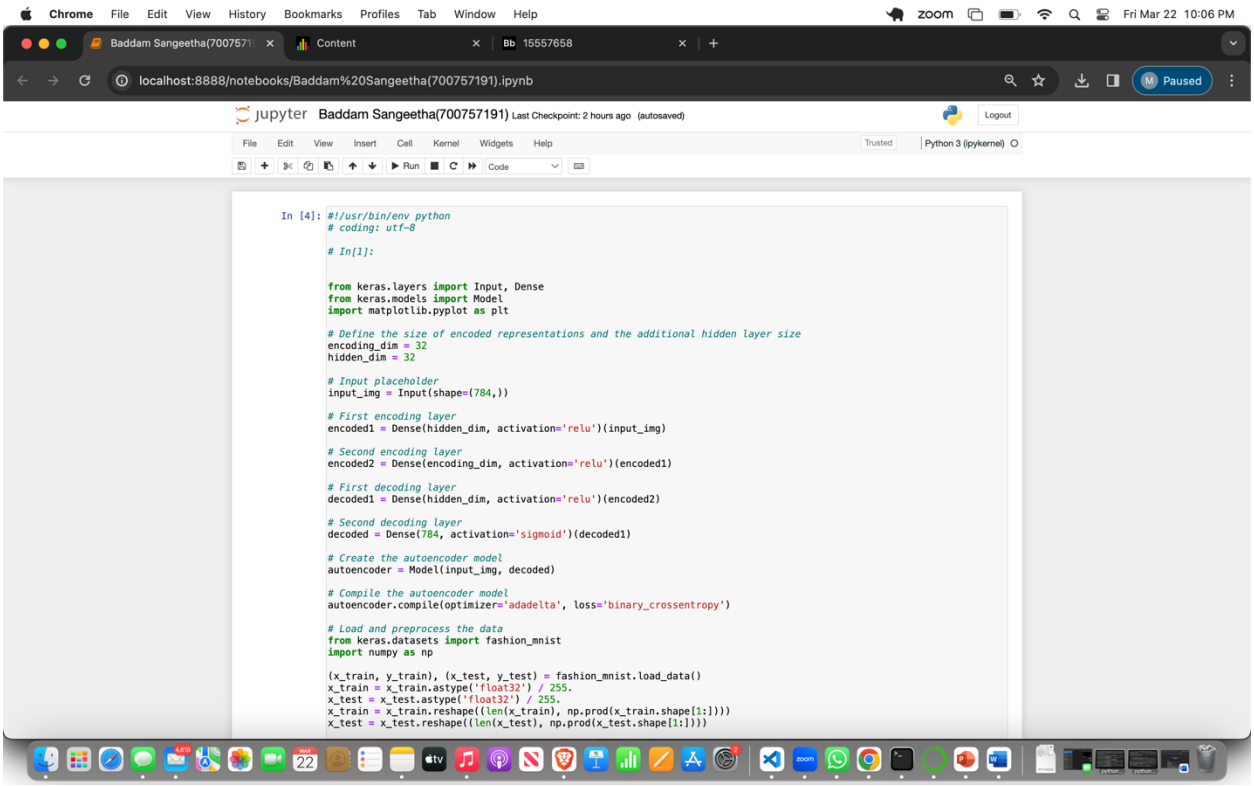
Spring 2024: CS5720 Neural Networks & Deep Learning – Autoencoders
Assignment- Autoencoders
Name: Baddam Sangeetha
STUDENT ID:700757191

GitHub link:
https://github.com/Sangeetha-Baddam/Assignment-8

Video link:
https://drive.google.com/file/d/1Jq67MamyNzLV3OI2Srmnc1Oj4nsiWVZX/view?usp=drive_link

Lesson Overview:
In this lesson, we are going to discuss types and applications of Autoencoder.
Programming elements:
1. Basics of Autoencoders
2. Role of Autoencoders in unsupervised learning
3. Types of Autoencoders
4. Use case: Simple autoencoder-Reconstructing the existing image, which will contain most important features of the image
5. Use case: Stacked autoencoder



