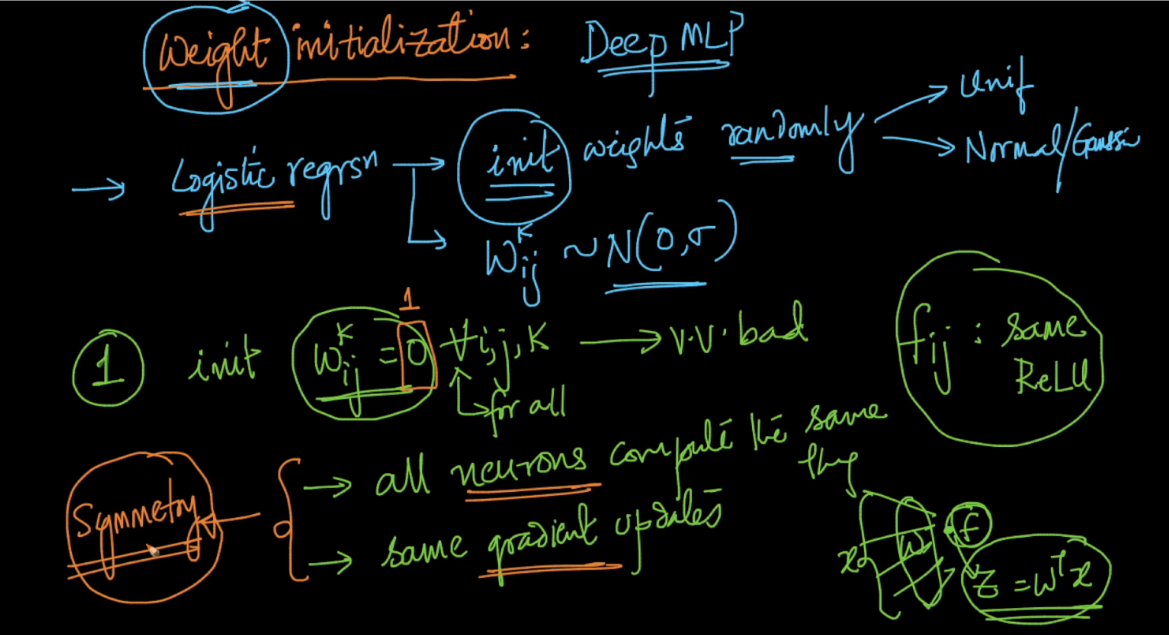
**Weight initialization**

In logistic regression we initialize weights randomly (uniformly or normal/gaussian distributed)

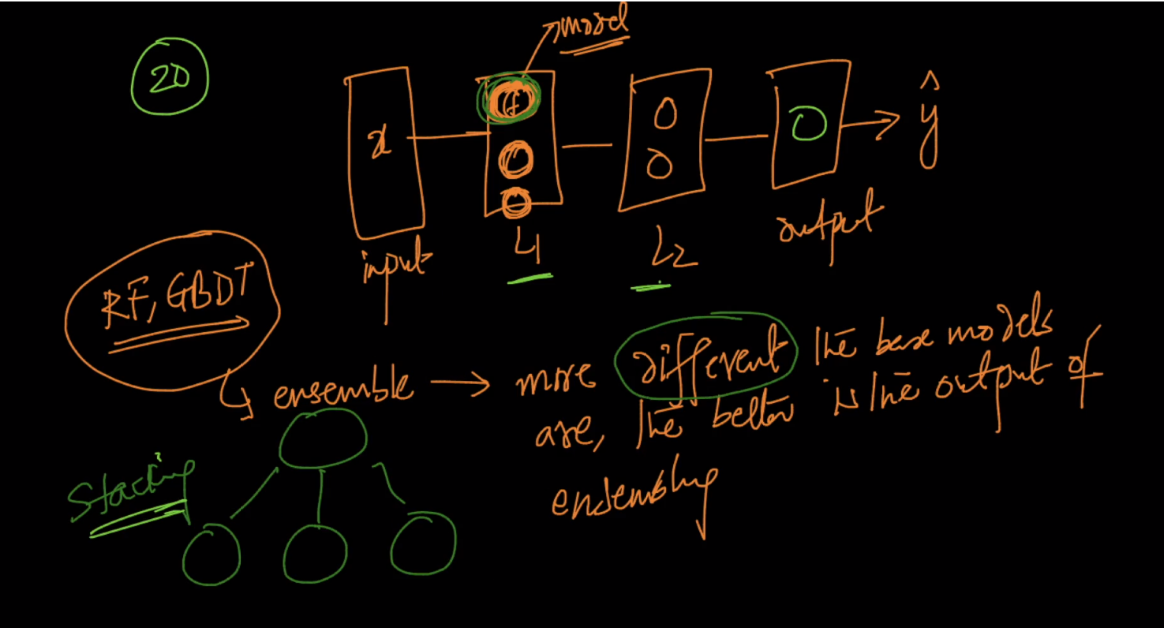
Or we initialize weights from standard normal distribution which is mean centric

