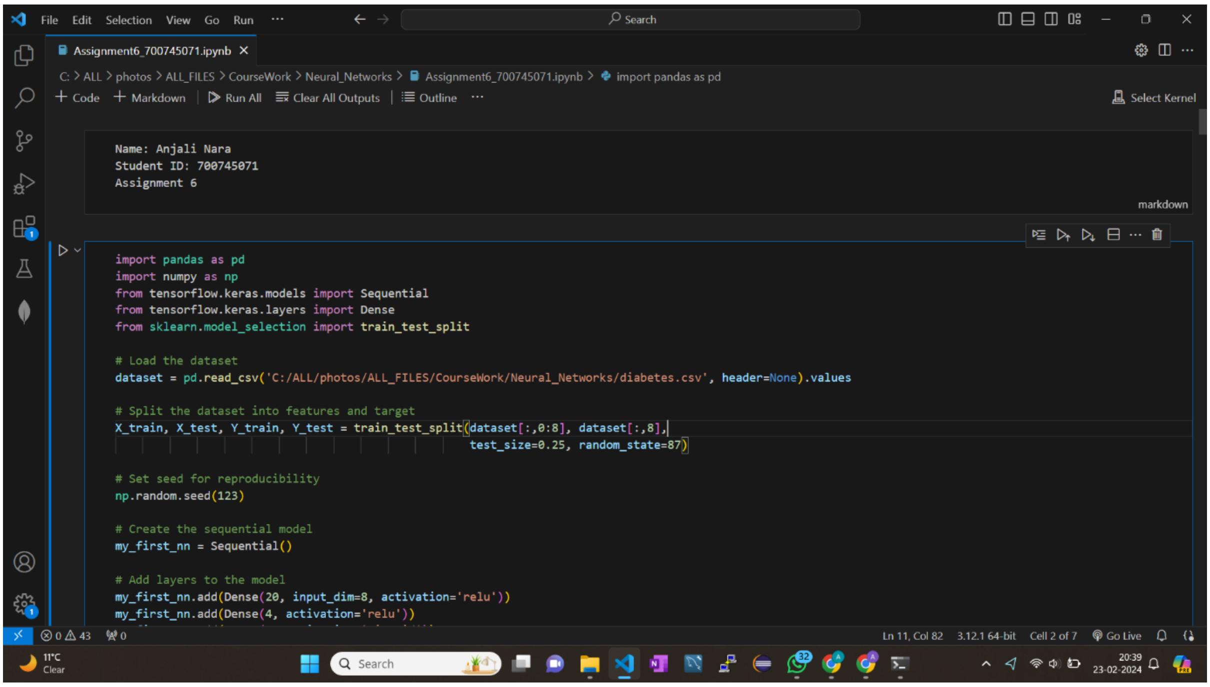
Assignment – 6

Name: Anjali Nara

Student Id: 700745071

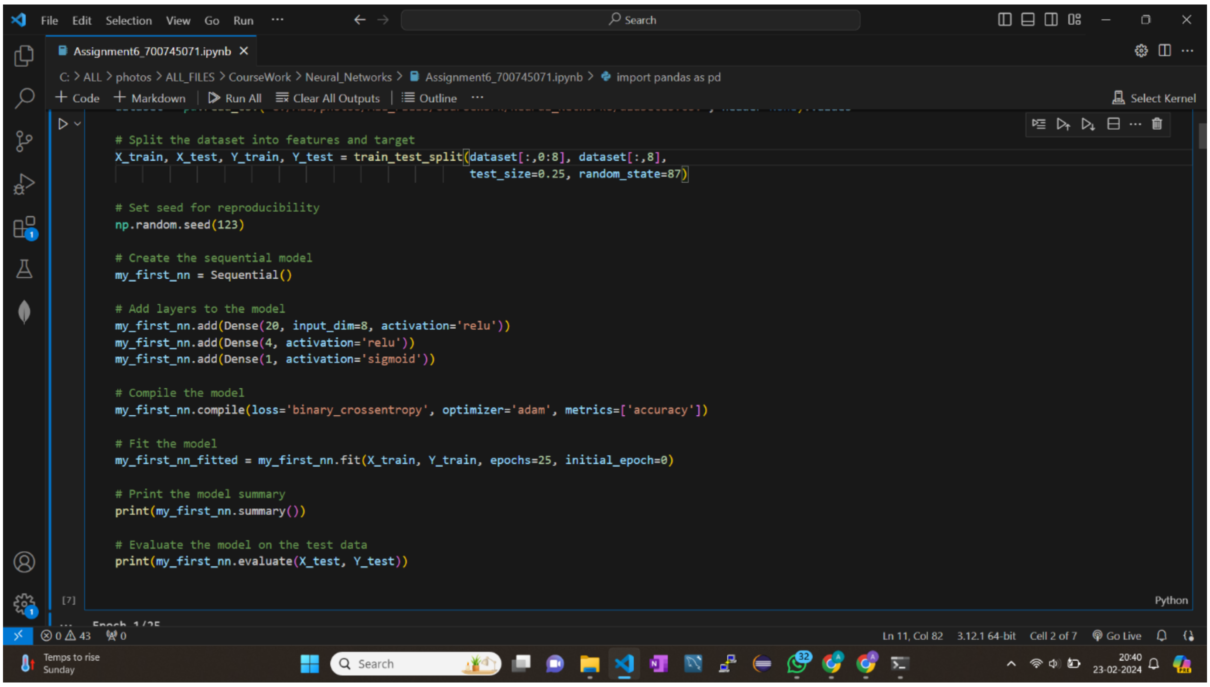
Github link: https://github.com/naraanjali/Assignment_6

