Dropout Layers

Dropout Layer is recent technique that solves the issue of Overfitting.

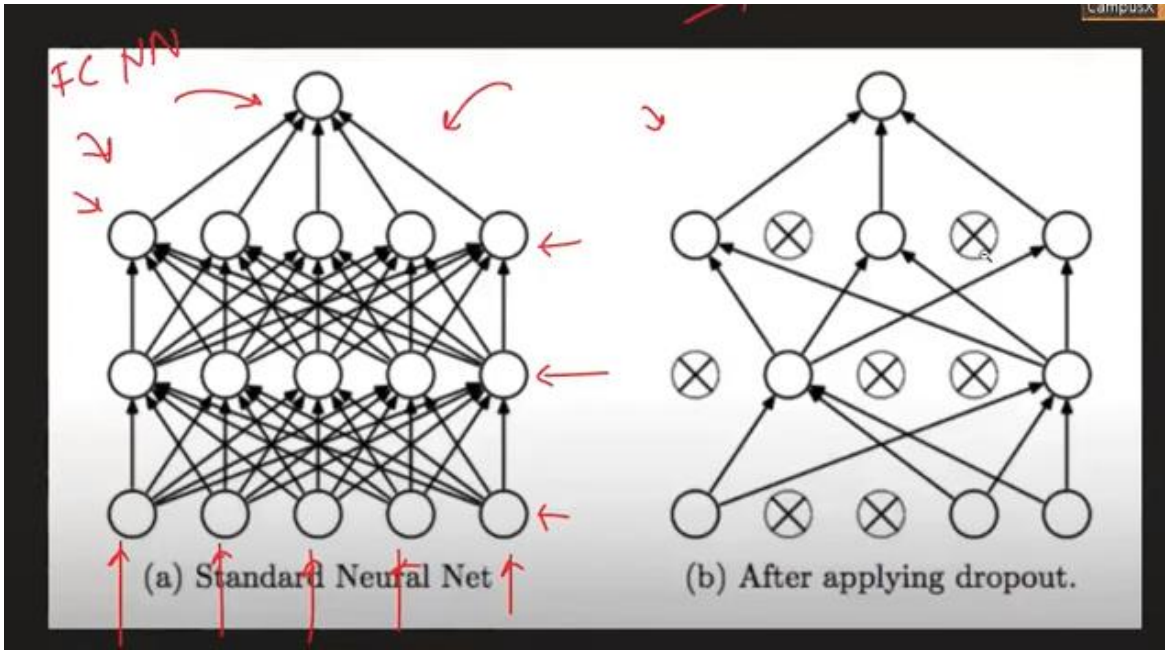
Neural Network are Prone to Overfitting because they are Very Complex

As they have Multiple Layers & each layer has multiple nodes.

Possible Solution of Overfitting

1. *Add More Data*
   Jitna zyaada data utna chance hai ki overfit na hoo
2. *Reduce Complexity of Network.*
3. *Early Stopping*
4. *Regularization L1 & L2*
5. *Dropout*

Dropout Layer → Nitish Srivastav ne banaya hai



Itna complex jbh architecture rehta hai tou overfit hota hie hai,

Dropout Layer ka jo concept hai: Randomly Input Layer & Hidden Layer ke Nodes ko Drop kardete hai

Drop ka mtlb kya hai?

Hum Har Epoch keliye Randomly Kuch Nodes Hidden Layer & Input Layer ke Node ko Turn Off kardenge.

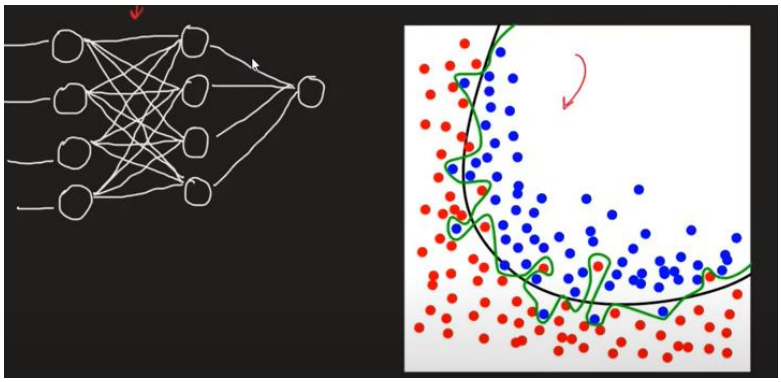
E.g 10 Epochs chalarhe hai then 10 Different Neural Network pr humm train krrhe hai.

So Har Epoch mai we are training on same data but on Different Neural Network.

By Breaking how It is improving Performance?

Dropout karnese 2% ka accuracy improve hota yeah proven hai.

Why Overfitting Happens in Neural Network?

