Machine Learning 2

Report for Assignment 3 (AutoEncoders and PCA)

Vikanksh Nath (MT19AI024)

Task 1

Objective: Implement a simple encoder-decoder neural network with one hidden layer using the following images (images of George Bush from the LFW Face database): http://vis-www.cs.umass.edu/lfw/lfw-bush.tgz

Use the MSE (Mean Square Error) loss and 70:20:10 train:val:test split to train the network.

Data-Preparation: The downloaded dataset contains 530 images of size (250 X 250) pixels with 3 channels (i.e. RGB images). So with the mentioned split, the number of training, validation and test images are For this task, all images are resized to (64 X 64) pixels with 3 channels. The size is reduced to avoid the out of memory error when working with 250 X 250 pixel images. This out of memory error occurs due to the large weight tensor W_large size of (187500 X 5000). W_large has the dimensions as input layer size X hidden layer size. If 250 X 250 image with 3 channels is used then input layer size becomes = 250 X 250 X 3 = 187500, so by practice to encode this large input vector number of hidden layer units should be around 5000. It results in W_large with size of (187500 X 5000). So, to prevent this out of memory error the all images are resized into (64 X 64) pixels with 3 channels. It results in a weight tensor of size (12288 X 1000) between the input layer and hidden layer. Details are in network architecture.

Moreover, our aim is to reconstruct the input with minimum MSE loss. So, investigating the network reconstruction capability with a different input size is justifiable.

With the given split we have 371 training, 106 validation and 53 testing samples.

Figure. Some of Training Images (pixel size: 64X64)

Network Architecture:

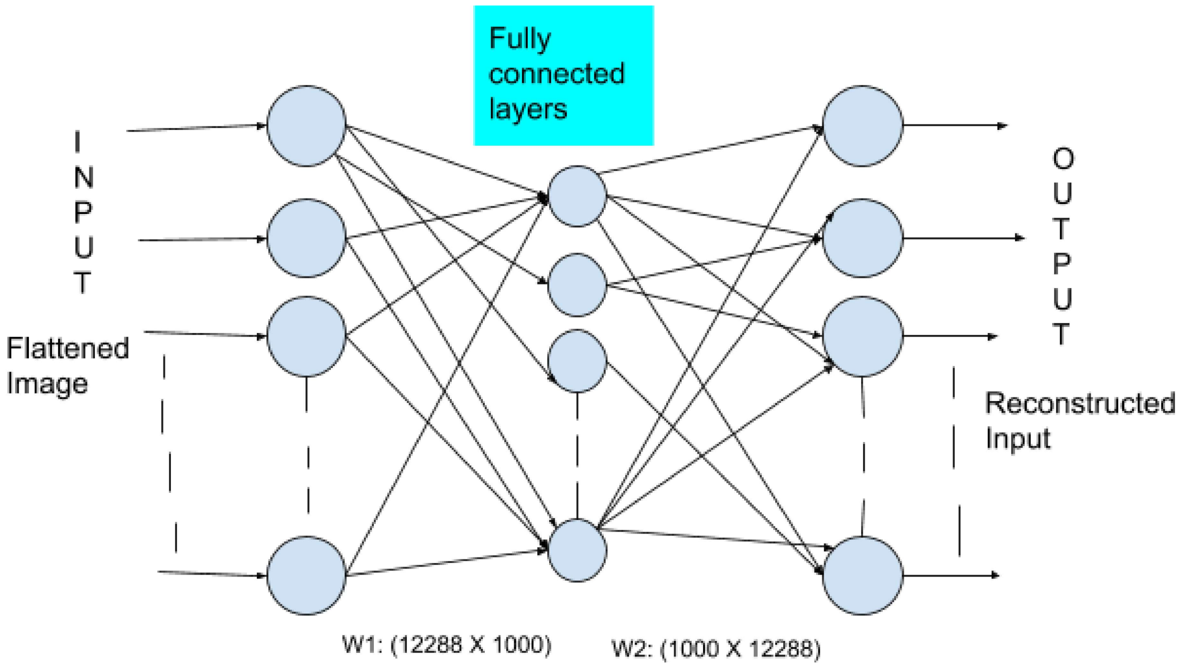

Figure . A simple encoder decoder network with a single hidden fully connected layer.

