AGENDA.

1) Regularization
   - l_1
   - l_2 } ML
   - max-norm
   - Dropout ✓

   } Why ? To avoid overfitting

2) Loss functions :-
   - Regression → MAE, MSE, huber loss, Pseudo huber loss
   - Classification → hinge loss, Cross entropy ← Binary, Categorical + sparse.

Linear Regression

$$y = \theta_0 1 + \theta_1 x_1 + \theta_2 x_2 + \dots \theta_n x_n = \theta^T x = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}^T \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

θ_0 → Bias

θ_1, θ_2 ... → parameters

total parameters $n \rightarrow \theta_1 \rightarrow \theta_n$

$1 \rightarrow \theta_0$

$n+1$

$n+1 \uparrow \Rightarrow$ Overfitting

solution → Huge amount which should contain almost all kind of patterns.

1) l_1 regularization | LASSO → Least Absolute Shrinkage & Selector operator

$$J_n(\theta) = J_0(\theta) + \alpha \sum_{i=1}^{m} |\theta_i|$$

$m \rightarrow 1 \rightarrow m$ ⇒ Not forcing bias

α → regularization factor $\in [0, 1]$

$\alpha \sum_{i=1}^{m} |\theta_i|$ → l_1 term

at some time during the training

$J_0(\theta) = 0 +$ noise $\Rightarrow$ error $\neq 0$

$\Downarrow$

weight update

$w = w - \eta \frac{\partial J}{\partial w}$

while training,

$J_n(\theta) = J_0(\theta) + L1 \Rightarrow w \uparrow$

while testing,

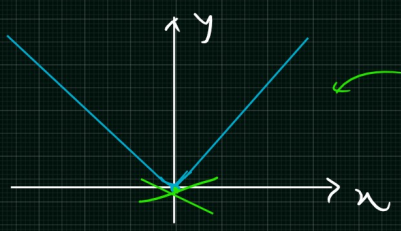
$J_n(\theta) = J_0(\theta)$

Drawback :-

i) Latency due to extra computation

ii) $J_n(\theta)$ is not differentiable everywhere

L_1 term $= \alpha \sum_{i=1}^{m} |\theta_i|$



Not differentiable at $x=0$

L_2 regularization | Ridge

$J_n(\theta) = J_0(\theta) + \frac{\alpha}{2} \sum_{i=1}^{m} (\theta_i)^2$

L_2 term

$\theta_i^2 \downarrow \frac{2\theta_i}{2}$

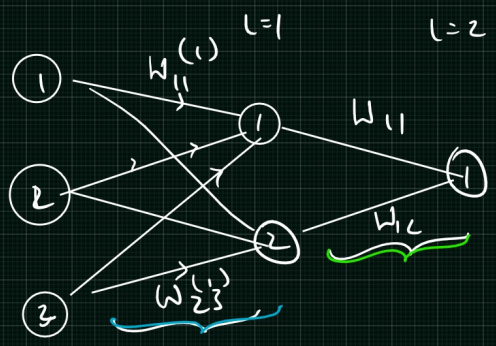
Training $\rightarrow$ $J_n(\theta) = J_0(\theta) + L_2$

Testing $\rightarrow$ $J_n(\theta) = J_0(\theta)$

Drawback :-

1) Latency due to extra computation.

2) α as an extra parameter has to be tuned.



$$L1 = \alpha \left\{ |W_{11}^{(1)}| + \cdots |W_{23}^{(1)}| + |W_{11}^{(2)}| + |W_{12}^{(1)}| \right\}$$

$$J_n(\theta) = J_0(\theta) + L1$$

$$L2 = \frac{\alpha}{2} \left\{ (W_{11}^{(1)})^2 + \cdots (W_{23}^{(1)})^2 + (W_{11}^{(2)})^2 + (W_{12}^{(2)})^2 \right\}$$

3) $l_1 - l_2$ Regularization

$$J_n(\theta) = J_0(\theta) + \overbrace{r\alpha \sum_{i=1}^{m} |\theta_i|}^{L1} + \overbrace{(1-r)\frac{\alpha}{2} \sum_{i=1}^{m} \theta_i^2}^{L2}$$

