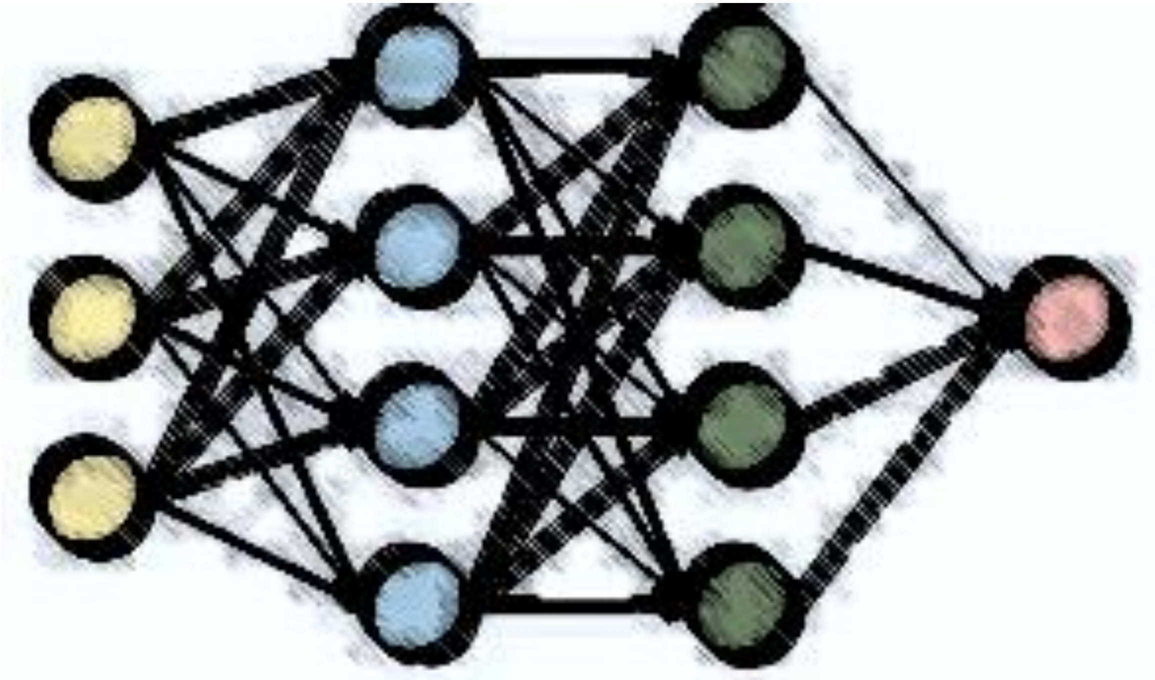
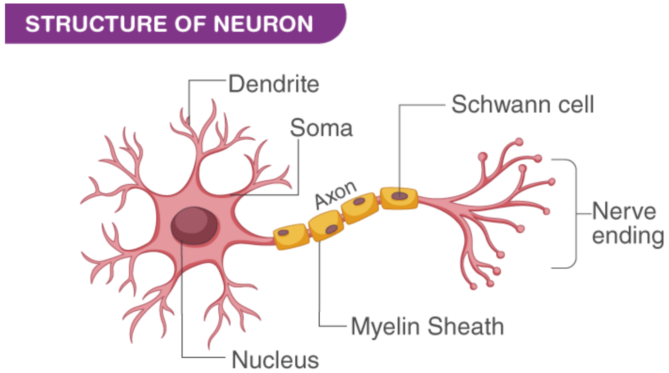
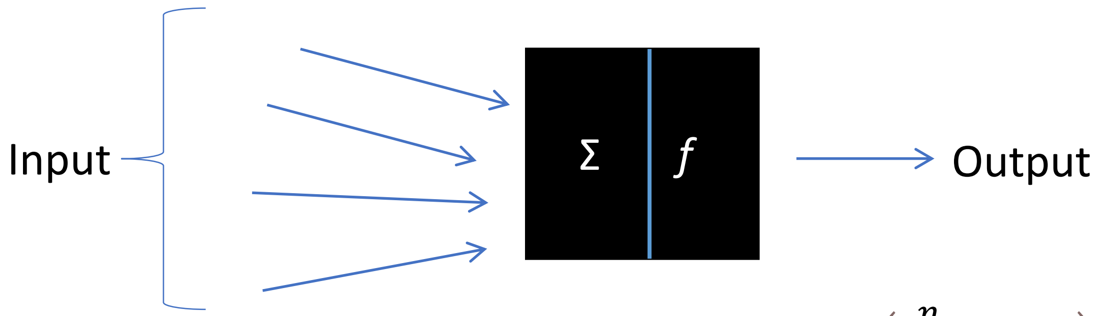
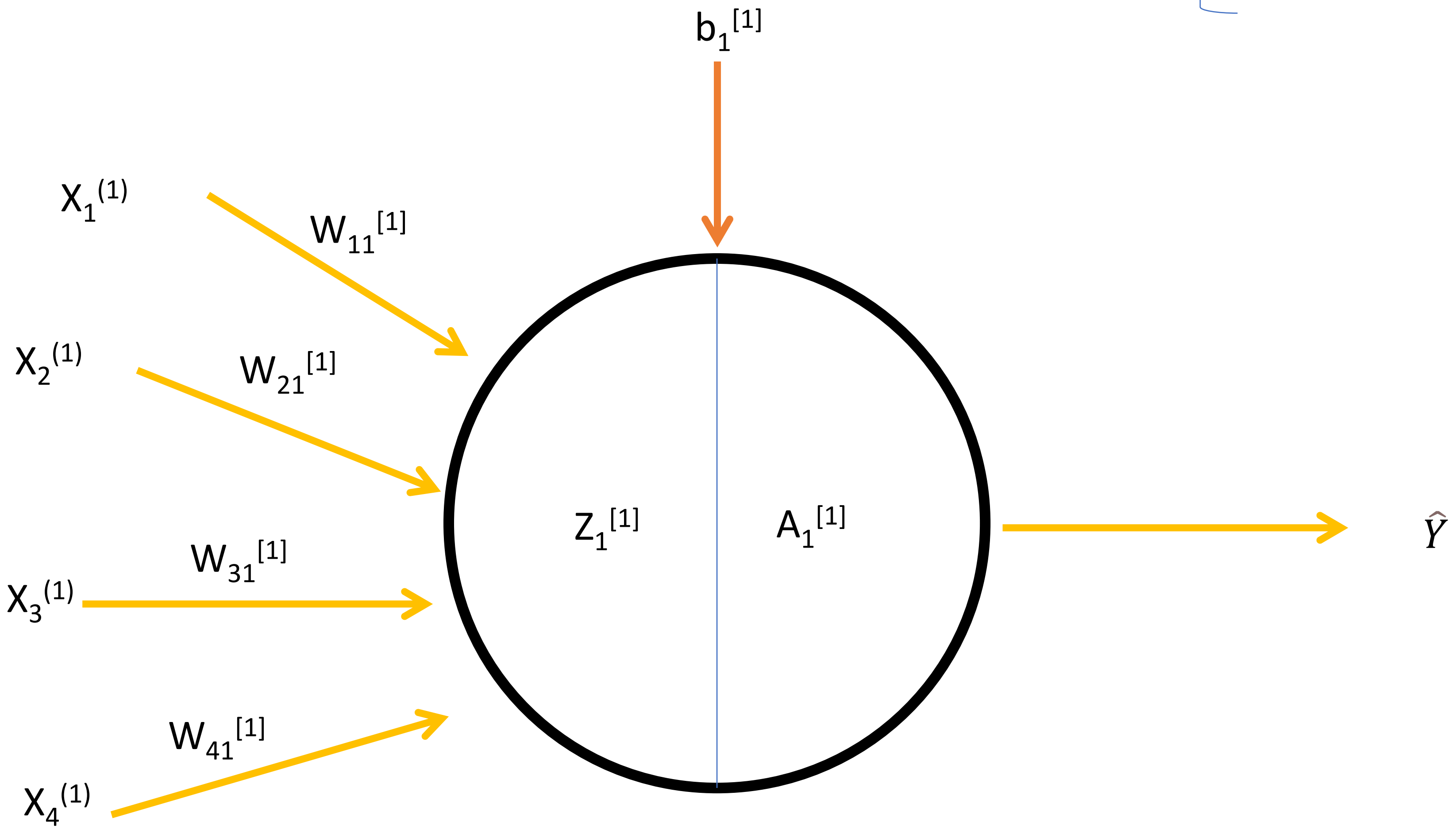
Neural Networks







