Lecture 2: Layers, Feed-Forward Networks, and Parameter Count

Main goal. Understand matrix notation for layers of neurons, how layers compose into feed-forward networks, and how to count network parameters.

1. Entrywise Activation

Recall:

$$\eta(x) = \sigma(\langle x, w\rangle + b)\,.$$

If

$$x = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{pmatrix},$$

we extend σ to vectors entrywise:

$$\boxed{\sigma(x) = \begin{pmatrix} \sigma(x_0) \\ \sigma(x_1) \\ \vdots \\ \sigma(x_{N-1}) \end{pmatrix}.}$$

This convention will allow us to write

$$\sigma(Wx + b).$$

2. Matrix Notation

Let

$$W \in \mathbb{R}^{N\times d}.$$

Its (j,k) entry is denoted by

$$W_{j,k}.$$

The jth row is

$$W_{j,:} \in \mathbb{R}^d,$$

and the kth column is

$$W_{:,k} \in \mathbb{R}^N.$$

We may write a matrix in terms of its columns:

$$W = \begin{pmatrix} | & & | \\ w_0 & \cdots & w_{d-1} \\ | & & | \end{pmatrix}.$$

The transpose is

$$W^T \in \mathbb{R}^{d\times N}.$$

If

$$W \in \mathbb{R}^{N\times d}, \qquad x \in \mathbb{R}^d,$$

then

$$Wx \in \mathbb{R}^N,$$

with

$$\boxed{(Wx)_j = \sum_{k=0}^{d-1} W_{j,k}x_k.}$$

Thus a matrix represents a linear map

$$\boxed{W : \mathbb{R}^d \to \mathbb{R}^N.}$$

3. A Matrix as Many Linear Functions

Suppose

$$W \in \mathbb{R}^{N\times d}.$$

Then

$$W = \begin{pmatrix} - & W_{0,:} & - \\ - & W_{1,:} & - \\ & \vdots & \\ - & W_{N-1,:} & - \end{pmatrix}.$$

Hence

$$Wx = \begin{pmatrix} \langle W_{0,:}, x\rangle \\ \langle W_{1,:}, x\rangle \\ \vdots \\ \langle W_{N-1,:}, x\rangle \end{pmatrix}.$$

Therefore,

$$\boxed{\text{one matrix-vector product} = \text{many inner products at once.}}$$

A useful identity is

$$\boxed{\langle x, Wy\rangle = \left\langle W^T x, y\right\rangle.}$$

4. A Layer of Neurons

Suppose we have d neurons:

$$\eta_j(x) = \sigma(\langle x, w_j\rangle + b_j), \qquad j = 0, \ldots, d-1.$$

Stack their outputs:

$$\ell(x) = \begin{pmatrix} \sigma(\langle x, w_0\rangle + b_0) \\ \sigma(\langle x, w_1\rangle + b_1) \\ \vdots \\ \sigma(\langle x, w_{d-1}\rangle + b_{d-1}) \end{pmatrix}.$$

Define

$$W = \begin{pmatrix} - & w_0^T & - \\ - & w_1^T & - \\ & \vdots & \\ - & w_{d-1}^T & - \end{pmatrix} \in \mathbb{R}^{d \times N}$$

and

$$b = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{d-1} \end{pmatrix} \in \mathbb{R}^d.$$

Then

$$Wx + b = \begin{pmatrix} \langle x, w_0 \rangle + b_0 \\ \langle x, w_1 \rangle + b_1 \\ \vdots \\ \langle x, w_{d-1} \rangle + b_{d-1} \end{pmatrix}.$$

Therefore,

$$\boxed{\ell(x) = \sigma(Wx + b).}$$

This is the key formula for a layer of neurons.

