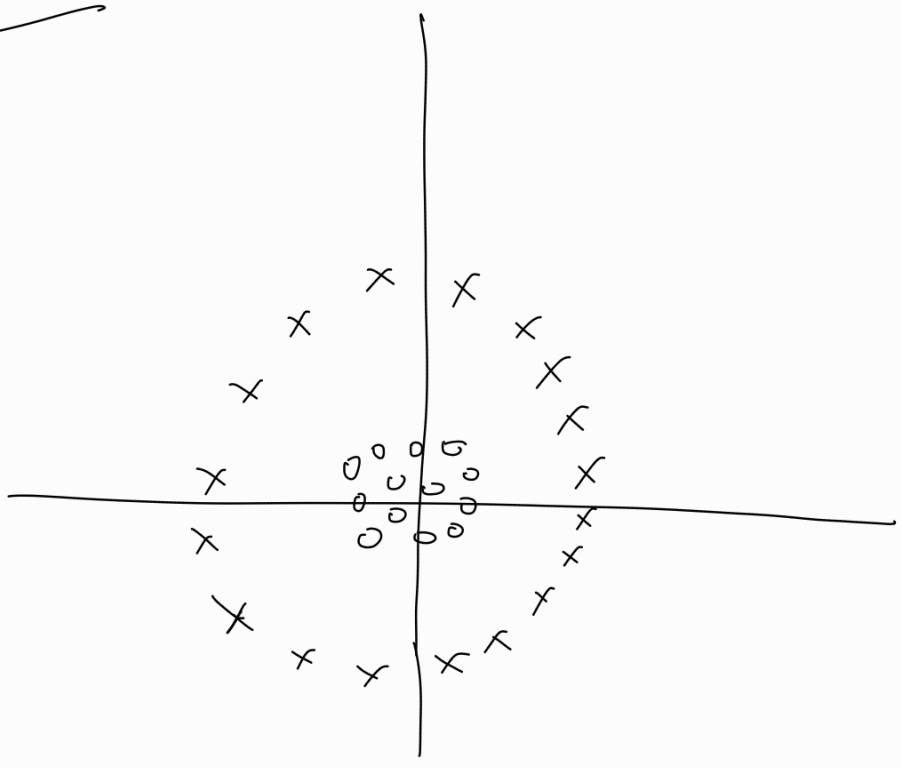
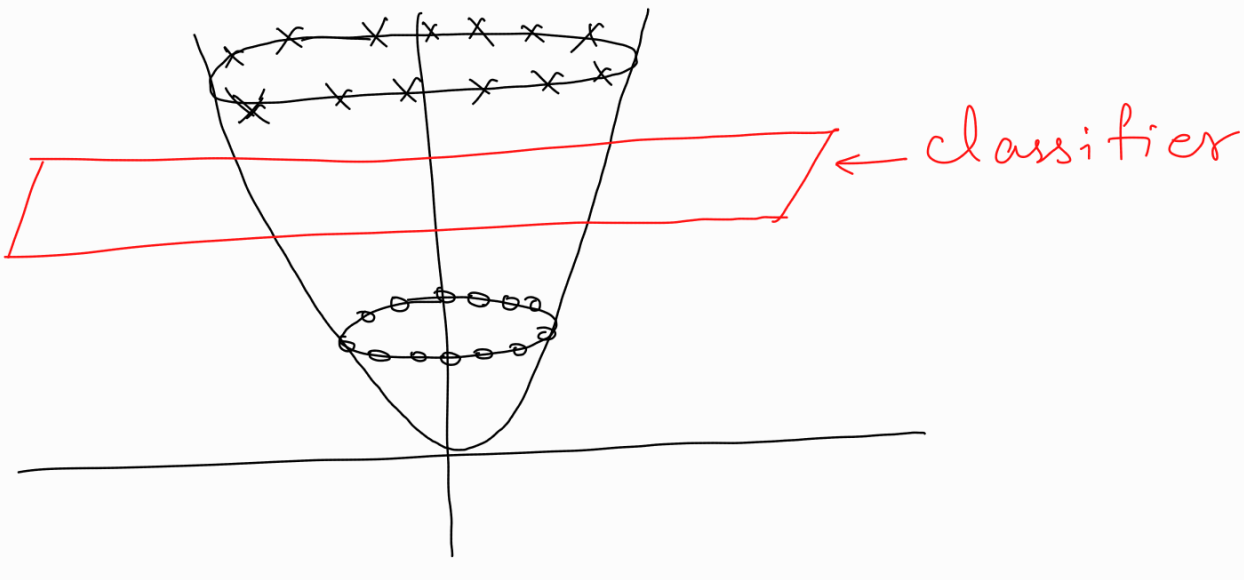
16/2/24