$$Output = f\left(\sum_{i=1}^{n} X_i\, W_i\right)$$



Current Layer

$$W_{j,k}^{[i]}$$

From,to

Current Layer

$$b_j^{[i](k)}$$

Node index in that layer

Current Layer

Training example

$$Z_j^{[i](k)}$$

Node index in that layer

Current Layer

Training example

$$a_j^{[i](k)}$$

Node index in that layer

Single Training Example (i=1) and four input features

Current Layer

One example

$$Z_1^{[1](1)} = W_{11}^{[1]} X_1^{(1)} + W_{21}^{[1]} X_2^{(1)} + \cdots W_{41}^{[1]} X_4^{(1)} + b_1^{[1]} - (1)$$

Node

We can write equation no 1 in Matrix form as:

$$Z_1^{[1]} = \left(W_1^{[1]}\right)^T X \;\boldsymbol{+}\; +b_1^{[1]} \qquad -(2)$$

1 x 1   1 x 4   4 x 1   1 x 1

$$Z_1^{[1]} = [Z_1^{[1])(1)}]$$

$$W_1^{[1]} = \begin{bmatrix} W_{11}^{[1]} \\ W_{21}^{[1]} \\ W_{31}^{[1]} \\ W_{41}^{[1]} \end{bmatrix} \in R^{4X1}$$

$$X = \begin{bmatrix} X_1^{(1)} \\ X_2^{(1)} \\ X_3^{(1)} \\ X_4^{(1)} \end{bmatrix} \in R^{4X1}$$

$$Z_1^{[1]} = \left(W_1^{[1]}\right)^T X \;\mathbf{+}\; + b_1^{[1]} \qquad - (2)$$

$$a_1^{[1](1)} = \sigma\left(Z_1^{[1]}\right) = \hat{Y} \qquad - (3)$$

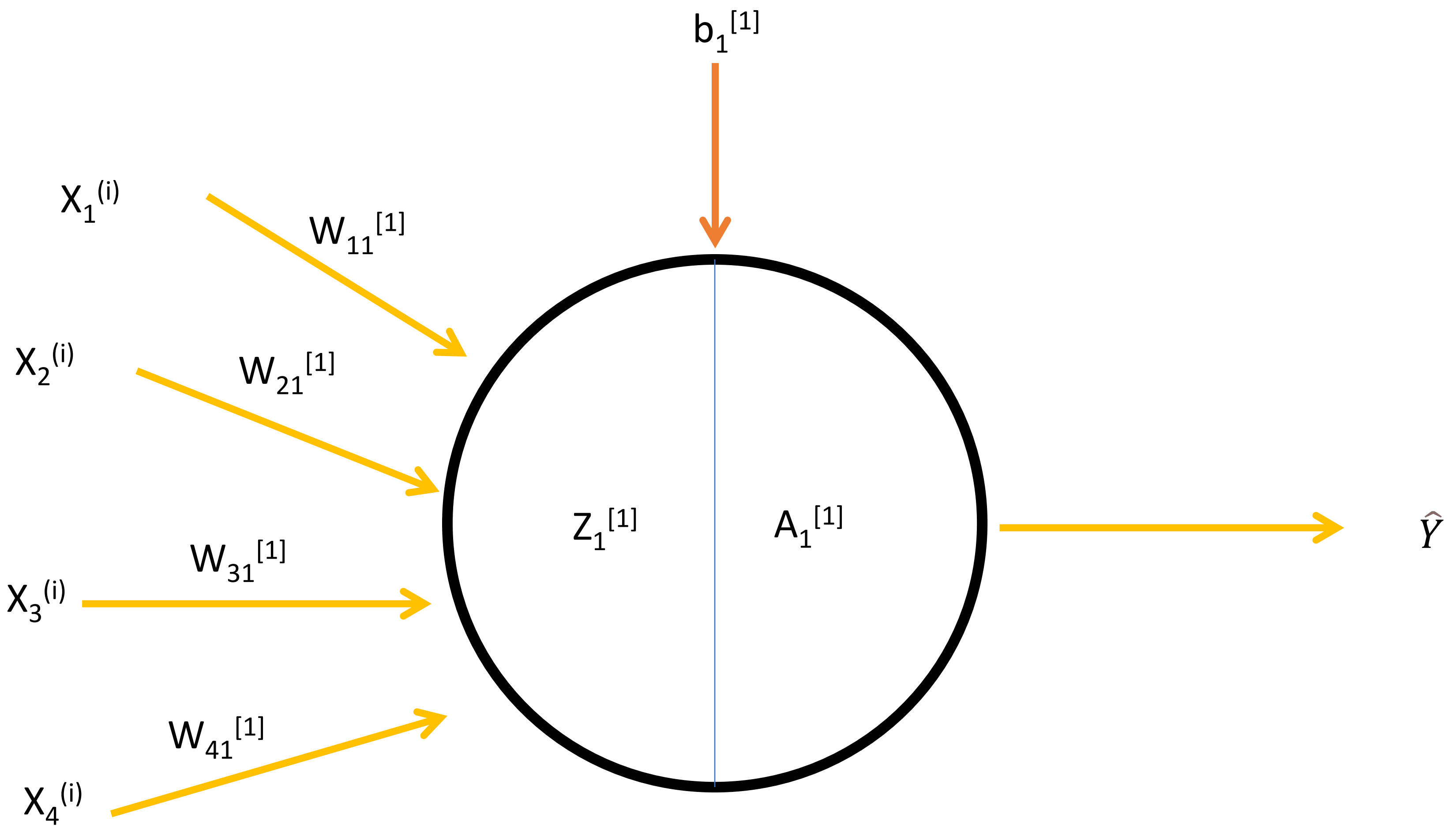
1 x 1

Activation Function

$$A_1^{[1]} = [a_1^{[1])(1)}]$$

Matrix Form

Multiple Training Examples (i = 1,2,3,4,5) and four features



For n number of features and m training examples we can write X as:

$$X = \begin{bmatrix} X_1^{(1)}\, X_1^{(2)} & \cdots & X_1^{(m)} \\ \vdots & \ddots & \vdots \\ X_n^{(1)} X_n^{(2)} & \cdots & X_n^{m} \end{bmatrix}$$