The input layer has 12288 units which is the size of our flattened input image. As, 64 X 64 X 3 = 12288. The output is nothing but the reconstructed input. So, the output layer has the same number of units as the input layer. The hidden layer has chosen to have 1000 units. So after loading the images in the tensor form, the images were flattened and fed to the network during training as well as testing. The reconstructed input i.e. output of the network is again reshaped into (64 X 64 X 3) and reconstructed images are obtained. The MSE loss is defined between the input and output vector of this network. It is worth noting that before feeding the flattened image to network as input, pixel values are re-ranged to [0,1] from [0,255]. This image normalization helps in optimal and fast training of the network.

Training-Phase: Adam is used as default solver, the 64 is used as default batch size. Different experiments are conducted to optimize the network performance for minimum validation loss. The results with different experiment are as follows:

Experiment 1

Only Adam solver with Early Stopping

Adam learning rate(lr)	Epochs taken	Optimal training loss	Optimal validation loss
0.0001	554	0.0021	0.0145
0.001	91	0.0040	0.0175
0.00001	> 1000	0.0079	0.0168
0.00005	912	0.0019	0.0143
default	89	0.0056	0.0178

Table . Loss in Experiment 1 with different learning rates

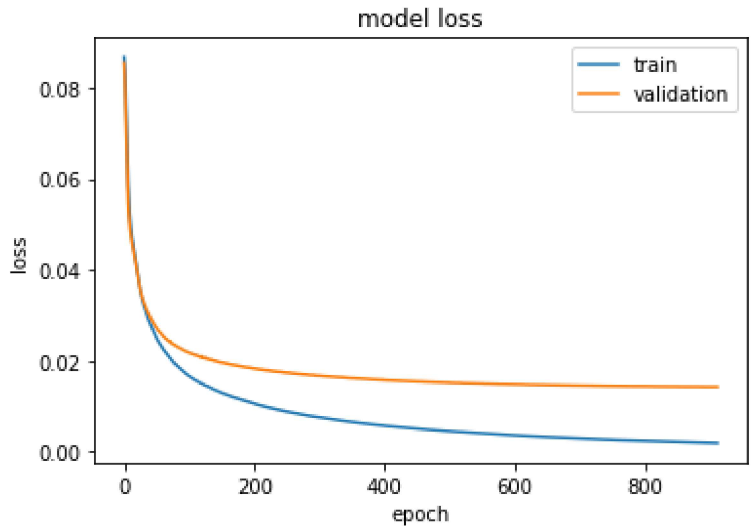

Figure . Loss in Experiment 1, lr = 0.00005

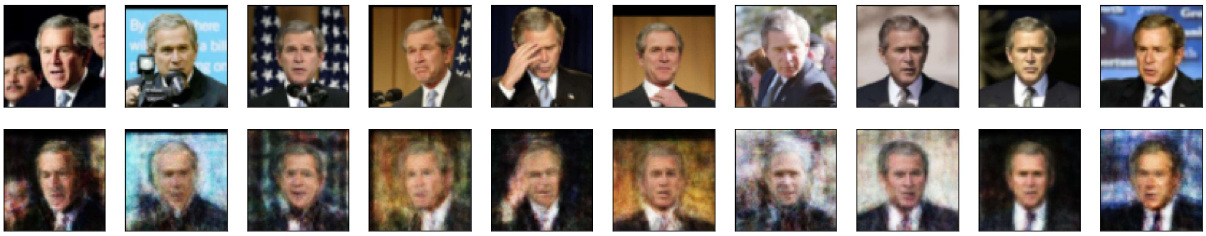

Figure . Reconstructed Validation Images (2nd Row) Experiment 1, lr = 0.00005

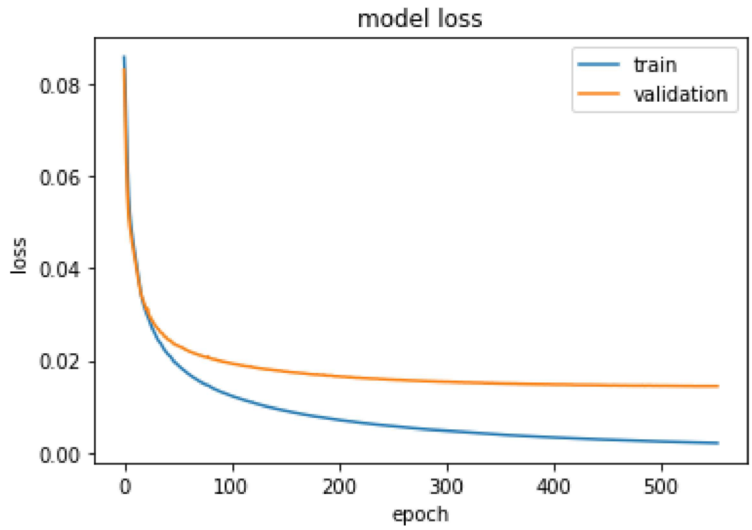

Figure . Loss in Experiment 1, lr = 0.0001

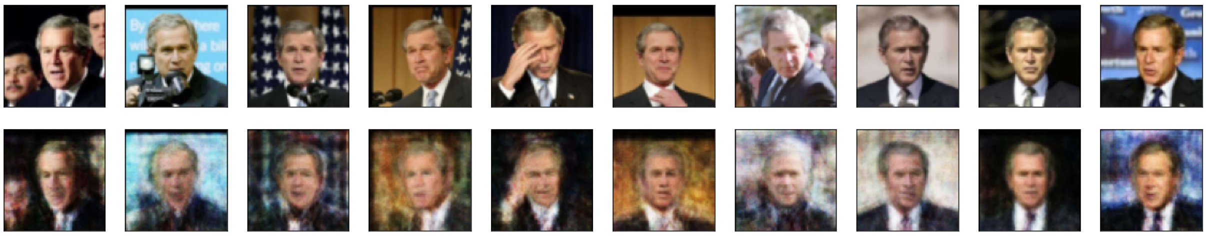

Figure . Reconstructed Validation Images (2nd Row) Experiment 1, lr = 0.0001

Experiment 2

Used “ReduceLROnPlateau(monitor='val_loss', factor=0.1, patience=7, verbose=1, epsilon=1e-4, mode='min')” a tensor flow module to modify learning rate, as it reduces learning rate when a metric has stopped improving. We monitored validation loss. I also employed Early Stopping.

Adam learning rate(lr)	Epochs taken	Optimal training loss	Optimal validation loss
0.0001	250	0.0066	0.0162
0.001	98	0.0053	0.0176
0.00001	447	0.0158	0.0214
0.00005	273	0.0092	0.0176
default	111	0.0051	0.0176

Table . Loss in Experiment 2 with different learning rates

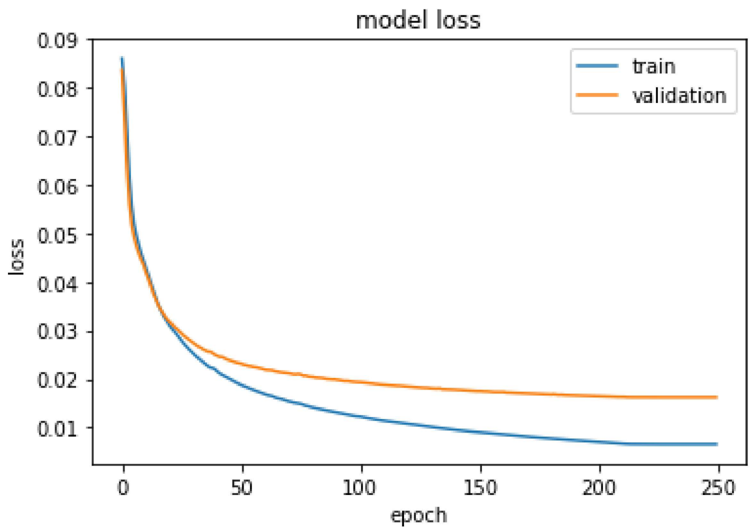

Figure . Loss in Experiment 2, lr = 0.0001

Figure . Reconstructed Training Images (2nd Row) Experiment 2, lr = 0.0001

Testing-Phase: Qualitative results on testing samples are shown below for optimal cases of training experiments.

