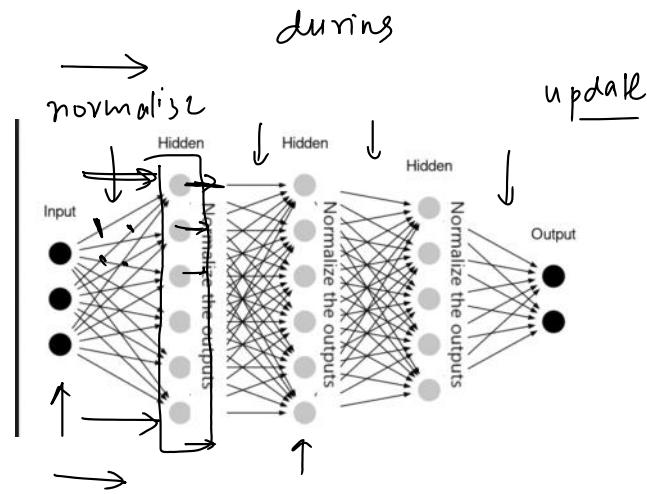


Batch Norm

03 January 2025 18:40

- Applied to Hidden Layers:
 - Typically applied to the hidden layers of a neural network, but not to the output layer.
- Applied After Linear Layers and Before Activation Functions:
 - Normalizes the output of the preceding layer (e.g., after nn.Linear) and is usually followed by an activation function (e.g., ReLU).
- Normalizes Activations:
 - Computes the mean and variance of the activations within a mini-batch and uses these statistics to normalize the activations.
- Includes Learnable Parameters:
 - Introduces two learnable parameters, gamma (scaling) and beta (shifting), which allow the network to adjust the normalized outputs.
- Improves Training Stability:
 - Reduces internal covariate shift, stabilizing the training process and allowing the use of higher learning rates.
- Regularization Effect:
 - Introduces some regularization because the statistics are computed over a mini-batch, adding noise to the training process.
- Consistent During Evaluation:
 - During evaluation, BatchNorm uses the running mean and variance accumulated during training, rather than recomputing them from the mini-batch.

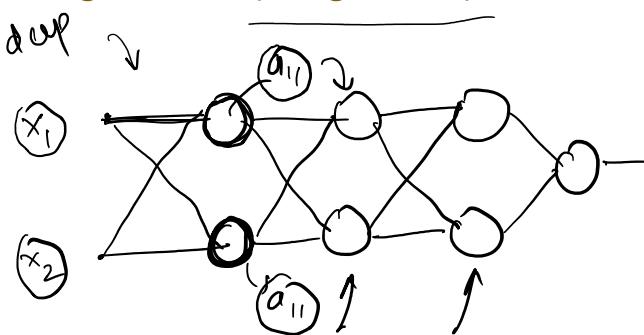


What is Batch Norm?

27 June 2022 11:00

Batch-Normalization (BN) is an algorithmic method which makes the training of Deep Neural Networks (DNN) faster and more stable.

- It consists of normalizing activation vectors from hidden layers using the mean and variance of the current batch. This normalization step is applied right before (or right after) the nonlinear function.