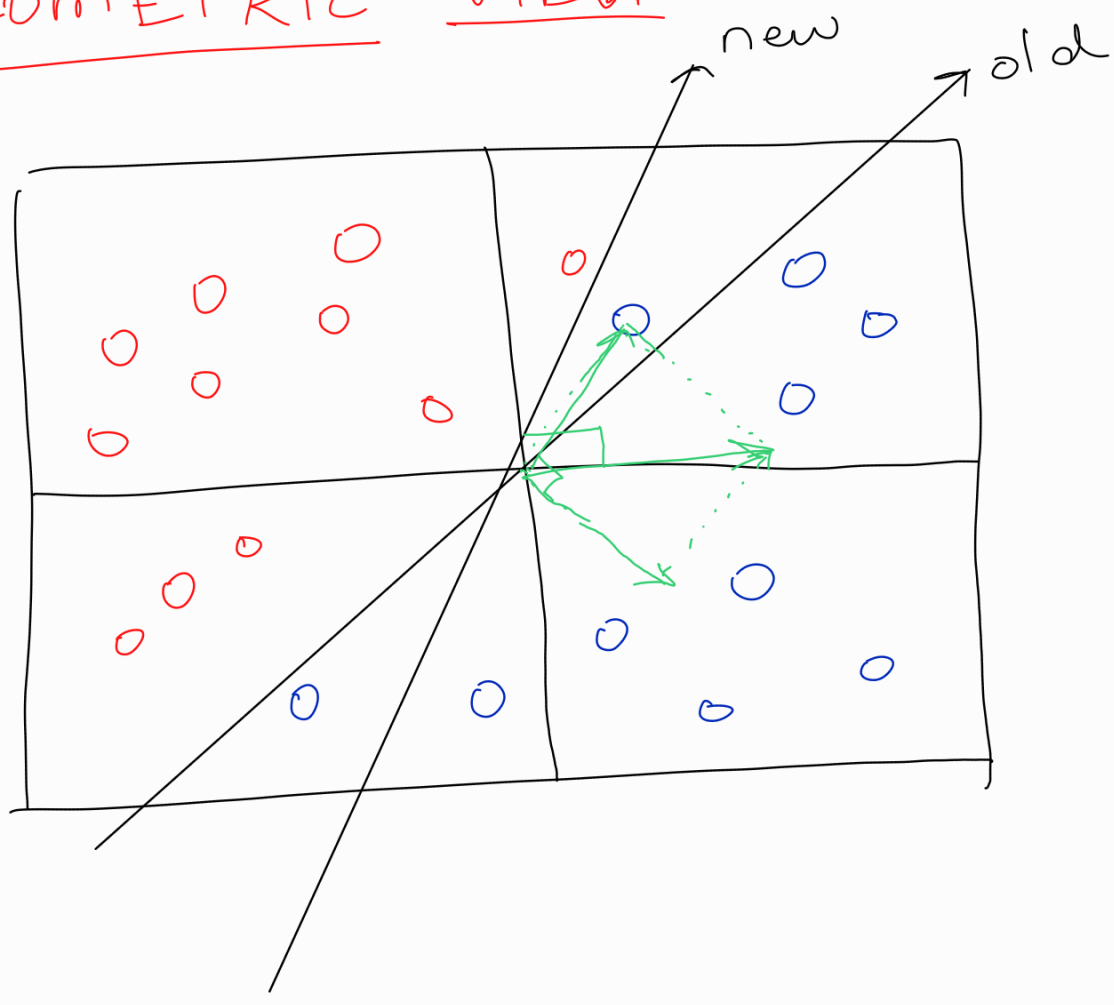


A simple trick can use linear classification for classifying this

$$\phi\begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} x \\ y \\ x^2 + y^2 \end{bmatrix}$$



