Statistical Methods in AI	Date:	*06/03/2019*
Instructor: *Dr. Ravi Kiran*	Submitted By:	Tarun Mohandas (2018201008)

Assignment 6: Convolutional Neural Networks

1 Question 1

1.1 LeNet Convolutional Neural Network

In this part of the question, we are expected to implement the LeNet Architecture for doing a forward propagation of Convolutional Neural Network.

1.1.1 Model

1. **Input**: $32 \times 32 \times 3$ image
2. **Output**: One of 10 classes
3. **Activation Function**: ReLU
4. **Output Function**: Softmax function
5. **Layers**:
   (a) Convolution Layer 1
   (b) Max Pooling Layer 1
   (c) Convolution Layer 2
   (d) Max Pooling Layer 2
   (e) Convolution Layer 3
   (f) Fully connected Layer
   (g) Output Layer
6. **Parameters**
   (a) Weights of the kernel
   (b) Weights of fully connected layers

1.1.2 Architecture Specific details

1. Convolution Layer 1
   (a) Input is $32 \times 32 \times 3 image$
   (b) 6 $5 \times 5 \times 3$ kernels (filters) for convoluting on the $32 \times 32 \times 3$ image
   (c) Output is 6 28×28 feature maps
   (d) Stride is 1
   (e) Padding is 0
2. Max Pooling Layer 1
   (a) Input is 6 28×28 feature maps
   (b) 2×2 max pooling window applied to all the feature maps
   (c) Stride is 2
   (d) Output is 6 14×14 matrices
3. Convolution Layer 2
   (a) Input is $14 \times 14 \times 6$ tensor
   (b) Output is 16 10×10 feature maps
   (c) 16 $5 \times 5 \times 6$ kernels (filters) for convoluting on the $14 \times 14 \times 6$ tensor
   (d) Stride is 1
   (e) Padding is 0
4. Max Pooling Layer 2
   (a) Input is 16 10×10 feature maps
   (b) 2×2 max pooling window applied to all the feature maps
   (c) Stride is 2
   (d) Output is 16 5×5 matrices
5. Convolution Layer 3
   (a) Input is $5 \times 5 \times 16$ tensor
   (b) Output is 120 1×1 feature maps
   (c) 120 $5 \times 5 \times 6$ kernels (filters) for convoluting on the $5 \times 5 \times 16$ tensor
   (d) Stride is 1
   (e) Padding is 0
6. Fully Connected Layer (120×84)
7. Fully Connected Layer (84×10)

1.2 Algorithm

1.2.1 Convolution operation

In the convolution operation a kernel is applied on the image matrix along with the depth of the input. The output of the convolution operation is a 2D feature map. The dimension of output feature map can be computed with the following formula:

$$W_{out} = \frac{W_{in} - F + 2P}{S} + 1$$
$$H_{out} = \frac{H_{in} - F + 2P}{S} + 1$$

where W_{out} is the width of the feature map, H_{out} is the height of the feature map, W_{in} is the width of the input and H_{in} is the height of the input, $F \times F$ is the dimension of the kernel, P is the padding and S is the stride of the operation. The following is the formula for convolution operation.

$$S_{ij} = (I * K)_{ij} = \sum_{a=\lfloor -\frac{m}{2} \rfloor}^{\lfloor \frac{m}{2} \rfloor} \sum_{b=\lfloor -\frac{n}{2} \rfloor}^{\lfloor \frac{n}{2} \rfloor} I_{i-a,j-b} K_{\frac{m}{2}+a,\frac{n}{2}+b}$$

where $m \times n$ is the dimension of the kernel and i and j are the indices of the feature map.

1.2.2 Max Pooling operation

In the max pooling operation, the pooling window is strided across the input and in each stride, the maximum of the window is taken and put in the output. This operation is just for reducing the dimension of the data. The alternative for this is average pooling, but in this assignment we have only worked on max pooling. This operation also helps in preventing overfitting and provides a regularized effect on the input.

1.2.3 Linear transformation

In the fully connected layers, we flatten the output from the convolution layer and make a linear transformation with the weight parameters. The following is the formula to do that.

$$z = Wx + b$$

The bias is added for better fitting the data points.

