Lecture 3: Feed-Forward Neural Networks and Expressivity

1. Why Must the Activation Be Nonlinear?

Consider a shallow scalar network

$$f(x) = b^1 + \sum_{j=0}^{d_0-1} w_j^1 \sigma(w_j^0 x + b_j^0).$$

Suppose the activation function is affine:

$$\sigma(y) = ay + c.$$

Then

$$\begin{aligned} f(x) &= b^1 + \sum_{j=0}^{d_0-1} w_j^1 \left[a(w_j^0 x + b_j^0) + c\right] \\ &= \left(\sum_{j=0}^{d_0-1} a w_j^1 w_j^0\right) x + \left[b^1 + \sum_{j=0}^{d_0-1} w_j^1 (a b_j^0 + c)\right]. \end{aligned}$$

Define

$$\widetilde{a} := \sum_{j=0}^{d_0-1} a w_j^1 w_j^0$$

and

$$\widetilde{c} := b^1 + \sum_{j=0}^{d_0-1} w_j^1 (a b_j^0 + c).$$

Then

$$\boxed{f(x) = \widetilde{a}x + \widetilde{c}.}$$

Thus all of the extra parameters have produced only another affine function.

Without nonlinearity, the network has very limited expressivity.

2. ReLU and Piecewise-Linear Functions

Theorem. *Let $\sigma : \mathbb{R} \to \mathbb{R}$ be the ReLU function*

$$\sigma(x) = \mathrm{ReLU}(x) = \max\{0, x\}.$$

Then every continuous piecewise linear function $f : \mathbb{R} \to \mathbb{R}$ of the form

$$f(x) = \begin{cases} a_0 x + b_0, & x \le \tau_1, \\ a_1 x + b_1, & \tau_1 \le x \le \tau_2, \\ \vdots & \\ a_{n-1} x + b_{n-1}, & \tau_{n-1} \le x \le \tau_n, \\ a_n x + b_n, & \tau_n \le x, \end{cases}$$

where

$$\tau_1 < \tau_2 < \cdots < \tau_n,$$

and $a_0, \ldots, a_n$ *and* $b_0, \ldots, b_n$ *are real numbers such that* f *is continuous, can be expressed by a shallow ReLU FNN whose single hidden layer contains* $n + 2$ *neurons.*

In other words, every continuous piecewise linear function $f : \mathbb{R} \to \mathbb{R}$ *whose slope changes finitely many times can be represented by a shallow ReLU FNN.*

Two useful identities are

$$\boxed{\mathrm{ReLU}(\gamma x) = \gamma\,\mathrm{ReLU}(x), \qquad \gamma > 0,}$$

and

$$\boxed{x = \mathrm{ReLU}(x) - \mathrm{ReLU}(-x).}$$

Then the piecewise linear function $f(x)$ can be written as

$$\boxed{f(x) = ReLU(a_0 x + b_0) - ReLU(-a_0 x - b_0) + \sum_{j=1}^{n} (a_j - a_{j-1})\,\mathrm{ReLU}(x - \tau_j).}$$

Interpretation:

- $a_0 x + b_0$ gives the initial line;
- $\mathrm{ReLU}(x - \tau_j)$ turns on at $x = \tau_j$;
- $a_j - a_{j-1}$ changes the slope at τ_j.

Hence a shallow ReLU network can represent a very rich class of piecewise-linear functions.

3. Why Depth?

Theorem. *Define the function* $g : \mathbb{R}^2 \to \mathbb{R}$ *by*

$$g(x_0, x_1) = \min\{0, \max\{x_0, x_1\}\}.$$

This function g *cannot be generated by a shallow ReLU FNN, but* g *can be obtained as a depth* $L = 2$ *ReLU FNN.*

The important message is

$$\boxed{\text{Depth can increase expressivity.}}$$

Proof. Part 1: shallow FNN cannot represent g.

We first recall a simple geometric property of shallow ReLU networks.

In one dimension, a shallow ReLU network has the form

$$f(x) = \sum_{j=1}^{m} c_j\,\mathrm{ReLU}(w_j x + b_j).$$

Each ReLU term has a turning point at

$$w_j x + b_j = 0.$$

Thus, adding ReLU neurons amounts to adding turning points.