Weight initialization in deep MLP:

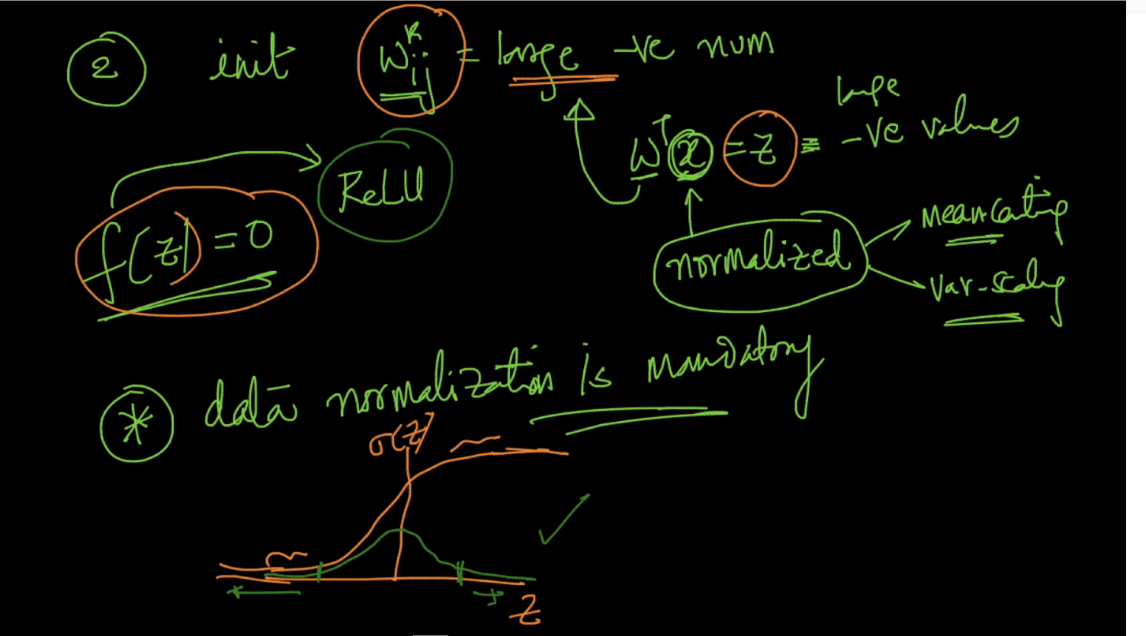
1. If we initialize all weights with 0 then it is very bad idea because all neurons compute the same thing and same gradient updation occur therefore it creates symmetry which is not good.



Why we want asymmetry because if all weights compute same thing then it doesn’t provide better output we can link it as in ensemble models more different the models are the better is the output of ensembling.



