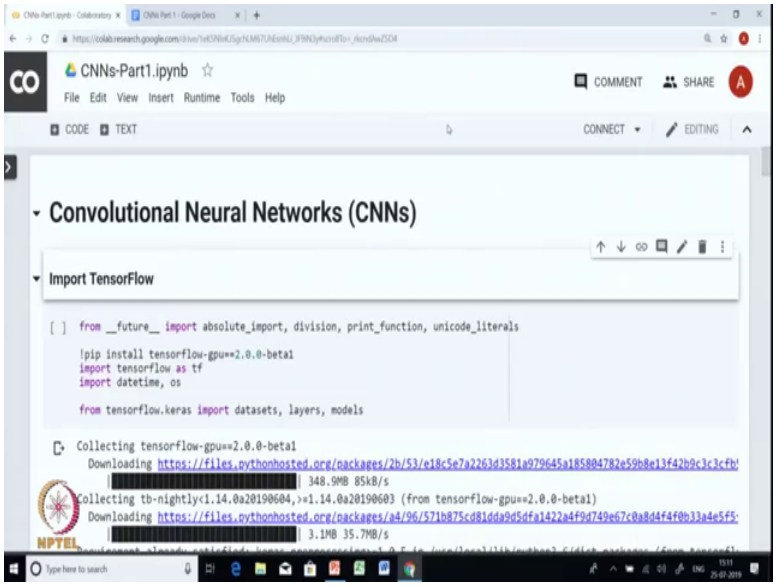
Practical Machine Learning
Dr. Ashish Tendulkar
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 20
Convolutional Neural Network: Part 1

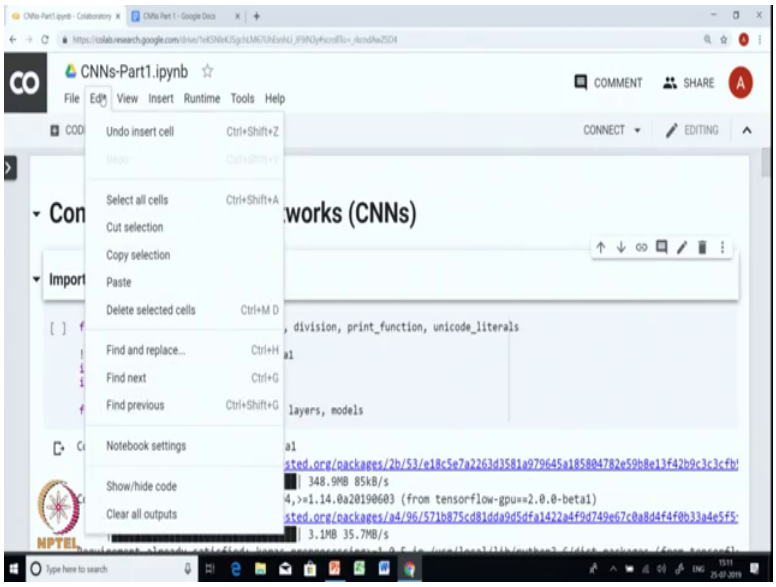
[FL] So far, in this course we have been building machine learning models with feed forward neural networks. Today, we will learn about a new neural network architecture called Convolutional Neural Networks or CNNs. Let us first Colab environment for CNNs.

(Refer Slide Time: 00:38)



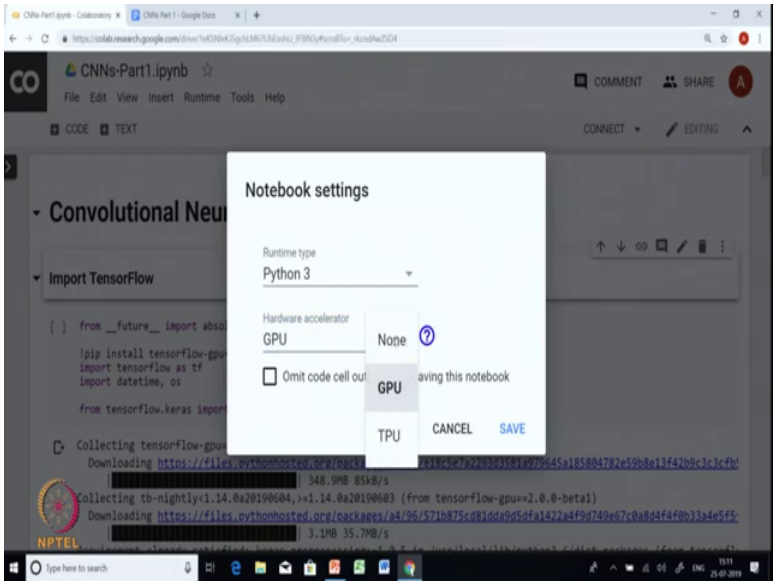
CNNs trains faster with GPUs. So, we will enable 3 GPUs in Colab like this.

