## horizontal line

Batch Normalization

21.03.2025

# 

# Batch Normalization

Batch Normalization is an algorithmic method of normalizing activation vectors (output after activation function) from hidden layers. Inputs are normalized also the activated output i.e. (sd=0 , mean =1). This makes the training faster.

# Why use Batch Normalization ?

In unnormalised data we need to take a small learning that takes too much epochs to reach local minima.

We cannot take large so that it overshoots in the Y\_axis direction as its contour is oval.

While in normalised data , its contour is equally falling in both axes. This helps to increase a good to approach the exact solution.

# Covariate Shift

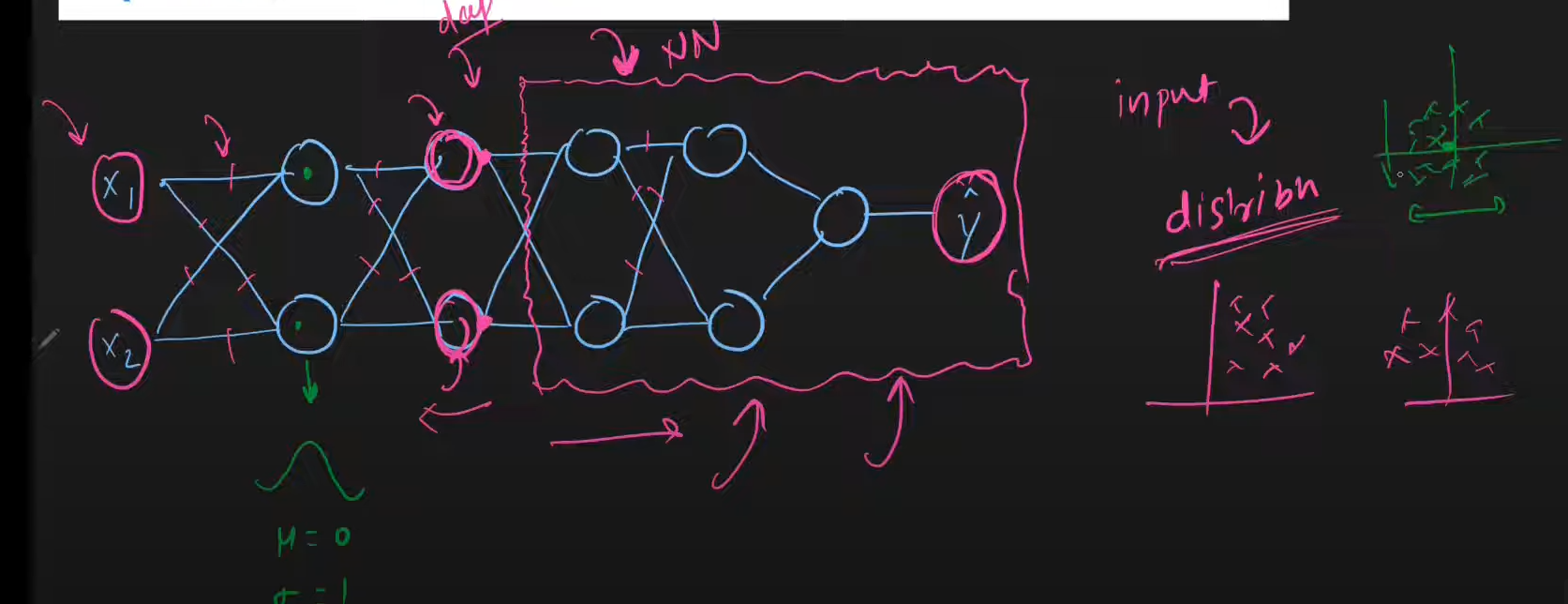
When the input distribution changes , it leads to incorrect output prediction while testing.

E.g. Training provided all red roses to classify them and between other flowers but in testing provided different kinds of roses as well like pink, white , yellow etc. that the model predicted in the category of not a rose.



Now Internal Covariate Shift is the change in distribution of activated outputs due to change in network parameters.