1. Initializing weights with large negative number then this create problem of dead activation.

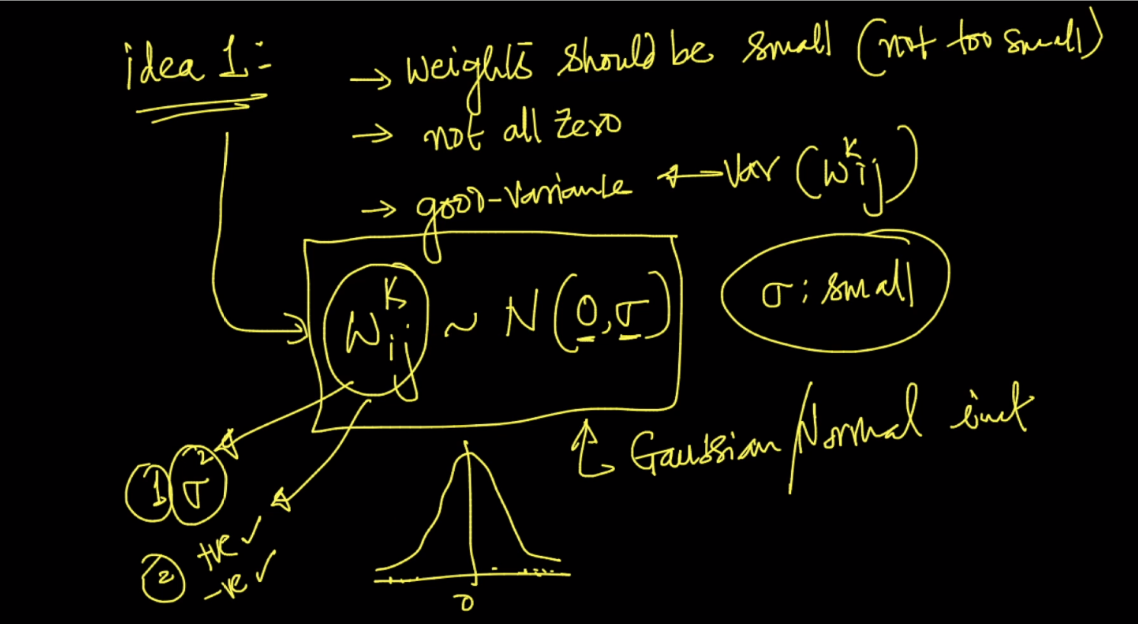


**For better initialization :**



* Weights should be small (not to small)
* Not all weights are zero
* Good -variance

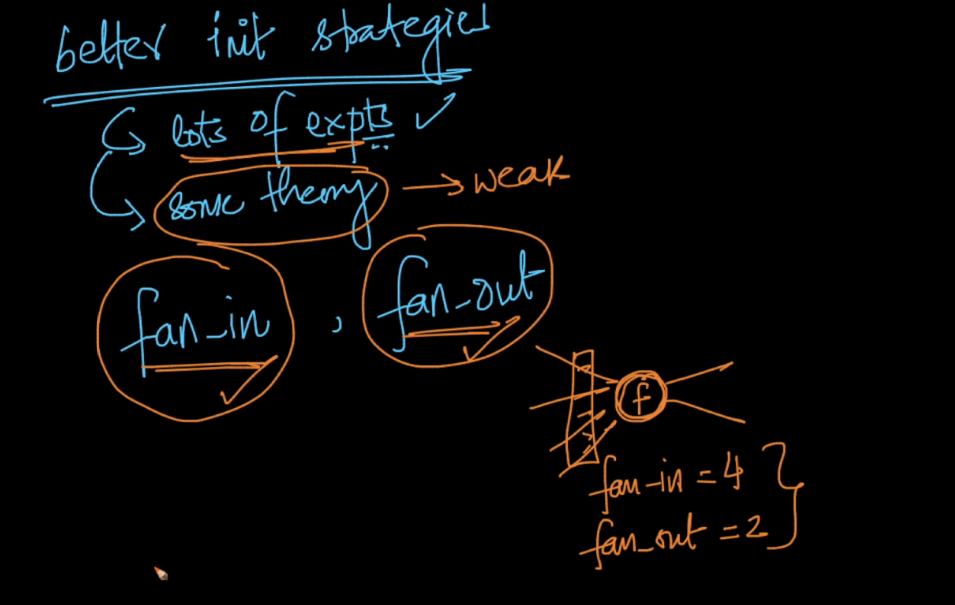
We can take weights as gaussian, normal distributed in this variance should be good but small.



