Deep Learning and Geometry (and Statistics)

김지수 (Jisu KIM)

딥러닝의 통계적 이해 (Deep Learning: Statistical Perspective), 2024년 2학기

This lecture note is very premature and a combination of many different things.

1 Review

1.1 Basic Model for Supervised Learning

- Input(입력) / Covariate(설명 변수) : $x \in \mathbb{R}^d$, so $x = (x_1, \ldots, x_d)$.
- Output(출력) / Response(반응 변수): $y \in \mathcal{Y}$. If y is categorical, then supervised learning is "classification", and if y is continuous, then supervised learning is "regression".
- Model(모형) :

  $$y \approx f(x).$$

  If we include the error ϵ to the model, then it can be also written as

  $$y = \phi(f(x), \epsilon).$$

  For many cases, we assume additive noise, so

  $$y = f(x) + \epsilon.$$

- Assumption(가정): f belongs to a family of functions $\mathcal{M}$. This is the assumption of a model: a model can be still used when the corresponding assumption is not satisfied in your data.
- Loss function(손실 함수): $\ell(y, a)$. A loss function measures the difference between estimated and true values for an instance of data.
- Training data(학습 자료): $\mathcal{T} = \{(y_i, x_i), i = 1, \ldots, n\}$, where (y_i, x_i) is a sample from a probability distribution P_i. For many cases we assume i.i.d., or x_i's are fixed and y_i's are i.i.d..
- Goal(목적): we want to find f that minimizes the expected prediction error,

  $$f^0 = \arg\min_{f \in \mathcal{F}} \mathbb{E}_{(Y,X)\sim P} \left[\ell(Y, f(X))\right].$$

  Here, $\mathcal{F}$ can be different from $\mathcal{M}$; $\mathcal{F}$ can be smaller then $\mathcal{M}$.
- Prediction model(예측 모형): f^0 is unknown, so we estimate f^0 by $\hat{f}$ using data. For many cases we minimizes on the empirical prediction error, that is taking the expectation on the empirical distribution $P_n = \frac{1}{n}\sum_{i=1}^n \delta_{(Y_i,X_i)}$.

  $$\hat{f} = \arg\min_{f \in \mathcal{F}} \mathbb{E}_{P_n} \left[\ell(Y, f(X))\right] = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \ell(Y_i, f(X_i)).$$

- Prediction(예측): if $\hat{f}$ is a predicted function, and x is a new input, then we predict unknown y by $\hat{f}(x)$.

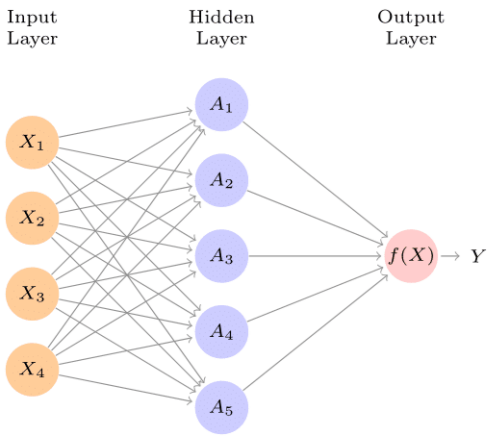

Figure 1: Neural network with a single hidden layer. The hidden layer computes activations $A_j = \sigma_j(x)$ that are nonlinear transformations of linear combinations of the inputs $x_1, \ldots, x_d$. Hence these A_j are not directly observed. The functions σ_j are not fixed in advance, but are learned during the training of the network. The output layer is a linear model that uses these activations A_j as inputs, resulting in a function $f(x)$. Figure 10.1 from [2].

1.2 Two Layer Neural Networks

A two-layer neural network takes an input vector of d variables $x = (x_1, x_2, \ldots, x_d)$ and builds a nonlinear function $f(x)$ to predict the response $y \in \mathbb{R}^D$. What distinguishes neural networks from other nonlinear methods is the particular structure of the model:

$$f(x) = f_\theta(x) = g\left(\beta_0 + \sum_{j=1}^{m} \beta_j \sigma(b_j + w_j^\top x)\right),$$

where $x \in \mathbb{R}^d, b_j \in \mathbb{R}, w_j \in \mathbb{R}^d, \beta_0 \in \mathbb{R}^D, \beta_j \in \mathbb{R}^D$. See Figure 1.

