Neural Networks

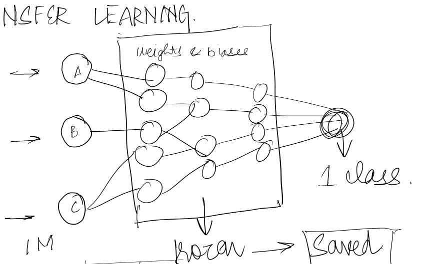
23 April 2023 11:16

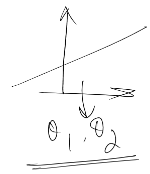
→ One of the most popular ML

→ Different types of NN:

- CNN : Image data
- RNN : Text data
- ANN : DL
- Transformers & Conformers
- FNN

* TRANSFER LEARNING

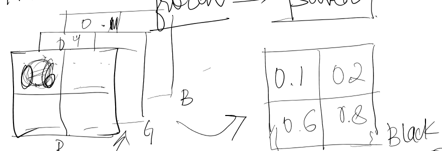




VGG-16

↳ for any img classification

Image Dataset Format



108×108×3 × 16 M → 100 → 16   1600.

Neural Networks

1. What is NN?
2. What NN does?
3. How NN fits on data?
4. Backpropogation (GFG Blog)
5. Variations

c/a BLACK BOX ⇒ because it is difficult to understand what happens inside a NN

EXAMPLE :

Drug → Illness

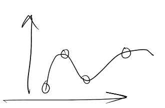
Dosage: Low → Not effec.; M → Effective; H → Not effective

⇒

⇒ No st line that can perfectly fit on such complex data

⇒ st line can predict only 2/3 dosages

* NN as shown can squiggle to any complex data -



. . . .

⭐ Concept & Composition of NN

NN consist of nodes & connections b/w nodes.

→ Node : represents parameter values that were estimated when NN was fit to a data

similar to slope & intercept of LR

→ NN starts w unknown parameters & these are estimated during fitting using a process c/a BACKPROPOGATION.

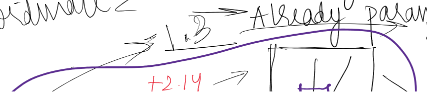
Assuming BP is completed.

Already parameters -mated.

x coordinate <

1.3 → $f(x) = 0.24$

+2.14



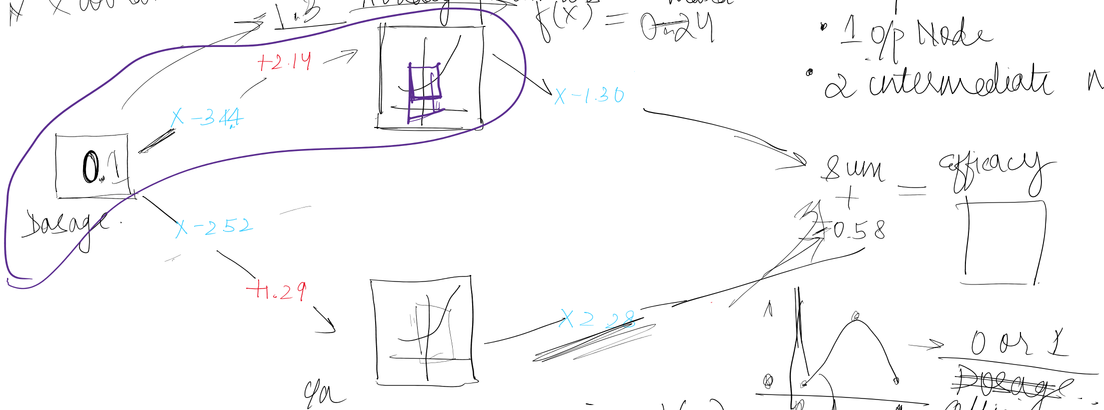
Components

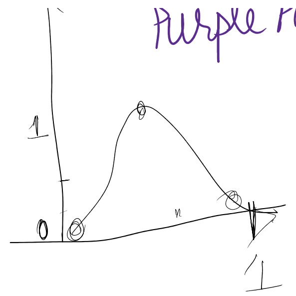
- 1 i/p Node
- 1 o/p Node

Purple Part

* x coordinate < 1.3 → Already parameters estimated · f(x) = ~~0.24~~ 0.24

• 1 i/p Node
• 1 o/p Node
• 2 intermediate nodes





CURVED LINES ? → via → Activation function f(x)