**Better initialization strategies :**

For this there are lot of experiments but some weak theory.

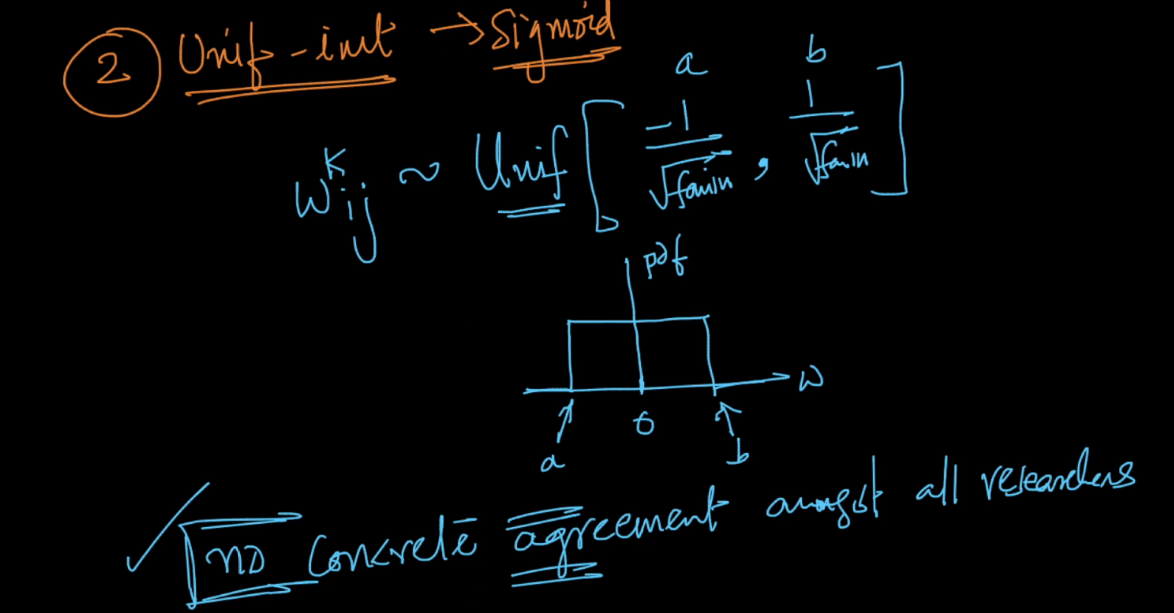
So in this initialization strategies we use fan-in, fan-out



1. Uniform-initialization :

In this we initialize weights with uniform distribution with range a,b whose values shown in below image .