- $\theta = \{[\beta, a_j, b_j, w_j] : j = 1, \ldots, m\}$ denotes the set of model parameters.
- $x_1, \ldots, x_d$ together is called an input layer.
- $A_j \coloneqq \sigma_j(x) = \sigma(b_j + w_j^\top x)$ is called an activation.
- $A_1, \ldots, A_m$ together is called a hidden layer or hidden unit; m is the number of hidden nodes.
- $f(x)$ is called an output layer.
- g is an output function. Examples are:
  - softmax $g_i(x) = \exp(x_i)/\sum_{l=1}^{D} \exp(x_l)$ for classification. The softmax function estimates the conditional probability $g_i(x) = P(y = i|x)$.
  - identity/linear $g(x) = x$ for regression.
  - threshold $g_i(x) = I(x_i > 0)$
- σ is called an activation function. Examples are:
  - sigmoid $\sigma(x) = 1/(1 + e^{-x})$ (see Figure 2)
  - rectified linear (ReLU) $\sigma(x) = \max\{0, x\}$ (see Figure 2)
  - identity/linear $\sigma(x) = x$
  - threshold $\sigma(x) = I(x > 0)$, threshold gives a direct multi-layer extension of the perceptron (as considered by Rosenblatt).

Activation functions in hidden layers are typically nonlinear, otherwise the model collapses to a linear model. So the activations are like derived features - nonlinear transformations of linear combinations of the features.

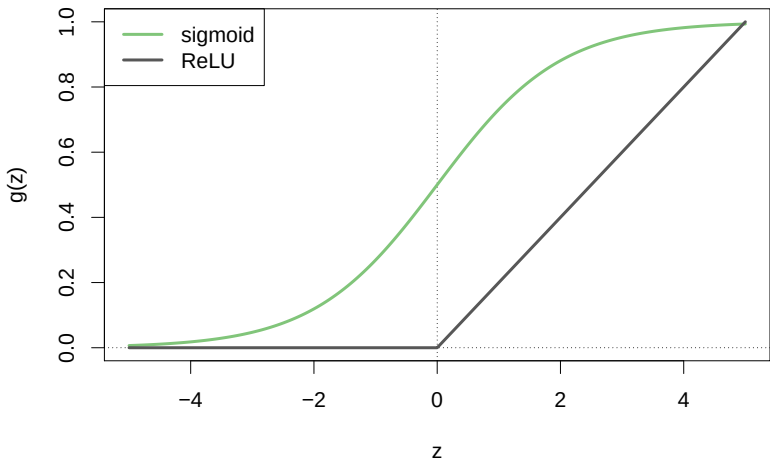

Figure 2: Activation functions. The piecewise-linear ReLU function is popular for its efficiency and computability. We have scaled it down by a factor of five for ease of comparison. Figure 10.2 from [2].

2 Goal

In this note, we explore the connection of geometry and topology to deep learning, in three perspectives.

3 Geometry and Topology as Regular condition

We will see one example from [5]. Suppose a two-layer neural network without bias term:

$$f_\theta(x) = \sum_{j=1}^{m} \beta_j \sigma(w_j^\top x),$$

where σ is a ReLU function. And suppose we use the gradient flow optimization:

$$\frac{d}{dt}\theta(t) = -\nabla_\theta L(\theta(t)).$$

Due to the ReLU function, the loss function L is invariant to the action T_α of rescaling the parameters as $w_j \mapsto \alpha^{-1} w_j$, $\beta_j \mapsto \alpha\beta_j$. Then, the orthogonality of the gradient with respect to the level set of the loss results as

$$\frac{d}{dt}\left(\|w_j\|_2^2 - \|\beta_j\|_2^2\right) = 0, \qquad \forall j,$$

which implies that

$$\|w_j(t)\|_2^2 - \|\beta_j(t)\|_2^2 = c_j, \qquad \forall j = 1, \ldots, m. \tag{1}$$

(1) identifies a manifold $\mathscr{H}(c)$, named invariant set, in the parameter space on which the training trajectory moves. See Figure (3).

