of trining very deep networks in view of the recent work on "Shattered Gradient Problem" by Balduzzi et al., 2017
- Check references in (Tsuchida et al. (2018)) for prior results of this type: Lee et al. (2017), Schoenholz et al. (2017), Poole et al. (2016) and Daniely (2016)

High-Dimensional Geometry

Before starting another topic, let's spend some time on high-dimensional geometry

- Arguments about high dimensions are often involved, mostly as heuristic, in NNs/ML papers
- But, one has to be careful
  - Usual low dimensional intuition does not work - counterintuitive
    - Volume of n-cube and n-ball is not where you would expect it
    - Many more vectors are orthogonal than you would think
- Better understanding of the high-dimensional geometry can lead to better statistical learning algorithms.

n-Cube

Define n-cube, $C^n(s)$, with side s, which is centered at the origin

$$C^n(s) = \{(x_1, \ldots, x_n) : -s/2 \leq x_i \leq s/2, 1 \leq i \leq n\}$$

Let $C^n = C^n(1)$ be the unit cube.

Lemma The cube C^n has diameter $\sqrt{n}$, but volume 1.

Algebraically this is trivial, but geometrically, this is counterintuitive(!) Try to visualize this. It will get worse.

Where is most of the volume concentrated?

n-Cube: Volume is concentrated in the very thin shell

Lemma For any $t > 0$,

$$\lim_{n\to\infty} \text{Vol}(C^n - C^n(1-t/n)) = \lim_{n\to\infty} (1-(1-t/n)^n) = 1 - e^{-t},$$

i.e., for any large fixed t, as $n \to \infty$, most of the volume is in the very thin shell of with

$$\frac{t}{n}.$$

Example For $t = 3$ and $n = 300$,

$$1 - (1 - 3/300)^{300} \approx 95\%$$

of the volume is in a shell of width 0.01

n-Cube: Volume is concentrated in the middle

Consider a hyperplane that goes through the middle (origin)

$$x_1 + \cdots + x_n = 0$$

Note that this hyperplane is perpendicular to the unit vector

$$\frac{1}{\sqrt{n}}(1, \ldots, 1)$$

Let A be the set of points that are at distance c from this hyperplane

$$A = \{(x_1, \ldots, x_n) : \frac{|x_1 + \cdots + x_n|}{\sqrt{n}} \leq c\}$$

Lemma For any $c > 0$, $\text{Vol}(C^n \cap A) \geq 1 - e^{-\Omega(c^2)}$

Proof We can view this volume as the probability, where $x_i, 1 \leq i \leq n$, are are i.i.d. and uniform on $[0, 1]$. Then apply the CLT.

How can this be?

- Most of the volume is both in the middle & thin shell?

n-Ball: Volume is concentrated in the very thin shell

Define an n-Ball and n-Sphere of radius r

$$B^n(r) = \{(x_1, \ldots, x_n) : x_1^2 + \cdots x_n^2 \leq r^2\}$$
$$S^{n-1}(r) = \{(x_1, \ldots, x_n) : x_1^2 + \cdots x_n^2 = r^2\}$$

Let $B^n = B^n(1)$ and $S^{n-1} = S^{n-1}(1)$ be the unit ball and unit sphere, respectively.

Lemma Most of the volume is in the thin shell of order t/n,

$$\lim_{n\to\infty} \frac{\mathrm{Vol}(B^n) - \mathrm{Vol}(B^n(1-t/n))}{\mathrm{Vol}(B^n)} = 1 - \lim_{n\to\infty}(1-t/n)^n = 1 - e^{-t}$$

Proof Follows from scaling

$$\mathrm{Vol}(B^n(r)) = r^n \mathrm{Vol}(B^n)$$

(Justify this)

Uniform Random Variables on n-Sphere & n-Ball

Let $X_i, i \geq 1$ be independent standard normal, $\mathcal{N}(0,1)$, random variables and let

$$\|\boldsymbol{X}\|_2 = \sqrt{X_1^2 + \cdots X_n^2}$$

Then,

$$\boldsymbol{Y} = \left(\frac{X_1}{\|\boldsymbol{X}\|_2}, \ldots, \frac{X_n}{\|\boldsymbol{X}\|_2} \right)$$

is uniformly distributed on unit sphere S^{n-1}.

Furthermore, if U is a uniform random variable on $[0,1]$, which is independent of $\boldsymbol{X}$, then

$$\boldsymbol{Z} = \left(\frac{U^{1/n} X_1}{\|\boldsymbol{X}\|_2}, \ldots, \frac{U^{1/n} X_n}{\|\boldsymbol{X}\|_2} \right)$$

is uniformly distributed inside the unit ball B^n.

