ASSIGNMENT 3 REPORT

N SRVIJAY RAM PRATHAP
2019H1030559G

TASK1:

PROBLEM STATEMENT:

Construct a good Denoising autoencoder to MNIST dataset.

IMPLEMENTATION:

Dataset:

Training data: 48,000 greyscale images of size 28*28.

Validation data: 12,000 greyscale images of size 28*28.

Model:

The model consists of three sections.
1. Encoder
2. Code
3. Decoder

Model is build using Convolution neural networks as the Convolution neural networks works better with image data. An Autoencoder is in which the output is expected to be same to the input given. In denoising autoencoder an image with noise is given as an input and the loss will be calculated according to the image without any noise. This helps the model to reduce the risk towards the learning of identity function.

For this introduced salt and pepper noise to our data which is also known as impulse noise. This noise introduces sharp and sudden disturbances in the image signal. The model is having the following layers in it.

Encoding:

The encoding architecture is composed of 3 Convolutional Layers and 3 Max Pooling Layers stacked one by one. “Relu” is used as the activation function in the convolution layers and padding is kept as "same". The three Convolution layers the number of nodes are in decreasing order as 64, 32 and 16.

Decoding:

Similarly in decoding architecture, the convolution layers will be used having same dimensions (in reverse manner 16,32,64) as the encoding architecture. But instead of 3 maxpooling layers, we will be adding 3 upsampling layers. Again the activation function will be same (Relu), and padding in convolution layers will be same as well.

Training:

To train the model we are using the train dataset of 48000 images.
- Number of epochs = 200
- Batch_size = 2048
- For regularization used earlystopping as callback with validation loss as a monitoring function.

ASSIGNMENT 3 REPORT

N SRVIJAY RAM PRATHAP
2019H1030559G

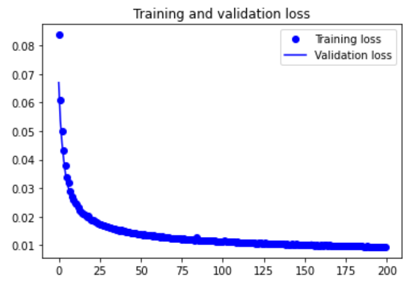
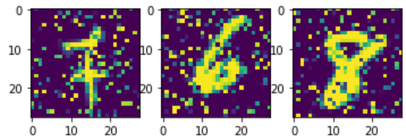


Figure 1: Training and validation accuracy

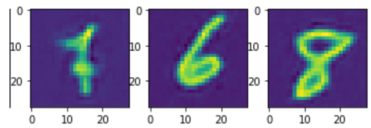
This graph show that the training loss and validation loss are almost similar so that the model is not overfitting to the training data.

RESULTS:

Input Image:



Output Image:



When a noise image is given as input the autoencoder successfully produced an image by denoising it. The final training loss is 0.0095 which indicates a good identity function property which is expected from an autoencoder.

Task 2:

Problem Statement:

To make a classification model on Fashion MNIST dataset to classify the images.

Implementation:

Dataset:

Train data: 54000 grayscale images of size 28*28.

Validation data: 6000 grayscale images of size 28*28.

Test data: 10000 grayscale images of size 28*28.

The model consists of convolution neural network to classify the images into labels. They are 10 output labels to classify into. The model consists of 3 convolution layers with maxpooling and dropout layers added to them. The activation function used all the convolution layers is 32, 64, 128. The activation function used for all the layers is "RELU". The dropout layers are used for regularization to reduce risk of the model to be overfitting.

The output of the convolution layers then passed to two more dense layers of 128 node and 10 node. The final dense layer is given an activation function of "softmax" and "relu" to the before layer.

- Loss function: Categorical cross entropy.
- Optimizer: Adam.

- Metric: Accuracy.

Training:

Using a dataset of size 48,000 greyscale image of size 28*28 to train the model. Parameter are as follow:

- Batch size = 256
- epochs = 20

The training and validation loss and accuracy can be observed as follows:

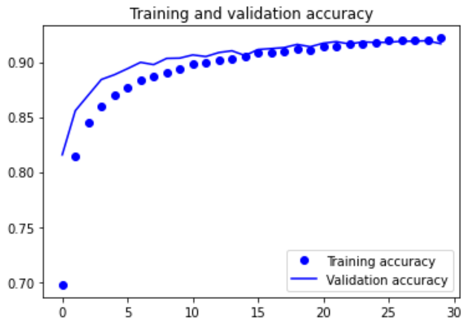


Figure 2: Training and validation accuracy of Model2

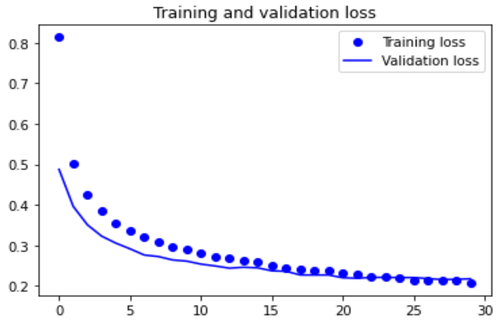


Figure 3: Training and validation loss of Model2

As per the training and validation loss and accuracy graphs are similar we can say that the model is not overfit to the train data. The train accuracy after 30 epochs is 0.9187 and train loss value is 0.2185.

Results:

To test the accuracy of the model 10,000 grayscale images of size 28*28 with original labels is used. The model predicted the label of all the images with an accuracy of 92.4 % and loss of 0.24 .

