Basics of Neural Networks

Junhyeok Byeon

Seoul National University

giugi2486@snu.ac.kr

March 31, 2022

ML as a function approximation

- Consider a phenomenon with empirical data: it takes inputs x_i and outputs y_i.
  - Determine whether a mail is spam mail or not.
  - Translating Korean to English and vice versa.
  - Get an image of some number and interpret its digit $(0,1,\cdots,9)$.

- Then, machine learning(ML) deals with the following question.

> Given a new data x_{new}, can we predict its outcome y_{new},
> based on empirical data $(\{x_i, y_i\})$?

- Suppose that that there exits an *underlying function* f such that

$$y_i = f(x_i) + \text{some error}.$$

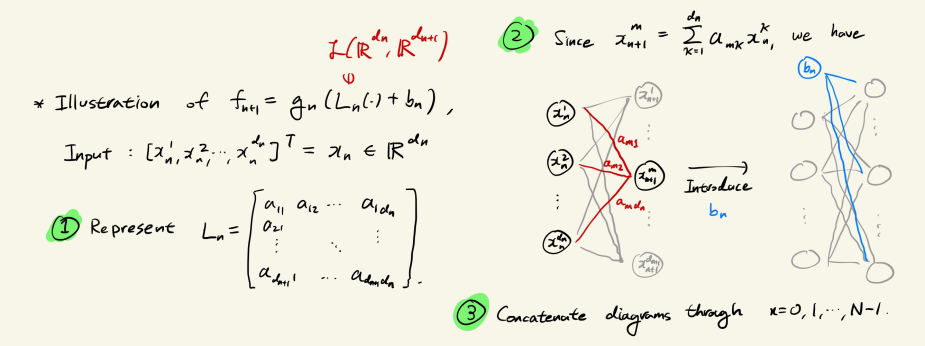
- Then, ML might be understood as a theory of finding suitable f subject to $(\{x_i, y_i\})$!

ML as a function approximation

- We approximate f by iterated composition of some 'good functions' and affine functions: [1] [2]

$$f_{n+1} = g_n(L_n \circ f_n + b_n),\ f_0 = \mathrm{Id},\ n = 0, 1, \cdots, N-1 \quad \Rightarrow \quad f_N \cong f? \tag{0.1}$$

Here f_N, equivalently $\{(g_n, L_n, b_n)\}_{n=0,\cdots,N-1}$, is called *N-layers Neural network(NN)*.



---

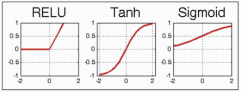
[1] $f : \mathbb{R}^n \to \mathbb{R}^m$ is called affine if it is of the form $L + C$ where $f : \mathbb{R}^n \to \mathbb{R}^m$ is linear and C is a constant.

[2] Here $g : \mathbb{R} \to \mathbb{R}$ is applied elementwise, e.g. $g([x, y]^T) := [g(x), g(y)]^T$.

ML as a function approximation

- Aforementioned good functions g_n are called *'activation function'*. Typical examples are:

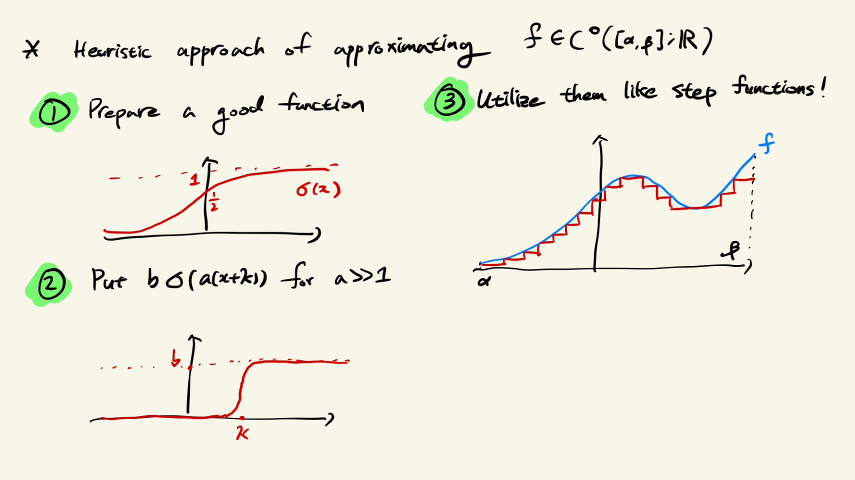
$$\mathrm{ReLU}(z) = \max(0, z), \qquad \tanh z = \frac{e^z - e^{-z}}{e^z + e^{-z}}, \qquad \sigma(z) = \frac{1}{1 + e^{-z}}.$$



- However, can f_N really can approximate various kind of f? In mathematical term:

For all $\varepsilon > 0$, is there N and $\{(g_n, L_n, b_n)\}_{n=0,\cdots,N-1}$ such that $\|f_N - f\| < \varepsilon$?

To get a grip, we may consider the following heuristic approach:



Some TMI: universal function approximation theorem

Theorem (Universal approximation Theorem, 2-layers NN ver.)

Let $\Omega \in \mathbb{R}^d$ be compact and $\sigma : \mathbb{R} \to \mathbb{R}$ be a 'good'(specified later) function. Consider a collection of functions of the following form:

$$f_\theta(x) = \sum_{i=1}^{N} u_i \sigma(\langle a_i, x\rangle + b_i),$$

which is parametrized by $\theta \in \Theta_N := \{(a_i, b_i, u_i)\}_{i=1,\cdots,N} \cong \mathbb{R}^{dN+2N}$. Then the class of functions

$$\bigcup_{N\in\mathbb{N}} \{f_\theta\}_{\theta\in\Theta_N}$$

is dense in $(C^0(\Omega), \|\cdot\|_\infty)$.