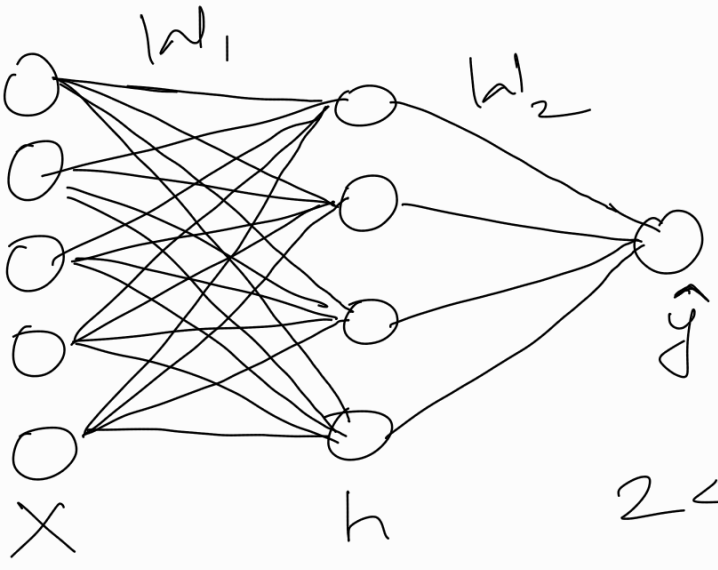
GEOMETRIC VIEW



1] Add the 2 vectors

2] New classifier $\perp$ to sum

3] Repeat



24 parameters

$$h = \phi(W_1^T x)$$

$$\hat{y} = \phi(W_2^T h)$$

$$L = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

$$W_1 = W_1 - \lambda \frac{\partial L}{\partial W_1}, \quad W_2 = W_2 - \lambda \frac{\partial L}{\partial W_2}$$

$$\hat{y} = \phi\left[W_2^T \cdot \phi(W_1^T x)\right]$$

$$\frac{\partial L}{\partial W_2} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial W_2}$$

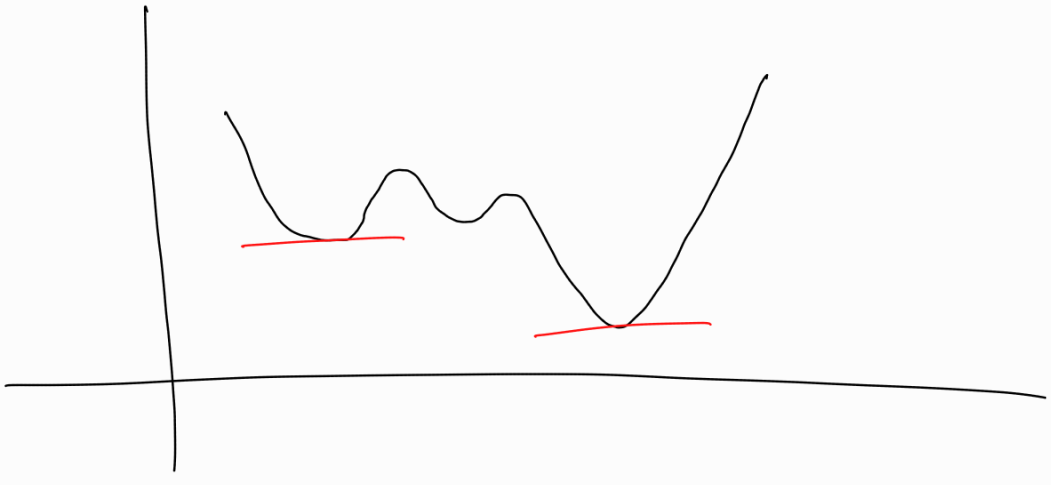
$$\frac{\partial L}{\partial W_2} = -2\Sigma(y - \hat{y}) \cdot \phi'(W_2^T h) \cdot h$$

$$\frac{\partial L}{\partial W_1} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial h} \cdot \frac{\partial h}{\partial W_1}$$