Task 3:

Problem Statement:

To classify Fashion MNIST dataset using the pertained model from task 1.

Implementation:

Dataset:

Train data: 54000 grayscale images of size 28*28.

Validation data: 6000 grayscale images of size 28*28.

Test data: 10000 grayscale images of size 28*28.

Model:

Let's say that Autoencoder we trained in task1 as AE. Constructed a model similar to encoder part of AE. Let's say this model is M3. To make it as a classification model added a flatten layer and dense

ASSIGNMENT 3 REPORT

N SRVIJAY RAM PRATHAP
2019H1030559G

layer with 10 nodes. This encoder layers that constructed similar to AE will be assigned to the trained weights of AE. This is known as knowledge transfer from one model to another model. As

MNIST dataset (D1) is similar to fashion MNIST (D2) , This knowledge transfer is used to learn the hidden Structures of the data.
By unfreezing all the layers to new data, all these layers will be again trained on D2 to finetune this model to the new dataset of D2. Output dense layer have 10 nodes with “softmax” activation to classify the data. All other layers are having a “RELU” activation only.

- Loss function: Categorical cross entropy.
- Optimizer: Adam.
- Metric: Accuracy.

Training:

The model is given with good weight initialization from AE from task1. All the layers are trained again on a dataset of 54,000 images of 28*28 size. For validation task 6,000 images of size 28*28 are used.

The following the training parameter:

- Batch size = 256
- Number of epochs = 20
- Call back = early stopping with patience = 10, monitor as “validation loss”.

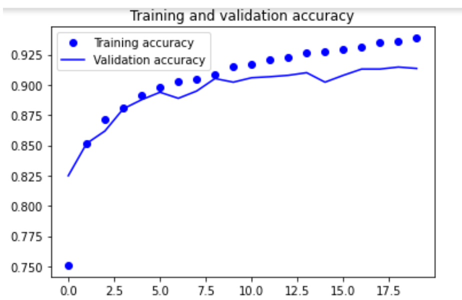


Figure 4: Training and validation accuracy of Model3

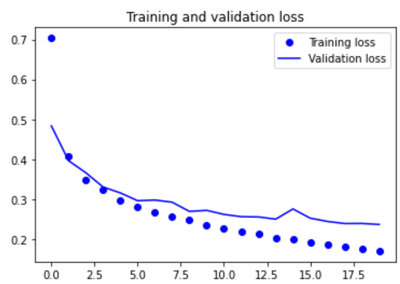


Figure 5: Training and validation loss of Model3

We can observe the training accuracy and validation accuracy started to divide as the validation accuracy remained same and training accuracy keeps on increasing. So stopped the training with only 20 epochs.

Results:

The model is tested with a dataset of 10,000 images of size 28*28 with true label values. This model gave an accuracy of 91.9% accuracy and a loss of 0.22.

ASSIGNMENT 3 REPORT

N SRVIJAY RAM PRATHAP
2019H1030559G

Comparing performance of M2 and M3:

	CNN Classifier in task2 (P)	Autoencoder as classifier(P’)
Train Accuracy	92%	93.3%
Train Loss	0.21	0.18
Test Accuracy	92.4%	91.9%
Test Loss	0.24	0.22

CNN Classifier in task2 (P):

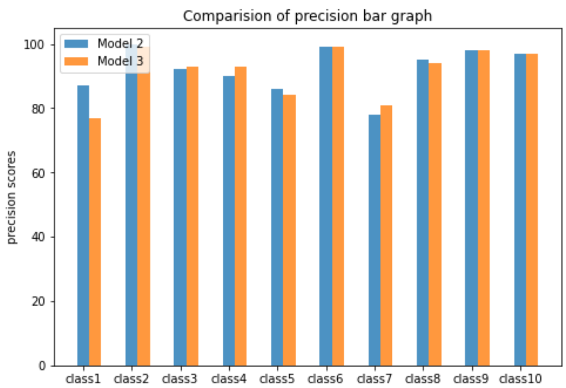
	precision	recall	f1-score	support
Class 0	0.87	0.89	0.88	1000
Class 1	1.00	0.98	0.99	1000
Class 2	0.92	0.82	0.87	1000
Class 3	0.90	0.96	0.93	1000
Class 4	0.86	0.90	0.88	1000
Class 5	0.99	0.97	0.98	1000
Class 6	0.78	0.77	0.78	1000
Class 7	0.95	0.98	0.96	1000
Class 8	0.98	0.99	0.99	1000
Class 9	0.97	0.97	0.97	1000
accuracy			0.92	10000
macro avg	0.92	0.92	0.92	10000
weighted avg	0.92	0.92	0.92	10000

Autoencoder as classifier (P’):

	precision	recall	f1-score	support
Class 0	0.77	0.87	0.81	1000
Class 1	0.99	0.98	0.99	1000
Class 2	0.93	0.82	0.87	1000
Class 3	0.93	0.93	0.93	1000
Class 4	0.84	0.92	0.88	1000
Class 5	0.99	0.98	0.98	1000
Class 6	0.81	0.72	0.76	1000
Class 7	0.94	0.96	0.95	1000
Class 8	0.98	0.98	0.98	1000
Class 9	0.97	0.96	0.97	1000
accuracy			0.91	10000
macro avg	0.92	0.91	0.91	10000
weighted avg	0.92	0.91	0.91	10000

Precison and Recall graphs of both models:

CNN Classifier in task2 (P):



Autoencoder as classifier (P’):