Assignment -6

Using CNN for determining the type of dress or item.

Name : Hari teja Narayanabhatla

Panther ID: 002566679

Group : MSA

**Business Question:** Humans can easily understand if an object is Shoe, dress, sandal, etc.

When it comes to an E-commerce website how does the machine know that the product

that’s been sold is a dress, coat, shoe, sandal, etc. In order to answer this question, We are

coming up with a model which can solve this problem and helps the machine identify

which product is being sold on their website?

**Dataset Overview:**

The datasets consist of several items like coats, shoes, sandals, trousers, shirts, formal shirt etc..

There are no null values and all these are image data with size of 28\*28 which means they are 2D

Images i.e and each pixel is a value from 0 to 255 describing the pixel intensity. 0 for white and 255

for black.

There are 60k training images and 10k test images along with class labels.

**Data Visualization:**

**A screenshot of a video game

Description automatically generated with medium confidence**

**Convolution Neural Network:**

Since a convolutional layer takes a 2-D image (2-D array) as input, we needn't convert this 2-D

images to 1-D arrays. If we had to feed these 2-D images to a DNN (hidden layer), which expects a 1D

array as input, we would have to convert these 2-D images to 1-D arrays before feeding them to a

hidden layer (DNN). Since each pixel is varying from 0-to 255 . we will normalize the data by making

the image binarized by dividing it with 255. This will make image pixels into 0 or 1 with 0 as black and

1 as white.

Graphical user interface, text, email

Description automatically generated

I have created a Sequential model where it takes the input and moves step by step and above is the

architecture. These are the explanation for all steps:

1. Using Conv2D, created the input layer which has 32 filters. Each filter try to identify

the important features like edge detection, vertical and horizontal regions and give an

the intermediate output which will be passed on to the next layer.

1. Created another Conv2D layer where the output from Step 1 is given as an input to this

The layer has a shape of (26,26,32). The shape of output from step 1 is different from

input image this is because of the filters that are used to understand important features.

1. We apply different filters using conv2d to prevent the loss of data information. Suppose use

only one layer with 64 filters. The input image gets processed multiple times and results in

a feature that is not necessary for the model to recognize. To prevent this we use multiple

Conv2d Layer so that actual information is not lost.

1. The resultant shape of the input image is (24,24,64). This output will be given as an input to

max pooling where only max values get selected which resembles the important feature.

1. Step 3 and step 4 are repeated to get clearer results or concrete results.
2. Result after pooling maps is flattened so that the data can be converted to a single dimension

instead of 2D which in turn helps in classifying the images.

1. The resultant flattened is passed to a dense layer with 128 neurons with an activation function

of SoftMax which gives a probabilistic score for each input image. Then the results are sent to

the output layer is made of the dense layer with 10 neurons since we have 10 different

objects in our dataset using the SoftMax activation function.

So the model is trained using 10 epochs which means the data will be sent over 10 times to get

accurate results or precise results

**Accuracy and results:**

The model gives an accuracy of 92.19 % on the test dataset. To visualize the predicted

labels. I have visualized the test images with their labels.

Diagram

Description automatically generated

**Graphical Results of accuracy and loss:**

**Chart, line chart

Description automatically generated**

**Fig 1: Accuracy of training and accuracy of test data**

**Chart, line chart

Description automatically generated**

**Fig 2: Train data loss and test data loss**

**Epoch Understanding and Conclusion:**

**Chart, line chart

Description automatically generated**

From the graph, we can conclude that the model gives the best accuracy when the model

is trained over 6 epochs but not for 10 epochs. We can also conclude that increasing the

number of epochs doesn’t guarantee that model will give the best accuracy, it’s about how

model architecture and what activation, optimizers are used to achieve the best accuracy.