SUB
DATE

CISC 867 : Deep Learning
Assignment #2
Name :- Hanan Fared Mohamed Omara.
ID :- 20 39 8559

[1] input = 500 * 500 * 3 , 100 hidden units is used
→ The shape of the weight matrix of this layer (without the bias)
= input * hidden units
= 500 * 500 * 3 * 100 = 75 million. $\xrightarrow{\text{shape}}$ (500*500*3 , 100)
→ The shape of the bias = 1 * 100 = (100 ,) where 100 is
the number of hidden units in the layer.

[2] 10 filters , size of kernel = 5 * 5
The number of parameters = (F_h * F_w * L + 1) * number of filter
= (5 * 5 * 3 + 1) * 10
= 760.

[3] left (vertical edge detector) = $\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$

Right (Horizontal edge detector) = $\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$

[4] The bias will disappears when average large numbers

$V_r = \dfrac{Vr}{1-\beta_r}$ when $r \to \infty$

and when it divide by small number will get larger.

SUB

DATE

$\boxed{5}$ The size batch m, input $z = (z^1, \dots, z^m)$

output $M = \frac{1}{m} \sum_{i=1}^{m} z_i$

$$\sigma^2 = \frac{1}{m} \sum_{i=1}^{m} (z_i - M)^2$$

normalization $\Rightarrow z_i = \frac{z_i - M}{\sqrt{\sigma^2 + \varepsilon}}$

applying scale shifitting $z_i^4 = \{ z_{i\,norm} + B$

$\Rightarrow$ The Two reasons for using The batch norimalization layer.

① Batch Norimalization is a General Technique That can be used to norimalize The input To a layer.

② it can be used with most network Types, such as Multi Layers Perceptrons, Convolutional and neural Network it Makes a network more stable during Training, it allows higher Learning rate and increase The speed at which network Train, it solves internal Convarient shift.

$\boxed{6}$

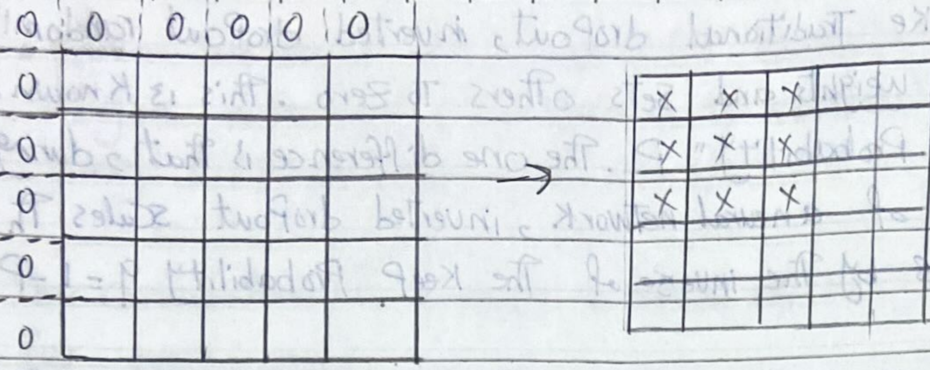
image size 256 * 256

first layer 32 feature filter size 3*3 Stride 1

the first layer has The Same Width and hight as The original image, next layer 3*3 Stride 2

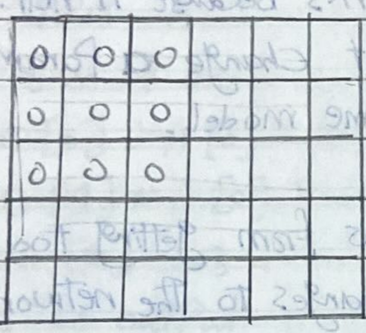
→ because first layer has The Same width and height of original image

∴ we will use Padding.



256 * 256

Convolution
⇓
when we applying the filter (3*3) we get the points in the Convolution output (5*5) [receptive] and when applying the Pooling window (3*3), we get the points that in the Pooling layer.