This is because if you assume the last two hidden layers , they get input from previous hidden layers which change every time the parameters of layers change. This causes problems in training i.e. unstable training , for that purpose we introduced gaussian distribution of all activated outputs.



To reduce internal covariate shift we use batch normalization and generally keep low.

# How Batch Normalization Works ?

It works on Mini-Batch GD , layer by layer.

We have two methods of applying batch normalization.

Z11 = w1cgpa + w2iq +b and g(Z11) = a11

1. Z11 —--> —--> g() —> a11
2. Z11 —--> g(Z11) —--> a11 —-->



Now see if you have 40 rows of data 3 columns cgpa , iq , placed.

You will take 4 rows in a batch , the input will be (4X2) . Weight (2X2) , Bias (1X2).

Combinely , (4X2)(2X2) + (1X2) will give us (4X2) matrix . 4 activations of 4 rows as Z11 and 4 activations of 4 rows as Z12.

Separately calculating and for both of them.

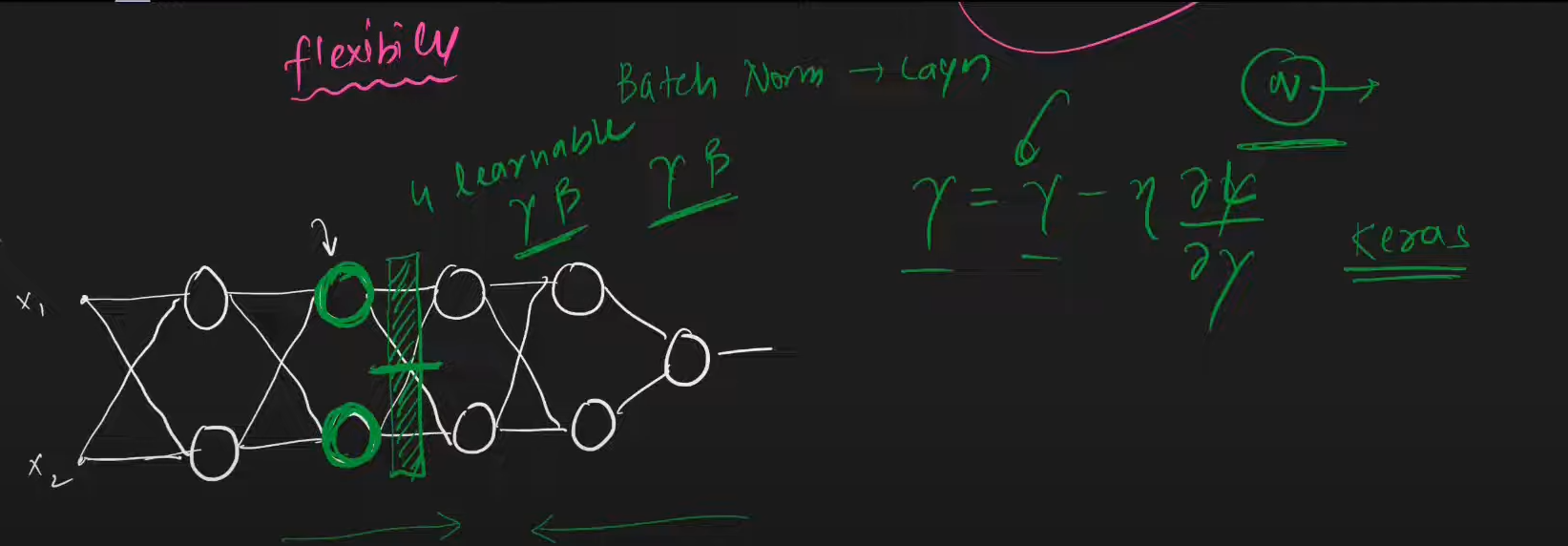
= 1/m Z11 and

Now , ) / +

Now it is suggested to have Z11 = + scale and shift , where and are trainable parameters . Every node has its own and .

Why add them this is just the reverse of what we have done earlier ?

This is because Every model does not require Batch Normalization just to provide flexibility.

