Lecture Notes:

Building a Neural Network From Scratch

Introduction:

- I'll assume an understanding of fundamental math concepts, like multivariable calculus & linear algebra, as well as a general understanding of the structure of neural networks, and programming.
- What I want to focus on is the precise mathematical formulation of a simple feedforward neural network, and then translate that into an actual model in Python.
- If you're shaky on some of the fundamental concepts, that's OK - you'll build an intuition moving forward.

Talk about problems

Implementation Architecture:

Before starting anything, look at data

Steps in Machine Learning:

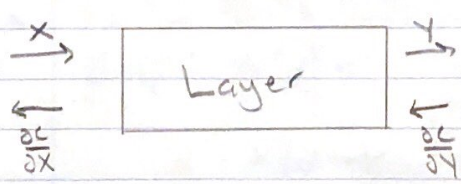
① Data → Model

② Compute the distance between desired and actual output

③ Adjust parameters of model

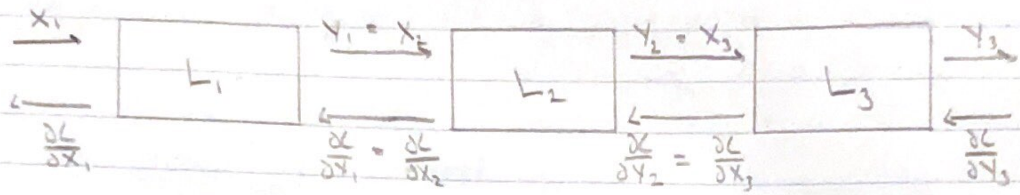
→ Repeat

- Goal is modular code ⇒ implement every layer separately



- With this design, we can think of each layer as an individual object

- Sequential Model: Output of one layer is input of the next

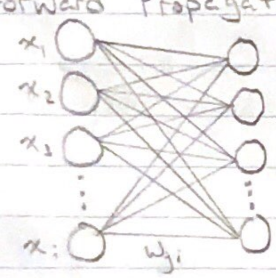


- This describes our Forward propagation and backward propagation algorithms

→ Implement Base Layer

Dense Layer: aka "Fully Connected"
→ Each input node is connected to each output node

Forward Propagation:



$y_1 = w_{11}x_1 + w_{12}x_2 + \dots + w_{1i}x_i + b_1$

$y_2 = w_{21}x_1 + w_{22}x_2 + \dots + w_{2i}x_i + b_2$

$y_3 = w_{31}x_1 + w_{32}x_2 + \dots + w_{3i}x_i + b_3$

$y_j = w_{j1}x_1 + w_{j2}x_2 + \dots + w_{ji}x_i + b_j$

⇒ Matrix Equation: $Y = WX + b$

$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_j \end{bmatrix}}_{j \times 1} = \underbrace{\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1i} \\ w_{21} & w_{22} & \cdots & w_{2i} \\ \vdots & \vdots & \ddots & \vdots \\ w_{j1} & w_{j2} & \cdots & w_{ji} \end{bmatrix}}_{j \times i} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \end{bmatrix}}_{i \times 1} + \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_j \end{bmatrix}}_{j \times 1}$$

Back propagation:

- Given a cost function C, we want to know how to change the parameters to reduce the error

$\Rightarrow$ Gradient Descent

Compute $\frac{\partial L}{\partial Y}$, we want $\frac{\partial L}{\partial W}$, $\frac{\partial L}{\partial b}$, and $\frac{\partial L}{\partial X}$

$$\underset{j \times 1}{\frac{\partial L}{\partial Y} = \begin{bmatrix} \frac{\partial L}{\partial y_1} \\ \frac{\partial L}{\partial y_2} \\ \vdots \\ \frac{\partial L}{\partial y_j} \end{bmatrix}} \xrightarrow{\text{We want:}} \underset{j \times i}{\frac{\partial L}{\partial W} = \begin{bmatrix} \frac{\partial L}{\partial w_{11}} & \frac{\partial L}{\partial w_{12}} & \cdots & \frac{\partial L}{\partial w_{1i}} \\ \frac{\partial L}{\partial w_{21}} & \frac{\partial L}{\partial w_{22}} & \cdots & \frac{\partial L}{\partial w_{2i}} \\ \vdots & \vdots & & \vdots \\ \frac{\partial L}{\partial w_{j1}} & \frac{\partial L}{\partial w_{j2}} & \cdots & \frac{\partial L}{\partial w_{ji}} \end{bmatrix}}$$

Using chain rule: $\frac{\partial L}{\partial W} = \frac{\partial L}{\partial Y} \cdot \frac{\partial Y}{\partial W}$

e.g. $$\frac{\partial L}{\partial w_{12}} = \frac{\partial L}{\partial y_1} \underbrace{\frac{\partial y_1}{\partial w_{12}}}_{= x_2} + \underbrace{\frac{\partial L}{\partial y_2} \frac{\partial y_2}{\partial w_{12}} + \cdots + \frac{\partial L}{\partial y_j} \frac{\partial y_j}{\partial w_{12}}}_{\text{these } = 0!}$$

$$\Rightarrow \frac{\partial L}{\partial w_{ji}} = \frac{\partial L}{\partial y_j} x_i$$