1.2.4 Activation

In this assignment, we use the ReLU activation function. The activation function is to provide a non-linearity to the inputs in order to better fit non-linearly separable data points. The following is the relu activation function.

Figure 1: Original image of the dog sitting

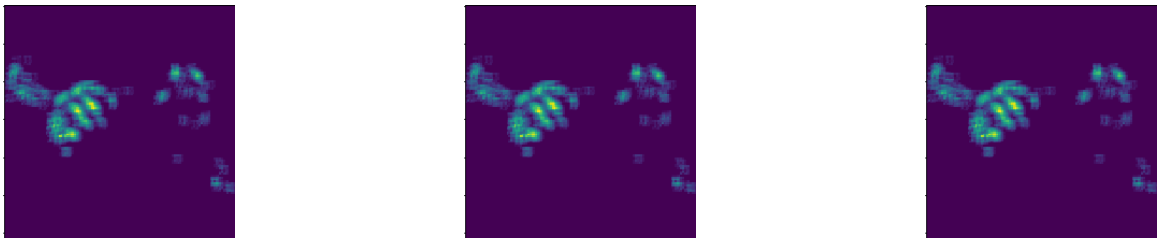

Figure 2: Images after Convolution Layer 1

$$a = max(z, 0)$$

1.2.5 Output Function

In the last part, we use an output function, which in our case is the softmax function. The softmax function gives the probability distribution of all the classes according to how likely the image can be. The following is the softmax function.

$$softmax(z) = \frac{e^{z_i}}{\sum_n e^{z_n}} \forall i \in n$$

1.3 Results

In this assignment, only doing the forward propagation of the network. The images of each intermediate results after every layer was captured to better understand what each layer did. The following are the results received. Figure 1 is the picture of the dog taken for forward propagating across the network.

The first convolution layer captures the specific parts of this image without generalizing it too much. In Figure 1 we can see that the convolution layer has clearly captured the specific information of dog and we can see the dog in a sitting position. The first max pooling layer just blurs out the output from the input as we can see in Figure 3. In figure 4 we see the images after the convolution layer 2. As we can see, the features of the dog has been generalized. It does not capture specific features of the dog as it is. By figure 6 we can see that by the third convolution layer, there is