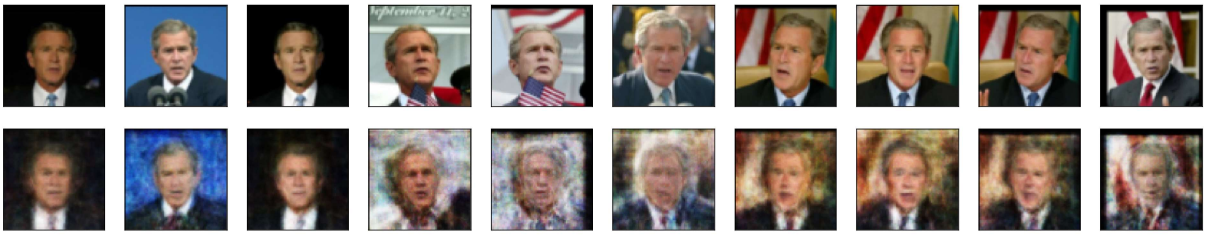

Figure . Reconstructed Test Images (2nd Row) Experiment 1, lr = 0.00005

Figure . Reconstructed Test Images (2nd Row) Experiment 1, lr = 0.0001

Figure . Reconstructed Test Images (2nd Row) Experiment 2, lr = 0.0001

Training Phase	**Training loss**	**Validation loss**	**Testing loss**
Exp.1 lr= 0.00005	0.0019	0.0143	0.014063602
Exp.1 lr= 0.0001	0.0021	0.0145	0.014048886
Exp.2 lr= 0.0001	0.0066	0.0162	0.016399488

Table . Testing Loss for most suitable configuration of Experiment 1 and 2.

Analysis on AutoEncoder Results :

From different experiments the one most suitable configuration is to use Adam solver with a learning rate of 0.0001 . As with other configurations validation loss as well as testing losses are higher. It is worth noting that with different run during reproduction of results, the number of epochs and losses may vary upto |5%|.

Moreover reconstructed testing images are shown, which indicates that visually images are more or less of the same quality with an optimally trained neural network from different configurations.

Task 2

PCA on Images :- PCA is performed on an “Image Matrix” having each flattened image as the row and and the pixel values as the columns. So in our case “Image Matrix” dimension is as (No. of Image Samples X 12288). So, different experiments were performed by changing the No. of Image Samples used to create Image Matrix and corresponding MSE loss with different top eigen vectors are shown as follows (detail qualitative image results and quantitative results are shown in submitted Notebook it self):

Experiment 1 (PCA):

All 530 samples used: Image Matrix (530 X 12288)

Some Results of Experiment 1 (PCA) are shown below.

No. of top eigenvectors	10	20	30	40	50	60
MSE loss	0.0263	0.0194	0.0155	0.0131	0.0112	0.0098

No. of top eigenvectors	10	110	210	310	360
MSE loss	0.0263	0.0057	0.0024	0.0011	0.0007

Table . MSE Loss in Experiment 1(PCA) with different top eigenvectors

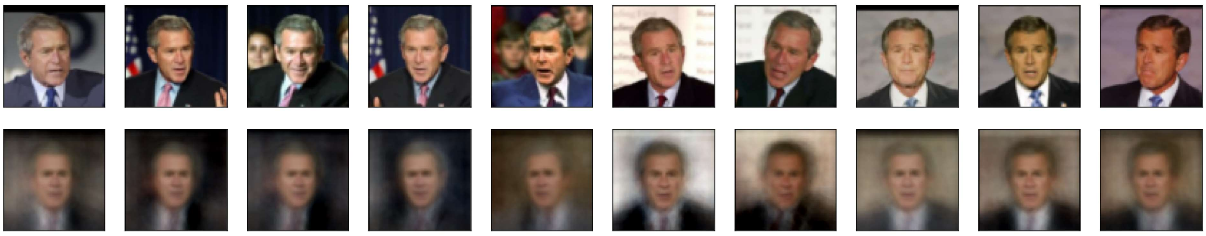

Figure . Reconstructed Images (2nd Row) from PCA with no of Top EigenVector: 10

Figure 3. Reconstructed Images (2nd Row) from PCA with no of Top EigenVector: 360

Experiment 2 (PCA):

All 371 training samples used: Image Matrix (371 X 12288)

Some Results of Experiment 2 (PCA) are shown below.