normalized

$$\begin{cases} m=0 \\ sd=1 \end{cases}$$

normalize

$$\begin{cases} m=0 \\ sd=1 \end{cases}$$

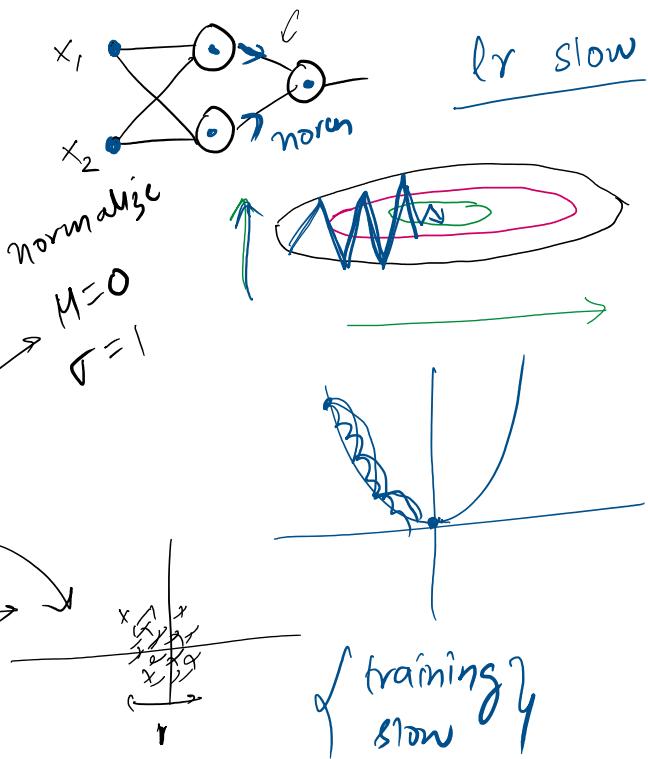
Why use Batch Norm?

