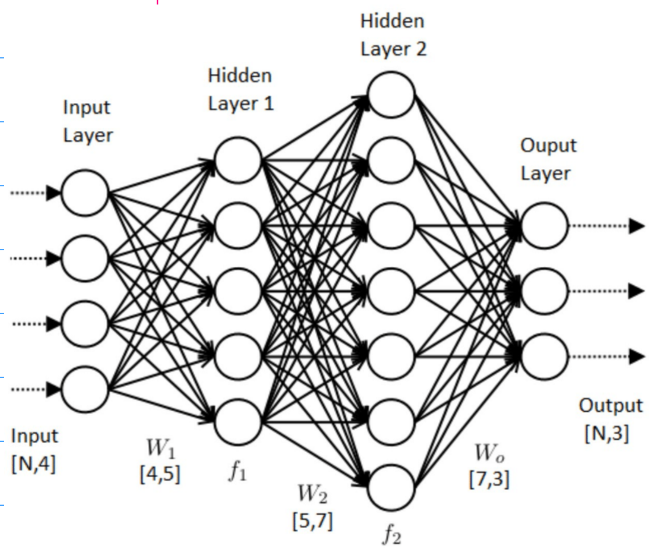
Activation functions → SOFTMAX and SIGMOID



output layer

sigmoid softmax

y_1 $y_a = y_1 / y_1+y_2+y_3$

y_2 $y_b = y_2 / y_1+y_2+y_3$

y_3 $y_c = y_3 / y_1+y_2+y_3$

$0 \leq y_1, y_2, y_3 \leq 1$

$0 \leq y_a, y_b, y_c \leq 1$

$y_a + y_b + y_c = 1$

* y_a, y_b, y_c gives the probability

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

sigmoid $$y = \frac{1}{1+e^{-z}}$$

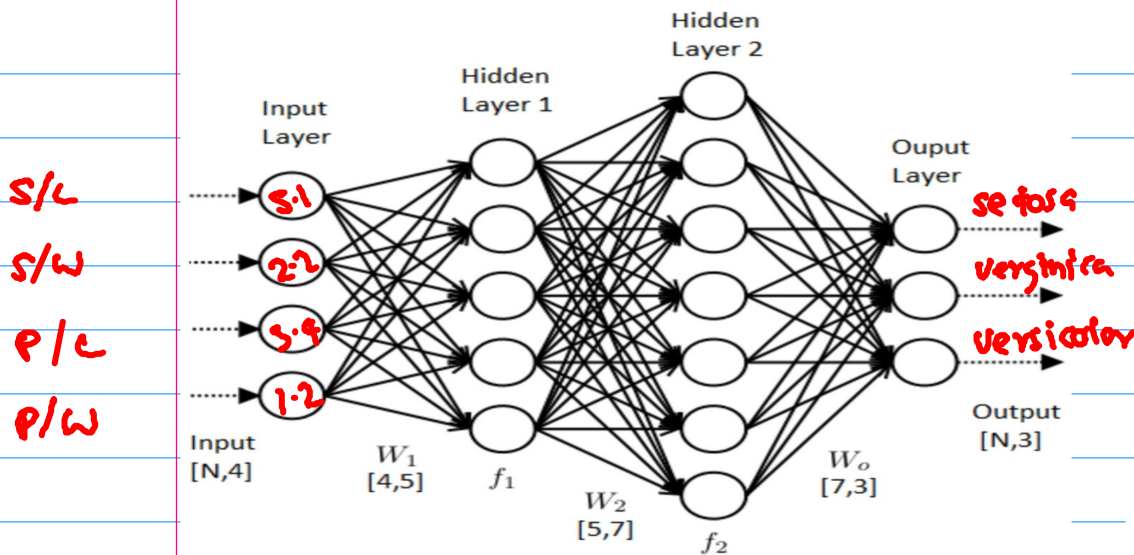
$$y_a = \frac{e^{z_1}}{\sum_{j=1}^{3} e^{z_j}}$$

$$y_1 = \frac{1}{1+e^{-z_1}} = e^{z_1}$$

$$y_2 = \frac{1}{1+e^{-z_2}} = e^{z_2}$$

$$y_3 = \frac{1}{1+e^{-z_3}} = e^{z_3}$$

Loss Functions :