5. Affine Maps and Composition

Define the affine map

$$A : \mathbb{R}^N \to \mathbb{R}^d$$

by

$$A(x) = Wx + b.$$

Then

$$\boxed{\ell = \sigma \circ A.}$$

Equivalently, define the augmented matrix

$$\widetilde{A} = (W \mid b) \in \mathbb{R}^{d \times (N+1)}.$$

Then

$$A(x) = \widetilde{A} \begin{pmatrix} x \\ 1 \end{pmatrix}.$$

Thus a neural-network layer is simply

$$\boxed{\text{affine map} \quad + \quad \text{entrywise nonlinearity.}}$$

6. Concrete Example

Take

$$W = \begin{pmatrix} 1 & -1 \\ 2 & 1 \end{pmatrix}, \qquad b = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad x = \begin{pmatrix} 2 \\ 1 \end{pmatrix}.$$

Then

$$Wx + b = \begin{pmatrix} 1 & -1 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 5 \end{pmatrix}.$$

Using ReLU,

$$\ell(x) = \begin{pmatrix} 2 \\ 5 \end{pmatrix}.$$

If instead

$$x = \begin{pmatrix} 0 \\ 2 \end{pmatrix},$$

then

$$Wx + b = \begin{pmatrix} -1 \\ 2 \end{pmatrix},$$

so

$$\ell(x) = \begin{pmatrix} 0 \\ 2 \end{pmatrix}.$$

End with

$$\boxed{x \longrightarrow \sigma(W_0 x + b_0) \longrightarrow \sigma(W_1 x + b_1) \longrightarrow \cdots}$$

Question:

$$\boxed{\text{What happens if we stack entire layers?}}$$

Lecture 2 (continued): Feed-Forward Networks and Parameter Count

Continue from layers to full feed-forward networks, dimensions, and parameter counting.

7. From One Layer to a Network

A single layer has the form

$$\ell_0(x) = \sigma(A_0(x)).$$

A second layer has the form

$$\ell_1(y) = \sigma(A_1(y)).$$

Thus

$$\ell_1(\ell_0(x)) = \sigma\left(A_1\left(\sigma(A_0(x))\right)\right).$$

In general, a feed-forward neural network has the form

$$\boxed{f = A_L \circ \sigma \circ A_{L-1} \circ \sigma \circ \cdots \circ \sigma \circ A_0.}$$

Each affine function is

$$A_j(y) = W_j y + b_j.$$

The network has:

- an input layer,
- hidden layers,
- a final affine output layer.

8. Dimensions and a Shallow Network

Consider

$$f : \mathbb{R}^3 \to \mathbb{R}^2$$

with one hidden layer of width 2.

Then

$$W_0 \in \mathbb{R}^{2\times 3}, \qquad b_0 \in \mathbb{R}^2,$$

and

$$W_1 \in \mathbb{R}^{2\times 2}, \qquad b_1 \in \mathbb{R}^2.$$

The network is

$$\boxed{f(x) = W_1\sigma(W_0 x + b_0) + b_1.}$$

Schematically,

$$\boxed{3 \text{ inputs} \longrightarrow 2 \text{ hidden neurons} \longrightarrow 2 \text{ outputs.}}$$

Parameter count:

$$W_0 : \ 2 \times 3 = 6, \qquad b_0 : \ 2,$$
$$W_1 : \ 2 \times 2 = 4, \qquad b_1 : \ 2.$$

Thus

$$\boxed{6 + 2 + 4 + 2 = 14}$$

parameters.

9. General Parameter Count

Suppose the input dimension is N and the layer widths are

$$d_0, d_1, \ldots, d_L.$$

Then

$$W_0 \in \mathbb{R}^{d_0 \times N}, \qquad b_0 \in \mathbb{R}^{d_0},$$

and for $j \geq 1$,

$$W_j \in \mathbb{R}^{d_j \times d_{j-1}}, \qquad b_j \in \mathbb{R}^{d_j}.$$

Therefore

$$\boxed{\#\text{parameters} = d_0(N+1) + \sum_{j=1}^{L} d_j(d_{j-1}+1).}$$

Indeed,

$$d_j d_{j-1} = \text{number of weights},$$

while

$$d_j = \text{number of biases}.$$

Thus,

$$\boxed{\text{large networks} \Longrightarrow \text{many large matrices.}}$$

This is one reason linear algebra becomes essential.