The preceding representations can be used to prove a variety of interesting results.

n-Ball: Volume is in the middle - around the equator

Lemma For any $c > 0$,

$$\frac{\mathsf{Vol}(B^n \cap \{|x_1| \leq c/\sqrt{n}\})}{\mathsf{Vol}(B^n)} \geq 1 - e^{-\Omega(c^2)}$$

Proof We will use the preceding construction of uniform random variables on a sphere. Note first that, with very high probability, $1 - O(e^{-\Omega(n)})$, we can chose a constant γ such that

$$\|\boldsymbol{X}\|_2 \geq \gamma\sqrt{n}$$

Then, for large n,

$$\begin{aligned}\frac{\mathsf{Vol}(B^n \cap \{|x_1| \leq c/\sqrt{n}\})}{\mathsf{Vol}(B^n)} &\geq 1 - \mathbb{P}\left[\frac{|X_1|}{\|\boldsymbol{X}\|_2} > c/\sqrt{n}\right] \\ &\geq 1 - \mathbb{P}[|X_1| > \gamma c] - O(e^{-\Omega(n)}) = 1 - e^{-\Omega(c^2)}\end{aligned}$$

Again, most of the volume is both in the middle (equator) & thin shell?

Randomly selected unit vectors are likely orthogonal

Lemma Two randomly selected unit vectors are orthogonal with very hight probability.

Proof Pick two independent unit vectors from two slides ago

$$\boldsymbol{Y}^i = \left(\frac{X_1^i}{\|\boldsymbol{X}^i\|_2}, \ldots, \frac{X_n^i}{\|\boldsymbol{X}^i\|_2} \right), \quad i = 1, 2.$$

Then, for any $\epsilon > 0$, as $n \to \infty$

$$\mathbb{P}[|\langle \boldsymbol{Y}^1, \boldsymbol{Y}^2 \rangle| > \epsilon] \approx \mathbb{P}[|\sum_j X_j^1 X_j^2| > \gamma \epsilon n] \to 0$$

- What is the probability of this happening in low dimensions: 2 or 3?
- This intuition can be used to explain the random projection dimensionality reduction: see Section 23.2 in [UML] book.

Volume is not monotonic in dimension

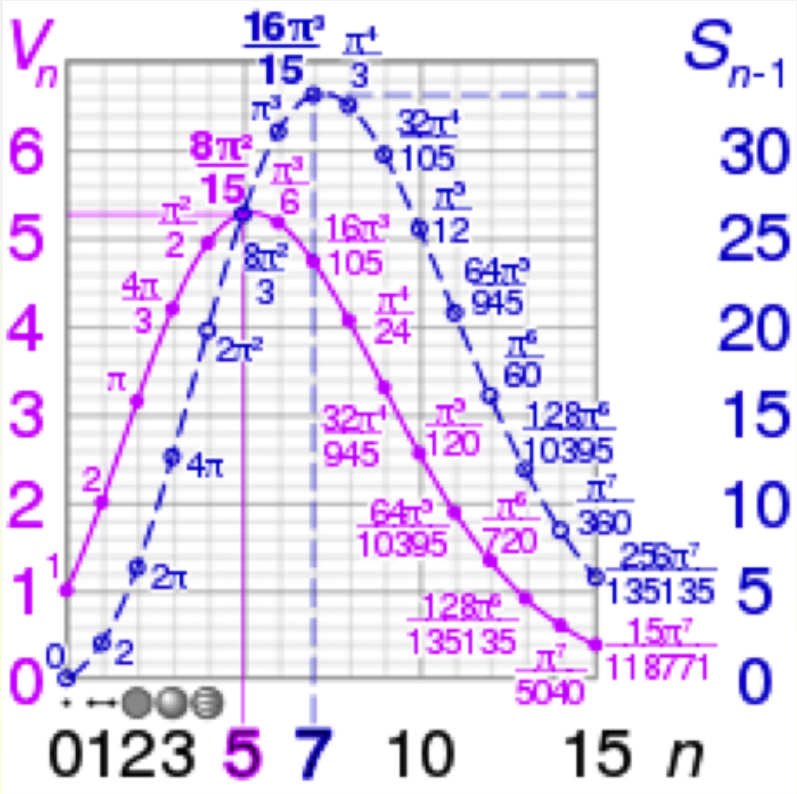
Lemma Volume of B^n is given by

$$\mathrm{Vol}(B^n) = \frac{\pi^{n/2}}{\Gamma(1+n/2)}$$

Proof Prove

$$\mathrm{Vol}(B^n) = \frac{2\pi}{n}\mathrm{Vol}(B^{n-2})$$

n-sphere, Wikipedia



Reading

- Mathematically oriented survey paper on ReLU
  - Neural Net Approximation, by DeVore, Hanin & Petrova, Dec 2020.
- Papers on the new topic: connection between NNs and Kernels
  - Invariance of Weight Distributions in Rectified MLPs, by Tsuchida et al., Jun 2018.
  - Kernel Methods for Deep Learning, by Cho & Saul, 2009.
  - Check the reference lists in these papers for additional readings
- References on high-dimensional geometry
  - SOME GEOMETRY IN HIGH-DIMENSIONAL SPACES, by Flaschka, 2010
  - Chapter 2 of the new book: Foundations of Data Science, by Blum et al., 2017
  - n-sphere, Wikipedia
- Recall books on Kernels: (available online through CU Library)
  1. Chapters 1, 2, 13-16 in: B. Schölkopf and A. J. Smola. *Learning with Kernels.* MIT Press, Cambridge, MA, 2002.
  2. Chapters 1, 5.3, 6, 7 in (this book is mathematically advanced) A. Berlinet and C. Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics.* Kluwer, 2004.

Have Fun!