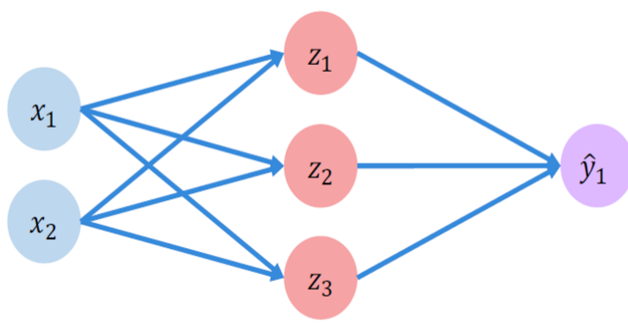


Actual	Predicted
1	0.3
0	0.2
0	0.5

$$\mathcal{L}\Big(\underbrace{f\big(x^{(i)}; W\big)}_{\text{Predicted}}, \underbrace{y^{(i)}}_{\text{Actual}}\Big)$$

(i) Regression problems : MSE (mean Squared Error)

$$X = \begin{bmatrix} 4, & 5 \\ 2, & 1 \\ 5, & 8 \\ \vdots & \vdots \end{bmatrix}$$



$$\underbrace{\begin{array}{cc} f(x) & y \\ \begin{bmatrix} 30 \\ 80 \\ 85 \\ \vdots \end{bmatrix} & \begin{bmatrix} 90 \\ 20 \\ 95 \\ \vdots \end{bmatrix} \end{array}}_{\text{Final Grades (percentage)}}$$

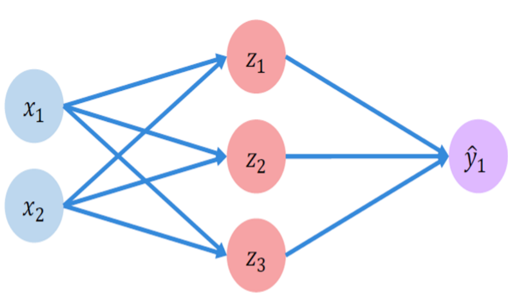
loss

$$J(W) = \frac{1}{n}\sum_{i=1}^{n}\Big(\underbrace{y^{(i)}}_{\text{Actual}} - \underbrace{f\big(x^{(i)}; W\big)}_{\text{Predicted}}\Big)^2$$

$$J(W) = \frac{1}{3}\left((90-30)^2 + (20-80)^2 + (95-85)^2\right)$$

(ii) classification : crossentropy loss

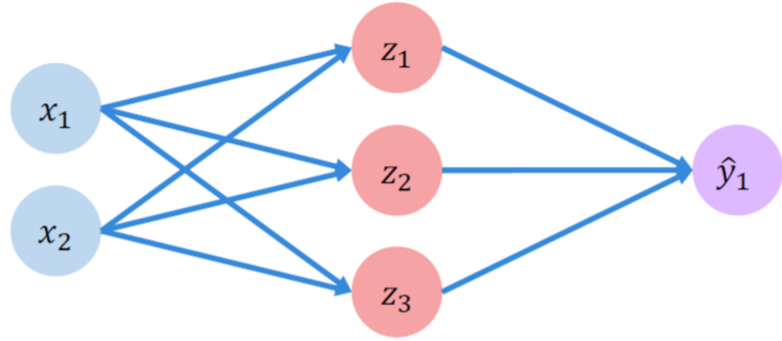
$$X = \begin{bmatrix} 4, & 5 \\ 2, & 1 \\ 5, & 8 \\ \vdots & \vdots \end{bmatrix}$$



$$\begin{array}{cc} f(x) & y \\ \begin{bmatrix} 0.1 \\ 0.8 \\ 0.6 \\ \vdots \end{bmatrix} & \begin{bmatrix} 1 \\ 0 \\ 1 \\ \vdots \end{bmatrix} \end{array}$$