$r \rightarrow$ controlling factor $\quad r \in [0,1]$

$r = 1 \Rightarrow$ L1 is dominating $\quad J_n(\theta) \rightarrow$ Lasso

$r = 0 \Rightarrow$ L2 is dominating $\quad J_n(\theta) \rightarrow$ Ridge

$r = (0,1)$

Presence of outliers

Outliers $\rightarrow$ exponentially less (rare) $<<$ less $\rightarrow$ L2

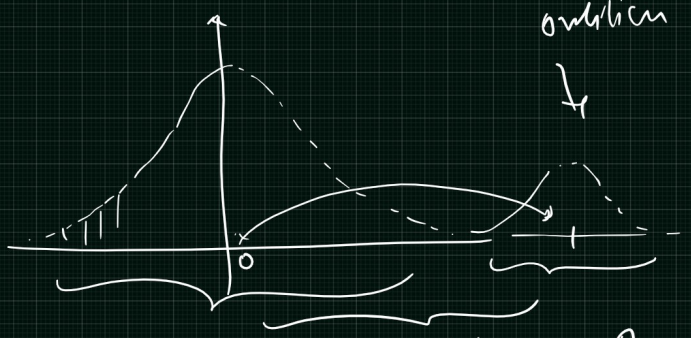
outliers $\rightarrow$ high $\rightarrow$ L1

RMSE =

$$\|V\|_k = \left(|V_0|^k + |V_1|^k + \cdots |V_n|^k \right)^{1/k} \Rightarrow k \text{ norm}$$

$$\|V\|_{k=1} = \left(|V_0| + |V_1| + \cdots |V_n| \right) = L1$$

$$\|V\|_{k=2} = \left(V_0^2 + V_1^2 + \cdots V_n^2 \right)^{1/2} = L2$$



$MSE = (y - \hat{y})^2$

$AE = |y - \hat{y}|$

$|d| = A \quad C > B > A$

$|d|^2 = B$

$|d|^n = C$

k is large ⇒ more sensitive to outliers

k is less ⇒ less sensitive

$\hat{y}_1^2 \quad \hat{y}_2^8$

$SE = 36$

$AE = 6$

$\hat{y}_1^2 \quad \hat{y}_2^{8.1}$

-0.1

$SE = 0.2$

$AE = 0.1$

Max-norm Regularization :-

↳ We scale weights.

W for incoming connection is constrained based on below condition

$$\|W\|_2 \leq r$$

l_2 norm ↳ max norm hyperparameter

After each training step

$$w' \leftarrow w - \eta \frac{\partial f}{\partial w} \longrightarrow w'' \leftarrow \frac{w' r}{\|W\|_2}$$