Proposition ([5, Lemma 1]). *The invariant set is homeomorphic to a cartesian product of hyperquadrics, one for each hidden neuron:*

$$\mathscr{H}(c) \cong \mathscr{Q}(c_1) \times \cdots \times \mathscr{Q}(c_m),$$

where

$$\mathscr{Q}(c_j) = \left\{(w, \beta) : \|w\|_2^2 - \|\beta\|_2^2 = c_j\right\}.$$

Then we can characterize the topological behavior of the invariant set $\mathscr{H}(c)$ where the optimization path lies on, also see Figure (3).

Theorem ([5, Corollary 2]). *If the output of the neural network is one dimension (in $\mathbb{R}$), its input dimension $d > 1$, and the initial parameter is that $\|w_j(0)\|_2^2 - \|\beta_j(0)\|_2^2 < 0$ for exactly m_- hidden neurons, then the set $\mathscr{H}(c)$ has 2^{m_-} connected components, and in particular disconnected when $m_- \geq 1$.*

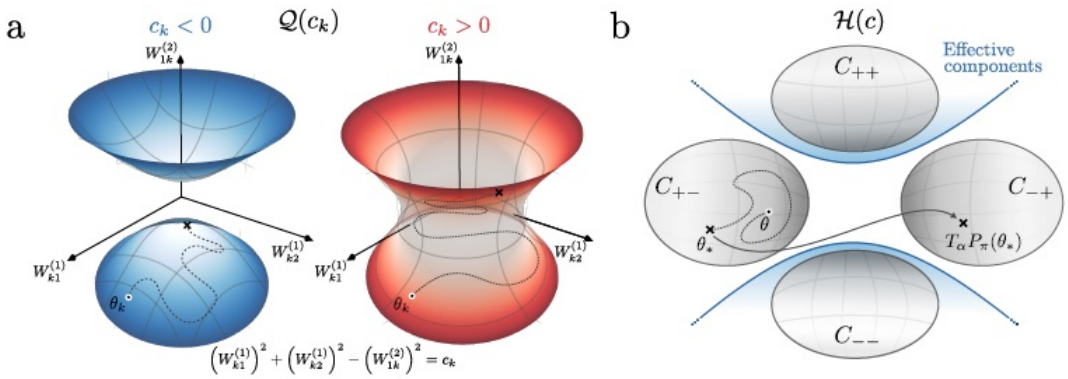

Figure 2: **a.** The invariant hyperquadric $\mathcal{Q}(c_k)$ of a neuron with two inputs ($d = 2$) and one output ($e = 1$) in the cases where $c_k < 0$ (left) and $c_k > 0$ (right). **b.** Depiction of the invariant set $\mathcal{H}(c)$ in the case where $l_- = 2$ so that there are $2^{l_-} = 4$ connected components. $C_{\pm\mp}$ denotes the connected component such that $s = (\pm 1, \mp 1)$. The blue lines separate the different effective components of $\mathcal{H}(c)$.

Figure 3: (a) Illustration of $\mathscr{Q}(c_j)$ when $c_j < 0$ and $c_j > 0$, and (b) Illustration of $\mathscr{H}(c)$. Figure 2 from [5].

Figure 1: Illustration of *Betti numbers* and *reach*. (a) A 2-manifold embedding in $\mathbb{R}^3$ with $\beta_0 = 1, \beta_1 = 0$. (b) A 2-manifold embedded in $\mathbb{R}^3$ with $\beta_0 = 1, \beta_1 = 3$. (c) A 1-manifold with large reach. (d) A 1-manifold with small reach, which is the radius of the dashed circle.

Figure 4: Illustration of Betti numbers and reach, Figure 1 from [7].

