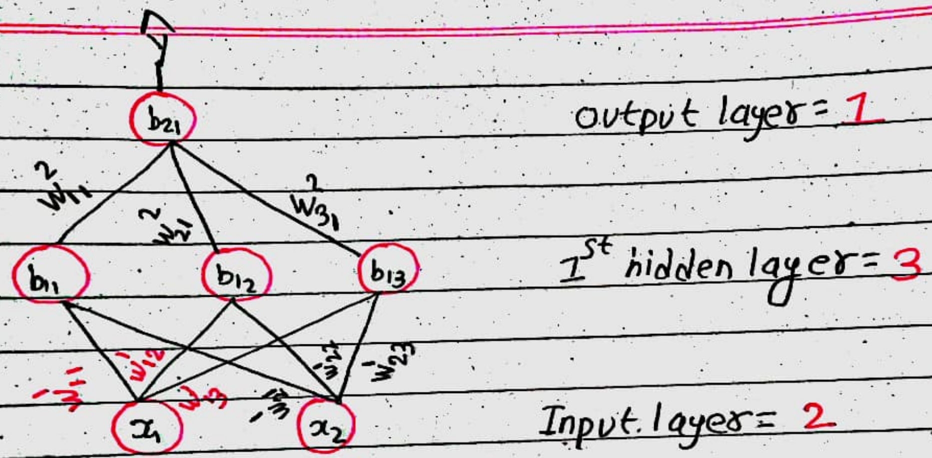
Dropout layers

Suppose hami sanga yo neural Network xa Jasle euta classification problem solve garxa.



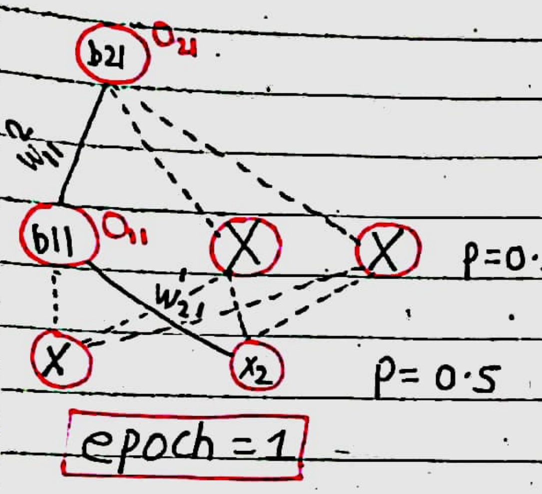
fully connected Neural Network

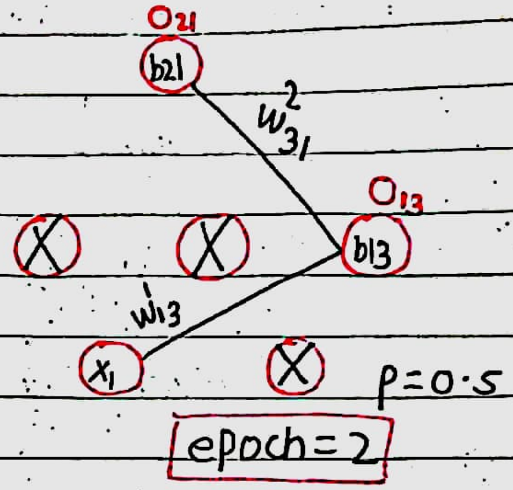
when we use this complex Architecture there will be the problem of the overfitting so, we use dropout in order to Solve this problem

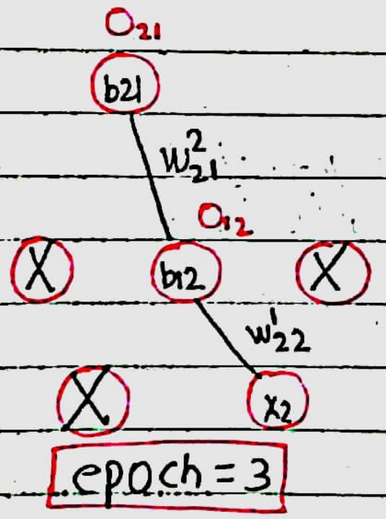
The concept of dropout:

