Topics

① weight Initialization techniques
② CNN (Convolution Neural network)
③ Drop outs

① weight Initialization

→ while we initializing weight in between Neurons researchers are Consider below 3 key points.

① weight should be small
② weight should not be same
③ weight should have good variable (range)

Based on above 3 points the techniq are

① Uniform Distribution
② Xavier / Glorot Initilization
③ Kaiming He Initilization

i) Uniform Distribution

The weights is $W_{ij} \approx$ Uniform Distribution $\left[\frac{-1}{\sqrt{inputs}}, \frac{1}{\sqrt{inputs}}\right]$

input → no. of inputs (features) to the NN

② Xavier / Glorot Initialization

There are two methods

① Xavier Normal init

$W_{ij} \approx N(0, \sigma)$

$$\sigma = \sqrt{\frac{2}{\sqrt{input + output}}}$$

② Xavier uniform

$W_{ij} \approx$ uniform Dis $\left[\frac{-\sqrt{6}}{\sqrt{i+o}}, \frac{\sqrt{6}}{\sqrt{i+o}}\right]$

③ Kaiming He Initialization

① $W_{ij} \approx N(0, \sigma)$

$$\sigma = \sqrt{\frac{2}{input}}$$

② He uniform

$W_{ij} \approx$ uniform Dist $\left[-\sqrt{\frac{6}{i}}, \sqrt{\frac{6}{i}}\right]$

④ Drop Out

→ This is used to overcome overfitting (or) Underfitting in model.

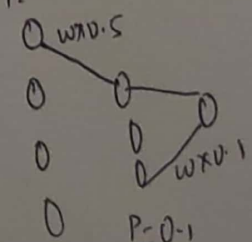
In training pipeline

→ P = 0.5 randomly 50% of neuron will deactivate

In testing data

P will multiple with new weight

P = 0.5



CNN

→ Common operations in CNN

→ ① Convert 0-255 to 0-1 (min-max scaling)

② Apply filter

→ Filter acts as a single template or pattern, which, when convolved across the input, finds similarities between the stored template & different location / region in the input image.

→ Filter values will update in back propagation

→ If image size = 6×6 & filter size = 3×3

o/p = 4×4 $\boxed{n-f+1}$

③ padding

→ In above process we are loosing the information lies, our image size is 6×6 & o/p is 4×4.

→ For not loosing the information we apply paddings

→ There are 2 process in paddings.

① Add zero's in

② Add boundary values.

$n - f + 2p + 1 = 6$

$6 - 3 + 2p = 6$

$p = 2/2 = 1$

what is Convolution layer

~~we have~~
~~G×G image~~
→ For our input image we apply filter and get output, to output we apply "Relu Activation" max(0,x)

→ Convolution + Relu Activation = Convolution

→ we apply relu becz it can derivative, so that it can be update filter weights.

pooling

① pooling is performed in NN to reduce variance and computation complexity.

(i) mean pooling :- This method smooths out the image and hence the sharp features may not be identify.

(ii) Max pooling :- Select the brighter pixel from the image. Used when background image is dark.

(iii) min pooling :- reverse of max pooling.

Faltenning