(Refer Slide Time: 00:47)



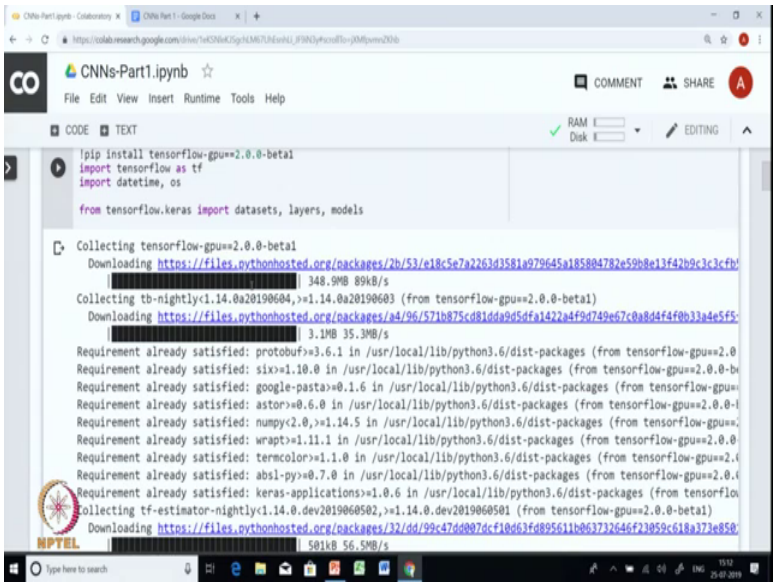
We will open Edit.

(Refer Slide Time: 00:54)



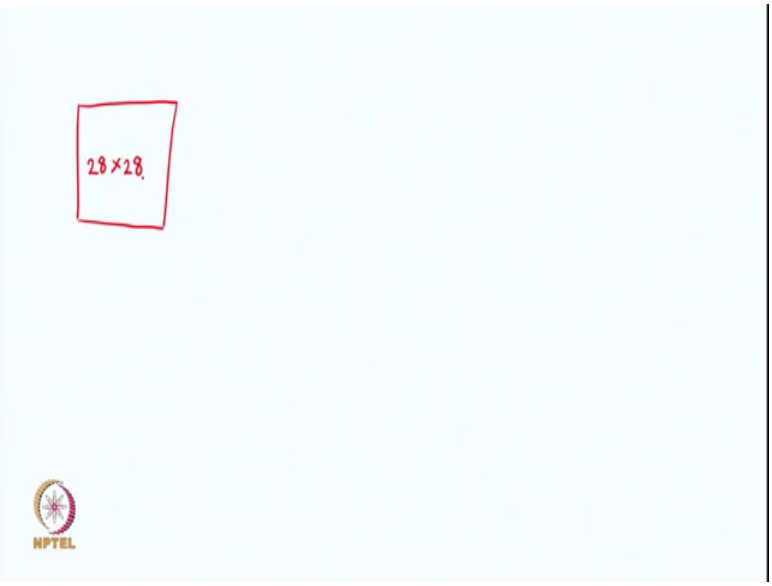
Go to the Notebook settings and in the Hardware accelerator we see like GPUs and we Save the settings. Now, the notebook is enabled to run with GPU run-time. Let us install and import all the necessary libraries in the colab run-time.

(Refer Slide Time: 01:11)



CNNs are used extensively in computer vision. They are also used for modeling temporal data. The key idea in CNNs is to capture local patterns in the data through convolution operation. It then down samples the resulting output with pulling layer. CNNs employ a series of convolution and pulling layers. Let us understand CNNs with MNIST Digit-Recognition example. In this case CNNs take an image as an input and predicts the corresponding digit as an output.