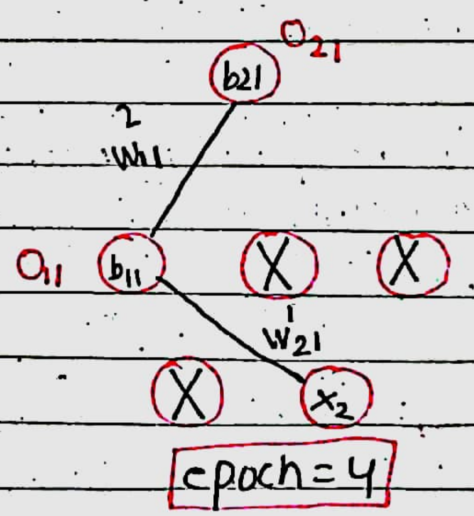
Suppose hami afno neural network epoch = 4 ko lagi train gardai xam. Jaba hamiley dropout use garxam hamro neural network ma It will randomly switch off Some of the nodes of input and hidden layer in each epoches.

aba harmo neural Network ko archicture herek epoch ko lagi different hunxa like down:-









Now, we have dataset as:

Age	purchase	Salary
5	0	0
20	1	200
10	0	50
30	1	70

Suppose we took batch = 2

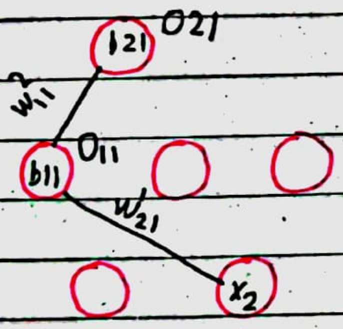
Batch size = $\frac{\text{Total row}}{\text{batch}} = \frac{4}{2} = 2$

x_1 Age	x_2 Purchase	y Sal
20	1	200
30	1	70

x_1 Age	x_2 Purchase	y Sal
5	0	0
10	0	50

At first, we initilize the parameters (All) of our Neural Network.

At epoch 1



During Forward propogation, we calculate

$O_{11} = x_2 \cdot w^1_{21} + b_{11}$

$O_{21} = O_{11} \cdot w^2_{11} + b_{21}$

(Batch 1 ko first row ko x_2 feature Janxa & calculate hunxa, 2nd row ko x_2 feature Janxa feri & calculate hunxa, Sabai row sakepaxi batch ko loss calculate hunxa ani back propogation vai (w^1_{21}, w^2_{11}, b_{11}, b_{21}) update hunxa.)

(Aba bhakar updated weights ma arko batch 2 ko dubai row ko x_2 feature NN ma Janxan, loss calculate hunxa feri weights update hunxa)

let's see how weight update takes place:
(Suppose we have linear activations to all nodes)

$$\boxed{w^2_{11_{new}} = w^2_{11_{old}} - \alpha \underbrace{\boxed{\frac{d\,loss}{dw^2_{11_{old}}}}}_{A}}$$

$[w^2_{11} \rightarrow O_{21} \rightarrow loss]$

$$\frac{d\,loss}{dw^2_{11}} = \frac{d\,loss}{dO_{21}} \times \frac{dO_{21}}{dw^2_{11}}$$

$$= \left\{ \sum_{i=1}^{2} \frac{d(y_i - \hat{y}_i)^2}{d(y-\hat{y})} \times \frac{d(y-\hat{y})}{dO_{21}} \right\} \times \frac{dO_{21}}{dw^2_{11}}$$

$$= \left\{ \sum_{i=1}^{2} 2(y_i - \hat{y}_i) \times (0-1) \right\} \times \frac{d[O_{11} \cdot w^2_{11} + b_{21}]}{dw^2_{11}}$$

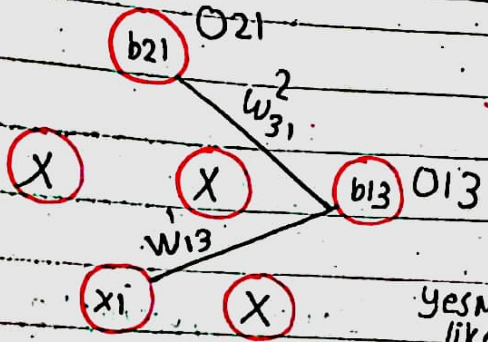
$$= \left\{ \sum_{i=1}^{2} 2(\hat{y}_i - y_i) \right\} \times O_{11}$$

$$= 2[(\hat{y}_1 - y_1) + (\hat{y}_2 - y_2)] \times O_{11}$$

$$= 2[(\hat{y}_1 - y_1) + (\hat{y}_2 - y_2)] \times [X_2 \cdot w^1_{21} + b_{11}]$$

$$\boxed{\therefore\; w^2_{11_{new}} = w^2_{11_{old}} - \alpha \times \left[2[(\hat{y}_1 - y_1) + (\hat{y}_2 - y_2)] \times [X_2 \cdot w^1_{21} + b_{11}] \right]}$$