↳ Building blocks of the final squiggle that fits on our data

which one you want to use
you can decide while building the NN

→ Sigmoid $f(x) = \frac{e^x}{e^x + 1}$

→ ReLU (Rectified Linear Unit) $f(x) = \max(0, x)$
↳ practical

→ Softplus: $f(x) = \log(1 + e^x)$

BACKPROPOGATION

→ helps evaluate weights & biases
most optimal value

* Basic Idea

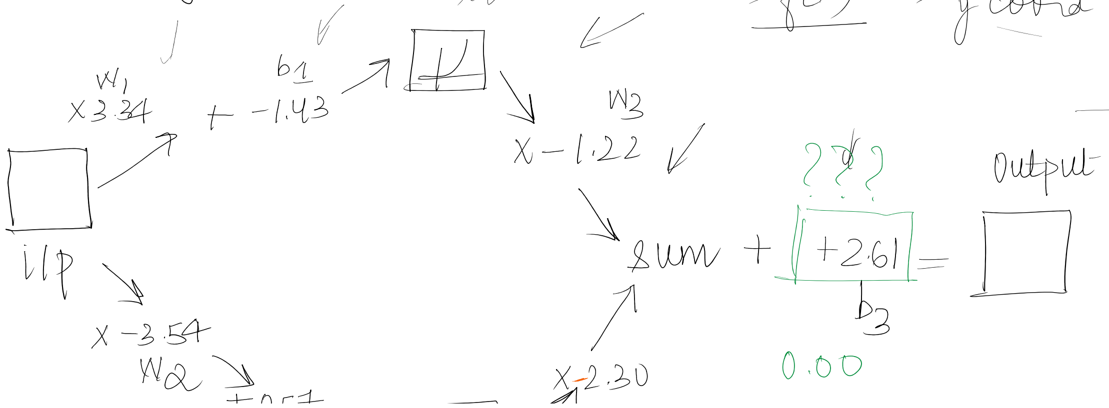
Main 2 steps:

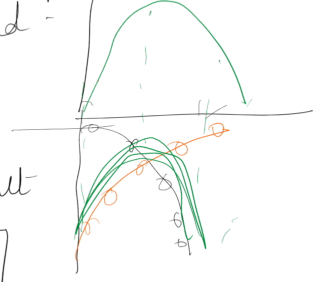
(1) Using chain rule calculate derivatives

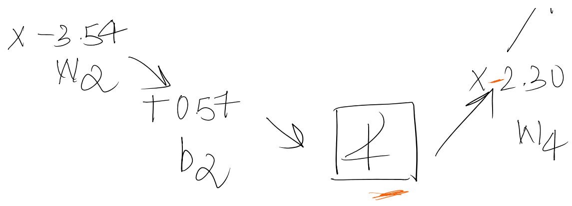
$$\frac{d\,SSR}{d\,bias} = \frac{d\,SSR}{d\,Predicted} \times \frac{d\,Predicted}{d\,Bias}$$

(2) Plugging derivative to Grad. Descent to optimize parameters

x c... → f(x) → y coord.