For 4 number of features and 5 training examples we can write X as:

$$X = \begin{bmatrix} X_1^{(1)}\, X_1^{(2)} & \cdots & X_1^{(5)} \\ \vdots & \ddots & \vdots \\ X_4^{(1)} X_4^{(2)} & \cdots & X_4^{(5)} \end{bmatrix}$$

We can write as:

$$Z_1^{[1](1)} = W_{11}^{[1]} X_1^{(1)} + W_{21}^{[1]} X_2^{(1)} + \cdots W_{41}^{[1]} X_4^{(1)} + b_1^{[1](1)}$$

$$Z_1^{[1](2)} = W_{11}^{[1]} X_1^{(2)} + W_{21}^{[1]} X_2^{(2)} + \cdots W_{41}^{[1]} X_4^{(2)} + b_1^{[1](2)}$$

$$\vdots \qquad \vdots$$

$$Z_1^{[1](5)} = W_{11}^{[1]} X_1^{(5)} + W_{21}^{[1]} X_2^{(5)} + \cdots W_{41}^{[1]} X_4^{(5)} + b_1^{[1](5)}$$

We can combine equations in Matrix form as:

$$Z_1^{[1](1)} = \left(W_1^{[1]}\right)^T X^{(1)} + \; + b_1^{[1](1)}$$

$$Z_1^{[1](2)} = \left(W_1^{[1]}\right)^T X^{(2)} + \; + b_1^{[1](2)}$$

$$Z_1^{[1](5)} = \left(W_1^{[1]}\right)^T X^{(5)} + \; + b_1^{[1](5)}$$

$$a_1^{[1](1)} = \sigma\left(Z_1^{[1](1)}\right)$$

$$a_1^{[1](2)} = \sigma\left(Z_1^{[1](2)}\right)$$

$$a_1^{[1](1)} = \sigma\left(Z_1^{[1](1)}\right)$$

$$a_1^{[1](2)} = \sigma\left(Z_1^{[1](2)}\right)$$

$$\vdots$$

$$a_1^{[1](5)} = \sigma\left(Z_1^{[1](5)}\right)$$

We can write as:

$$\underbrace{Z_1^{[1]}}_{1 \times 5} = \underbrace{\left(W_1^{[1]}\right)^T}_{1 \times 4} \underbrace{X}_{4 \times 5} + \underbrace{+b_1^{[1]}}_{1 \times 5} \qquad -(3)$$

n = 4 features and m = 5 Training examples

$$Z_1^{[1]} = \left(W_1^{[1]}\right)^T X \ + \ +b_1^{[1]} \qquad -(3)$$

1 x 5 **1 x 4** **4 x 5** **1 x 5**

$$Z_1^{[1]} = [Z_1^{[1])(1)} Z_1^{[1])(2)} \dots Z_1^{[1])(5)}]$$

$$b_1^{[1]} = [b_1^{[1])(1)} b_1^{[1])(2)} \dots b_1^{[1])(5)}]$$

$$A_1^{[1]} = \sigma\left(Z_1^{[1]}\right) = \hat{Y} \qquad -(4)$$

1 x 5 **1 x 5**

Generalize Form:

$$Z_1^{[1]} = \left(W_1^{[1]}\right)^T X \;\mathbf{+}\; +b_1^{[1]} \qquad - (5)$$

1 x m	*1 x n*	*n x m*	*1 x m*

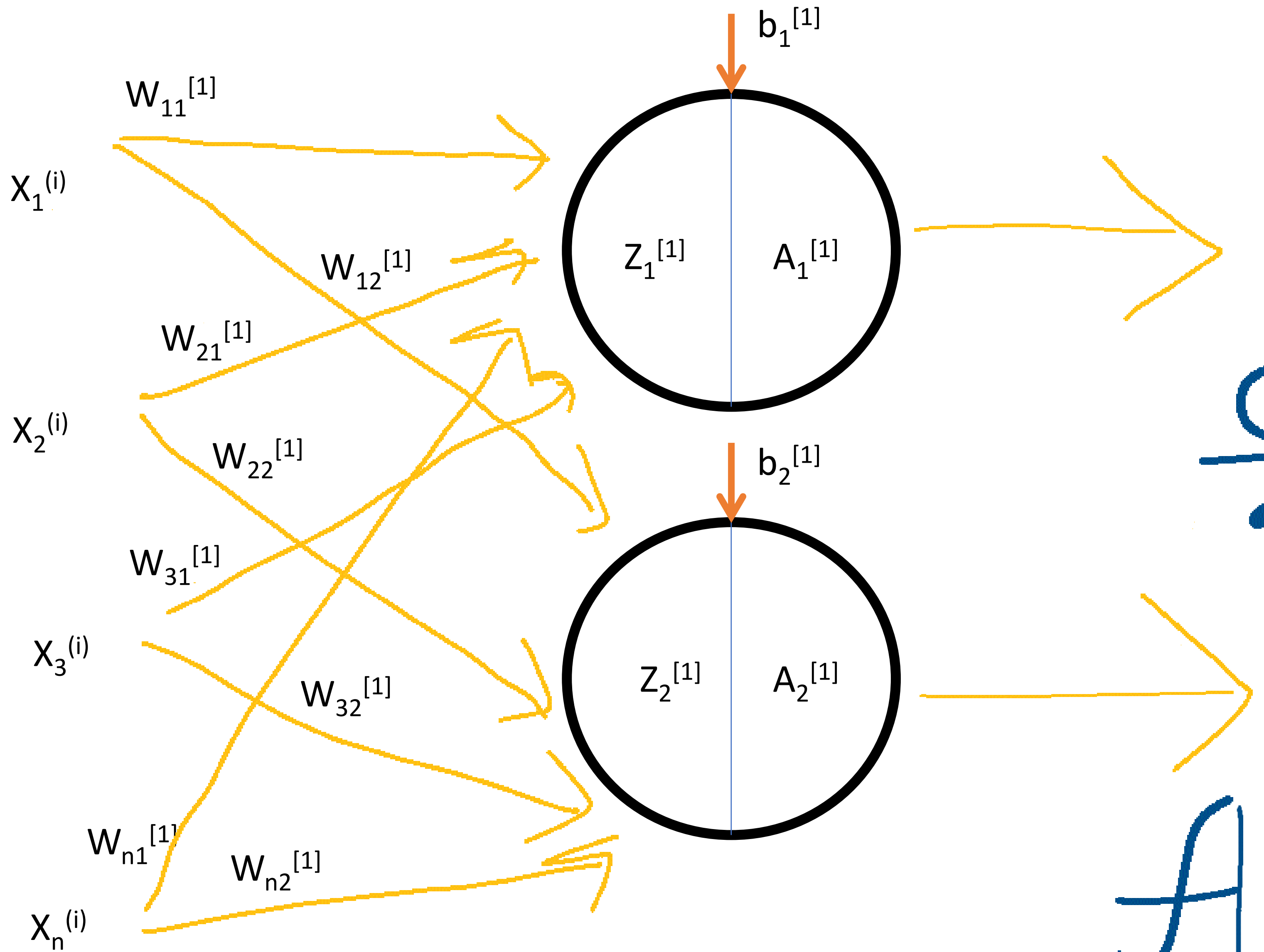
This is called as Single Layer Perceptron –Having Single node