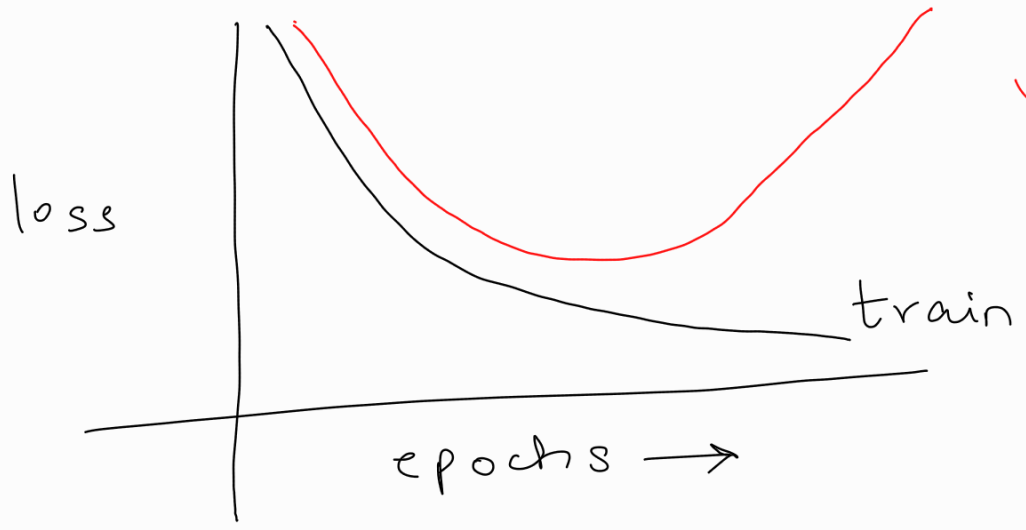
$$\frac{\partial L}{\partial W_1} = -2 \sum (y - \hat{y}) \cdot \phi'(W_2^T h) W_2 \cdot \phi'(W_1^T X) \cdot X$$

Loss function of Linear Regression was convex.

This is non-convex, you can't be sure of going to global minimum. People still achieve good performance.



Q] When to stop?

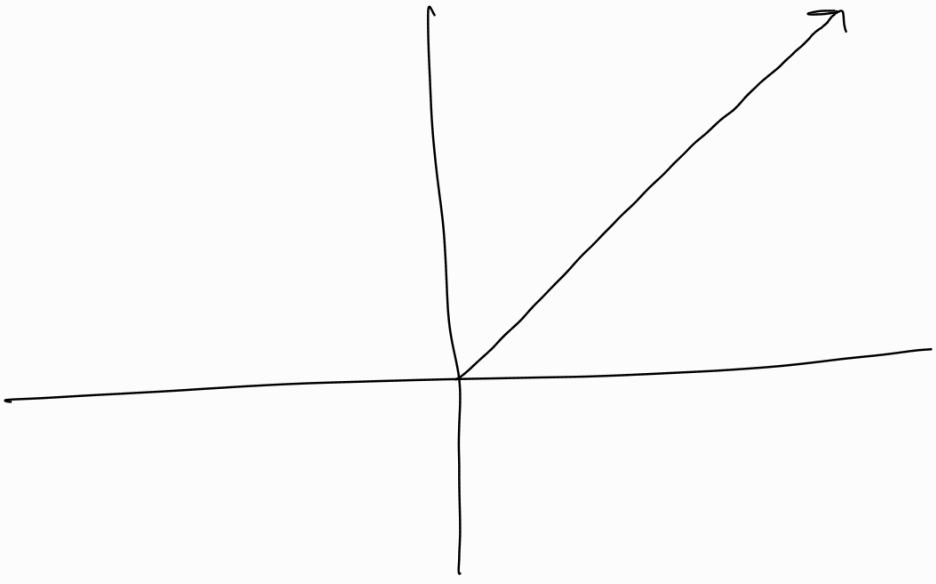


→ Stop when you get best performance on validation

→ This is called early stopping.

Most Common Non-Linearities

1] ReLU (Rectified Linear Unit)

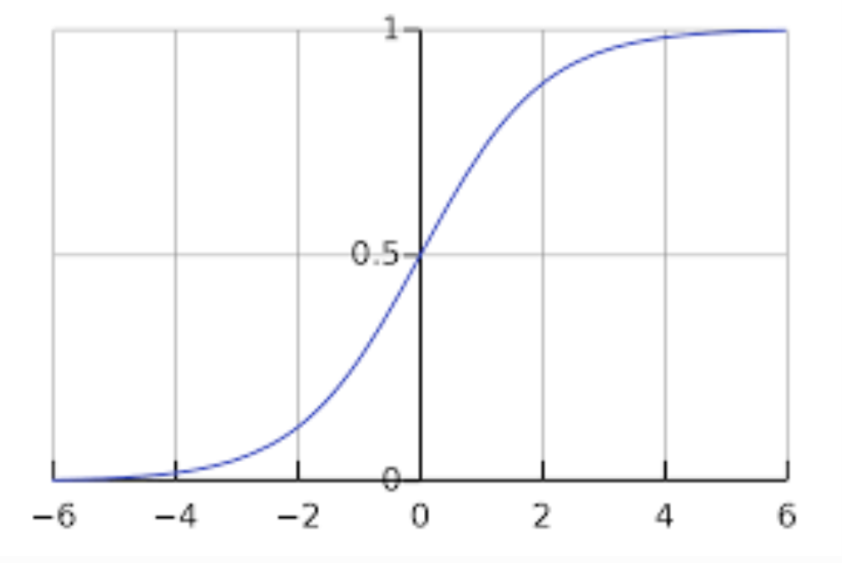


This is not differentiable at 0.