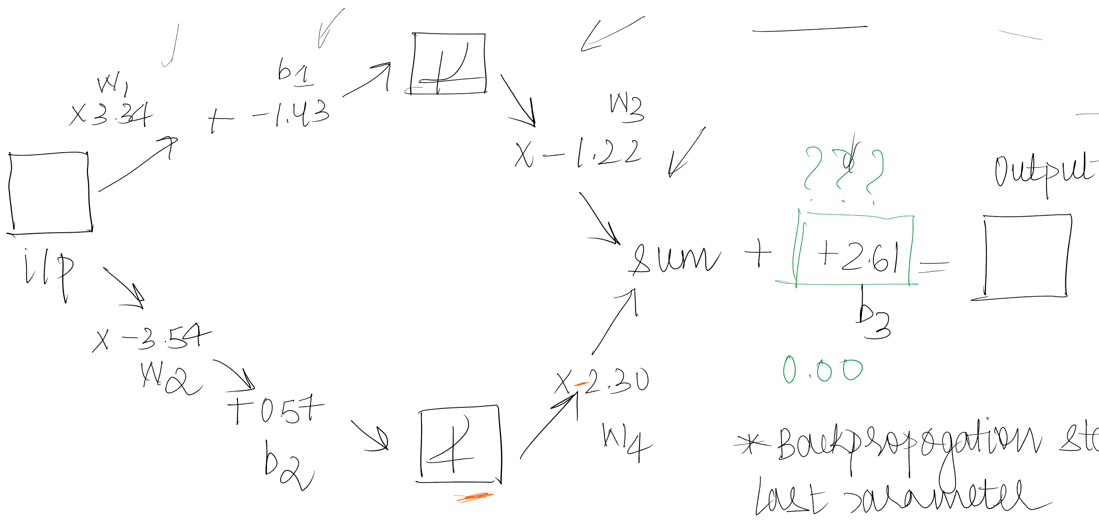


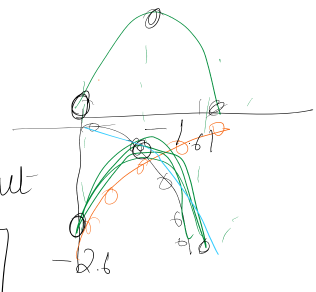


* Backpropogation starts in the last parameter

* while understanding the main concepts, we start in estimating Bias, b_3
* In order to estimate last value we assume we already have optimal values for all other w & b.
* Dosage (0 — 1)
  Efficacy (0, 1)

UNDERSTANDING PREDICTION OF B_3





* Backpropogation starts in the last parameter

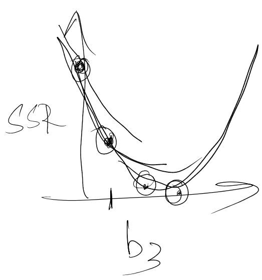
Residual: Difference b/w Observed & predicted value

Residual = Obs − Pred

$$SSR = (0 - -2.6) = 2.6$$

$$= 1 - (-1.61) = 2$$

$$= 0 - (-2.61) =$$



$$= (0 - -2.6)^2 + (1 - (-1.61))^2 + (0 - (-2.61))^2$$

$$= \boxed{20.4} \rightarrow SSR \text{ w } b_3 = 0$$

$\boxed{\text{let } b_3 = 4}$

$\boxed{\text{let } b_3 = 4}$

$$SSR = (0 - -1.6)^2 + (1 - -0.61)^2 + (0 - -1.61)^2$$
$$= 7.8$$

$\boxed{b_3 = 2}$

$SSR = 1.1$

$b_3 = 3$

$SSR = 0.48$

Generalizing.

$$SSR = (ob_1 - pred_1)^2 + (ob_2 - pred_2)^2 \ldots$$
$$= \sum_{i=1}^{n=3} (ob_i - pred_i)^2$$

$$pred_i = \text{green squiggle}_i = \text{blue}_i + \text{orange}_i + b_3$$

$$\frac{d(-x)}{dx} = -1$$

$$\frac{dSSR}{db_3} = \frac{dSSR}{d\,\text{predicted}} \times \frac{d\,\text{Predicted}}{db_3}$$

$$\frac{dSSR}{d\text{Predicted}} = \frac{d}{d\text{Predicted}} \sum_{i=1}^{3} (obs_i - pred_i)^2$$

$$= \sum_{i=1}^{n=3} 2 \times (obs_i - pred_i) \times \underbrace{\frac{d}{d\text{Predicted}} (\overset{0}{obs_i} - \overset{1}{pred_i})}_{-1}$$

$$\frac{dSSR}{db_3} = \sum_{i=1}^{3} -2 \times (obs_i - pred_i) \times \frac{d\text{Pred}}{db_3}$$

$$\frac{d\text{Pred}}{db_3} = \frac{d\,\text{green squiggle}}{db_3}$$

$$= \frac{d}{db_3} [\text{blue} + \text{orange} + b_3] = 1$$

$$\frac{dSSR}{db_3} = \sum_{i=1}^{n=3} -2 \times (obs_i - pred_i)$$

$$\frac{lr}{gn}$$