The same idea extends to two dimensions. A shallow 2D ReLU network has the form

$$f(x_0, x_1) = \sum_{j=1}^{m} a_j \,\mathrm{ReLU}(w_{j0} x_0 + w_{j1} x_1 + b_j),$$

where the turning points, also called kinks, occur at the (x_0, x_1) values for which the input to the ReLU satisfies

$$w_{j0} x_0 + w_{j1} x_1 + b_j = 0.$$

In two dimensions, these kinks form lines.

Each ReLU neuron can create a kink along the line

$$w_{j0} x_0 + w_{j1} x_1 + b_j = 0.$$

Across this line, the gradient/slope of the neuron changes by a constant vector.

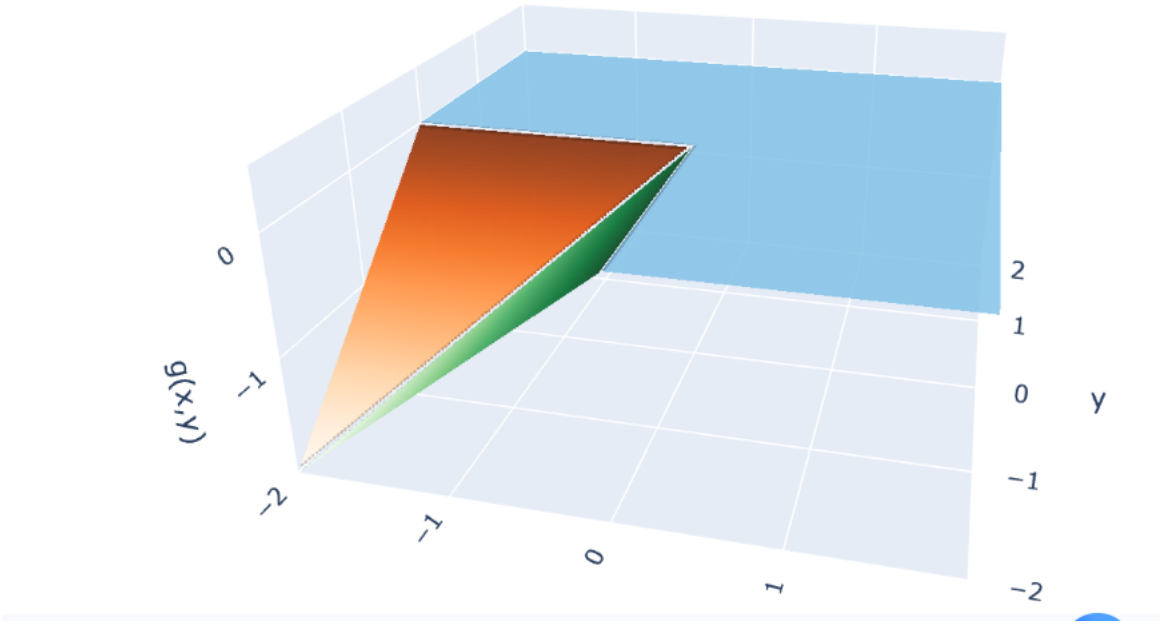
In particular, a shallow ReLU network cannot create a kink along only half of a line.

Now consider

$$g(x_0, x_1) = \min\{0, \max\{x_0, x_1\}\}.$$

we have

$$g(x_0, x_1) = \begin{cases} x_0, & x_1 \leq x_0 < 0, \\ x_1, & x_0 < x_1 < 0, \\ 0, & \max x_0, x_1 >= 0 \end{cases}$$



As shown in the figure, the kink of g consists of only half of a line: it appears on the negative half-axis, but not on the positive half-axis. Therefore g cannot be represented by a shallow ReLU network .

Part 2: deep FNN can represent g. Use the identity

$$\max\{x_0, x_1\} = x_0 + \mathrm{ReLU}(x_1 - x_0).$$

Also,

$$\min\{0, z\} = -\mathrm{ReLU}(-z).$$

Therefore

$$\begin{aligned} g(x_0, x_1) &= -\mathrm{ReLU}\left(-\max\{x_0, x_1\}\right) \\ &= -\mathrm{ReLU}\left(-x_0 - \mathrm{ReLU}(x_1 - x_0)\right). \end{aligned}$$

This gives a depth-2 ReLU representation. □