The meaning of 'good' σ is as follows:

$$\left[\mu \in \mathcal{M}(\Omega) \text{ such that } \int_\Omega \sigma(\langle a, x\rangle + b)d\mu(x) = 0 \text{ for all } a, b\right] \implies \mu = 0.$$

Fact. The sigmoid function is a good function.

Proof.

- Let $\mathcal{S} := \text{span}(\{\sigma(\langle a_i, x\rangle + b_i)\}_{a\in\mathbb{R}^d, b\in\mathbb{R}})$. Suppose $\text{clos}(\mathcal{S}) \neq C^0(\Omega)$.

- Pick $0 \neq g \in C^0(\Omega)\backslash\text{clos}(\mathcal{S})$ and define a bounded linear functional

$$L : \text{clos}(\mathcal{S}) \oplus \text{span}(g) \to \mathbb{R}, \quad L[s + \lambda g] = \lambda, \quad \forall s \in \text{clos}(\mathcal{S}), \lambda \in \mathbb{R}.$$

- Use the Hahn-Banach theorem to extend L to $\bar{L} : C^0(\Omega) \to \mathbb{R}$. Then by the Riesz Representation Theorem, $\bar{L}$ is represented by some nonzero measure $\mu_{\bar{L}}$:

$$\bar{L}(h) = \int_\Omega h d\mu_{\bar{L}}.$$

- However, since $\bar{L} = 0$ on $\text{clos}(\mathcal{S})$ and σ is good, we have $\mu_{\bar{L}} = 0$, a contradiction. □

Remark. The proof is not constructive; it does not allow us to construct explicit NN.

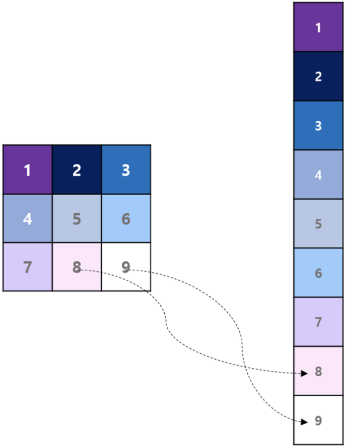
Backpropagation

- Therefore, one may expect f to be approximated by some good NN!
- But.. how can we find such NN?
- Recall: N-layers NN is characterized by $\{(g_n, L_n, b_n)\}_{n=0,\cdots,N-1}$.
- Observation: For fixed N and g_n, f_N is characterized by finite number of parameters!
- Thus, the problem of finding f reduce to the following optimization problem:

  Minimize an error function, defined on $\{(L_1, b_1, \cdots, L_N, b_N)\} \cong \mathbb{R}^D$.

- (Stochastic) Gradient method is used, and it is called backpropagation in NN context.

For special kind of weighted sum, NN is called as a *Convolutional Neural Network*.

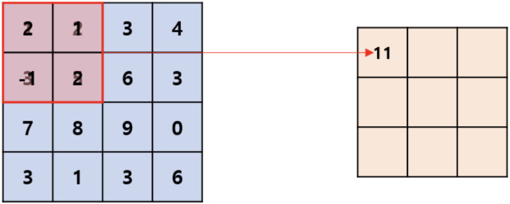


Convolutional Neural Network

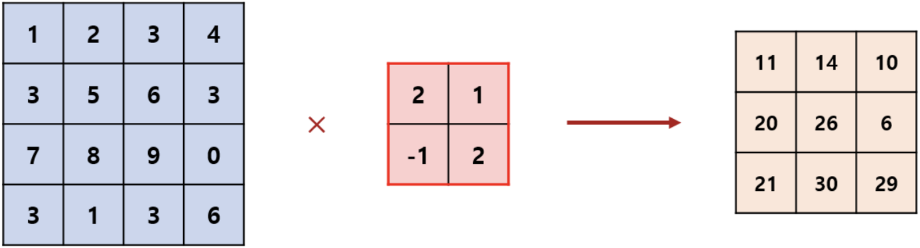
1	2	3	4
3	5	6	3
7	8	9	0
3	1	3	6

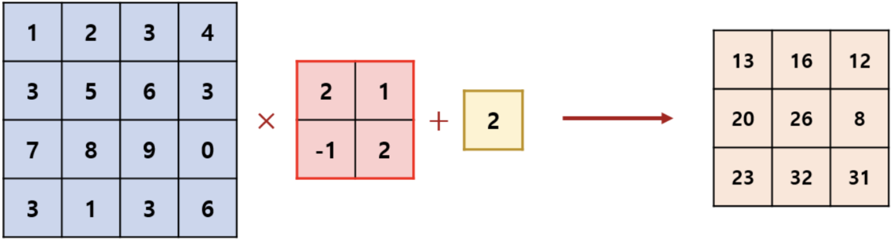
×

2	1
-1	2



$$(1 \times 2) + (2 \times 1) + (-1 \times 3) + (5 \times 2) = 11$$


1 2 3 4
3 5 6 3
7 8 9 0
3 1 3 6
×
2 1
-1 2
11 14 10
20 26 6
21 30 29


1 2 3 4
3 5 6 3
7 8 9 0
3 1 3 6
×
2 1
-1 2
+
2
13 16 12
20 26 8
23 32 31

The End