Video link: https://drive.google.com/file/d/1UAclAOm95MFYAX-yWaGh35NiQzFQdXgQ/view?usp=sharing


File Edit Selection View Go Run ...
Search
Assignment6_700745071.ipynb
C: > ALL > photos > ALL_FILES > CourseWork > Neural_Networks > Assignment6_700745071.ipynb > import pandas as pd
+ Code + Markdown | Run All Clear All Outputs | Outline ...
Select Kernel
markdown
Ln 11, Col 82 3.12.1 64-bit Cell 2 of 7 Go Live

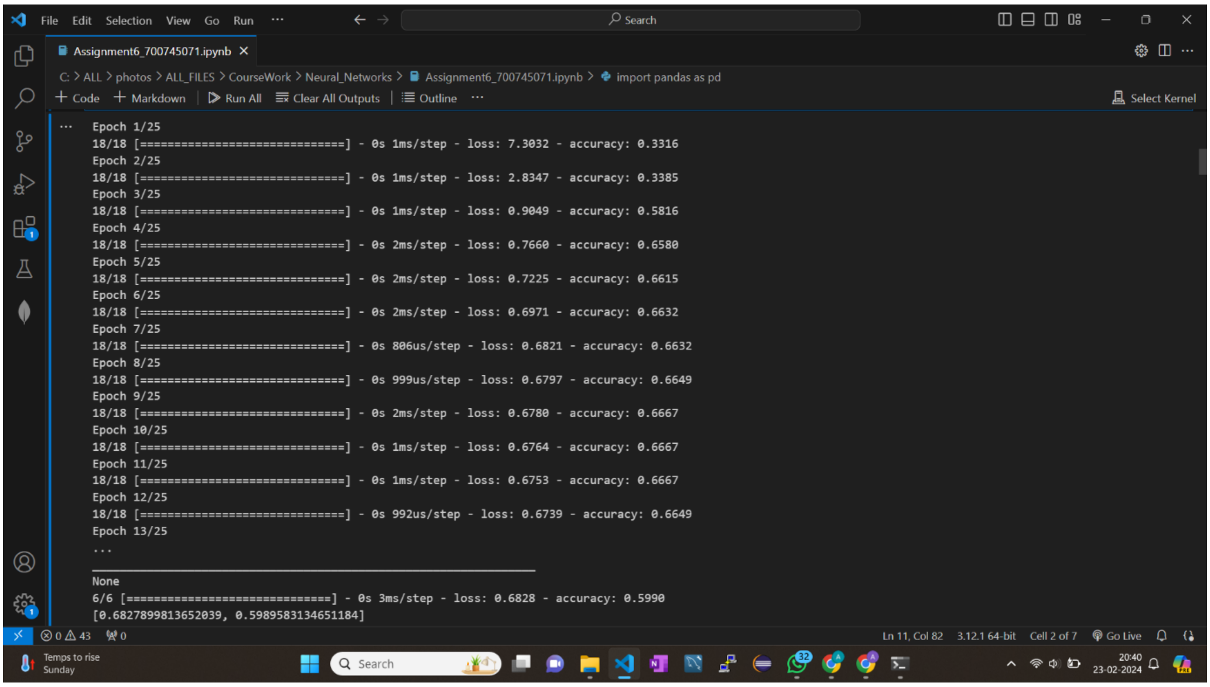
```
Name: Anjali Nara
Student ID: 700745071
Assignment 6
```

```python
import pandas as pd
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from sklearn.model_selection import train_test_split

# Load the dataset
dataset = pd.read_csv('C:/ALL/photos/ALL_FILES/CourseWork/Neural_Networks/diabetes.csv', header=None).values

# Split the dataset into features and target
X_train, X_test, Y_train, Y_test = train_test_split(dataset[:,0:8], dataset[:,8],
                                                    test_size=0.25, random_state=87)

# Set seed for reproducibility
np.random.seed(123)

# Create the sequential model
my_first_nn = Sequential()

# Add layers to the model
my_first_nn.add(Dense(20, input_dim=8, activation='relu'))
my_first_nn.add(Dense(4, activation='relu'))
```