Example of Single Neuron

And, OR and XoR implementation using NN done on board in a class

MultiPerceptron

Multi Perceptron



1st Hidden Layer

$\hat{Y}1$

$Z^{[1]} = W^{[1]}X + B^{[1]}$

$\hat{Y}2$

$A^{[1]} = \sigma(Z^{[1]})$

Multi-Layer Perceptron

For Node 2 we can write

$$Z_2^{[1]} = \left(W_2^{[1]}\right)^T X \ + +b_2^{[1]}$$

Hence, for First Node, Second Node and P nodes

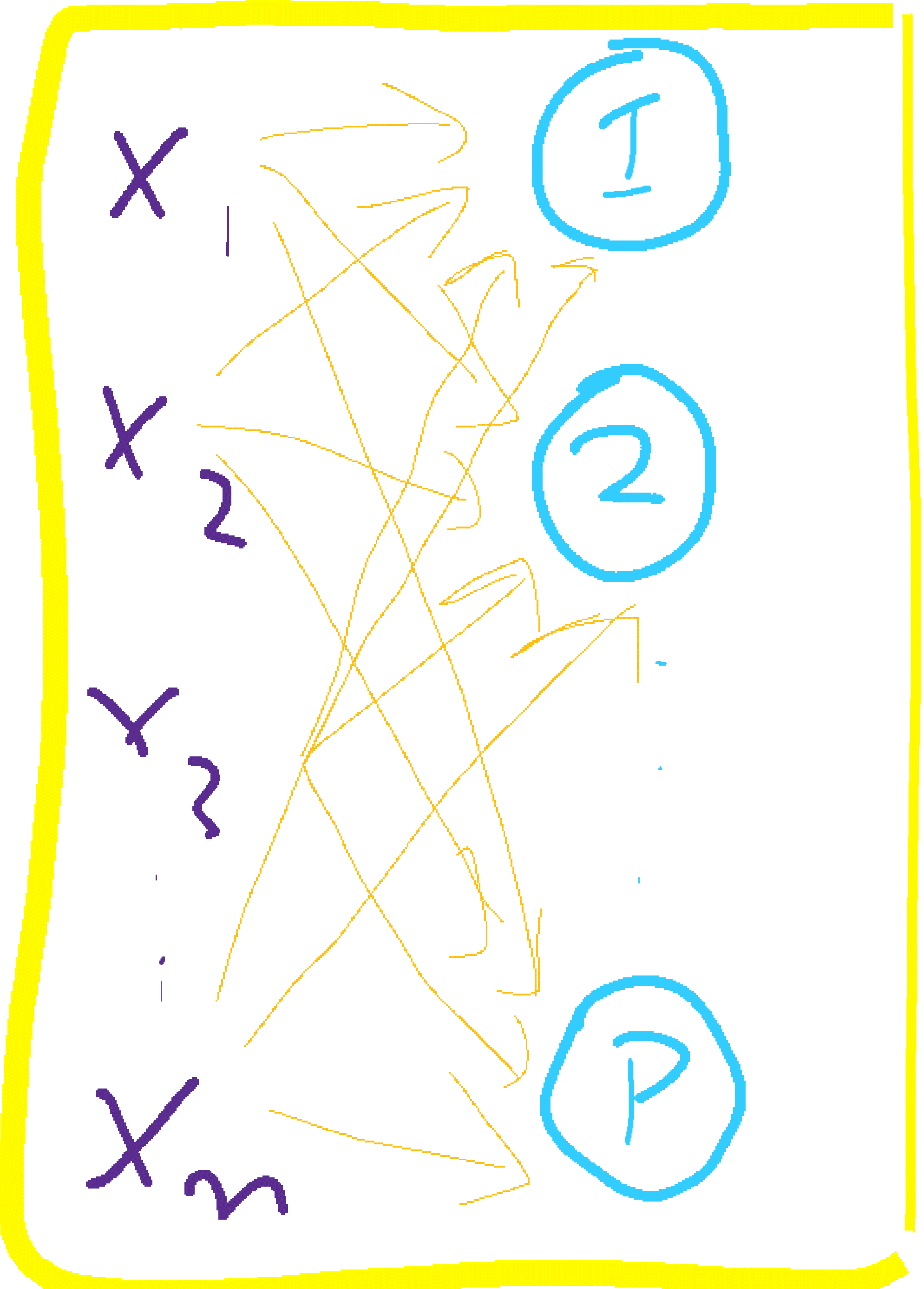
$$Z_1^{[1]} = \left(W_1^{[1]}\right)^T X \ + +b_1^{[1]}$$

$$Z_2^{[1]} = \left(W_2^{[1]}\right)^T X \ + +b_2^{[1]}$$

$$Z_p^{[1]} = \left(W_p^{[1]}\right)^T X \ + +b_p^{[1]}$$

$$Z^{[1]} = \left(W^{[1]}\right)^T X \ + +B^{[1]} \qquad (6)$$

p x m **p x n** **n x m** **p x m**



where,

Current layer

Training example

$$Z^{[1]} = \begin{bmatrix} Z_1^{[1]} \\ Z_2^{[1]} \\ . \\ . \\ Z_p^{[1]} \end{bmatrix} = \begin{bmatrix} Z_1^{[1](1)} & Z_1^{[1](2)} & \cdots & Z_1^{[1](m)} \\ \vdots & & \ddots & \vdots \\ Z_P^{[1](1)} & Z_P^{[1](2)} & \cdots & Z_P^{[1](m)} \end{bmatrix}$$

Current node

$$W^{[1]} = \begin{bmatrix} W_1^{[1]} \\ W_2^{[1]} \\ . \\ . \\ W_p^{[1]} \end{bmatrix}^T = \begin{bmatrix} W_{11}^{[1]} & \cdots & W_{n1}^{[1]} \\ \vdots & \ddots & \vdots \\ W_{1P}^{[1]} & \cdots & W_{nP}^{[1]} \end{bmatrix}$$

$$B^{[1]} = \begin{bmatrix} b_1^{[1]} \\ b_2^{[1]} \\ . \\ . \\ b_p^{[1]} \end{bmatrix} = \begin{bmatrix} b_1^{[1](1)} & b_1^{[1](2)} & \cdots & b_1^{[1](m)} \\ \vdots & & \ddots & \vdots \\ b_P^{[1](1)} & b_P^{[1](2)} & \cdots & b_P^{[1](m)} \end{bmatrix}$$