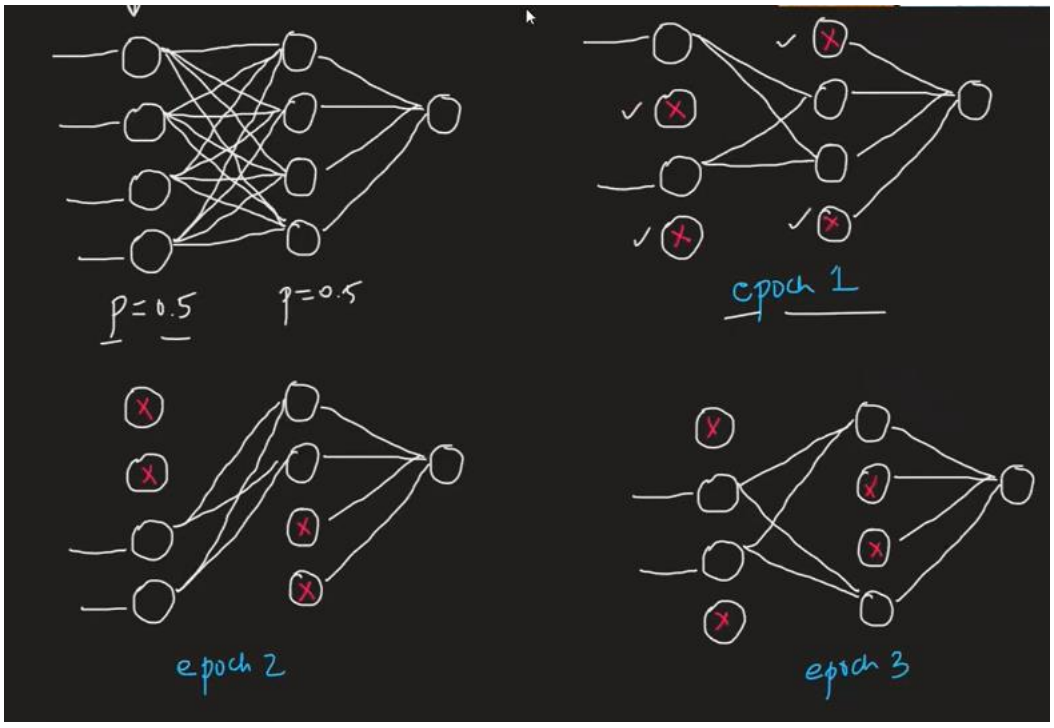


How to Avoid Overfitting?

1. Reduce Number of Nodes,

2. Else Nodes mai esa change karo ki vo Biased ya 1 particular pattern pe focus na rheah sbh patterns pe focus rheah.

This is How Even Dropout Layer Works.



As we can see No. of Nodes reduce horhe

& Even Pattern bhi if you see different hai Har Epochs mai

Kisi Ek Neuron Ke Weights pr Focussed nahi hai , sbh Har Neuron ko Importance deraha hai.

Let’s Compare Dropout with Random Forest.

It is connected with Random Forest.

Random Forest Multiple DT ka combination

Random forest samples rows with replacement (bootstrap samples) to remove correlation between the decision trees. Think about it, if you didn’t do this even though you create each split based on only a subset of features, your trees would end up looking fairly similar

it is a way to reduce bias. If you're training a Random Forest with 100 trees, then you will grow these trees with (potentially) 100 different training sets. You can achieve the "wisdom of the crowds" since there is a crowd formed by these training sets.

2 way se data share krte hai

Column Sampling or else Row Sampling.

1st DT mai 50% bss Data bheje

2nd DT mai vps randomly 50% data denge

Ese saare Decision Tree Jo honge vo ek dusre se bhaut different honge

Yeahi Kindoff Logic Dropout Layer ka bhi hai.

For Every Epoch, We are training a Different Neural Network,

Every Epoch has different Neural Network Architecture.

Jaise harbaar RF mai alag alag DT train krte hai,

Vaise alag alag NN ko train krte hai , & unka intelligence millake decision making krte hai.

That’s the reason Dropout use karnese performance improve hota hai & overfitting reduce hota hai.

----------→

<u>Dropout bss training ke tym lagta hai</u>

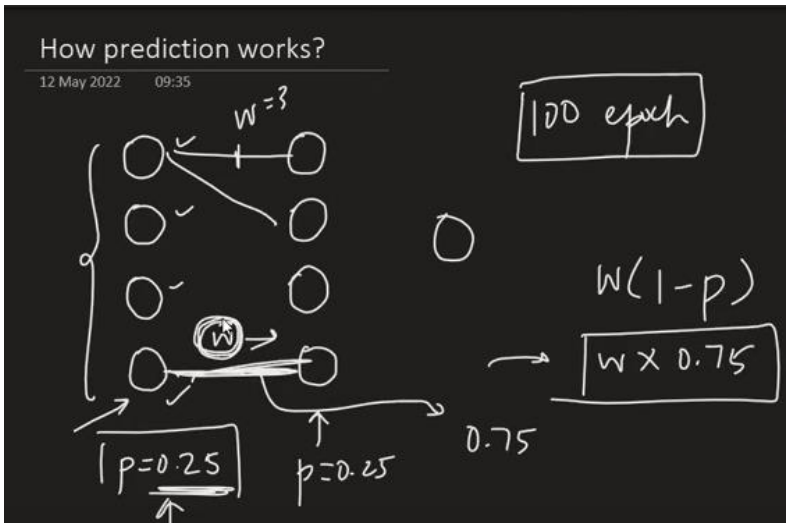
Neurons Drop bss Training ke tym hota hai

&

Testing ke time Saare Neurons Available hote hai

& we

Example:



Jaise Dropout laga hai 0.25 ka

Then testing ke time Weight kya hota hai ?

Testing ke time Weight hojayega:

=W(1-DropoutValue)

=W * (1 – 0.25)

=W* 0.75

It is Very Logical

Har Neuron ka gayab rehneka Probability is 0.25 hai & Rehneka

0.75

Suppose 100 epochs hai then 75 times rahega vo & 25 times gayab rahega.