This suggests that the optimum might be unreachable from the initialization.
[7] has the following results:

Theorem ([7, Main Theorem]). *Let $\mathcal{M} \subset \mathbb{R}^D$ be a d-dimensional manifold ($d \leq D$). There exists a ReLU network classifier g with depth at most $O\left(\log \beta + \log(\tau^{-1})\right)$ and size at most $O\left(\beta^2 + \tau^{-d^2/2}\right)$, such that with large probability, the true risk of g is small. β is the sum of Betti numbers and τ is the reach of $\mathcal{M}$.*

4 Geometry as Embedding space

I will first start with constant curvature space (though I am not defining what constant curvature means).

4.1 Constant curvature space

- Let $\mathbb{E}^d$ be the usual $\mathbb{R}^d$ with the usual ℓ^2-metric

$$d_{\mathbb{E}^d}(x, y) = \|x - y\|_2 .$$

  This is the space with constant curvature 0.

- Let $\mathbb{S}^d$ be the d-dimensional sphere in $\mathbb{R}^{d+1}$ as

$$\mathbb{S}^d := \left\{x \in \mathbb{R}^{d+1} : \|x\|_2 = 1\right\} .$$

Figure 5: https://web.colby.edu/thegeometricviewpoint/2016/12/21/tessellations-of-the-hyperbolic-plane-and-m-c-escher/

Define the distance $d_{\mathbb{S}^d}$ as

$$d_{\mathbb{S}^d}(x,y) = \arccos\langle x,y\rangle = \arccos\left(\sum_{i=1}^{d} x_i y_i\right) \in [0,\pi].$$

This is the space with constant curvature $+1$.

- Let $\mathbb{H}^d$ be the open ball in $\mathbb{R}^d$ as
$$\mathbb{H}^d := \left\{x \in \mathbb{R}^d : \|x\|_2 < 1\right\}.$$
Define the distance $d_{\mathbb{H}^d}$ as
$$d_{\mathbb{H}^d}(x,y) = \operatorname{arccosh}\left(1 + 2\frac{\|x-y\|_2^2}{\left(1-\|x\|_2^2\right)\left(1-\|y\|_2^2\right)}\right).$$
This is called the Poincare Ball model. This is the space with constant curvature -1. See Figure (5).

Our interest is $\mathbb{H}^d$. This type of space is called hyperbolic space. Hyperbolic space is well-suited for embedding tree-structured data: one key property is its exponential growth of the volume:

$$Vol_{\mathbb{E}^d}(\mathcal{B}_{\mathbb{E}^d}(x,r)) \sim Cr^d, \qquad \text{while} \qquad Vol_{\mathbb{H}^d}(\mathcal{B}_{\mathbb{H}^d}(x,r)) \sim C_1\exp(C_2 r),$$

where for $x \in \mathcal{X}$ and $r > 0$, $\mathcal{B}_{\mathcal{X}}(x,r) = \{y \in \mathcal{X} : d_{\mathcal{X}}(x,y) < r\}$ is the ball in $\mathcal{X}$ centered at x and radius r.

Hence the hyperbolic space naturally reflects the exponential growth of nodes in a tree, which is challenging to capture in Euclidean spaces. The geometry of hyperbolic space allows for compact representations of hierarchical relationships, preserving distances and structural similarities more effectively. This enables embeddings with lower distortion and better scalability for tree-like data, such as taxonomies, knowledge graphs, or biological hierarchies.

Suppose we have a tree data, and we are embedding it into a hyperbolic space. One result in is that those distances are equivalent: then those distances are equivalent:

Theorem ([3, Theorem 1]). *Suppose $\mathcal{T}$ is a tree and $x_1,\ldots,x_n \in \mathcal{T}$. Then we can construct a map $\psi : \mathcal{T} \to \mathbb{H}^d$ so that*

$$d_{\mathcal{T}}(x_i,x_j) \sim d_{\mathbb{H}^d}(\psi(x_i),\psi(x_j)).$$

Some references for embedding tree data into hyperbolic space:

- https://medium.com/@nathan_jf/treerep-and-hyperbolic-embeddings-41312c98b264
- https://meiji163.github.io/post/combo-hyperbolic-embedding/

5 Geometry and Topology as Information

This is a recently growing area in deep learning. Though, there are currently not many connection between utilizing information Geometry/Topology and statistical theory. I suggest two area:

- Geometric Deep Learning: this is a field of deep learning that extends deep learning techniques to non-Euclidean domains such as graphs, manifolds, and other structured data by leveraging their geometric properties and symmetries. It is commonly regarded as an extension of Graph Neural Network, but in fact about more general utilization of geometry and symmetry. Good references to start are:
  - https://geometricdeeplearning.com/
  - Michael M. Bronstein, Joan Bruna, Taco Cohen, Petar Veličković, Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges, 2021. https://arxiv.org/abs/2104.13478
- Topological Data Analysis (TDA): this is a field that uses tools from topology to extract, analyze, and interpret the shape and structure of data, focusing on its topological features like connectedness and holes across multiple scales. TDA usually refers to mapper and persistent homology, and with respect to applications to machine learning or deep learning, it is mostly about persistent homology. Good references to start are:
  - https://jkim82133.github.io/321.621A/2023F/
  - Larry Wasserman, Topological Data Analysis, 2018. https://www.annualreviews.org/doi/abs/10.1146/annurev-statistics-031017-100045/
  - Frédéric Chazal, Bertrand Michel, An introduction to Topological Data Analysis: fundamental and practical aspects for data scientists, 2021. https://www.frontiersin.org/articles/10.3389/frai.2021.667963/
  - Felix Hensel, Michael Moor, Bastian Rieck, A Survey of Topological Machine Learning Methods, 2021. https://doi.org/10.3389/frai.2021.681108

References

[1] Herbert Federer. *Geometric measure theory*. Die Grundlehren der mathematischen Wissenschaften, Band 153. Springer-Verlag New York Inc., New York, 1969.

[2] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning—with applications in R*. Springer Texts in Statistics. Springer, New York, [2021] ©2021. Second edition [of 3100153].

[3] Ya-Wei Eileen Lin, Ronald R. Coifman, Gal Mishne, and Ronen Talmon. Hyperbolic diffusion embedding and distance for hierarchical representation learning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 21003–21025. PMLR, 2023.

[4] Partha Niyogi, Stephen Smale, and Shmuel Weinberger. Finding the homology of submanifolds with high confidence from random samples. *Discrete & Computational Geometry*, 39(1-3):419–441, 2008.

[5] Marco Nurisso, Pierrick Leroy, and Francesco Vaccarino. Topological obstruction to the training of shallow relu neural networks. *CoRR*, abs/2410.14837, 2024.

[6] A. Singer and H.-T. Wu. Vector diffusion maps and the connection Laplacian. *Comm. Pure Appl. Math.*, 65(8):1067–1144, 2012.

[7] Jiachen Yao, Mayank Goswami, and Chao Chen. A theoretical study of neural network expressive power via manifold topology. *CoRR*, abs/2410.16542, 2024.

6 Appendix: Reach

For this lecture note, for $A \subset \mathbb{R}^d$, we use the notation $A^r \coloneqq \left\{x \in \mathbb{R}^d : d(x, A) < r\right\}$ for the r-offset of A.

Definition of reach

First introduced by Federer [1], the reach is a regularity parameter defined as follows. Given a closed subset $A \subset \mathbb{R}^d$, the medial axis of A, denoted by $\mathrm{Med}(A)$, is the subset of $\mathbb{R}^d$ composed of the points that have at least two nearest neighbors on A. Namely, denoting by $d_A(x) = d(x, A) = \inf_{q \in A} \|q - x\|$ the distance function to A,

$$\mathrm{Med}(A) = \left\{ x \in \mathbb{R}^d | \exists q_1 \neq q_2 \in A, ||q_1 - x|| = ||q_2 - x|| = d(x, A) \right\}. \tag{2}$$

The reach of A is then defined as the minimal distance from A to $Med(A)$. See Figure