$$J(W) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{y^{(i)}}_{\text{Actual}} \log\Big(\underbrace{f\big(x^{(i)}; W\big)}_{\text{Predicted}}\Big) + (1 - \underbrace{y^{(i)}}_{\text{Actual}}) \log\Big(1 - \underbrace{f\big(x^{(i)}; W\big)}_{\text{Predicted}}\Big)$$

$$X = \begin{bmatrix} 4, & 5 \\ 2, & 1 \\ 5, & 8 \\ \vdots & \vdots \end{bmatrix}$$



$$f(x) = \begin{bmatrix} 0.1 \\ 0.8 \\ 0.6 \\ \vdots \end{bmatrix} \qquad y = \begin{bmatrix} 1 \\ 0 \\ 1 \\ \vdots \end{bmatrix}$$

$$J(\boldsymbol{W}) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{y^{(i)}}_{\text{Actual}} \log\Big(\underbrace{f\big(x^{(i)};\boldsymbol{W}\big)}_{\text{Predicted}}\Big) + \underbrace{(1-y^{(i)})}_{\text{Actual}} \log\Big(1-\underbrace{f\big(x^{(i)};\boldsymbol{W}\big)}_{\text{Predicted}}\Big)$$

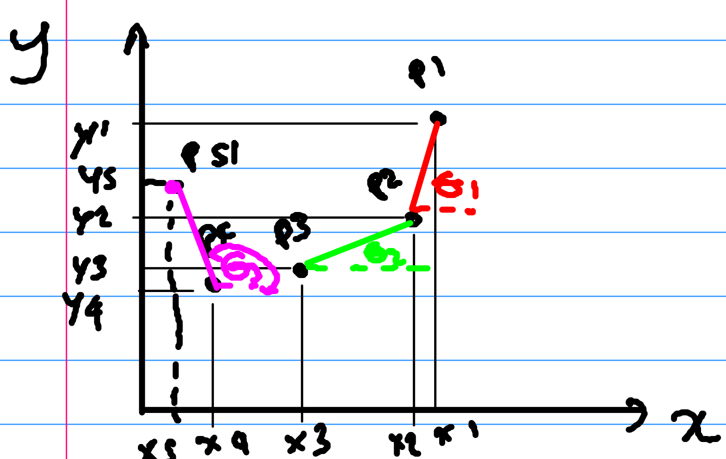
$$J(w) = \frac{1}{3}\Big[(1\log(0.1)) + \cancel{(1-1)\log(1-0.1)}^{\,0}$$
$$+ (\cancel{0\log(0.8)}^{\,0}) + (1-0)\log(1-0.8)$$
$$+ (1\log(0.6)) + \cancel{(1-1)\log(1-0.6)}^{\,0}$$

Actual Predicted

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\frac{1}{3}\Big[1\,\cancel{\log(1)}^{\,0} + \cancel{(1-1)}^{\,0}\log(1-1)$$
$$+ \cancel{0\log(0)}^{\,0} + (1-0)\,\cancel{\log(1-0)}^{\,0}$$
$$+ \cancel{0\log(0)}^{\,0} + (1-0)\,\cancel{\log(1-0)}^{\,0}\Big]$$

The Gradient



$$\tan\theta_1 = m_1 = \frac{y_1 - y_2}{x_1 - x_2} \quad +ve$$

$$\tan\theta_2 = m_2 = \frac{y_2 - y_3}{x_2 - x_3} \quad +ve$$

$$\tan\theta_3 = m_3 = \frac{y_4 - y_5}{x_4 - x_5} \quad -ve$$

Generally → $\tan\theta = m = \dfrac{\delta y}{\delta x}$ ← small change in y / small change in x

Gradient Decent Algorithm (loss optimization)