$w_1 \to 0$

$w_2 \to 0$

$w_n \to 0$

$\sum w^2 \leq r$

conditions :-

1) if $\|W\|_2 = r$ ⇒

$$w'' \leftarrow \frac{w' r}{\|W\|_2}$$

$w'' \leftarrow w'$ No regularization or scaling

2) if $\|W\|_2 < r$

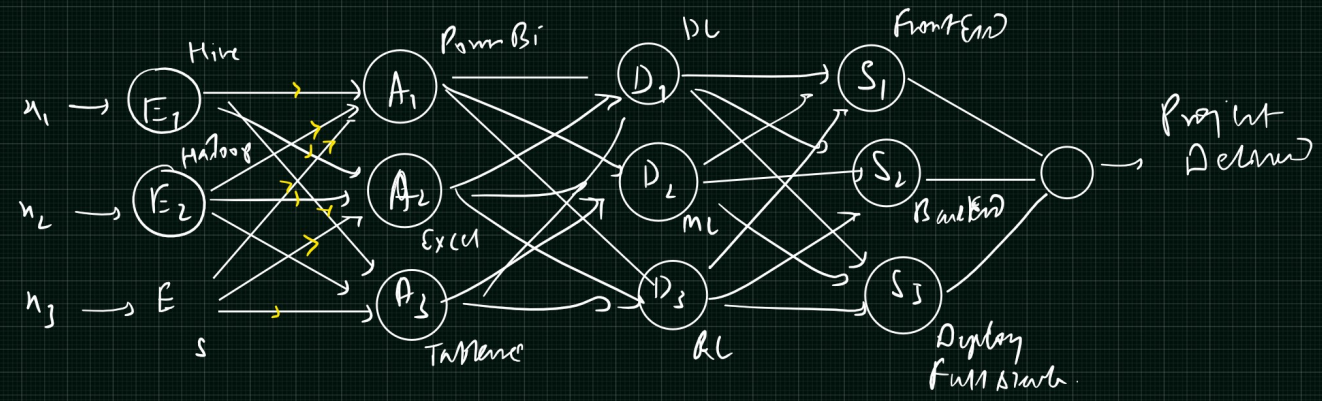
$$w'' \leftarrow \frac{w' r}{\|W\|_2}$$

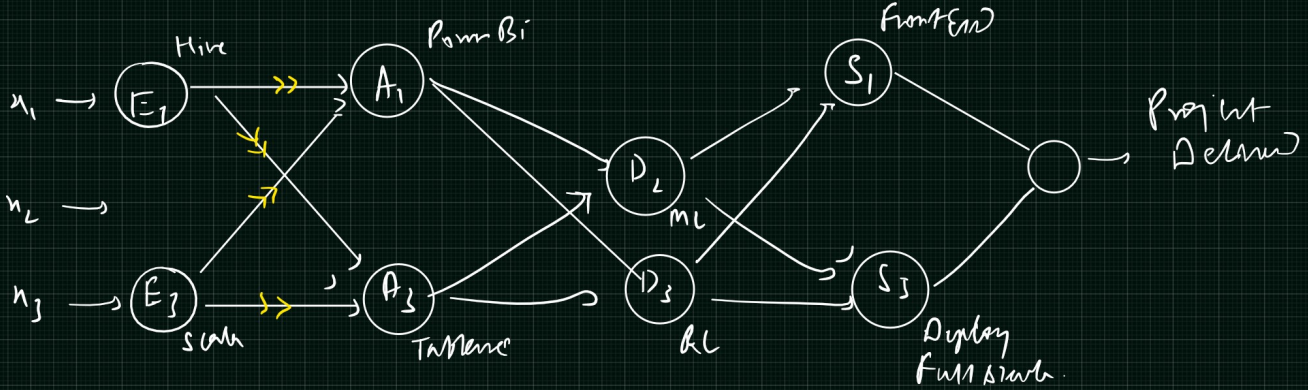
r > denominator

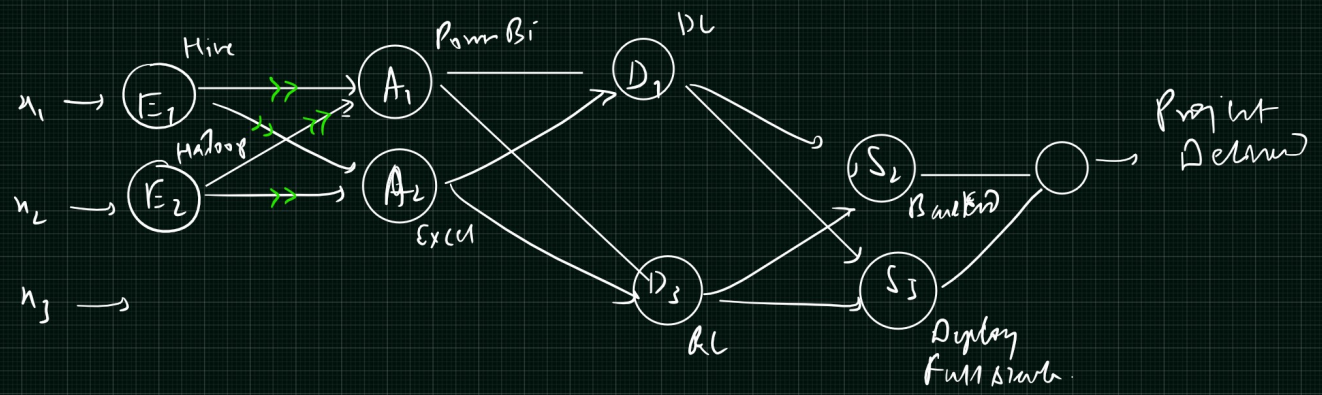
⇒ Scaling up.