(Refer Slide Time: 02:02)



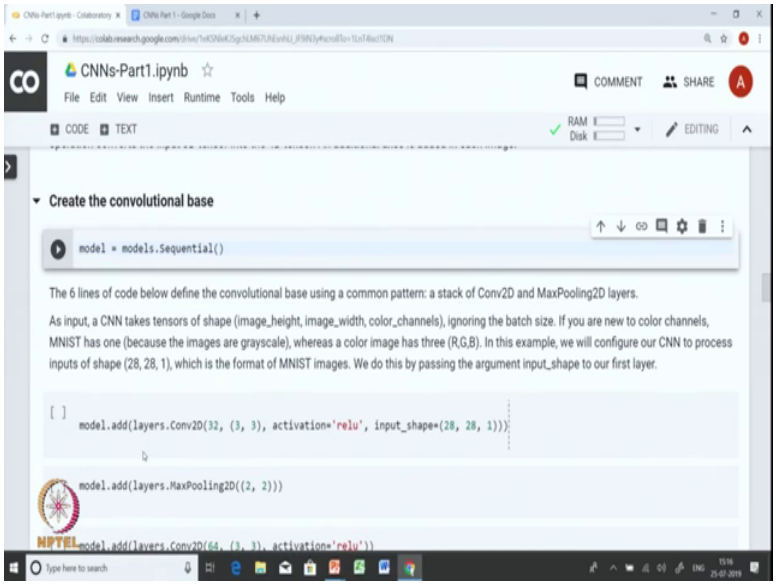
The image is encoded as a 3D tensor with access corresponding to height, width and depth of an image. In case of MNIST data we are dealing with grayscale images and hence the depth is one for us while height and width are both 28. We will download and pre-process MNIST handwritten digital mission data set just as before except for a small change.

Here we reshape each image into a 4D tensor. This is done to satisfy the input requirements of the CNNs. So, the reshaping is done with a reshape command that you can see here on your screen. Both train images and test images tensors are reshaped into 4D tensors.

We can observe that the reshaping operation converts the 3D tensors into 4D tensors. The shape of the train and test tensors for each image changes from 28 height of 28 and width of 28 to height and width of 28 each and a depth of 1. Now that we have prepared data for training let us build our model. We will first create the convolution base. We using equation model.

Let us add a convolution layer to the model.

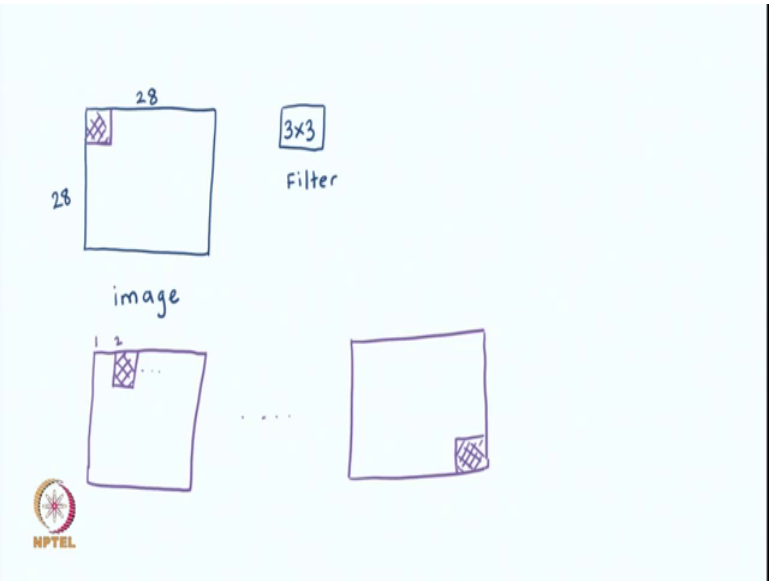
(Refer Slide Time: 03:50)



Let us understand about convolution operation. The convolution layer is added using layers.Conv2D command. The convolution operation involves a filter which captures

local pattern and applies it on the image. The filter is a 3D tensor of a specific width, height and depth. Each entry in the filter is a real number. The entries in the filter are learnt during CNN transition. The filter slides across width, height and depth stopping at all possible positions to extract a local 3D patch of surrounding feature.

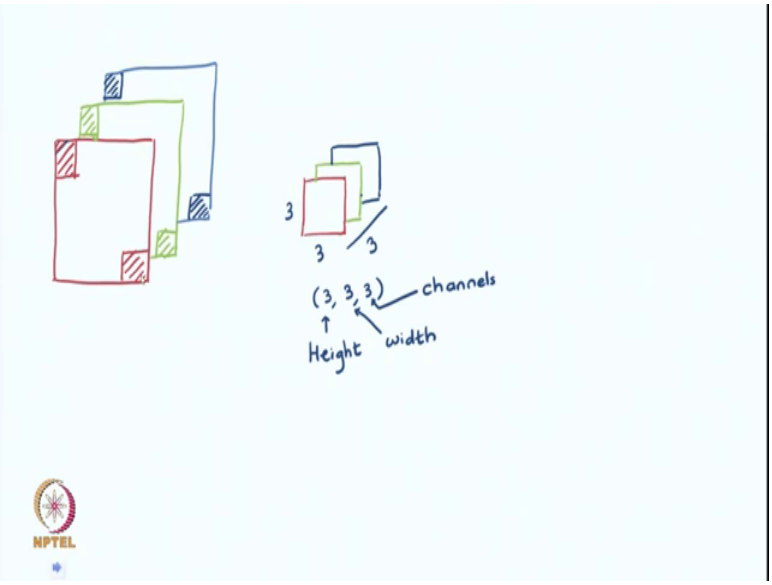
(Refer Slide Time: 04:41)