Similary for epoch 2



During forward propogation, we calculate

$$O13 = x1 \cdot w^1_{13} + b13$$
$$O21 = O13 \cdot w^2_{31} + b21$$

yesma last updated weight xa vane tesmai train huni vayo like b21

(Similarly yesma pani agi Jastai hunxa. Batch 1 ko row 1 bata x_1 feature Janxa, tespaxi row 2 bata x_1 feature Janxa. dubai ko ⇑ palai pilo calculate hunxa. loss calculate hunxa, backpropagation ma parameters (b_{21}, b_{13}, w^2_{31}, w^1_{13}) update hunxa)

(feri bhakar update vayeko parameters ma batch 2 ko data train garinxa, ⇑ garinxa, loss calculate garinxa and feri back propagation use gari parameters update garinxa)

yesari harek epoch ma naya naya archicture form hunxa kunai node off kunai on vayera. ani weights pani Sabai update vai rako hunxa

Dropout ratio (p):

Suppose I choose dropout ratio as 0.5 i.e $\boxed{p=0.5}$. yesko matlab yesle randomly harek epoch ma 50% Neuron lai off gardinxa.

Why does dropout works?

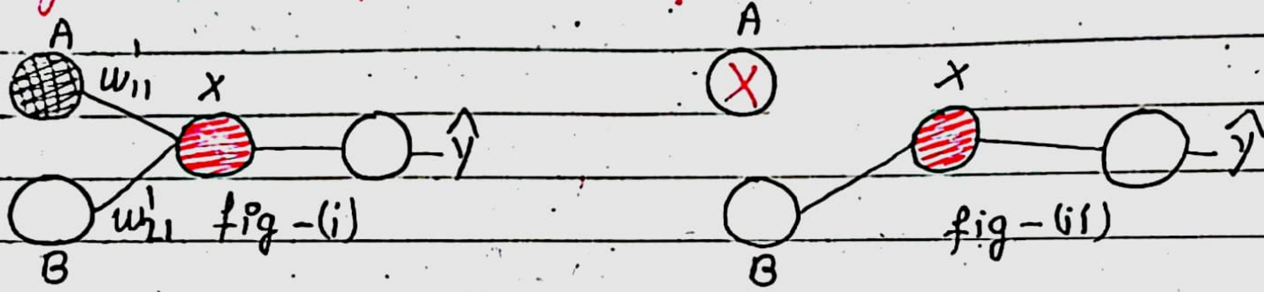
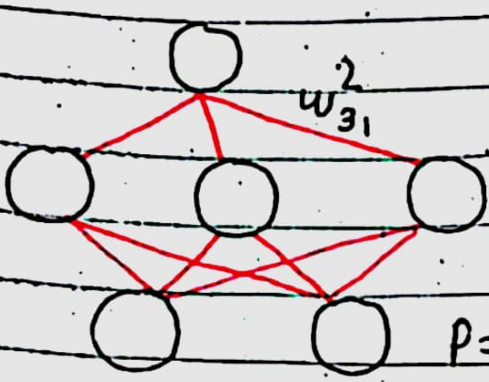


fig-(i) fig-(ii)

A= no of rooms , B= no of gardens

When we feed our data to this NN. Node X more focuses on feature A. Since, no of rooms have strong corelation **to the** price of house than no of gardens. due to this reason Node X assign higher weights value for w_{11}' and due to this reason there may be overfitting problem because it won't give priority to feature B and can't learn proper pattern. but when we we drop out as in figure (ii) the Node X also look for feature B and also give time to lear patterns from feature B which prevents w from overfitting.

How prediction happens?



$P = 0.25$

$P = 0.25$

This dropout only happens during training. Jaba Sabai data train vayera hami ley afno best weights pauxam. Tyo weight lai nai we gardenam hamle testing ko lagi

Testing ko time ma sabai Node available hunxa. So, what will be the weights valve ?

It was not guranteed the w^2_{31} was available or not during training (w^2_{31} was taken as a example it also worked for others). Suppose the valve we got $w^2_{3.1}$ is w

now, for testing $w^2_{31} = w(1-p)$

$= w \times 0.75$

Why this $w \times 0.75$?

=> yo $w^2_{3.1}$ training ko time ma active hune probablity 0.75 matra xa, 0.25 probablity ma [redacted] 1 layer node 3 bata kunai data na gako huna skxa tei vara. hami traning time ma active hune probablity sanga multiply garera testing garxam

(This all is handled by keras)