Dropout layers :-

Data science Team







n → no-of neurons

2^n → network is trained.

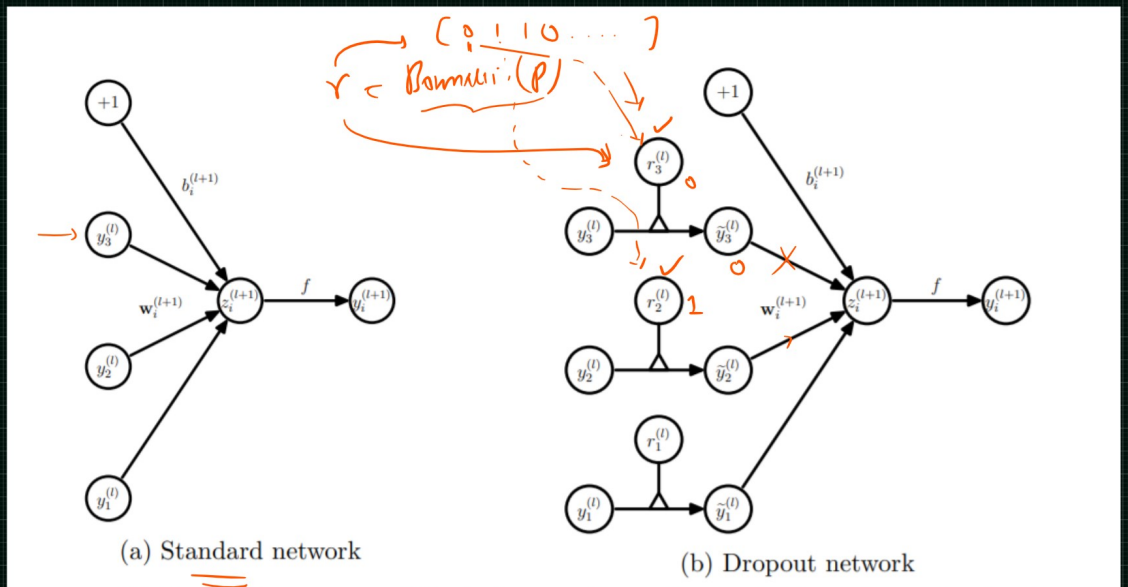
This section describes the dropout neural network model. Consider a neural network with L hidden layers. Let $l \in \{1, \ldots, L\}$ index the hidden layers of the network. Let $\mathbf{z}^{(l)}$ denote the vector of inputs into layer l, $\mathbf{y}^{(l)}$ denote the vector of outputs from layer l ($\mathbf{y}^{(0)} = \mathbf{x}$ is the input). $W^{(l)}$ and $\mathbf{b}^{(l)}$ are the weights and biases at layer l. The feed-forward operation of a standard neural network (Figure 3a) can be described as (for $l \in \{0, \ldots, L-1\}$ and any hidden unit i)

$$
\begin{aligned}
z_i^{(l+1)} &= \mathbf{w}_i^{(l+1)} \mathbf{y}^l + b_i^{(l+1)}, \\
y_i^{(l+1)} &= f(z_i^{(l+1)}),
\end{aligned}
$$

where f is any activation function, for example, $f(x) = 1/(1 + \exp(-x))$.