Let us understand about convolution operation on handwritten digit. So, this is our handwritten digit encoded in 28 x 28 image. Since it is a grayscale image, the depth is 1. The convolution operation defines a filter of specific height and width.

Let us say our filter is of size 3 x 3. We take the filter and we position it at different places in the image. If we slide the filter we get a new position of the filter. We keep doing this across the length and breadth of the image till the final positioning of the filter. Since it was a grayscale image we had depth of 1. Let us try to understand how the convolution happens with images that are colored.

(Refer Slide Time: 07:20)



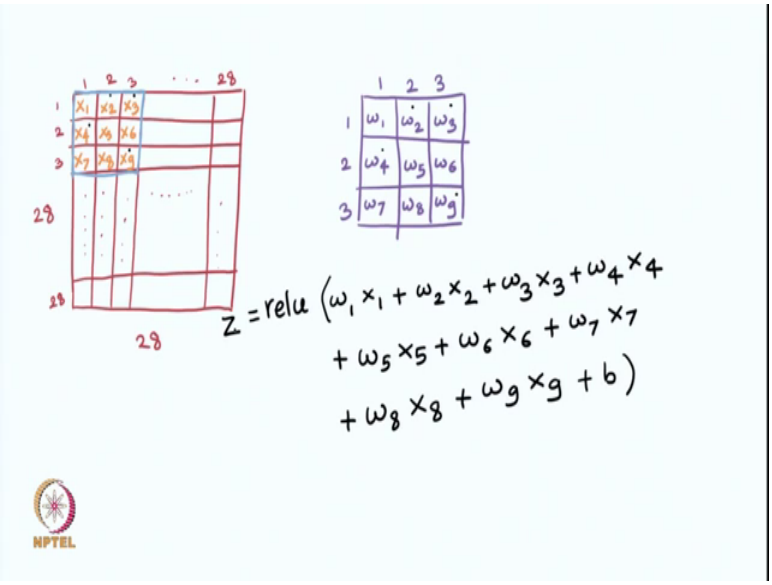
In case of colored images we have three channels; for every image we have three channels red, green and blue. This is red channel, and there is a green channel and then there is a blue channel. So, let us understand how a filter is defined on an image with multiple channels.

So, we will have a filter which also has three channels. So, we have let us say 3 x 3 filter which has got depth of 3. So, this 3 x 3 filter is represented as a 3D tensor. In this case the depth is 3: red, green and blue.

For each position we slide the filter across width and height of the colored image until we get to the last portion, until we get to the last patch.

We transform every such patch with a convolution filter into a number or 1D array. Let us understand what happens when you position the filter at a particular patch of an image.

(Refer Slide Time: 10:58)



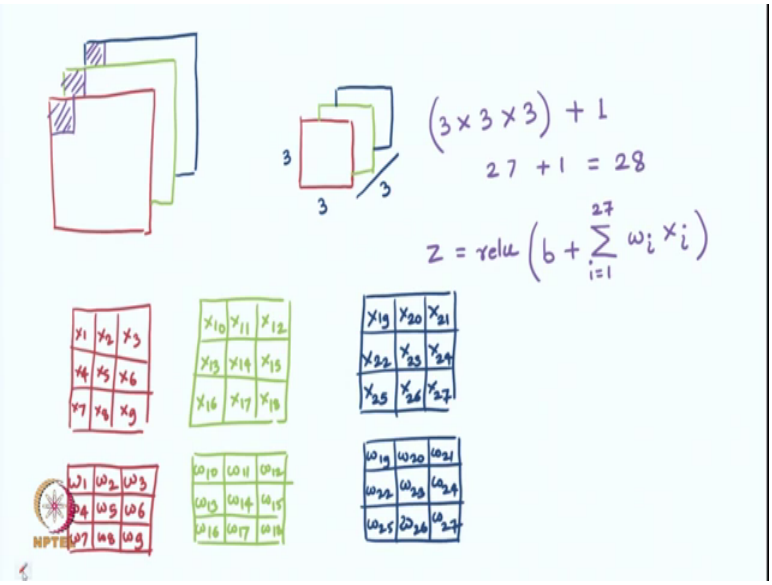
Let us say this is our handwritten digit from MNIST dataset which has got 28 rows and 28 columns which has got height of 28 and width of 28.