30 June 2022 16:08

	cgpa	iq	placed
7	70	1	
8	80	0	
9	90	1	
6	60	0	

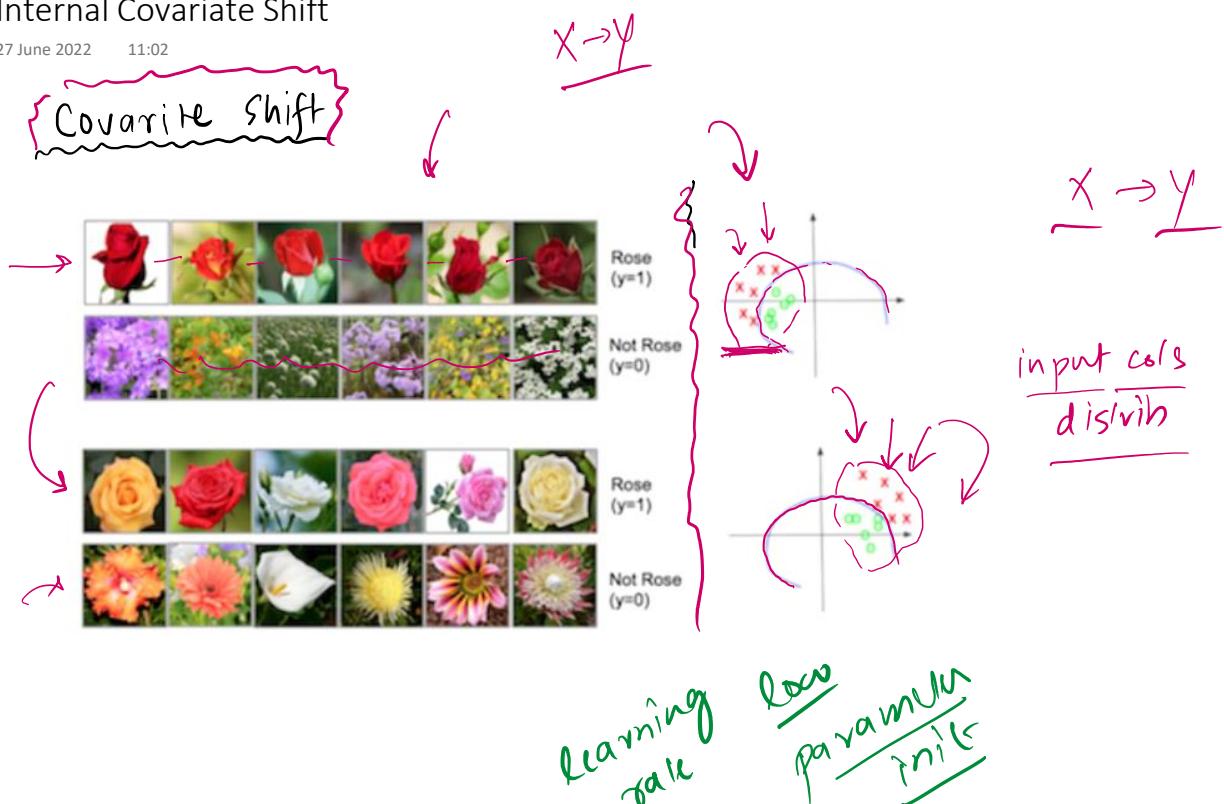
in | $x^{\alpha} x x x x$

cgpa | \rightarrow



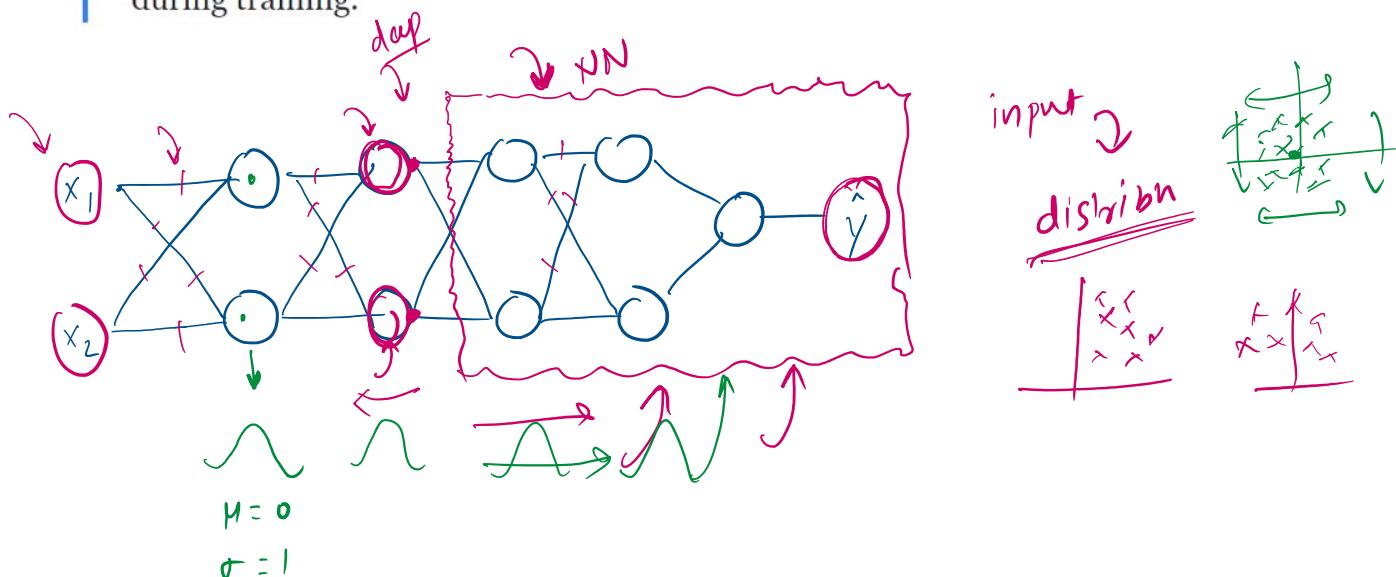
Internal Covariate Shift

27 June 2022 11:02



The authors' precise definition is:

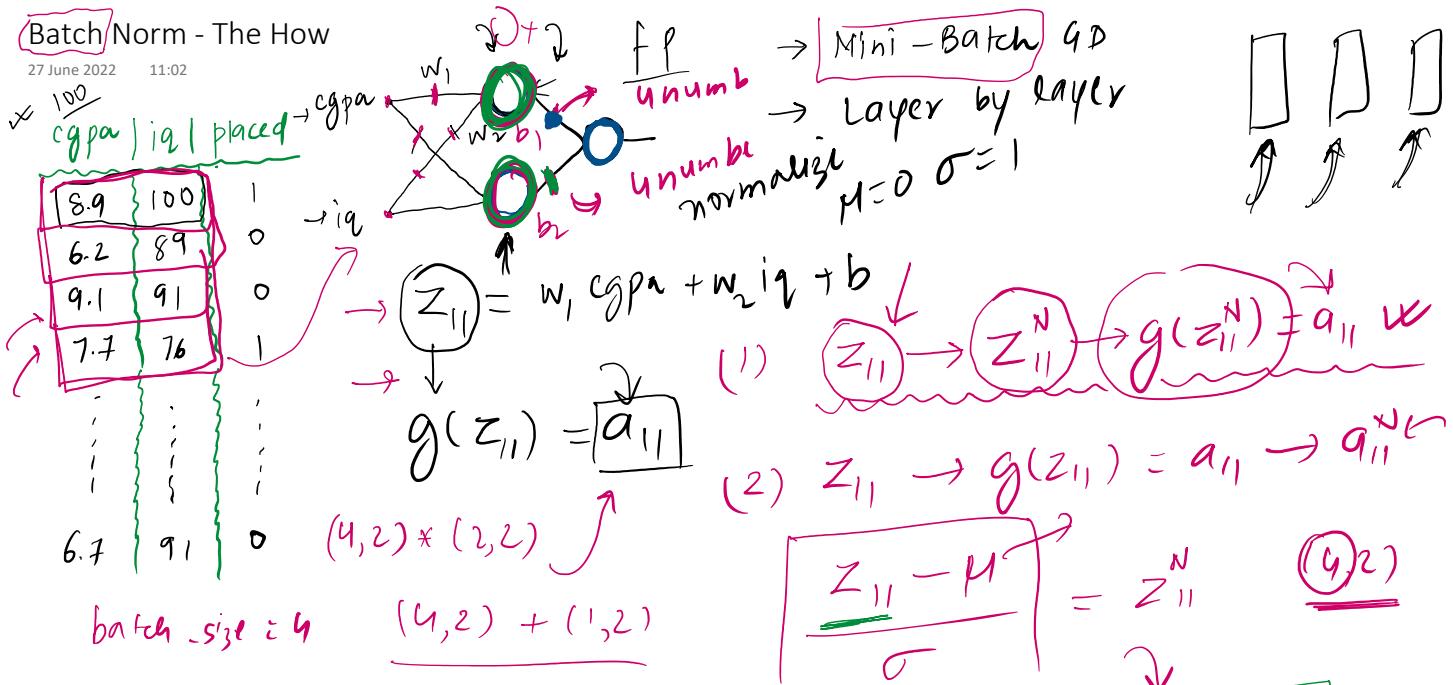
We define Internal Covariate Shift as the change in the distribution of network activations due to the change in network parameters during training.