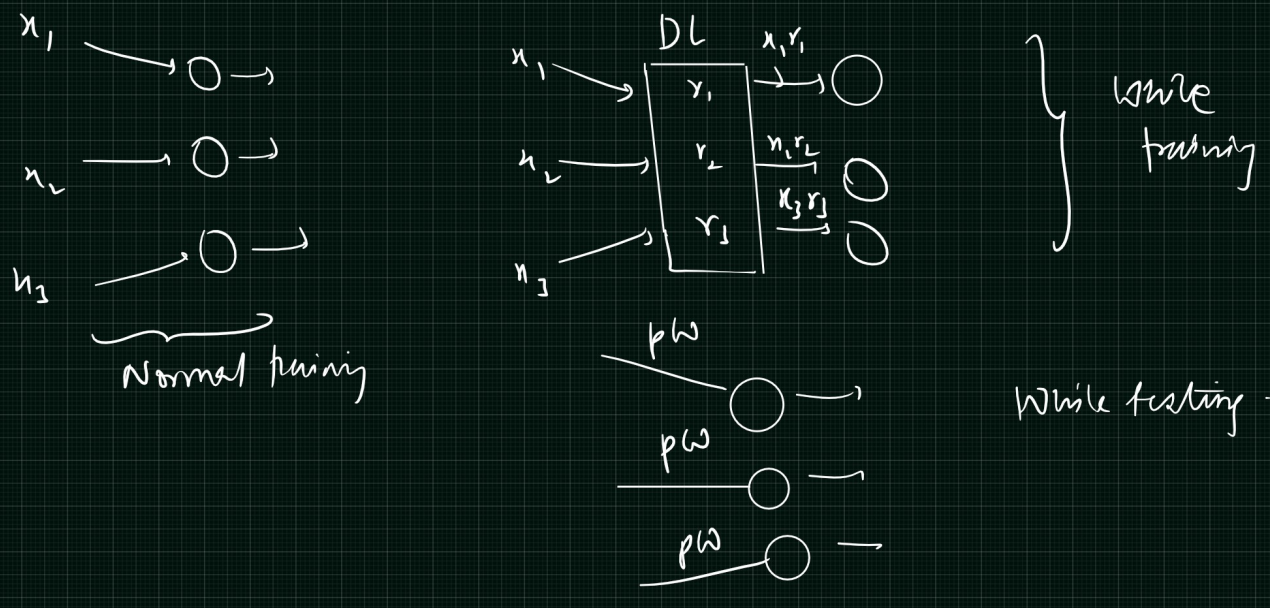
With dropout, the feed-forward operation becomes (Figure 3b)

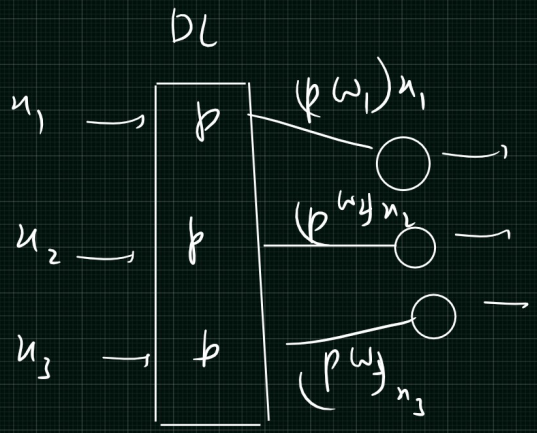
$$
\begin{aligned}
r_j^{(l)} &\sim \text{Bernoulli}(p), \\
\tilde{\mathbf{y}}^{(l)} &= \mathbf{r}^{(l)} * \mathbf{y}^{(l)}, \\
z_i^{(l+1)} &= \mathbf{w}_i^{(l+1)} \tilde{\mathbf{y}}^l + b_i^{(l+1)}, \\
y_i^{(l+1)} &= f(z_i^{(l+1)}).
\end{aligned}
$$