So, $$\frac{\partial L}{\partial W} = \underset{j \times i}{\begin{bmatrix} \frac{\partial L}{\partial y_1} x_1 & \frac{\partial L}{\partial y_1} x_2 & \cdots & \frac{\partial L}{\partial y_1} x_i \\ \frac{\partial L}{\partial y_2} x_1 & \frac{\partial L}{\partial y_2} x_2 & \cdots & \frac{\partial L}{\partial y_2} x_i \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial L}{\partial y_j} x_1 & \frac{\partial L}{\partial y_j} x_2 & \cdots & \frac{\partial L}{\partial y_j} x_i \end{bmatrix}} = \underset{j \times 1}{\frac{\partial L}{\partial Y}} \underset{1 \times i}{X^T}$$

$$\frac{\partial L}{\partial b} = \begin{bmatrix} \frac{\partial L}{\partial b_1} \\ \frac{\partial L}{\partial b_2} \\ \vdots \\ \frac{\partial L}{\partial b_j} \end{bmatrix}_{j \times 1}$$

ex) $\frac{\partial L}{\partial b_1} = \frac{\partial L}{\partial y_1} \underbrace{\frac{\partial y_1}{\partial b_1}}_{=1} + \underbrace{\frac{\partial L}{\partial y_2} \frac{\partial y_2}{\partial b_1} + \dots + \frac{\partial L}{\partial y_j} \frac{\partial y_j}{\partial b_1}}_{=0}$

Thus, $\frac{\partial L}{\partial b} = \frac{\partial L}{\partial Y}$

Finally, we need $\frac{\partial L}{\partial X}$ to pass back to previous input

$$\frac{\partial L}{\partial X} = \begin{bmatrix} \frac{\partial L}{\partial x_1} \\ \frac{\partial L}{\partial x_2} \\ \vdots \\ \frac{\partial L}{\partial x_i} \end{bmatrix}_{i \times 1}$$

ex) $\frac{\partial L}{\partial x_1} = \frac{\partial L}{\partial y_1} \frac{\partial y_1}{\partial x_1} + \frac{\partial L}{\partial y_2} \frac{\partial y_2}{\partial x_1} + \dots + \frac{\partial L}{\partial y_j} \frac{\partial y_j}{\partial x_1}$

$= \frac{\partial L}{\partial y_1} w_{11} + \frac{\partial L}{\partial y_2} w_{21} + \dots + \frac{\partial L}{\partial y_j} w_{j1}$

So, $\underset{i \times 1}{\frac{\partial L}{\partial X}} = \underset{i \times j}{W^T} \underset{j \times 1}{\frac{\partial L}{\partial Y}}$

Activation Layer:

Computing our activation function element-wise

$x_1 \longrightarrow \bigcirc \longrightarrow y_1 = F(\ )$

$x_2 \longrightarrow \bigcirc \longrightarrow y_2 = F(\ )$

$x_3 \longrightarrow \bigcirc \longrightarrow y_3 = F(\ ) \quad \Rightarrow \quad Y = F(X)$

$\vdots$

$x_i \longrightarrow \bigcirc \longrightarrow y_i = F(\ )$

$$\frac{\partial L}{\partial Y} = \underset{i \times 1}{\begin{bmatrix} \frac{\partial L}{\partial y_1} \\ \frac{\partial L}{\partial y_2} \\ \vdots \\ \frac{\partial L}{\partial y_i} \end{bmatrix}} \xrightarrow{\text{We want}} \frac{\partial L}{\partial X} = \underset{i \times 1}{\begin{bmatrix} \frac{\partial L}{\partial x_1} \\ \frac{\partial L}{\partial x_2} \\ \vdots \\ \frac{\partial L}{\partial x_i} \end{bmatrix}}$$

ex) $$\frac{\partial L}{\partial x_1} = \frac{\partial L}{\partial y_1}\frac{\partial y_1}{\partial x_1} + \underbrace{\frac{\partial L}{\partial y_2}\frac{\partial y_2}{\partial x_1}}_{=0} + \dots + \underbrace{\frac{\partial L}{\partial y_i}\frac{\partial y_i}{\partial x_1}}_{=0}$$

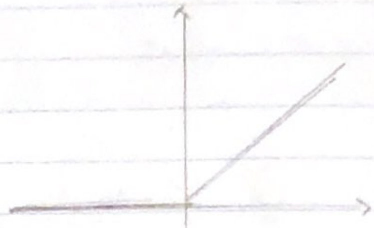
$$\Rightarrow \frac{\partial L}{\partial x_1} = \frac{\partial L}{\partial y_1} \cdot F'(x_1); \quad \text{So,} \quad \frac{\partial L}{\partial X} = \frac{\partial L}{\partial Y} \odot F'(X)$$

Hadamard Product,
i.e., element-wise multiplication

* Note: We need a nonlinear activation function, otherwise our network just computes a linear combination of the input.