Batch Norm - The How

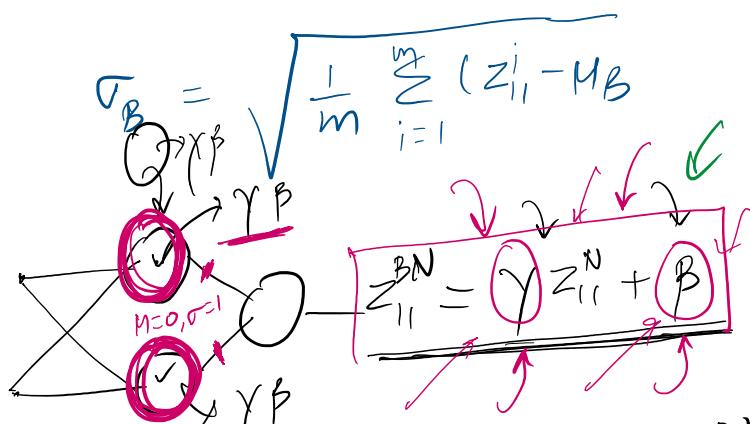
27 June 2022 11:02



$$\mu_B = \frac{1}{m} \sum_{i=1}^m z_{ii}^i$$

$$m = 4$$

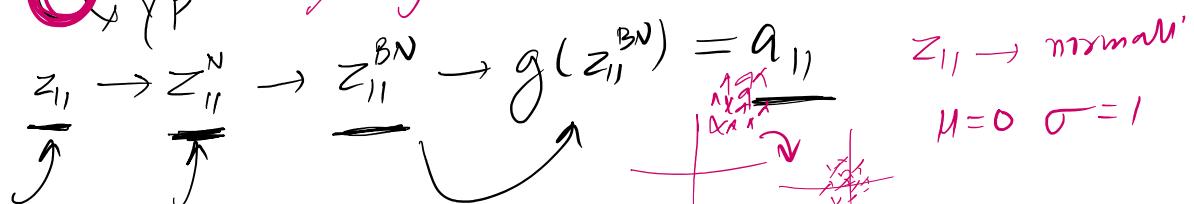
$$z_{ii}^i = \frac{z_{ii}^i - \mu_B}{\sigma_B + \epsilon}$$



$$[0-1] \quad \mu = 0 \quad \sigma = 1 \quad [w, b]$$

learnable parameters

$$[\gamma = 1, \beta = 0]$$



$$\gamma = \gamma + \epsilon \quad \beta = \mu$$

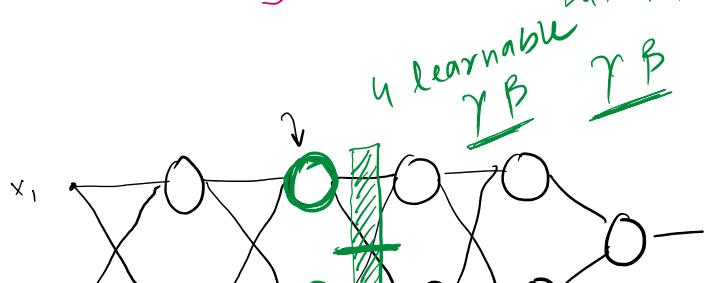
flexibility

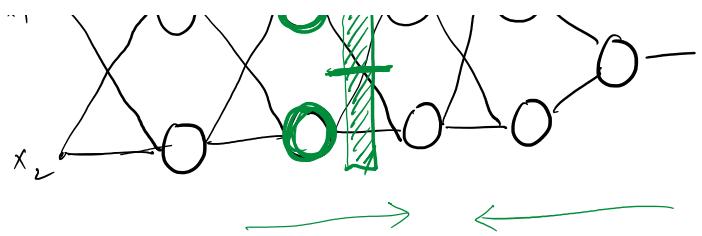
$$z_{11} \rightarrow z_{11}^N \rightarrow z_{11}^{BN}$$

