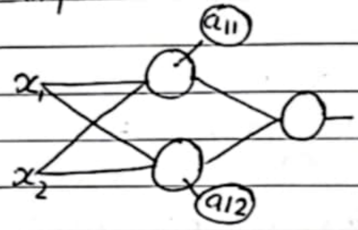
normalize = $\begin{bmatrix} mean \to 0 \\ Sd \to 1 \end{bmatrix}$

Batch Normalization – It is used to speed up Neural network training

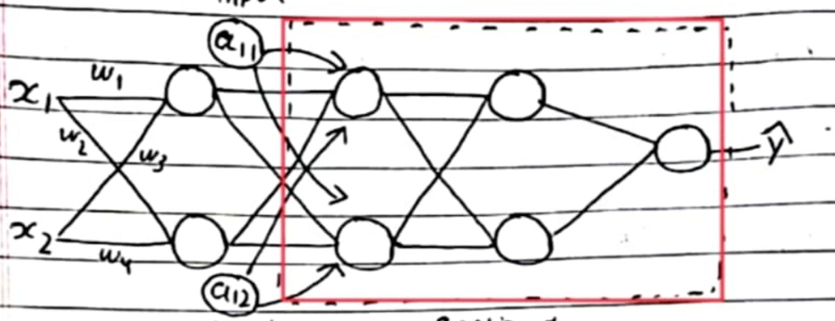
It normalizes the activation vectors from hidden layer using mean and variance of current batch

Example:



We normalize the input x_1, x_2 (It's normal case) but In Batch normalization process we also normalize the $[a_{11}, a_{12}]$.

Concept:

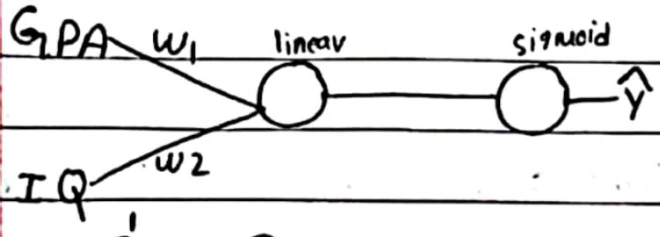


→ Section 1 lai xutai individual neural network socham. Section 1 ko first layer ko nodes ley input a_{11} and a_{12} linxa. a_{11} and a_{12} values after one batch of training change vairakxa ra yesko distribution ne change hunxa kina vane Euta batch ko training vaye pani weight update hunxa. a_{11} ra a_{12} (w_1, w_2, w_3, w_4) ma depend xa ra harek batch ko training vaye pani $(w_1, w_2, w_3$ ra $w_4)$ pani change hunxa so, section 1 ko nodes ley different distribution input linxa Tale training stable hunna ra model ley na chani kura pani learn garxa ra slow hunxa so, we use concept of batch normalization.

Jaba a_{11} ra a_{12} normalize hunxa yesko values between 0 and 1 ko range ma hunxa so, Section 1 ko layer nodes ley stable data pauxan ra training faster hunxa.

[Internal covariance shift – change of distribution of activation due to change of network parameters during training]

GPA	IQ	Placement
8.9	100	1
6.2	25	0
8.3	2	0
9.2	70	1



$$z^1_{11} = 8.9 \times w_1 + 100 \times w_2 + b_{11}$$
$$= M^1$$

[z^1_{11} — 1 → first row batch ko; 11 → first node; first layer]

[M^1 → linear output of first row]

Similarly we will have M^2, M^3, M^4

Our batch size is 4 so, in one batch we have 4 rows so, we will get 4 outputs [M^1, M^2, M^3, M^4]

Now, we will normalize them using formula

$$z^i_{nor} = \frac{m^i - u}{\sigma} \quad [\text{ i is row no of batch}]$$

$$\therefore u = \frac{1}{n} \sum_{i=1}^{n} m^i$$

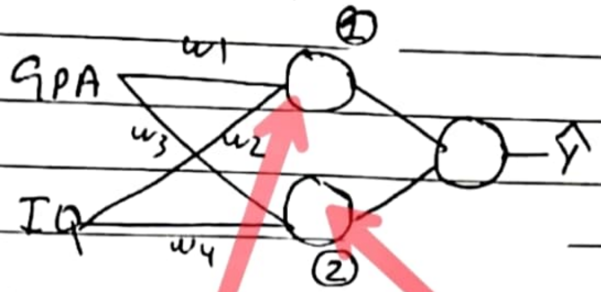
$$\Rightarrow \frac{[M^1 + M^2 + M^3 + M^4]}{4 \text{ (batchsize)}} \quad [\text{n is batch size}]$$

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{2} (m^i - u)}$$

Now, let's do for $z^1_{nor} = \frac{m^1 - u}{\sigma + \varepsilon}$ (ε is small value if σ is 0 the ε will prevent it from undefined)

Similary we will get ($z^1_{nor}, z^2_{nor}, z^3_{nor}, z^4_{nor}$)

If it has multiple nodes like below figure



GPA	IQ	Plac
1	2	1
10	11	0

we have to calculate μ and σ seperately for 1, 2 nodes and do normalize.

Node 1 outputs:

→ $1 \cdot w_1 + 2 \times w_3 + b_{11}$

→ $10 \cdot w_1 + 11 \times w_3 + b_{11}$

(calculate seperate μ and σ and normalize)

Node 2 outputs

$1 \cdot w_2 + 2 \times w_4 + b_{12}$

$10 \cdot w_2 + 11 \times w_4 + b_{12}$

(calculate seperate μ and σ and normalize)

Now,

In depth Intution

$$\boxed{z'_{11} \rightarrow z'_{11}nor \rightarrow P'_{11} \rightarrow g(P'_{11}) \rightarrow a'_{11}}$$

(z'_{11} first layer ko first node ley batch ko first row ko output)

Here $\boxed{P = \gamma \cdot z + \beta}$ (γ and β are learnable parameters whose values can be find during training)

Jaile normalize garda ramro accuracy na aauna sakxa tei vara γ and β learnable parameter rakheko yo pani w and b update vako jasari update hunxa during back propagation like $\boxed{\gamma_{new} = \gamma_{old} - \frac{d\ Loss}{d\gamma_{old}}}$

Generally $\gamma = 1$ and $\beta = 0$ during first time.