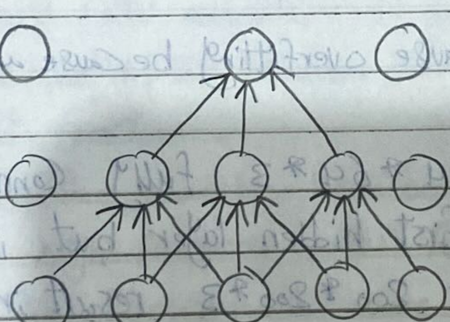


Pooling layer

Pooling layer

Convolution layer

input image



The size of the receptive field is <u>5 X 5</u> or we can say 5*5*3

[7] input data block in a Convolutional network has dimension C*H*W

= 96 * 128 * 128

↳ channels ↳ spatial dimension

applying Conv, filter = D*C * HF * WF = 128 * 26*7*7

stride 2, Pad 3

→ dimension of the output = $\frac{\text{size} - \text{filter} + 2P}{S} + 1$

$= \frac{128-7+(2*3)}{2} + 1 = 64.5$ ∴ $\boxed{\text{output} = 128 * 64 * 64}$

SUB
DATE

[8] → Just like Traditional dropout, inverted dropout randomly keeps some weights and sets others to zero. This is known as the " Keep Probability " P. The one difference is that, during the Training of a neural network, inverted dropout scales the activations by the inverse of the keep probability q = 1 - P.

→ inverted Dropout is how Dropout is implemented in Practice in the various deep learning Frameworks because it helps to define the model once and just change a parameter to run Train and test on the same model.

→ This prevents network's activations from getting too large, and does not require any changes to the network during evaluation.

[9] because it cause overfitting because of large number of Parameters
for example ~ 64 * 64 * 3 fully connected layer need 12288 weights in the first hidden layer but in Real life image have at least 200 * 200 * 3 result in 120 000 or 225 * 225 * 3 which result in 151875 weights in the first hidden layer also this layer require
- Large storage to store the weights
- Long Time to train
- long Time to classify an input image.
- Also when the Place of class or object is changed it can't identify it.

Nile

SUB
DATE

10 • Given Two arrays A[] and B[] Consisting of N and M integers respectively, The task is To construct a Convolution array C[] of size (N+M-1).

• The Convolution of 2 arrays is defined as $C[i+j] = \Sigma(a[i] * b[j])$ for every i and j

⇒ our Example:

input: $A[] = \{4, 1, -1, 3\}$; $B[] = \{-2, 1\}$

size of array, $C[] = N+M-1 = 4+2-1 = 5$ ⇒ (Length)

$C[0] = A[0] * B[0] = 4 * -2 = -8$

$C[1] = A[0] * B[1] + A[1] * B[0] = 4*1 + 1*-2 = 2$

$C[2] = 3$; $C[3] = -6$; $C[4] = 3$

∴ The output $= \{-8, 2, 3, -6, 3\}$

11

we use The learning rate adaptor to handle This Problem, every Time The loss begins to Platea, The learning rate decreases by a set fraction.

→ The belief is That The model has become Caught in region with The "high learning rate" Reducing The learning rate will allow The optimizer To more efficiently Find The minimum in The loss surface. At This time, one might be concerned about Converging to local minimum.

12

because Conv. layer is more flexible than fully Connected because it isn't densely Connected The input doesn't affected To all output nodes. So, The number of weights Per layer is small which helps alot with high dimensional input. Such as image Processing. ~~and They assume That~~ also The Convolution layers are Explicit hierarchical representation

Nile

SUB
DATE

of features. The best Thing in CNN architecture is no need for feature extraction.

- Reduces overfitting :- if The model is massively overfitting you can Start adding dropout in small pieces.
- Translation invariant.

[13] • The term "dropout" refers to dropping out The nodes (input and hidden layer) in a neural network. All The forward and backwards connections with a dropped node are Temporarily removed, Thus creating new network architecture out of Parent network. The nodes are dropped by Probability of P

- in the original implementation of dropout layer, during Training a unit in layer is selected with a Keep Probability (1 - drop Probability). This creates a Thinner architecture.

- During The inference (test), we do not use a dropout layer, This means That all The units are Considered during The Prediction step. but, because of Taking all units from a layer, The final weights will be larger Than expected and To deal with This is Problem, weights are first scaled by The chosen dropout rate. With This, The network would be able to make accurate Predictions.

⇒ To be more Precise, if a unit is retained with Probability P during training, The outgoing weights of That unit are multiplied by P during The Prediction Stage.