No. of top eigenvectors	10	20	30	40	50	60
MSE loss	0.0262	0.0191	0.0151	0.0125	0.0106	0.0091

No. of top eigenvectors	10	90	120	150	180
MSE loss	0.0263	0.0062	0.0043	0.0031	0.0022

Table . MSE Loss in Experiment 2(PCA) with different top eigenvectors

Figure 3. Reconstructed Images (2nd Row) from PCA with no of Top EigenVector: 90

Experiment 3 (PCA):

All 106 validation samples used: Image Matrix (106 X 12288)

Some Results of Experiment 3 (PCA) are shown below.

No. of top eigenvectors	10	20	30	40	50	60
MSE loss	0.0223	0.0147	0.0104	0.0074	0.0052	0.0036

No. of top eigenvectors	10	70	80	90
MSE loss	0.0223	0.0024	0.0014	0.0007

Table . MSE Loss in Experiment 3(PCA) with different top eigenvectors

Figure . Reconstructed Images (2nd Row) from PCA with no of Top EigenVector: 20

Figure . Reconstructed Images (2nd Row) from PCA with no of Top EigenVector: 50

Experiment 4 (PCA):

All 53 testing samples used: Image Matrix (53 X 12288)

Some Results of Experiment 4 (PCA) are shown below.

No. of top eigenvectors	10	15	20	25	30	35
MSE loss	0.0192	0.0136	0.0098	0.0071	0.0050	0.0033

No. of top eigenvectors	10	40	45
MSE loss	0.0192	0.0020	0.0009

Table . MSE Loss in Experiment 4(PCA) with different top eigenvectors

Figure . Reconstructed Images (2nd Row) from PCA with no of Top EigenVector: 15

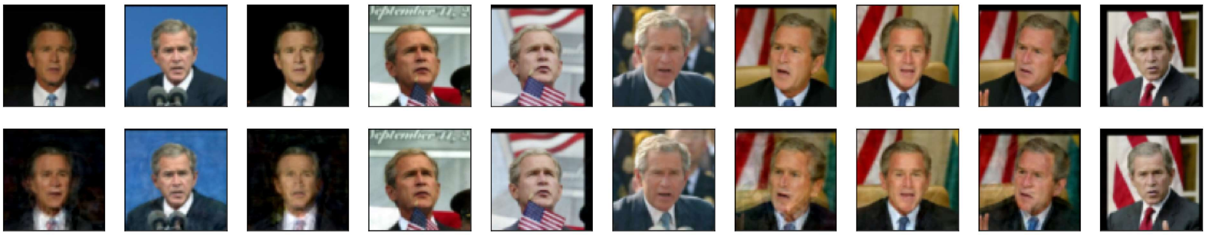

Figure . Reconstructed Images (2nd Row) from PCA with no of Top EigenVector: 45

Analysis on PCA results : It is found that for every experiment the losses are reduced and quality of reconstructed images increases with increase in number of top eigenvectors. For an Image Matrix of size (No. of Image Samples X 12288) the total number of eigenvectors are equal to “No. of Image Samples”. When the number of top eigenvectors reaches towards “No. of Image Samples” i.e. reconstruction with almost all eigenvectors (PCA components) then the quality of reconstructed images improves and they tend to become the same as Input Images with very less MSE losses.

Comparison of AutoEncoder and PCA on Testing Samples:

Figure . Reconstructed Images (2nd Row) from PCA with no of Top EigenVector: 15

MSE loss = 0.0136

Figure . Reconstructed Test Images (2nd Row) Experiment 1, lr = 0.0001

MSE loss =`0.014048886`

As seen from above figures and whole analysis that for top 10 eigenvectors the PCA on testing samples give a MSE loss (0.0136) which is similar to optimal MSE testing loss (0.014048886) with Autoencoders. For these similar losses, the reconstructed images by PCA and Autoencoders are visually of the same quality. But with PCA the quality of reconstructed images can be improved significantly with increment of top eigenvectors (PCA components) while in AutoEncoders despite of optimal training it is very difficult to reconstruct the images of same quality as Input images and with very low MSE loss as it is possible in case of PCA.