Chrome File Edit View History Bookmarks Profiles Tab Window Help

Baddam Sangeetha(700757 | Content | 15557658

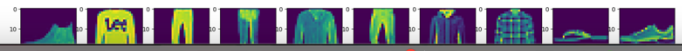
localhost:8888/notebooks/Baddam%20Sangeetha(700757191).ipynb

jupyter Baddam Sangeetha(700757191) Last Checkpoint: 2 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

```python
# Train the autoencoder
history = autoencoder.fit(x_train, x_train, epochs=5, batch_size=256, shuffle=True, validation_data=(x_test, x_test))


# Predict and visualize one of the reconstructed test data
decoded_imgs = autoencoder.predict(x_test)

n = 10  # Number of digits to display
plt.figure(figsize=(20, 4))
for i in range(n):
    # Display original images
    ax = plt.subplot(2, n, i + 1)
    plt.imshow(x_test[i].reshape(28, 28))

# Display reconstructed images
    ax = plt.subplot(2, n, i + 1 + n)
    plt.imshow(decoded_imgs[i].reshape(28, 28))

plt.show()

# Visualize the loss and accuracy
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
```

```
Epoch 1/5
235/235 [==============================] - 1s 3ms/step - loss: 0.6935 - val_loss: 0.6935
Epoch 2/5
235/235 [==============================] - 1s 3ms/step - loss: 0.6935 - val_loss: 0.6934
Epoch 3/5
235/235 [==============================] - 1s 3ms/step - loss: 0.6934 - val_loss: 0.6934
Epoch 4/5
235/235 [==============================] - 1s 3ms/step - loss: 0.6933 - val_loss: 0.6933
Epoch 5/5
235/235 [==============================] - 1s 3ms/step - loss: 0.6933 - val_loss: 0.6932
313/313 [==============================] - 0s 379us/step
```



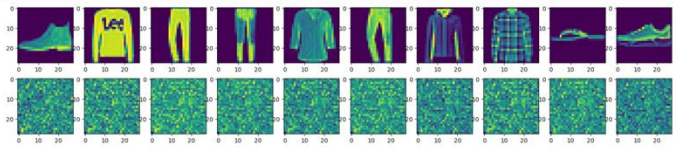
Chrome File Edit View History Bookmarks Profiles Tab Window Help

Baddam Sangeetha(700757 | Content | 15557658

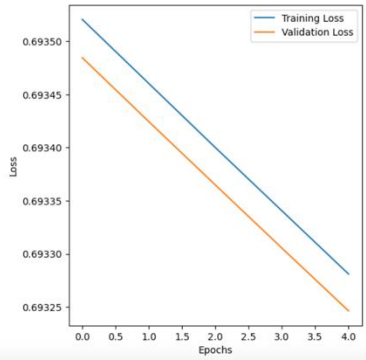
localhost:8888/notebooks/Baddam%20Sangeetha(700757191).ipynb

jupyter Baddam Sangeetha(700757191) Last Checkpoint: 2 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

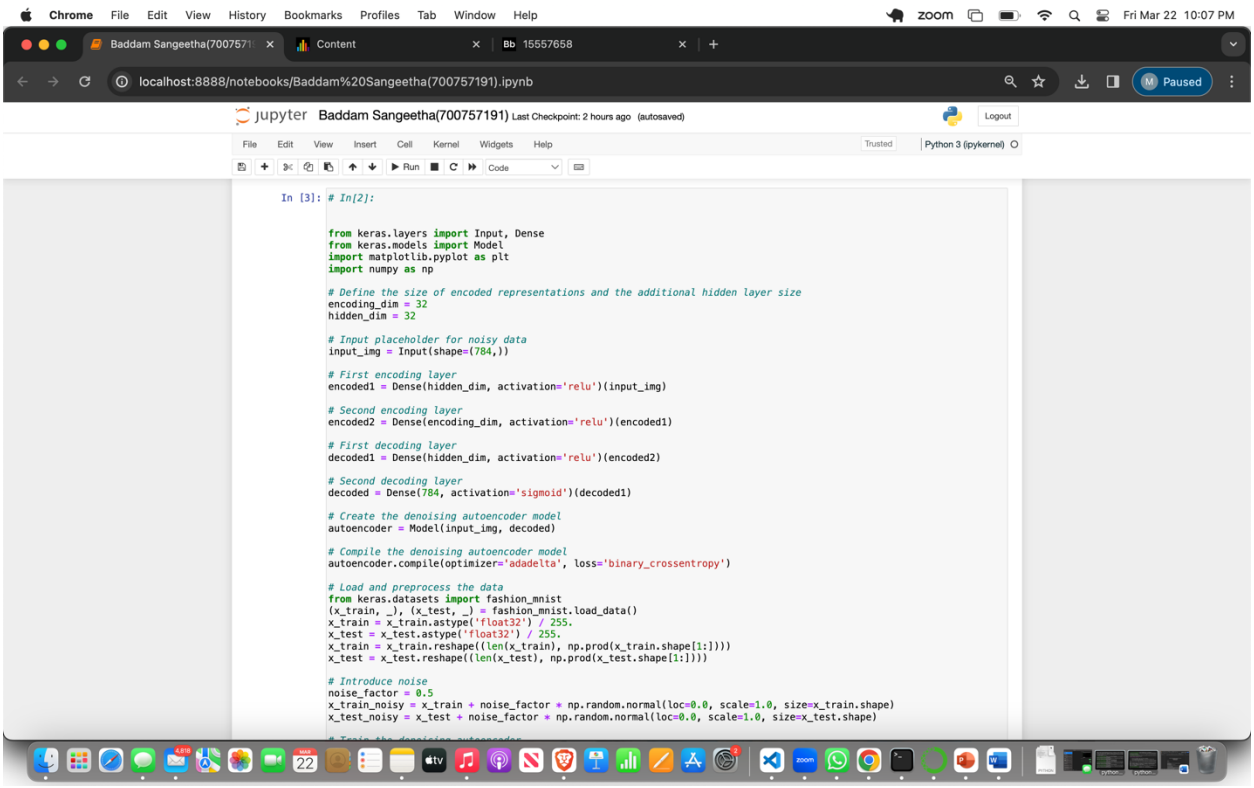


```
Out[4]: <matplotlib.legend.Legend at 0x2803a67d0>
```