And,

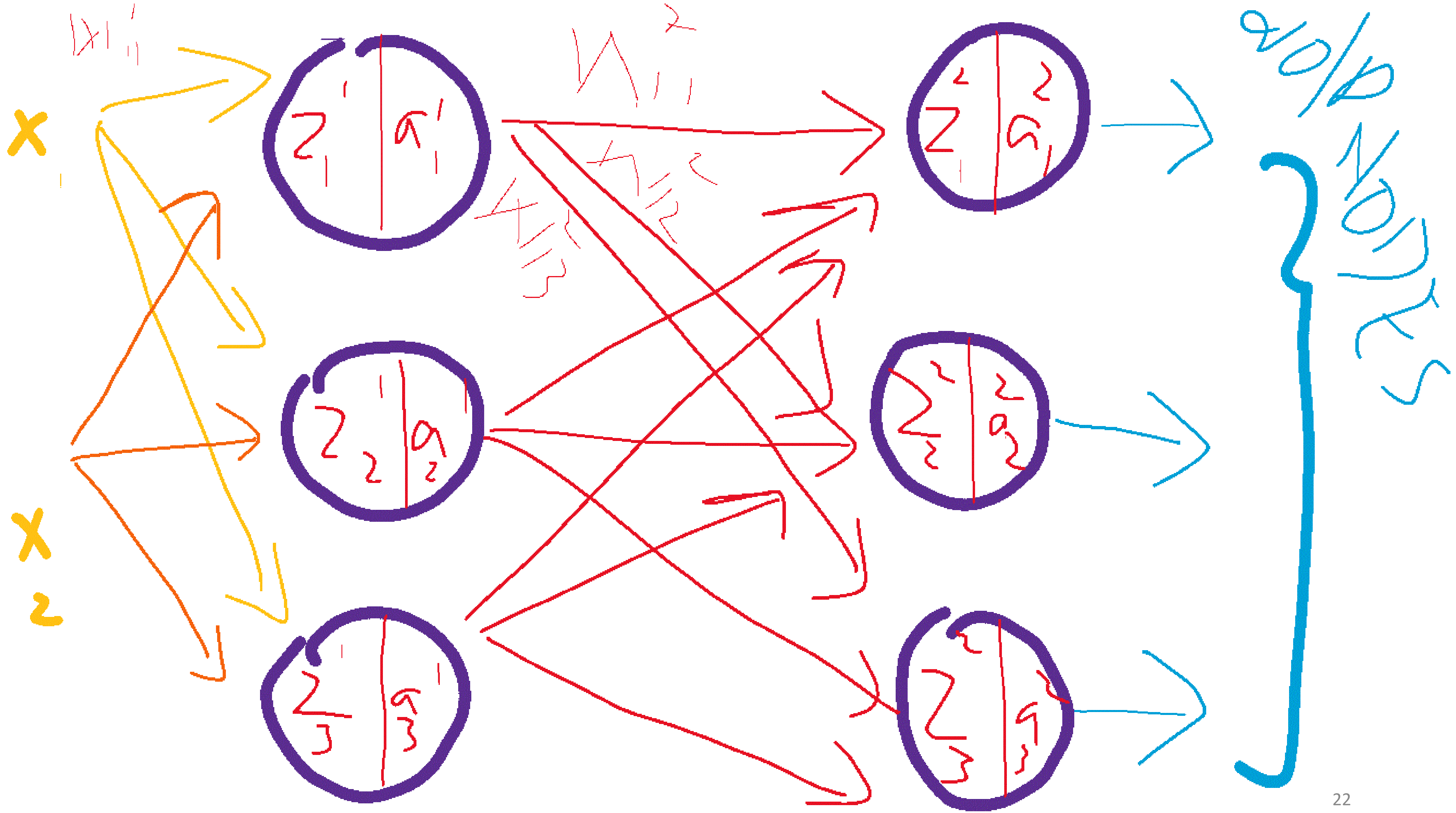
$$A^{[1]} = \sigma\left(Z^{[1]}\right) \qquad -(7)$$

