dl 3

Image Classification: is the task of tacking an input image and outputting a class or a probabilty of classes that best describe the image.

Softmax:

Softmax is primarily used in the output layer of a neural network, especially in classification tasks where the goal is to assign a probability distribution over multiple classes.

Softmax takes a vector of real numbers as input and transforms it into a probability distribution by exponentiating each element and normalizing the results. This ensures that the sum of the output probabilities is equal to 1.

ReLU introduces non-linearity to the network by replacing negative input values with zeros and leaving positive values unchanged.

Softmax is used in the output layer to produce class probabilities for classification, while ReLU is employed in hidden layers to introduce non-linearity and enable the network to model complex relationships in the data

An optimizer in the context of machine learning and neural networks is an algorithm or method used to adjust the parameters of a model in order to minimize the error or loss function during the training process.

types:SGD,ADam,RMSprop

Sparse Categorical Cross-Entropy (Sparse CCE) is a loss function used in machine learning and deep learning, particularly in multi-class classification problems

so in the code we use mnist dataset which contain over 10000 images and the we split it into training and testing after that

we show the shape of the data which id (60000,28,28)

here 28\*28 are the pixel parameters

so the given input shape is (28,28,1) where 1 is images presented in grayscale so the input channel is 1

then we change the shape of the xtrain and xtest

(60000, 28, 28, 1) we will get this as output

here the datatype of dataset is int so we changed into float value .

after that we scale the xtrain and x test to 0 to 1 by dividing it by 255 which is the range of grayscale

now we define the model architecture

model=Sequential([

Conv2D(28,kernel\_size=(3,3),input\_shape=input\_shape),

MaxPooling2D(pool\_size=(2,2)),

Flatten(),

Dense(200,activation="relu"),

Dropout(0.3),

Dense(10,activation="softmax")

])

we use sequential model here for first layer we use conv2d with 28 neurons and with filter size 3\*3

then we use maxpooloing with the matrics of it is 2

then we flatten the pooling layer output and pass it to the hidden layer where we use dense layer with 200 neuron after getting the result of it we do the droput od some neurons by 0.3 which is 30%

then softmax for output layer

model.compile(optimizer='adam',loss='sparse\_categorical\_crossentropy',metrics=['accuracy'])

model.fit(xtrain,ytrain,epochs=10)

then we compile the model using adam optimizing function and we took loss func as sparse categorical crossentropy and metrics used as accuracy.as it is a standard metrics

then we fit the data with epochs 10 as it will increse the accuracy

then

image=xtrain[5]

plt.imshow(np.squeeze(image),cmap='gray')

plt.show()

here we choose the input of index 5 and we sueeze it means reduce the single dimension entries will be removed

image=image.reshape(1,input\_shape[0],input\_shape[1],input\_shape[2])

predict\_model=model.predict([image])

print("predicted class : {}".format(np.argmax(predict\_model)))

now to predict the class we reshape the image with four parameteres

then we predict the reshape image here we use np.argmax it is a func who gives max probabilty of a class