In class programming:

1. Add one more hidden layer to autoencoder
2. Do the prediction on the test data and then visualize one of the reconstructed version of that test data. Also, visualize the same test data before reconstruction using Matplotlib
3. Repeat the question 2 on the denoisening autoencoder
4. plot loss and accuracy using the history object



Chrome File Edit View History Bookmarks Profiles Tab Window Help Fri Mar 22 10:07 PM

Baddam Sangeetha(700757 | Content | 15557658

localhost:8888/notebooks/Baddam%20Sangeetha(700757191).ipynb

jupyter Baddam Sangeetha(700757191) Last Checkpoint: 2 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

```
for i in range(n):
    # Display noisy images
    ax = plt.subplot(3, n, i + 1)
    plt.imshow(x_test_noisy[i].reshape(28, 28))

    # Display original images
    ax = plt.subplot(3, n, i + 1 + n)
    plt.imshow(x_test[i].reshape(28, 28))

    # Display reconstructed images
    ax = plt.subplot(3, n, i + 1 + 2 * n)
    plt.imshow(decoded_imgs[i].reshape(28, 28))

plt.show()

# Visualize the loss and accuracy
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
```

```
Epoch 1/10
235/235 [==============================] - 1s 3ms/step - loss: 0.6936 - val_loss: 0.6936
Epoch 2/10
235/235 [==============================] - 1s 3ms/step - loss: 0.6936 - val_loss: 0.6935
Epoch 3/10
235/235 [==============================] - 1s 3ms/step - loss: 0.6935 - val_loss: 0.6934
Epoch 4/10
235/235 [==============================] - 1s 3ms/step - loss: 0.6934 - val_loss: 0.6933
Epoch 5/10
235/235 [==============================] - 1s 3ms/step - loss: 0.6933 - val_loss: 0.6932
Epoch 6/10
235/235 [==============================] - 1s 3ms/step - loss: 0.6932 - val_loss: 0.6931
Epoch 7/10
235/235 [==============================] - 1s 3ms/step - loss: 0.6931 - val_loss: 0.6931
Epoch 8/10
235/235 [==============================] - 1s 3ms/step - loss: 0.6930 - val_loss: 0.6930
Epoch 9/10
235/235 [==============================] - 1s 3ms/step - loss: 0.6930 - val_loss: 0.6929
Epoch 10/10
235/235 [==============================] - 1s 3ms/step - loss: 0.6929 - val_loss: 0.6928
313/313 [==============================] - 0s 421us/step
```

Chrome File Edit View History Bookmarks Profiles Tab Window Help Fri Mar 22 10:07 PM

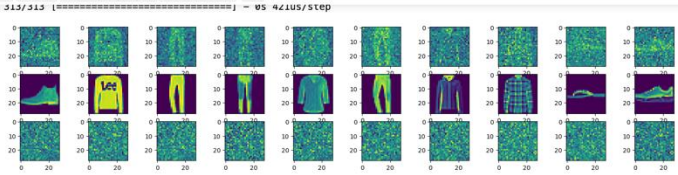
Baddam Sangeetha(700757 | Content | 15557658

localhost:8888/notebooks/Baddam%20Sangeetha(700757191).ipynb

jupyter Baddam Sangeetha(700757191) Last Checkpoint: 2 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

```
313/313 [==============================] - 0s 421us/step
```



```
Out[3]: <matplotlib.legend.Legend at 0x292dbe7d0>
```