It is piecewise differentiable.

Gradient at 0 is 0

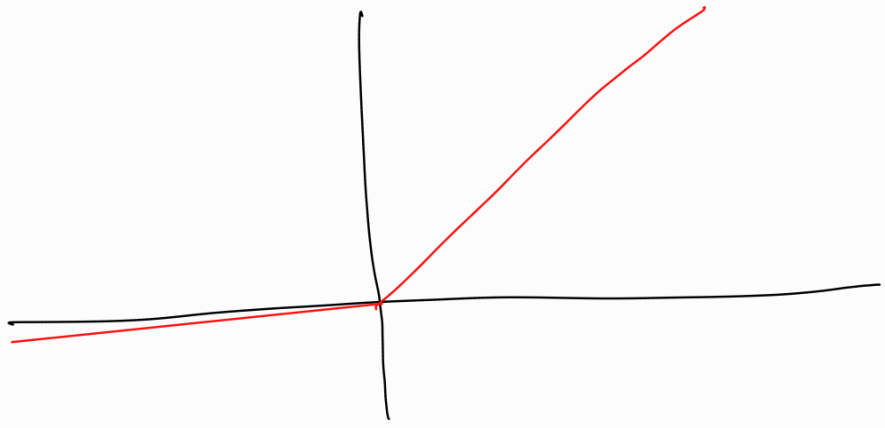
2] Sigmoid



$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

$$\sigma'(x) = \sigma(x)\left[1 - \sigma(x)\right]$$

3] Leaky ReLU



4] tanh

Mostly used in RNNs.

---

2 - class classification

Apply sigmoid on output

Apply loss on sigmoidal o/p

y_i is either 0 or 1

p_i is predicted value

for $y_i = 0$ → ← for $y_i = 1$

$$L = -\sum (1 - y_i) \log (1 - p_i) + y_i \log p_i$$

here y_i is actual value

Binary Cross Entropy Loss Function

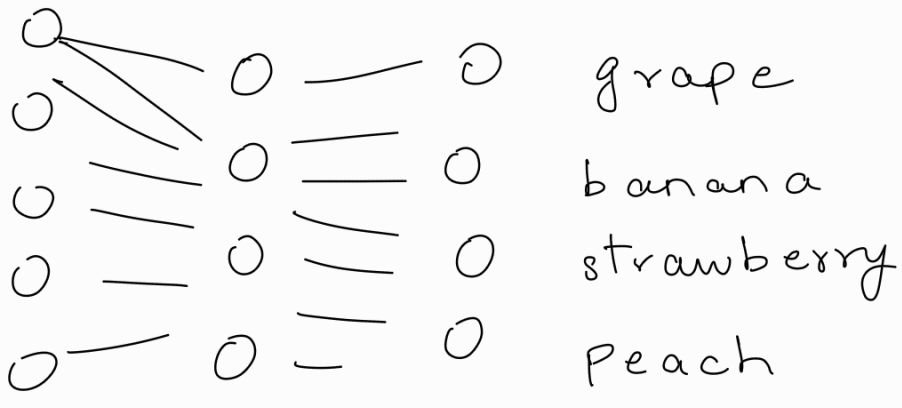
→ We want p_i to be close to y_i

When $p_i = y_i$, $L = 0$

For multi-class classification, we apply loss on each class o/p and take mean gradient. This is a case where multiple classes are correct

---

Multi-class single correct



We use 1-hot embedding

4 vectors:
$$\begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$

→ Predictions are called logits.

→ We use softmax : $\frac{e^{l_i}}{\sum_{i=1}^{c} e^{l_i}}$

Consider logits $\begin{bmatrix} 5 \\ 7 \\ 2 \\ 1 \end{bmatrix}$

Applying softmax:

$$5 \rightarrow \frac{e^5}{e^7+e^5+e^2+e^4}$$

$$7 \rightarrow \frac{e^7}{e^7+e^5+e^2+e^4}$$

$$2 \rightarrow \frac{e^2}{e^7+e^5+e^2+e^4}$$

Mi

→ Heightens maximum value.

