Batch Normalisation

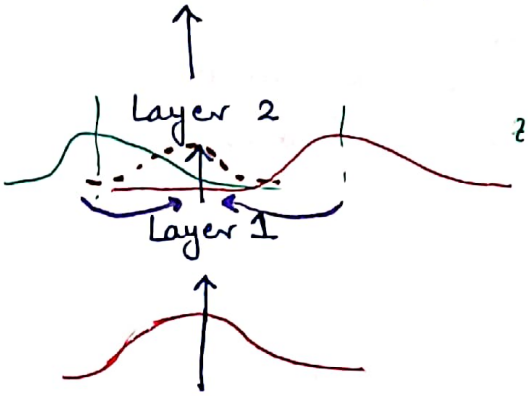
Even if we do proper weight initialisation & choose a right activation function → still we may encounter gradient problem

↓

ICS : Internal Covariate Shift

- change in distribution of network
- activation o/p due to change in network parameters during training

Let's say we have a bell shaped distribution in Layer1. If we move to Layer2 there may be a shift in the curve or also in the distribution range of the curve due to the changes in the value of z. Hence we need to bring the distribution in a similar form in layer2 as well.



$z = Wx + b$

Why?

→ It has been experimentally proven {Le Cun et al, 1998 & Wiesler and Ney 2011}

"The network converges faster if inputs to the layer are ~~whitened~~ widened i.e if they are normalised ⇒ linearly transformed to zero mean ($\mu = 0$) and unit variance" ($\sigma = 1$)

Our Expectation:

- fix distribution for each layer ⇒ reduction in Internal Covariate Shift (ICS) ⇒ faster convergence ⇐ fast training ⇐ efficient use of resources ⇐ Save dollars.

↓

Batch Normalisation solves this (BN) : In 2015 Sergey Ioffe & Christian Szegedy have published this paper

↓

[proposed extra set of operations which can be performed before or after the activation layer]

BN Algorithm (While Training)

1) Calculate the batch mean

$$\mu_B = \frac{1}{m_B}\sum_{i=1}^{m_B} x^i$$

μ_B → batch mean; m_B → number of samples in the batch; x^i → input

2) Calculate the batch variance

$$\sigma_B^2 = \frac{1}{m_B}\sum_{i=1}^{m_B}(x^i - \mu_B)^2$$ Batch variance

3) $$\hat{x}^{(i)} = \frac{x^i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

$\hat{x}^{(i)}$ ↓ Normalising x^i

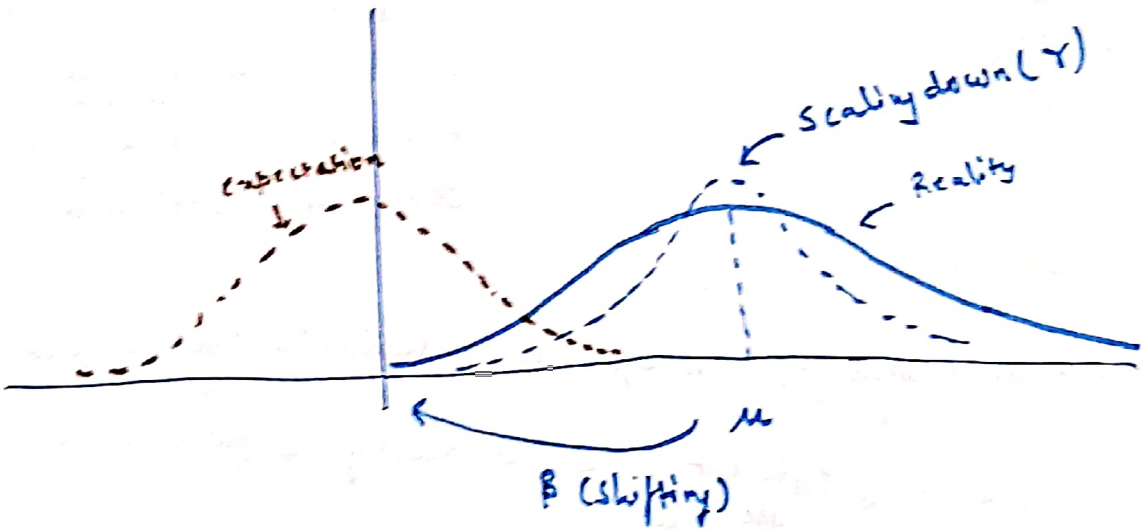
ϵ → this has been introduced because σ_B can be zero.

Smoothing Term $\epsilon = 10^{-7}$ To avoid zero division error.

4) $$z^i = \gamma \otimes \hat{x}(i) + \beta$$

γ → scaling Parameter; β → shifting Parameter

Learnable Parameters] shift & scale the distribution



Calculate overall mean (μ) and standard deviation (σ) by using moving average on μ_{BA} & σ_{BA}

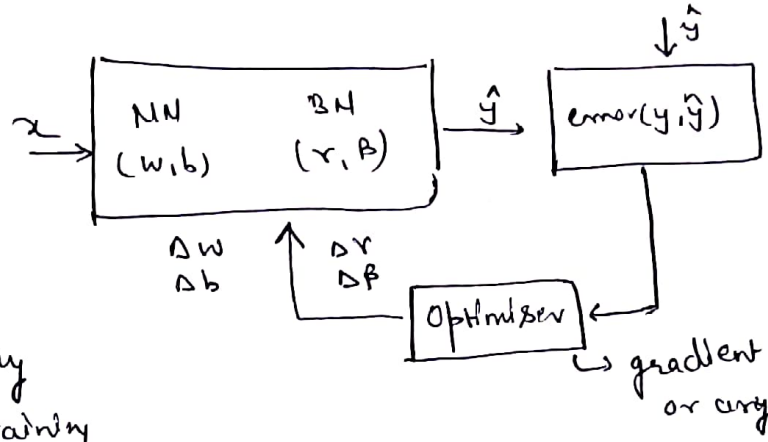
While prediction:- (i) $$x^{(i)} = \frac{x^{(i)} - \mu}{\sqrt{\sigma^2 + \epsilon}}$$ (ii) $$z^{(i)} = \gamma \hat{x}^{(i)} + \beta$$

① → layer 1 $\xrightarrow{z}$ BN $\xrightarrow{\hat{z}}$ activation function $\xrightarrow{a}$ layer 2

② → layer 1 → actn $\xrightarrow{a}$ BN $\xrightarrow{\hat{a}}$ layer 2

③ Two approaches to apply Batch Normalisation.

In normal neural network trainable parameters are just weights and biases where as in a neural network with batch normalisation we have two extra parameters named γ (gamma) and β (Beta) that needs training.



μ, σ will be Calculated Internally after completion of training

- Extra parameters but not trainable our cost function is dependent on 4 parameters.

$C(w, b, \gamma, \beta)$ → updated by back propogation

Disadvantages

① It increases the complexity of the network.

② Number of learnable parameters increased

③ Run time penalty due to complex network ⇒ slow prediction.

④ Training time is increased but convergence will be faster.

Advantages

① You don't need scaling of data if you are using Batch Normalisation as a 1st layer.

② It converges faster despite having two extra learnable parameter.

③ It helps to reduce the vanishing and exploding gradient issue drastically

④ It doesn't get affected by choice of activation function and weight initialisation technique.