It is useful for sigmoid activation function



1. Xavier/glorot initialization :

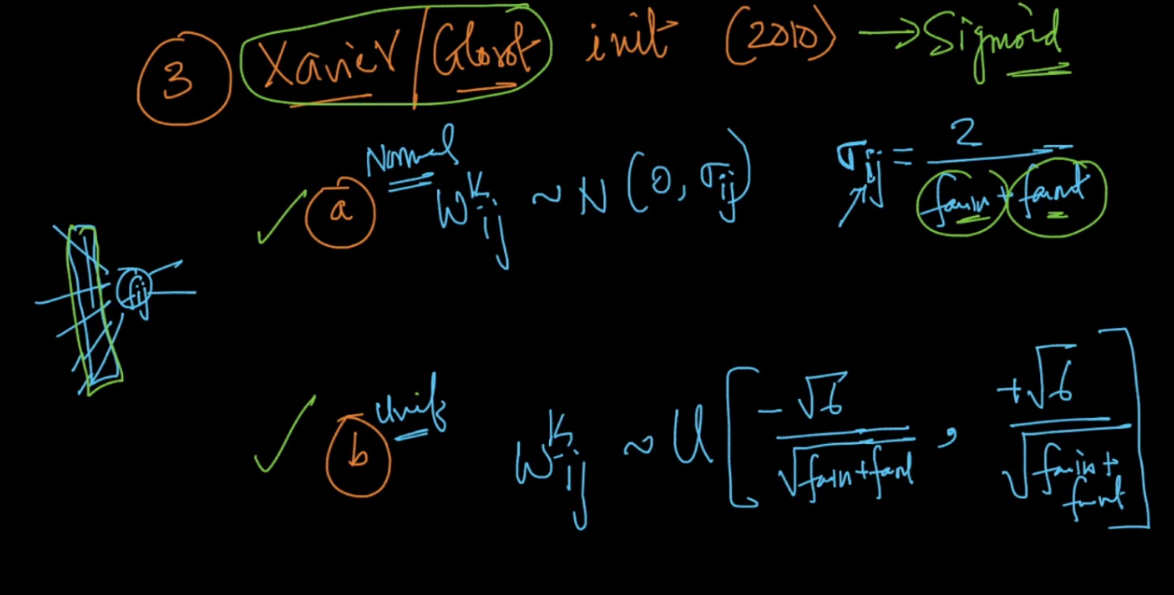
In this we can initialize in two ways normal xavier/glorot initialization and uniform xavier/glorot initialization.

1. In this we initialize with normal distribution where mean is 0 and

sigma = sqrt(2/(fan\_in +fan\_out)) , note : misprint in image

1. In this we initialize with uniform distribution as shown below.

It is useful in sigmoid activation function.

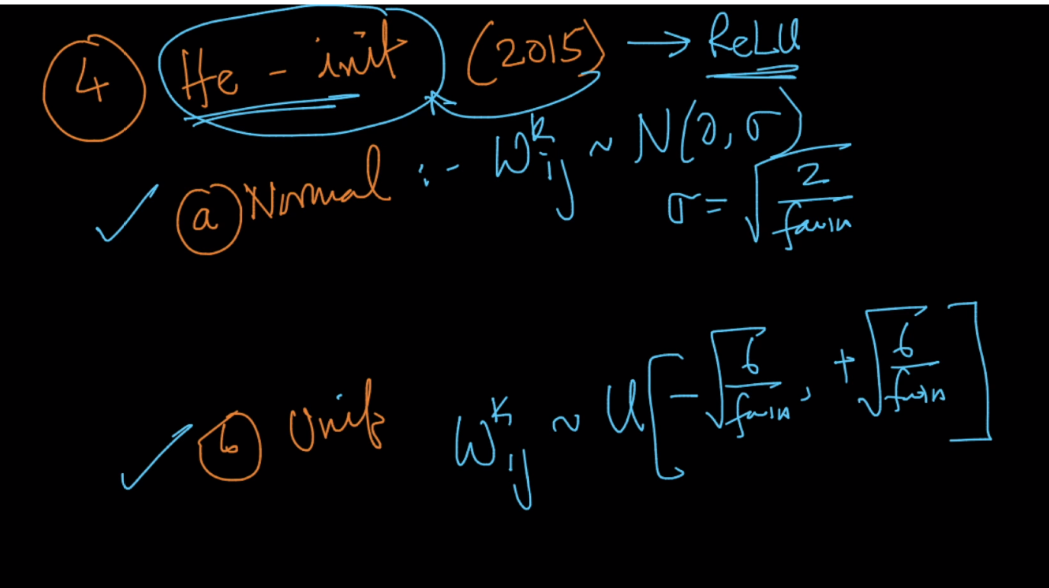


1. He-initialization :
2. Normal : we initialize weights with normal distribution with mean = 0 and

Sigma = sqrt(s/fan\_in)

1. Uniform : we inititalize weights with uniform distribution as shown below.

It is useful in ReLu



Comments :