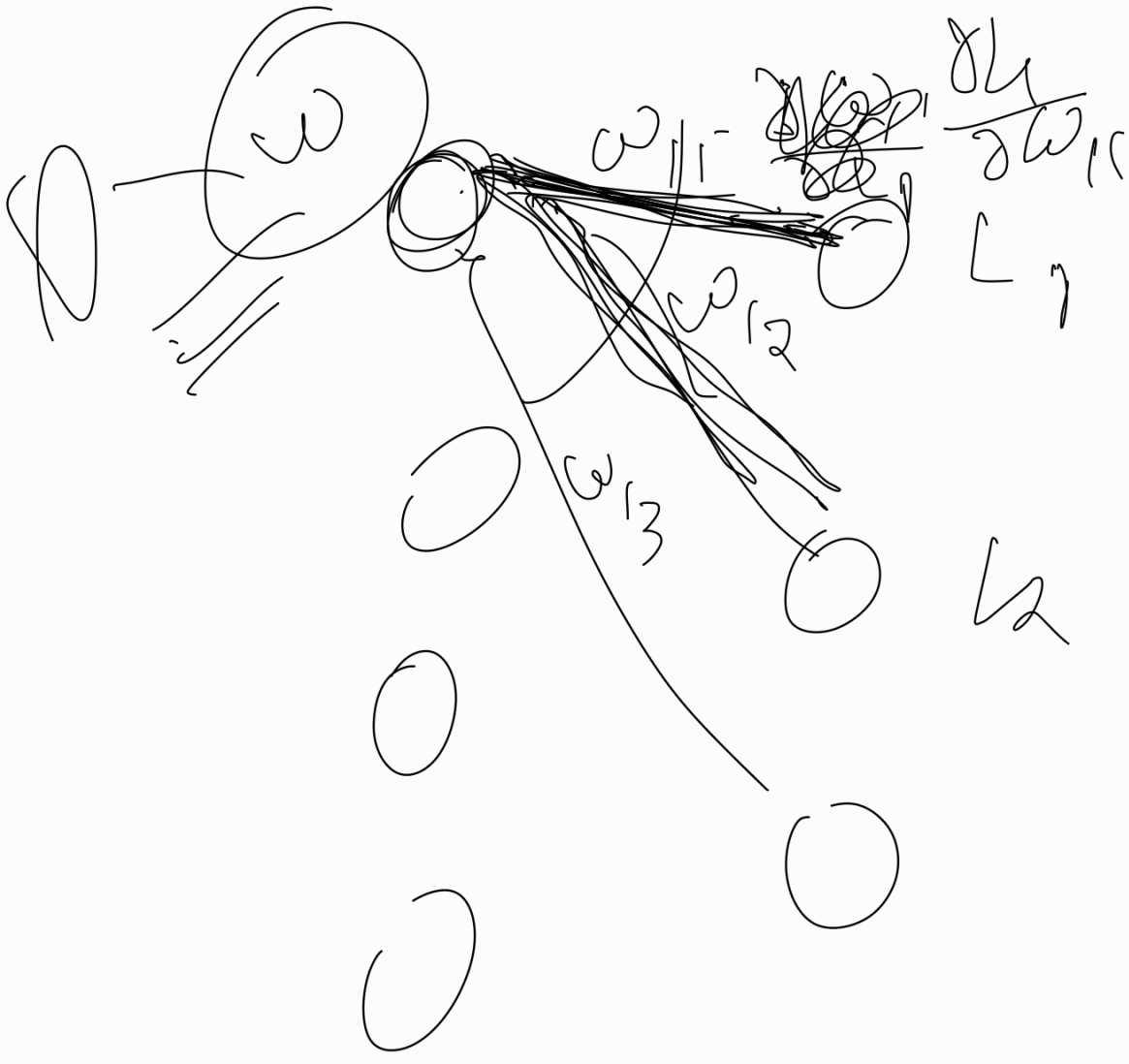
→ Brings value b/n 0 and 1.

→ Apply cross entropy loss function.

$$L = -\log(p_g)$$

on the ground truth class

→ If your value of the expected class is close to 0, the loss will be very high. (Note that you are applying loss function after applying softmax)


ω
ω_{11}
$\frac{\partial L_1}{\partial \omega_{11}}$
L_1
ω_{12}
ω_{13}
L_2