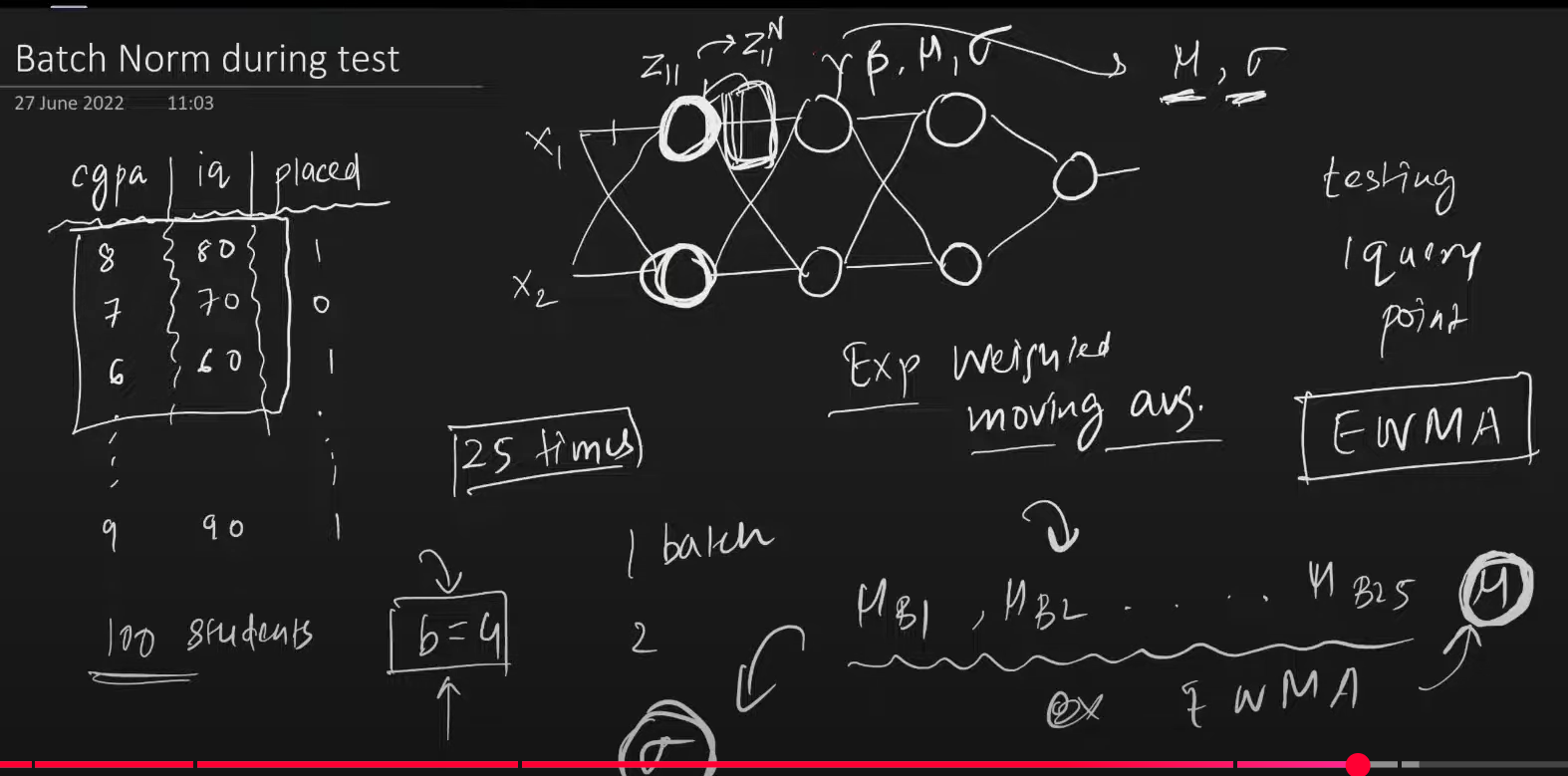
This is for deep hidden layers.

We consider batch normalization as a layer.

# 

# 

# Batch Normalization during Testing



# Advantages

1. Stable
2. Faster
3. Provide Regularization and dropouts
4. Weight initializations impact reduce