Definition ([1, 4.1 Definition])**.** The reach of a closed subset $A \subset \mathbb{R}^d$ is defined as

$$\tau_A = \inf_{q \in A} d\left(q, \mathrm{Med}(A)\right) = \inf_{q \in A, x \in \mathrm{Med}(A)} ||q - x||. \tag{3}$$

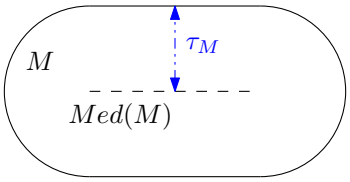

Figure 6: The medial axis of a set M is the set of points that have at least two nearest neighbors on the set M, and the reach is the distance between the set and its medial axis.

Definition. Some authors refer to τ_A^{-1} as the *condition number* [4, 6]. From the definition of the medial axis in (2), the projection $\pi_A(x) = \arg\min_{p \in A} \|p - x\|$ onto A is well defined outside $Med(A)$. The reach is the largest distance $r \geq 0$ such that π_A is well defined on the r-offset. Hence, the reach condition can be seen as a generalization of convexity, since a set $A \subset \mathbb{R}^d$ is convex if and only if $\tau_A = \infty$.

In the case of submanifolds, one can reformulate the definition of the reach in the following manner.

Theorem ([1, Theorem 4.18])**.** *For all submanifold $M \subset \mathbb{R}^d$,*

$$\tau_M = \inf_{q_1 \neq q_2 \in M} \frac{\|q_1 - q_2\|_2^2}{2d(q_2 - q_1, T_{q_1} M)}. \tag{4}$$

This formulation has the advantage of involving only points on M and its tangent spaces, while (3) uses the distance to the medial axis $Med(M)$, which is a global quantity.

The ratio appearing in (4) can be interpreted geometrically, as suggested in Figure 7. This ratio is the radius of an ambient ball, tangent to M at q_1 and passing through q_2.

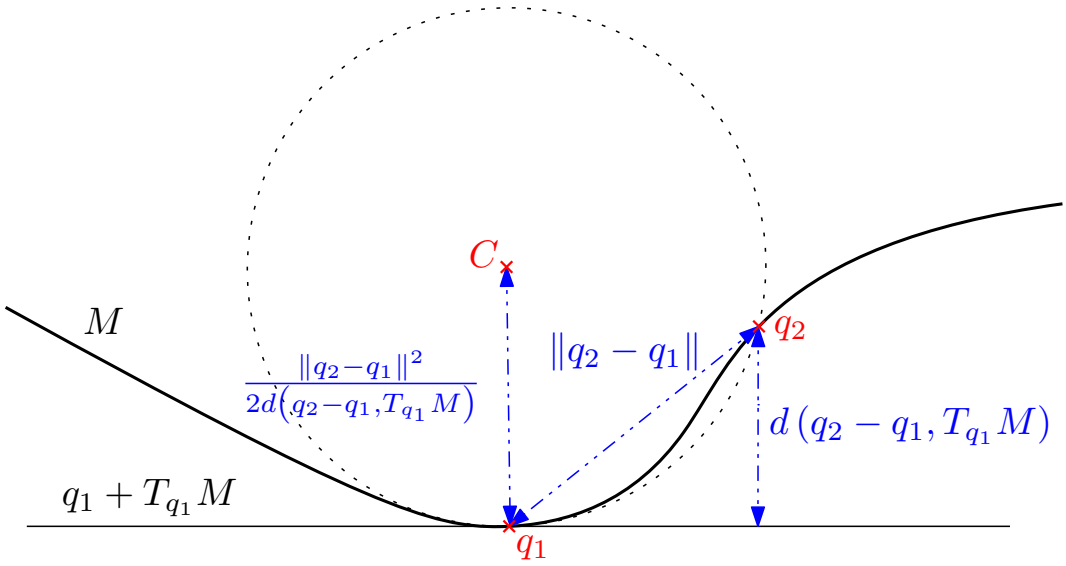

Figure 7: Geometric interpretation of quantities involved in (4).