We define a 3 x 3 filter. Filter has 9 entries and each entry is a weight let us say $w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8$ and w_9. We take this filter and we position it at this particular patch of the image.

Let us say the image has the following inputs the patch of the image also has nine positions. So, We position this particular filter on top of this image. So, we perform linear combination of the weight in the filter with the feature value in the image patch. So, in this case we have $w_1x_1 + w_2x_2 + \ldots + w_9x_9 + b$.

You can see that this particular equation is very very is a similar to equation that we saw in feed forward neural network and we apply a Relu activation on this linear combination which gives us a variable Z. Z is a scalar quantity for a given positioning of a patch on the image. Let us try to understand how the 3D patch works.

(Refer Slide Time: 15:30)



Let us say this is the image and we have corresponding filter. Remember, we have the same number of channels in the filter as the input image since the input image has three channels we also have three channels in the filter. So, let us say this is 3 x 3 filter with 3 channels. So, when you position this particular filter in the image; let us say this is the first position of the filter. So, this filter is positioned across the channel.

We have red patch which has got inputs $x_1, \ldots, x_9$. Then we have a green patch which also has let us call this inputs as and we have a blue patch. So, we have a patch which has got 27 inputs (9 for each channel). We also have corresponding filters there is a filter for blue channel one for green and one more for the red channel and the corresponding weights are $w_1, w_2, w_3, w_4, \ldots, w_{27}$.

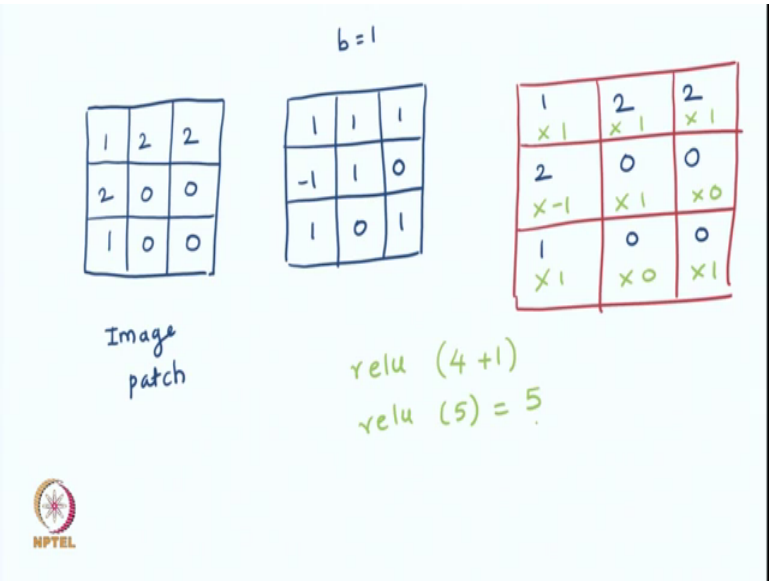
So, we have 27 weights in the filter, 9 corresponding to each of the 3 channels. We have one bias unit for this particular filter. So, we have 3 x 3 x 3 parameters and one bias. So, which gives us 28 weights per filter. We calculate the linear combination of weight and the corresponding value in the image patch and add a bias to it and apply an activation like Relu to get a scalar number.

So, this is how we apply a 3D filter to a 3D image. This filter generalizes to number of channels more than 3 in exactly the same manner. Only thing that will be different is we

will essentially get number of channels which are same as the input and that will also increase the number of parameters for a filter.

Let us take a concrete example of a 3 x 3 filter on a grayscale image.

(Refer Slide Time: 21:54)



Let us say this is a patch extracted from a grayscale image and this is the filter which is also which is a 3 x 3 filter has got 10 parameters 9 parameters corresponding to the positions and one is a bias term. Let $b = 1$. The way the linear combination works is as follows. We will write the values in the image in blue and each of which is multiplied by the weight of the parameter in the filter.

So, we super impose a filter on the patch and multiply the number in the image patch with the weight of a parameter in the filter. The linear combination comes to 4 + 1 and if you apply Relu on it. Relu of a positive number is the positive number. So, we get 5 as an answer from this particular patch of the image. So, 5 signifies the strength of this local pattern as represented by the filter at this particular image patch.

So far we studied about convolution operation we learned how filters are created and how you how we evaluate a filter at a position at a particular position in the image. We slide the filter across the image at all possible positions and assess the strength of local pattern represented by the filter at each position.