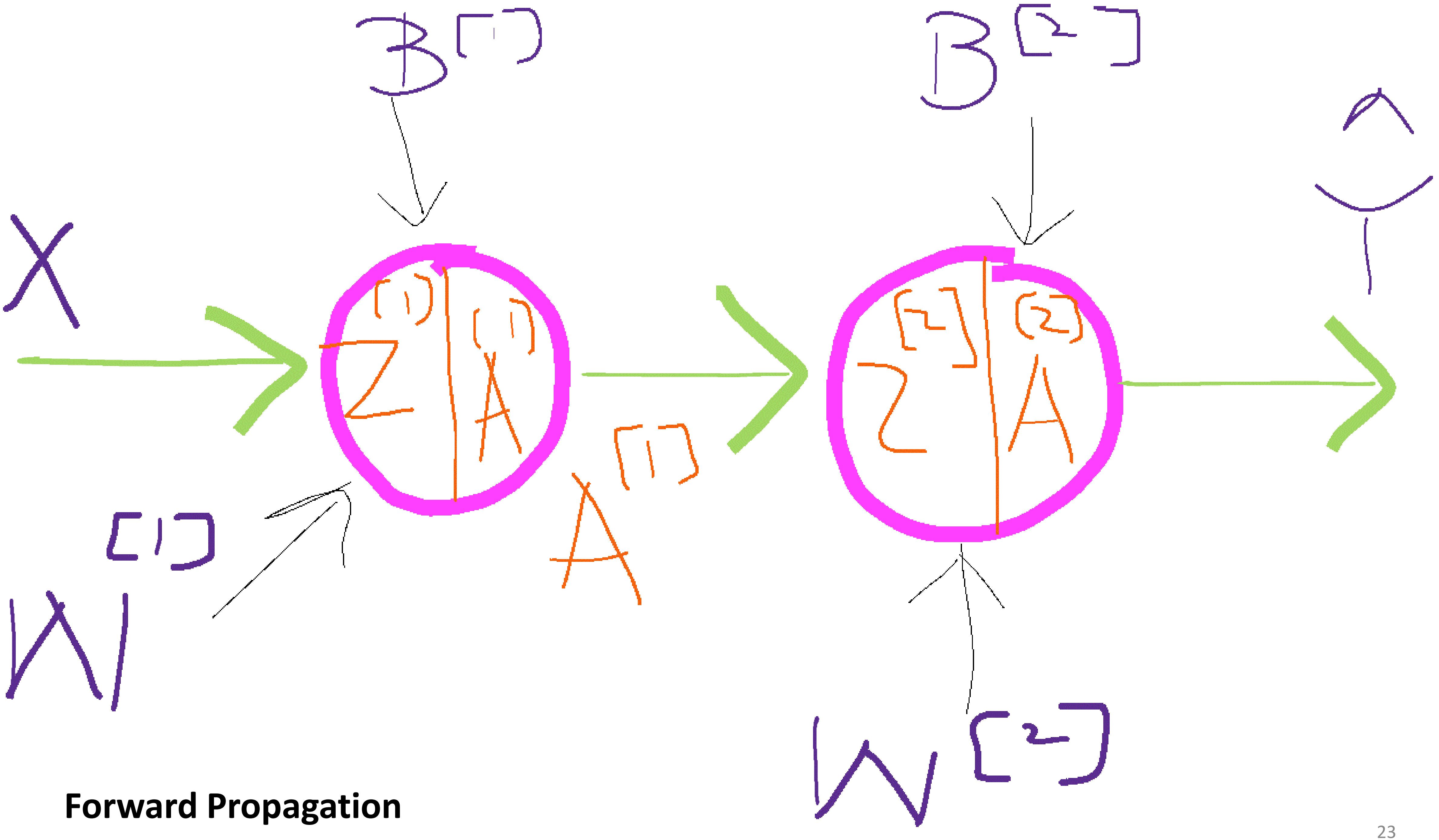
p x m **p x m**

where,

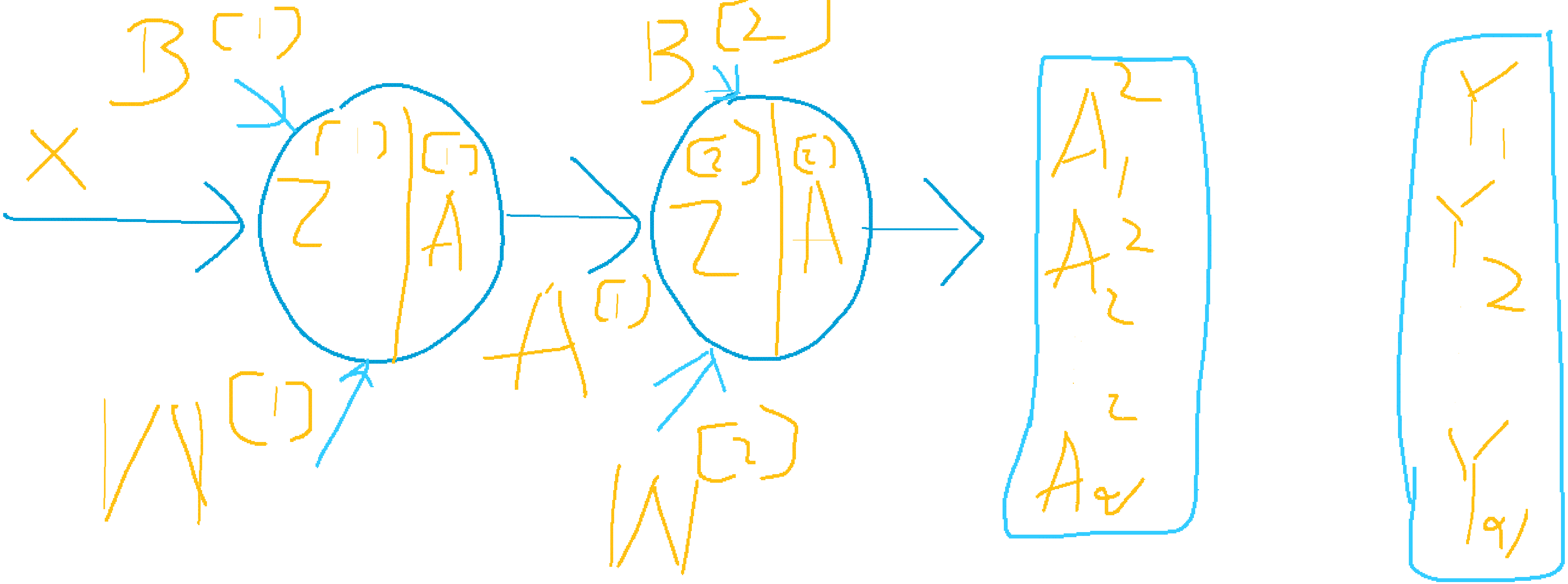
$$A^{[1]} = \begin{bmatrix} a_1^{[1]} \\ a_2^{[1]} \\ . \\ . \\ a_p^{[1]} \end{bmatrix} = \begin{bmatrix} a_1^{[1](1)} & a_1^{[1](2)} & \cdots & a_1^{[1](m)} \\ & \vdots & \ddots & \vdots \\ a_P^{[1](1)} & a_P^{[1](2)} & \cdots & a_P^{[1](m)} \end{bmatrix}$$

Multi-Layer Perceptron


x_1
x_2
W^1_{11}
W^2_{11}
W^2_{12}
W^2_{13}
z^1_1 | a^1_1
z^1_2 | a^1_2
z^1_3 | a^1_3
z^2_1 | a^2_1
z^2_2 | a^2_2
z^2_3 | a^2_3
o/p nodes



Forward Propagation



Loss = (A-Y)

A – Predicted

Y – Actual Labels

For 1st Layer

$$A^{[1]} = \sigma\left(Z^{[1]}\right) \qquad - (8)$$

p x m p x m

where,

$$Z^{[1]} = \left(W^{[1]}\right)^T X \ + \ +B^{[1]}$$

p x m p x n n x m p x m

For 2nd Layer

$$A^{[2]} = \sigma\left(Z^{[2]}\right) \qquad - (9)$$

q x m q x m

where,

$$Z^{[2]} = \left(W^{[2]}\right)^T A^{[1]} + \ +B^{[2]}$$

q = output nodes

q x m q x p p x m q x m

Let q = 2 i-e two output nodes

In this case: Dimension of $A^{[2]}$ is 2 x m

$$A^{[2]} = \begin{bmatrix} A_1^2 \\ A_2^2 \end{bmatrix} = \begin{bmatrix} A_1^2 \\ 1 - A_2^2 \end{bmatrix}$$

Sigmoid output represents probabilities

True output:

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} Y_1 \\ 1 - Y_1 \end{bmatrix}$$

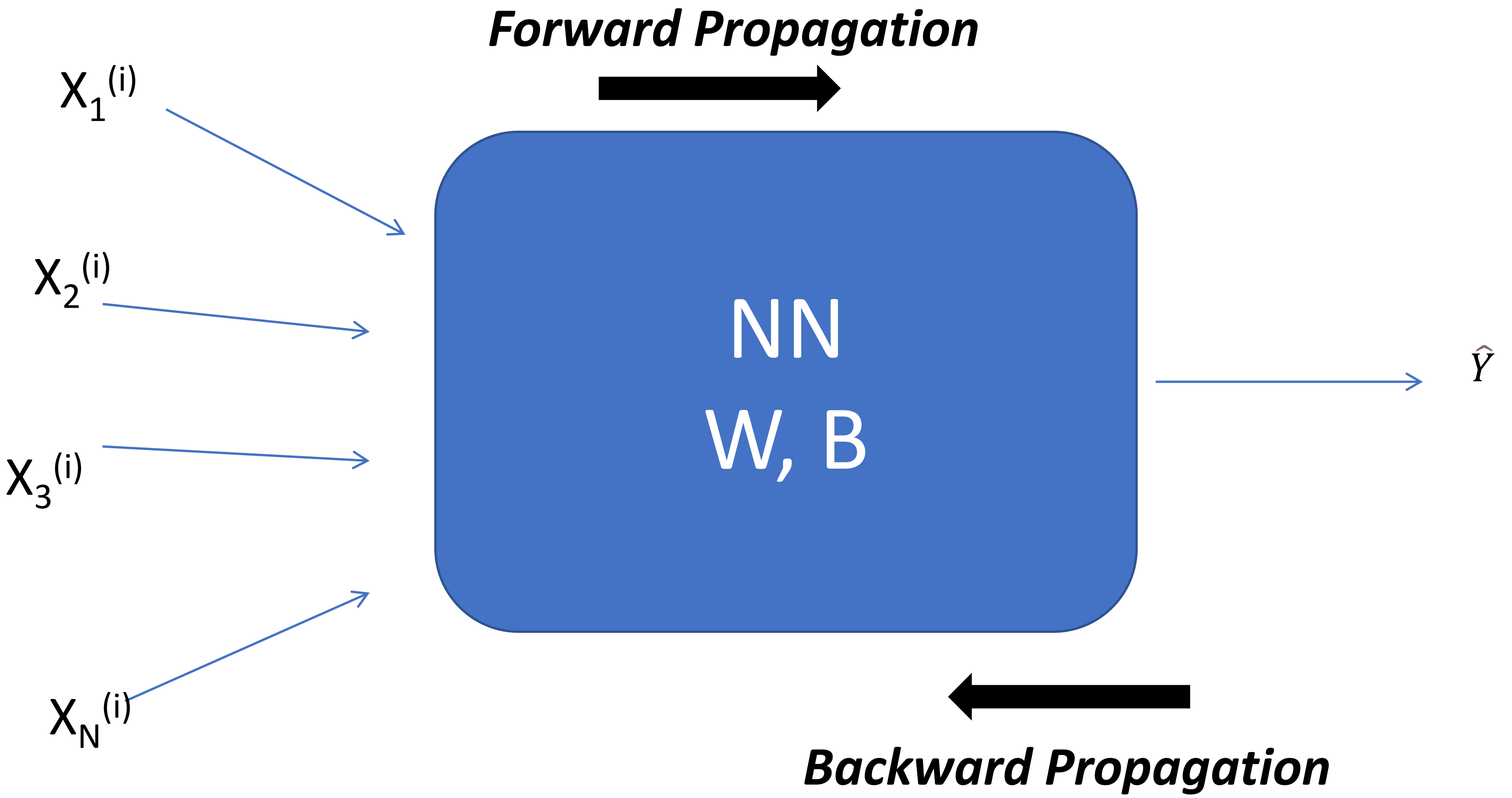
Loss Function:

$$L = -\sum_{i=1}^{q} Y_i log A_i^{[2]}$$

Since q = 2, binary class

Explore : Playground.tensorflow.org

Back Propapgation



Forward Propagation
$X_1^{(i)}$
$X_2^{(i)}$
$X_3^{(i)}$
$X_N^{(i)}$
NN
W, B
$\hat{Y}$
Backward Propagation

Back Propagation (BP)

- We train NN – Forward propagation.

- Backpropagation- Transforming information from output to input.

- We actually tune the weighs and bias during Backpropagation.

Updating weights as:

$$w^{new} = w^{old} - \alpha \frac{dl}{dw}$$

- Updating bias as:

$$B^{new} = B^{old} - \alpha \frac{dl}{dB}$$

$\frac{dl}{dw}$ $\frac{dl}{dB}$ Both are important to calculate and adjust W and B

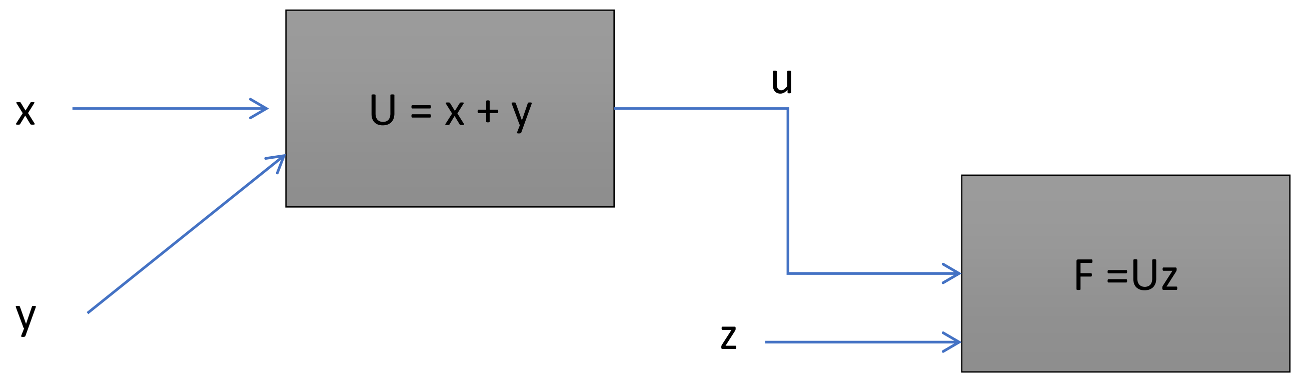
The concept of Computation Graph

- Directed graphs where the nodes correspond to the mathematical operation.

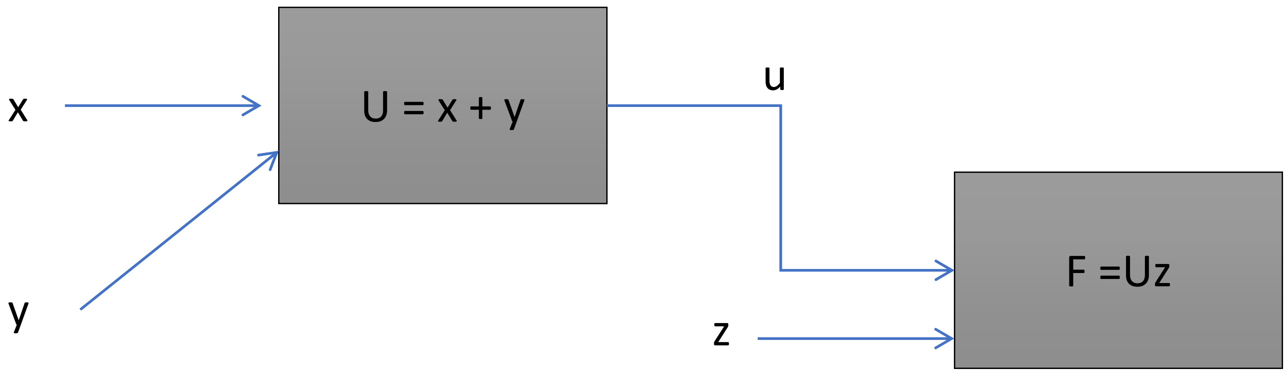
$$F = (x + y)z$$

x, y and z are the forward propagating values.

We can show as,



X = 1, Y = 2 and Z = 3
F = 9



X = 1, Y = 2 and Z = 3
F = 9

$$Now\ find, \frac{\partial F}{\partial x}, \frac{\partial F}{\partial y} \text{ and } \frac{\partial F}{\partial z}$$

$$U = x + y\ and\ F = Uz \text{ and } \mathrm{F = xz + yz}$$

$$\frac{\partial F}{\partial x} = \left[\frac{\partial F}{\partial u}\ \frac{\partial u}{\partial x}\right] = z \qquad \frac{\partial F}{\partial z} = x + y = u$$