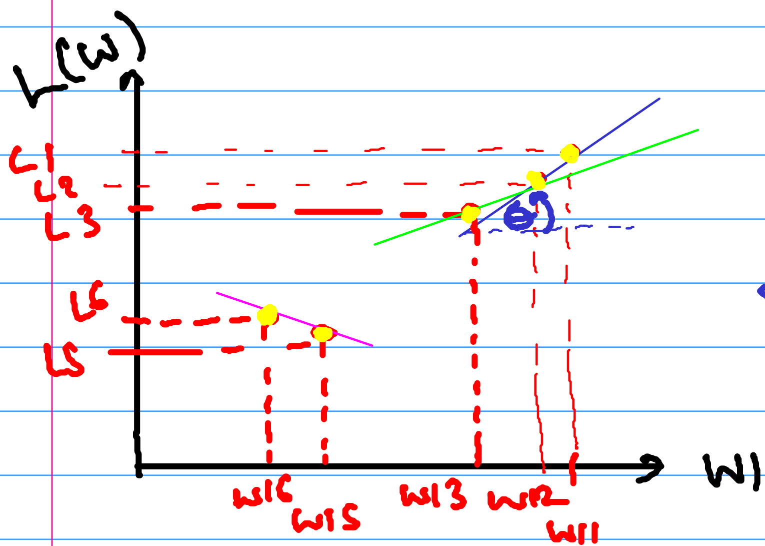
Compute gradient, $\frac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}}$

Learning Rate

Update weights, $\boldsymbol{W} \leftarrow \boldsymbol{W} - \eta \frac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}} \rightarrow \Delta W$

updated weights   Previous weights   Gradient Vector

$$W = \begin{bmatrix} w1 \\ w2 \\ w3 \\ . \\ . \\ wn \end{bmatrix} \quad -\eta \frac{\partial J(W)}{\partial W} = \begin{bmatrix} \Delta w1 \\ \Delta w2 \\ \Delta w3 \\ . \\ . \\ \Delta wn \end{bmatrix}$$