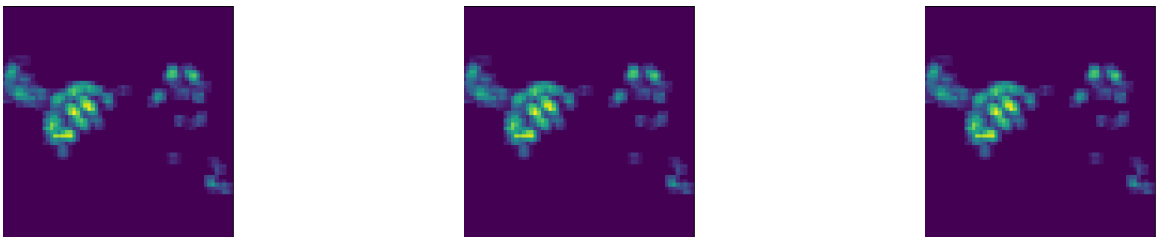

Figure 3: Images after Max Pooling layer 1

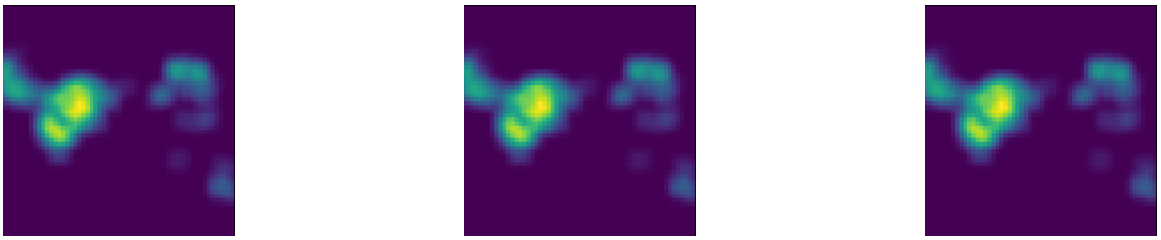

Figure 4: Images after Convolution Layer 2

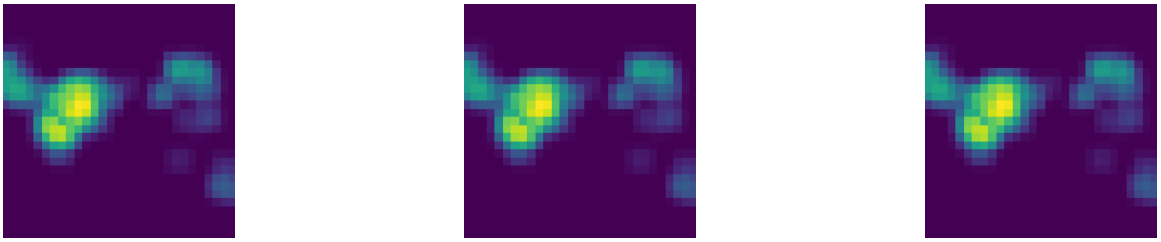

Figure 5: Images after Max Pooling layer Layer 2

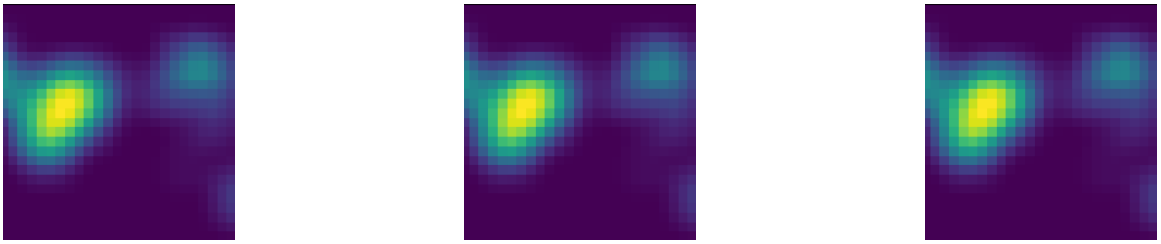

Figure 6: Images after Convolution layer Layer 3

only generalized feature that is being captured, which captured the body, tail, head and almost captured the hand.

2 Question 2

2.1 What are the number of parameters in 1st convolutional layers ?

The kernel dimensions are $5 \times 5 \times 3$ and there are 6 such kernels. So number of parameters $= 5 \times 5 \times 3 \times 6 = 450$

Answer: 450

2.2 What are the number of parameters in pooling operation?

Since there are no parameters to be learnt, the number of parameters of any max pooling layer is 0.

Answer: 0

2.3 Which of the following operations contain most number of parameters?

The most number of parameters is 3^{rd} the convolution layer where the total number of parameters is $120 \times 5 \times 5 \times 16 = 48120$ where we have 120 $5 \times 5 \times 16$ kernels.

Answer: 3^{rd} convolution operation.

2.4 Which operation consume most amount of memory?

Answer: Fully connected layer costs the most amount of memory because of large matrix multiplications.

3 Question 3

3.1 5. Write an explanation of what you did.

In the Question 3 I explored the basics of Tensor flow and Keras (with base as tensor flow) framework to build convolution neural network of 3 layers. Neural network layers were built using tf.nn library in barebone tensor flow. In Keras, the sequential model was explored where the model and the optimization function has to be initialized by us and trained on CIFAR10 dataset. In cifar 10 dataset, I got a 78.3% test accuracy and 82% train accuracy. The following is the model I used to train the CIFAR10 dataset.

1. Convolution Layer 1

2. Activation Layer 1 (ReLU)
3. Convolution Layer 2
4. Activation Layer 2 (ReLU)
5. Max pooling layer 1
6. Dropout Layer 1 (p = 0.25)
7. Convolution Layer 3
8. Activation Layer 3 (ReLU)
9. Convolution Layer 4
10. Activation Layer 4 (ReLU)
11. Max pooling layer 2
12. Dropout Layer 2 (p = 0.25)
13. Dense Layer 1
14. Activation Layer 5 (ReLU)
15. Dropout Layer 3 (p = 0.25)
16. Dense Layer 2
17. Activation Layer 6 (Softmax)