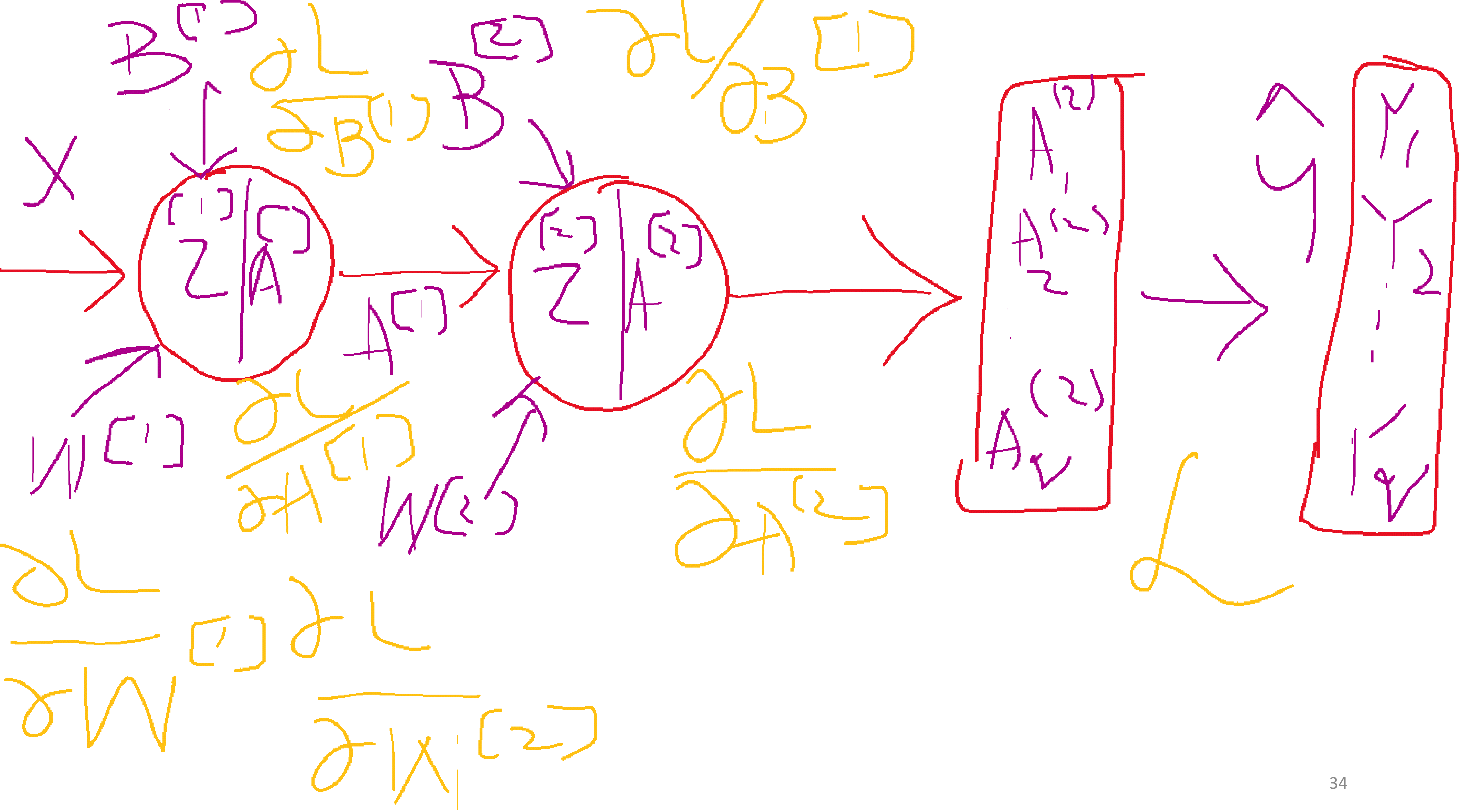
$$\frac{\partial F}{\partial y} = z \cdot 1 = z$$

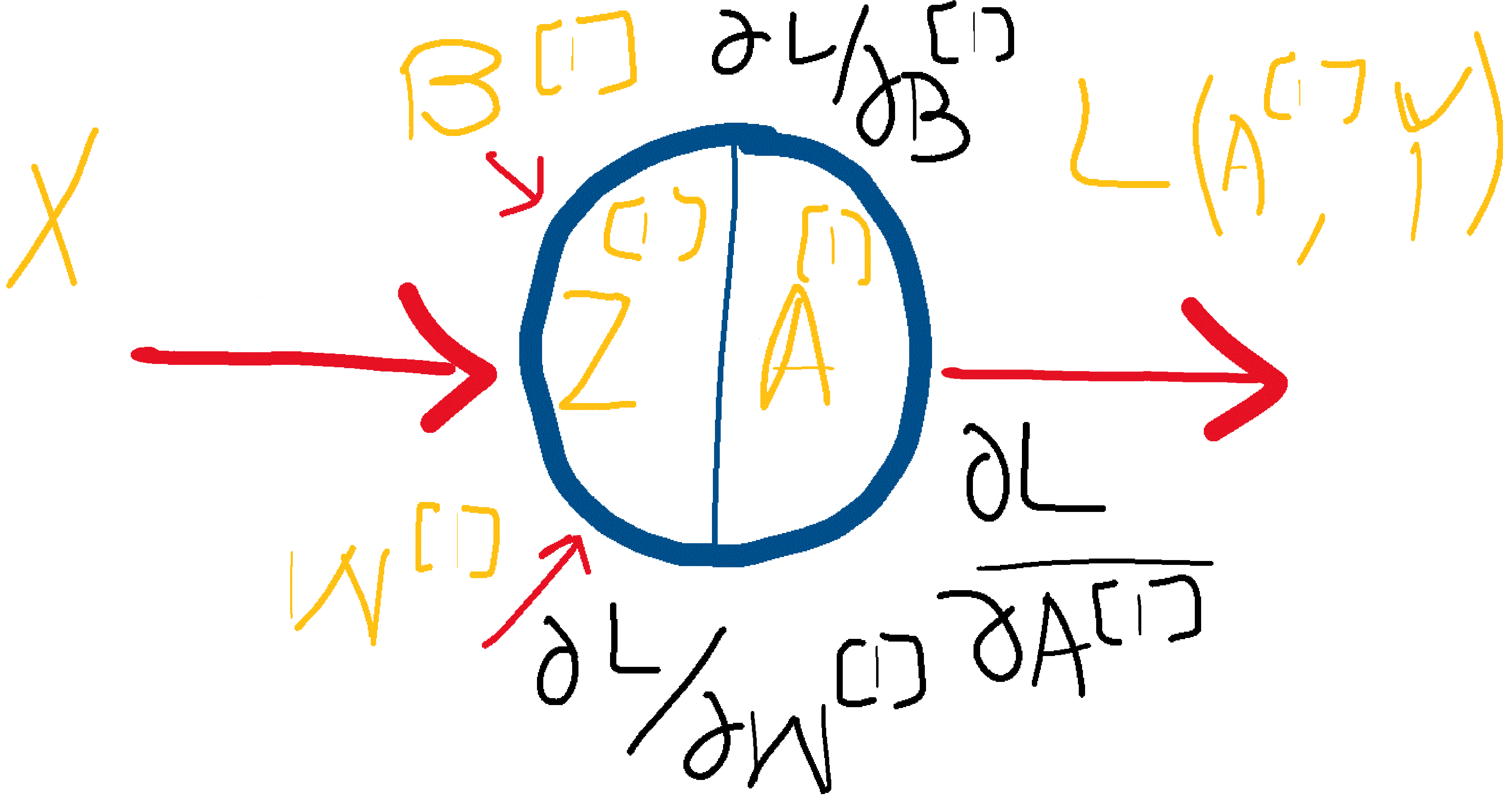
Example 2: $F = 5(XY + Z)$

Self try:

Example 3: $F = 8[\ (x + y)(y+z) + yz]$

- Self try:


X
B[1]
∂L/∂B[1]
B[2]
∂L/∂B[1]
Z[1] | A[1]
A[1]
Z[2] | A[2]
W[1]
∂L/∂A[1]
W[2]
∂L/∂A[2]
A_1^(2)
A_2^(2)
A_N^(2)
ŷ
y_1
y_2
y_N
L
∂L/∂W[1]
∂L/∂W[2]


X
$B^{[1]}$
$\partial L / \partial B^{[1]}$
$L(A^{[1]}, Y)$
$Z^{[1]}$
$A^{[1]}$
$W^{[1]}$
$\frac{\partial L}{\partial A^{[1]}}$
$\partial L / \partial W^{[1]}$

Board Derivation:

Discussed in a class.

End