(a) Standard network (b) Dropout network

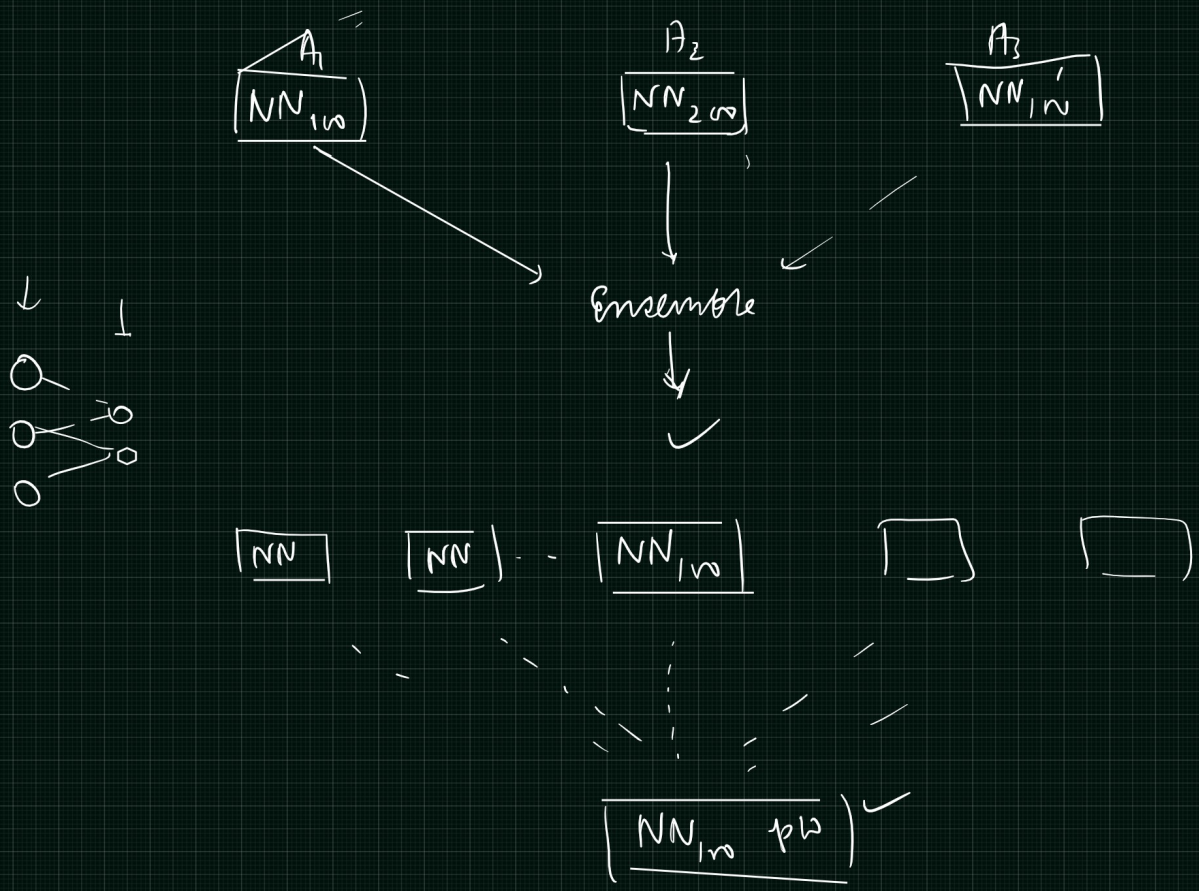
Figure 3: Comparison of the basic operations of a standard and dropout network.





$p \rightarrow$ keeping probability =
probability of keeping the neurons

$p \in [0.5, 1]$



Loss functions.

$\frac{\partial C}{\partial w}$ $\quad \nabla_\theta J(\theta)$

A) Regression :-

(i) MAE mean absolute error | l_1 loss | l_1 norm loss fn

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|y^{(i)} - \hat{y}^{(i)}|$$

Disadvantage :
Not continuously differentiable at every point.

(ii) MSE mean squared error | l_2 loss | l_2 norm loss fn

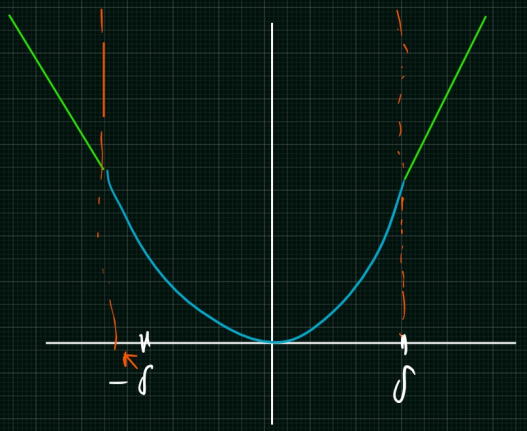
$$MSE = \frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} - \hat{y}^{(i)}\right)^2$$

Drawbacks :-
Sensitive to outliers

(iii) Huber loss

$$L_\delta(y, \hat{y}) = \begin{cases} \frac{1}{2}(y-\hat{y})^2 & \text{for } |y-\hat{y}| \le \delta \\ \delta\left(|y-\hat{y}| - \frac{1}{2}\delta\right), & \text{otherwise} \end{cases}$$

$$|y-\hat{y}| \le \delta$$
$$\Rightarrow \quad -\delta \le (y-\hat{y}) \le \delta$$