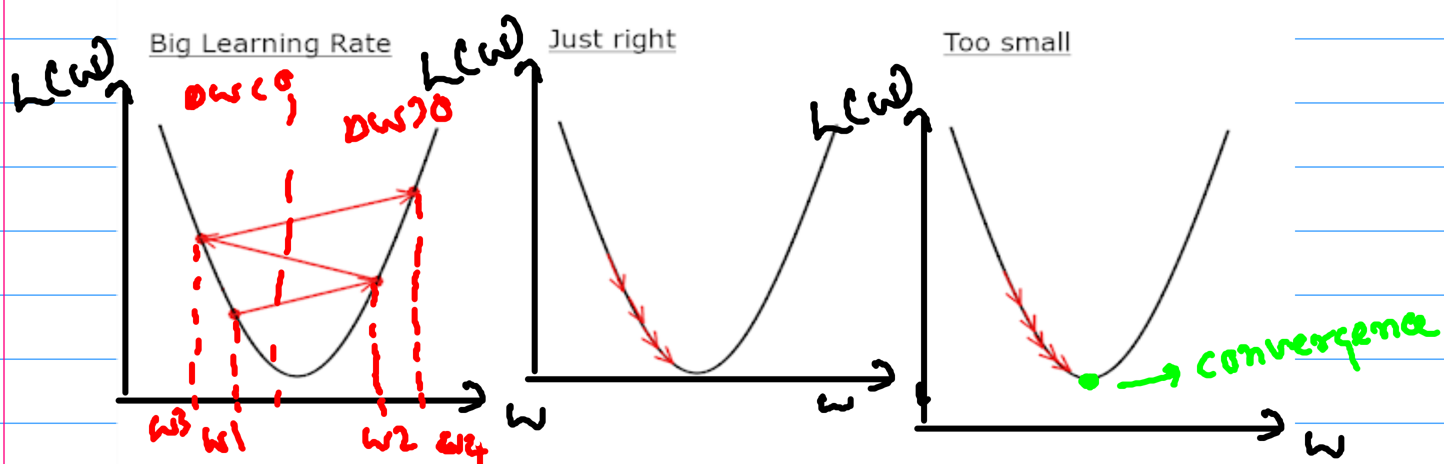
$W12 \rightarrow W11 - \eta \Delta W$

$$\tan\theta = m = \frac{\partial J(W)}{\partial W_1}$$

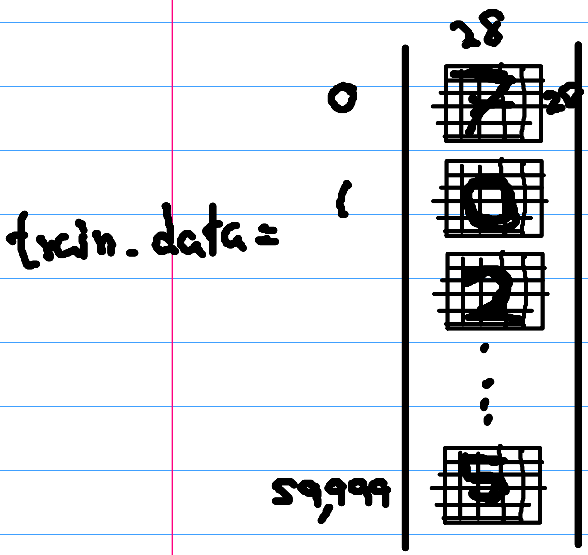
The individual influence (rate of change) of W1 to the L(W)

Learning Rate

$W \rightarrow W - \eta \, \Delta W \rightarrow$ gradient



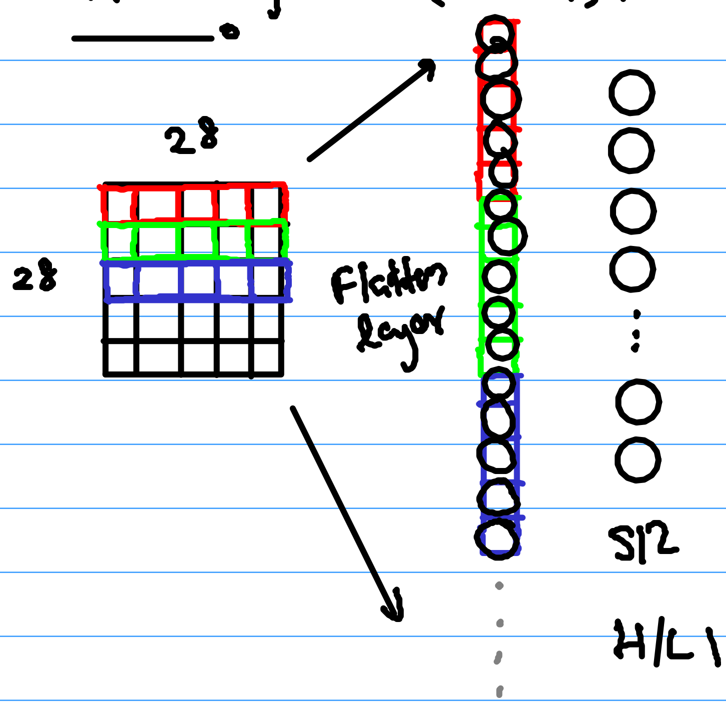
MNIST DATASET



train_target = $\begin{bmatrix} 7 \\ 0 \\ 2 \\ \vdots \\ 5 \end{bmatrix}$

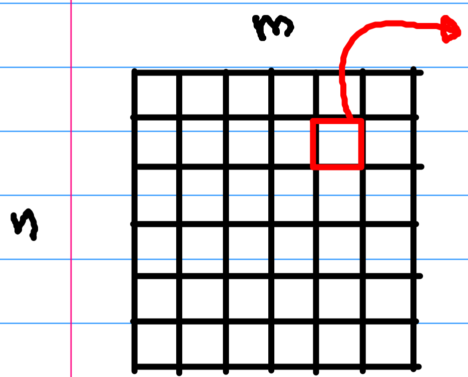
Flatten layer

$(28 \times 28), 1 = 784, 1$



Images in Computers

resolution = m x n



(i) color Images

□ → R G B

8 bit x 3

0-255 0-255 0-255

ex→ (255,255,0)

(0,0,0)

(255,255,255)

(179, 56, 250)

(ii) Gray Scaled Image

□ ↓ 8 bit (0-255)

0 - Black

255-white

(iii) B/W (Binary) Images

□ → 0/255