Batch Norm \rightarrow Layer

$$\gamma = \gamma - \eta \frac{\partial \gamma}{\partial \gamma}$$

keras

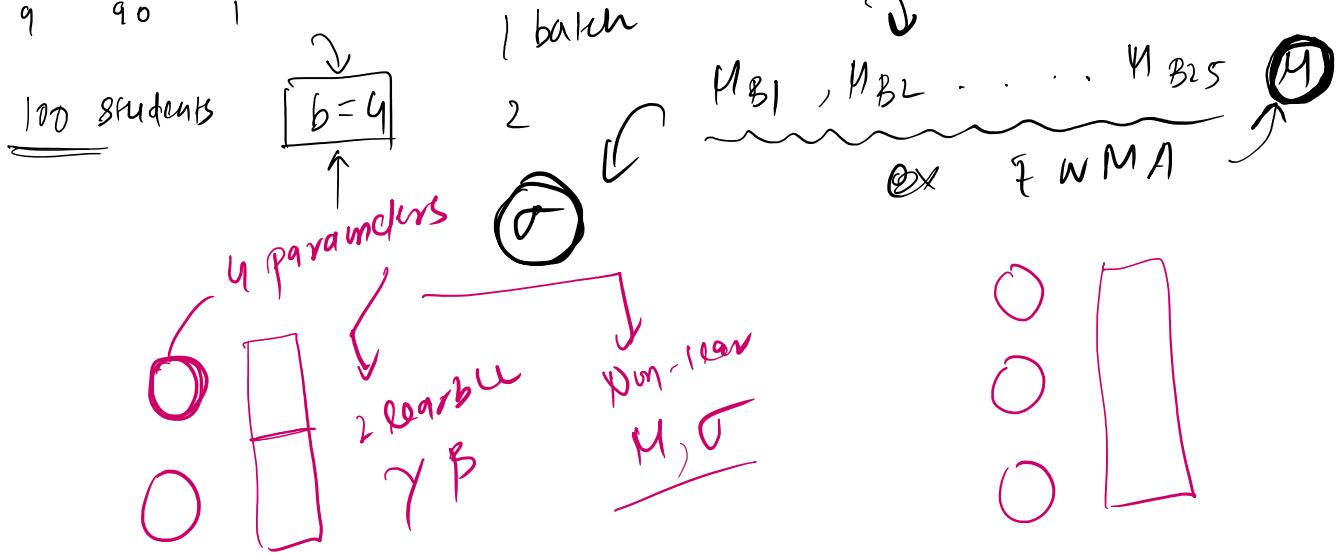
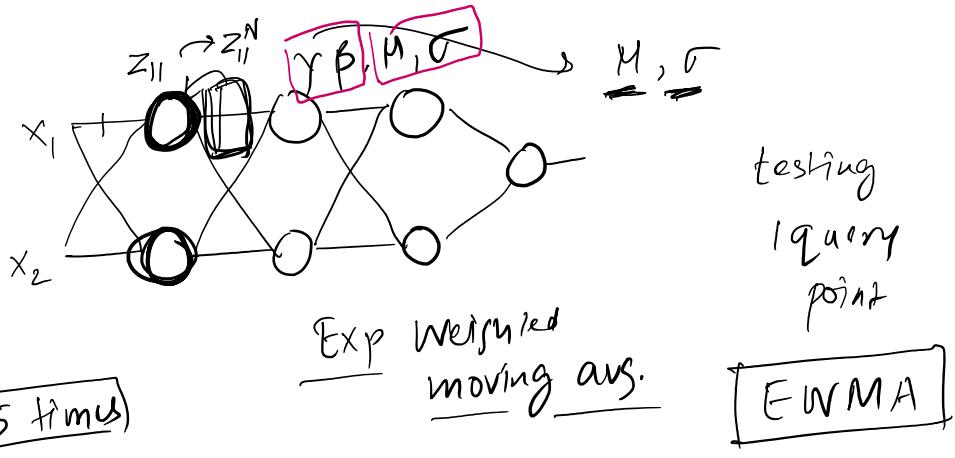




Batch Norm during test

27 June 2022 11:03

cgpa	ia	placed
8	80	1
7	70	0
6	60	1
:	:	:
9	90	1

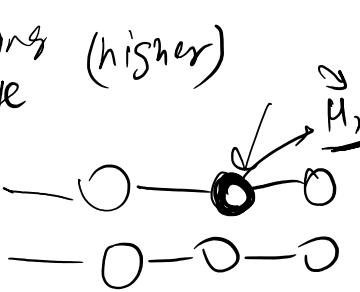


$$3 \times 4 = 12$$

$$6 \quad 6$$

Advantages

27 June 2022 11:02

- 1) stable → hyper ~~para~~ → wider range of robustness
- 2) faster → learning rate (higher)
batch
randomness
noise
overfitting
- 3) Regularizer →
↓
Dropout
- 4) weight init impact
reduce