(iii) Pseudo Huber loss
Refer Notebook

B) For Classification

(i) Hinge loss (for Binary classification)

$$L = \max\left(0, (1 - t \cdot \hat{y})\right)$$

$\hat{y}$ → predicted value

t → target

where $t = \{+1, -1\}$

$+1$ → Class A

-1 → Class B

Case 1: A^{+1} $(2, 2)$

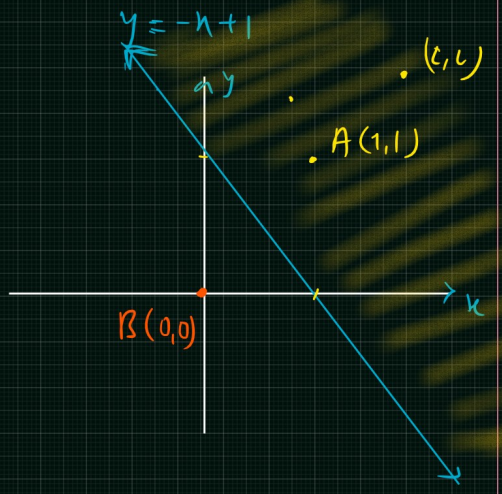
$f(x, y) = \hat{y} = 2 + 2 - 1 = 3$

$L = \max(0, 1 - (+1)\,3)$

$= \max(0, 1 - 3)$

$= \max(0, -2)$

$L = 0$ ⇒ correct prediction



$y + x - 1 = 0$

$\hat{y} = f(x, y) = 0$

Case 2: $\hat{y} = f'(x, y) = -3$ (Assumption)

$L = \max(0, 1 - (+1)(-3))$

$= \max(0, 1 + 3)$

$= \max(0, 4)$

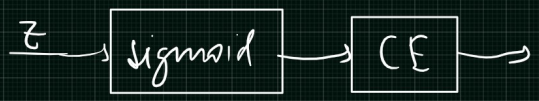
$L = 4$ ⇒ Wrong prediction

Cross Entropy Loss

$$CE = -\sum_{i=1}^{C} p \log(q)$$

(p: actual probabilities; q: predicted probabilities)

Binary cross Entropy loss

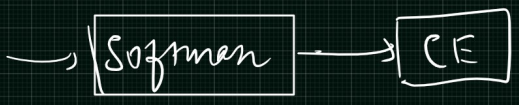


$$BCE = -\sum_{i=1}^{2} y \log(\sigma(z))$$

$$= -\sum_{i=1}^{2} y \log \hat{y}$$

$$BCE = -y \log \hat{y} - (1-y) \log(1-\hat{y})$$

Categorical cross entropy loss class ≥ 2



$$CCE = -\sum_{i=1}^{n} y_i \log(\text{softmax}(z_i))$$

$$= -\sum_{i=1}^{n} y_i \log(\hat{y})$$

if $n = 3$.

				sum
actual value	y_A	y_B	y_C	1
predicted value	$\hat{y}_A$	$\hat{y}_B$	$\hat{y}_C$	1

$$CCE = -y_A \log(\hat{y}_A) - y_B \log(\hat{y}_B) - y_C \log(\hat{y}_C)$$

	y_A	y_B	y_C
actual	0	1	0
pred	0_1	1_1	0_1
	$\hat{y}_A$	$\hat{y}_B$	$\hat{y}_C$

$$CCE = -\underbrace{0 \cdot \log(0)}_{0} - \underbrace{1 \log(1)}_{0} - \underbrace{0 \log(0)}_{0}$$

$$CCE = 0$$

pred 0.7 0.3 0

$$CCE = -1 \log(\hat{y}_B)$$

$$= -1 \cdot \log(0.3)$$

$$= -(-0.5)$$

$$CCE = +0.5$$

0 1 → -ve

Categorical cross entropy

y_A	y_B	y_C
0	1	0

$y_A = 1$
$y_B = 2$
$y_C = 3$

— Sparse categorical cross entropy

$y_A = 1$
$y_B = 2$
$y_C = 3$

sparse array → [... lot of zeros, few non zero values]

class →

10^6 ↓

$[0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0]$
$[0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$

$\{(0,3) \to 1$
$(1,2) \to 1\}$