ReLU: $g(x) = \max(0, x)$

$$g'(x) = \begin{cases} 0, & x \leq 0 \\ 1, & \text{else} \end{cases}$$



Softmax: $$y_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$

$$\frac{\partial L}{\partial x_k} = \frac{\partial L}{\partial y_1}\frac{\partial y_1}{\partial x_k} + \frac{\partial L}{\partial y_2}\frac{\partial y_2}{\partial x_k} + \cdots + \frac{\partial L}{\partial y_n}\frac{\partial y_n}{\partial x_k}$$

If $k = i$: $$\frac{\partial y_i}{\partial x_k} = \frac{e^{x_i}\left(\sum_{j=1}^{n} e^{x_j}\right) - e^{x_i}e^{x_i}}{\left(\sum_{j=1}^{n} e^{x_j}\right)^2}$$ ← Quotient Rule

$$= \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} - \left(\frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}\right)^2$$

$$= y_i(1 - y_i)$$

If $k \neq j$: $$\frac{\partial y_i}{\partial x_k} = \frac{-e^{x_k}e^{x_i}}{\left(\sum_{j=1}^{n} e^{x_j}\right)^2} = -y_k y_i$$

$$\Rightarrow \frac{\partial L}{\partial X} = \begin{bmatrix} \frac{\partial L}{\partial y_1}\frac{\partial y_1}{\partial x_1} + \cdots + \frac{\partial L}{\partial y_n}\frac{\partial y_n}{\partial x_1} \\ \frac{\partial L}{\partial y_1}\frac{\partial y_1}{\partial x_2} + \cdots + \frac{\partial L}{\partial y_n}\frac{\partial y_n}{\partial x_2} \\ \vdots \\ \frac{\partial L}{\partial y_n}\frac{\partial y_n}{\partial x_1} + \cdots + \frac{\partial L}{\partial y_n}\frac{\partial y_n}{\partial x_n} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_2}{\partial x_1} & \cdots & \frac{\partial y_n}{\partial x_1} \\ \frac{\partial y_1}{\partial x_2} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_n}{\partial x_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_1}{\partial x_n} & \frac{\partial y_2}{\partial x_n} & \cdots & \frac{\partial y_n}{\partial x_n} \end{bmatrix} \begin{bmatrix} \frac{\partial L}{\partial y_1} \\ \frac{\partial L}{\partial y_2} \\ \vdots \\ \frac{\partial L}{\partial y_n} \end{bmatrix}$$

$$= \begin{bmatrix} y_1(1-y_1) & -y_1 y_2 & \cdots & -y_1 y_n \\ -y_2 y_1 & y_2(1-y_2) & \cdots & -y_2 y_n \\ \vdots & \vdots & \ddots & \vdots \\ -y_n y_1 & -y_n y_2 & \cdots & y_n(1-y_n) \end{bmatrix} \begin{bmatrix} \frac{\partial L}{\partial y_1} \\ \frac{\partial L}{\partial y_2} \\ \vdots \\ \frac{\partial L}{\partial y_n} \end{bmatrix}$$

$$= \left(\begin{bmatrix} y_1 & y_1 & \cdots & y_1 \\ y_2 & y_2 & \cdots & y_2 \\ \vdots & \vdots & \ddots & \vdots \\ y_n & y_n & \cdots & y_n \end{bmatrix} \odot \begin{bmatrix} 1-y_1 & -y_2 & \cdots & -y_n \\ -y_1 & 1-y_2 & \cdots & -y_n \\ \vdots & \vdots & \ddots & \vdots \\ -y_1 & -y_2 & \cdots & 1-y_n \end{bmatrix} \right) \cdot \frac{\partial L}{\partial Y}$$

$$= \left(\underbrace{\begin{bmatrix} y_1 & y_1 & \cdots & y_1 \\ y_2 & y_2 & \cdots & y_2 \\ \vdots & \vdots & \ddots & \vdots \\ y_n & y_n & \cdots & y_n \end{bmatrix}}_{A} \odot \left(I - \begin{pmatrix} y_1 & y_2 & \cdots & y_n \\ y_1 & y_2 & \cdots & y_n \\ \vdots & \vdots & \ddots & \vdots \\ y_1 & y_2 & \cdots & y_n \end{pmatrix} \right) \right) \cdot \frac{\partial L}{\partial Y}$$

$$= \left(A \odot (I - A^T) \right) \cdot \frac{\partial L}{\partial Y}$$

Cost Function: Mean-Squared Error

Given desired output Y^*, actual output Y,

$$Y^* = \begin{bmatrix} y_1^* \\ y_2^* \\ \vdots \\ y_n^* \end{bmatrix}, \qquad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix},$$

$$C = \frac{1}{n} \sum_{i=1}^{n} (y_i^* - y_i)^2$$

ex) $\frac{\partial C}{\partial y_1} = \frac{1}{n} \frac{\partial}{\partial y_1} \left[(y_1^* - y_1)^2 + \underbrace{(y_2^* - y_2)^2 + \ldots + (y_n^* - y_n)^2}_{=0} \right]$

$$= \frac{2}{n}(y_1 - y_1^*)$$

$$\Rightarrow \frac{\partial C}{\partial Y} = \frac{2}{n}(Y - Y^*)$$