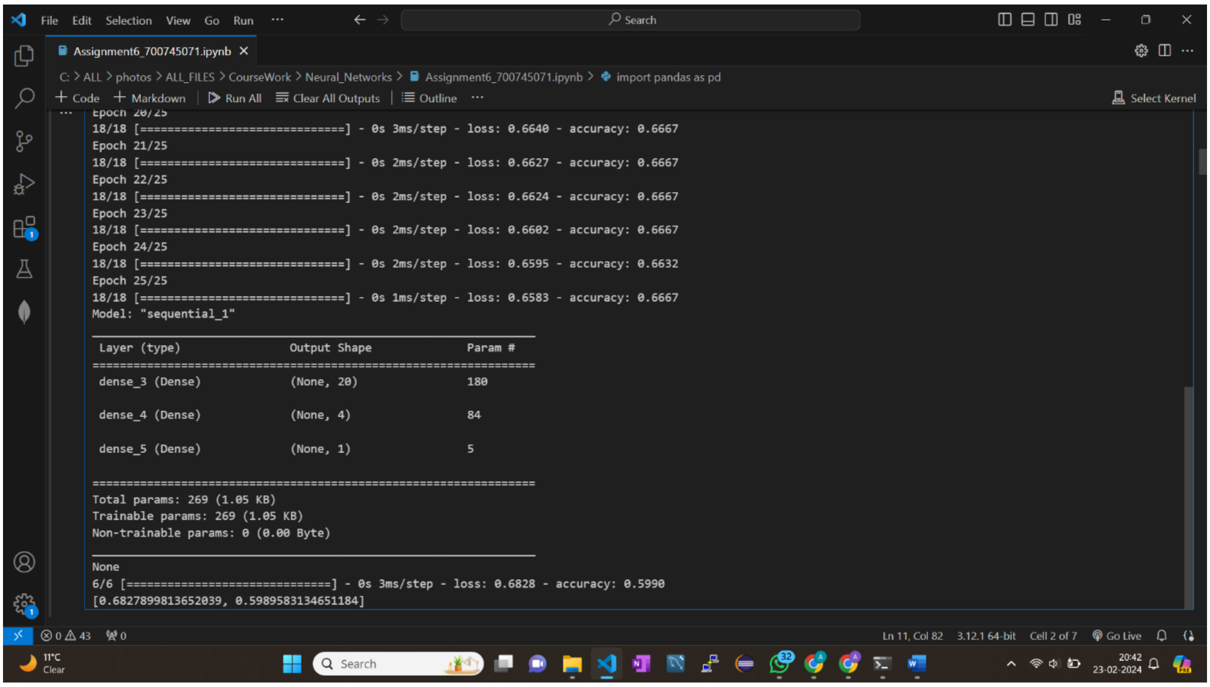

File Edit Selection View Go Run ...
Search
Assignment6_700745071.ipynb
C: > ALL > photos > ALL_FILES > CourseWork > Neural_Networks > Assignment6_700745071.ipynb > import pandas as pd
+ Code + Markdown | Run All Clear All Outputs | Outline ...
Select Kernel
[7]
Python
Ln 11, Col 82 3.12.1 64-bit Cell 2 of 7 Go Live

```python
# Split the dataset into features and target
X_train, X_test, Y_train, Y_test = train_test_split(dataset[:,0:8], dataset[:,8],
                                                    test_size=0.25, random_state=87)

# Set seed for reproducibility
np.random.seed(123)

# Create the sequential model
my_first_nn = Sequential()

# Add layers to the model
my_first_nn.add(Dense(20, input_dim=8, activation='relu'))
my_first_nn.add(Dense(4, activation='relu'))
my_first_nn.add(Dense(1, activation='sigmoid'))

# Compile the model
my_first_nn.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

# Fit the model
my_first_nn_fitted = my_first_nn.fit(X_train, Y_train, epochs=25, initial_epoch=0)

# Print the model summary
print(my_first_nn.summary())

# Evaluate the model on the test data
print(my_first_nn.evaluate(X_test, Y_test))
```

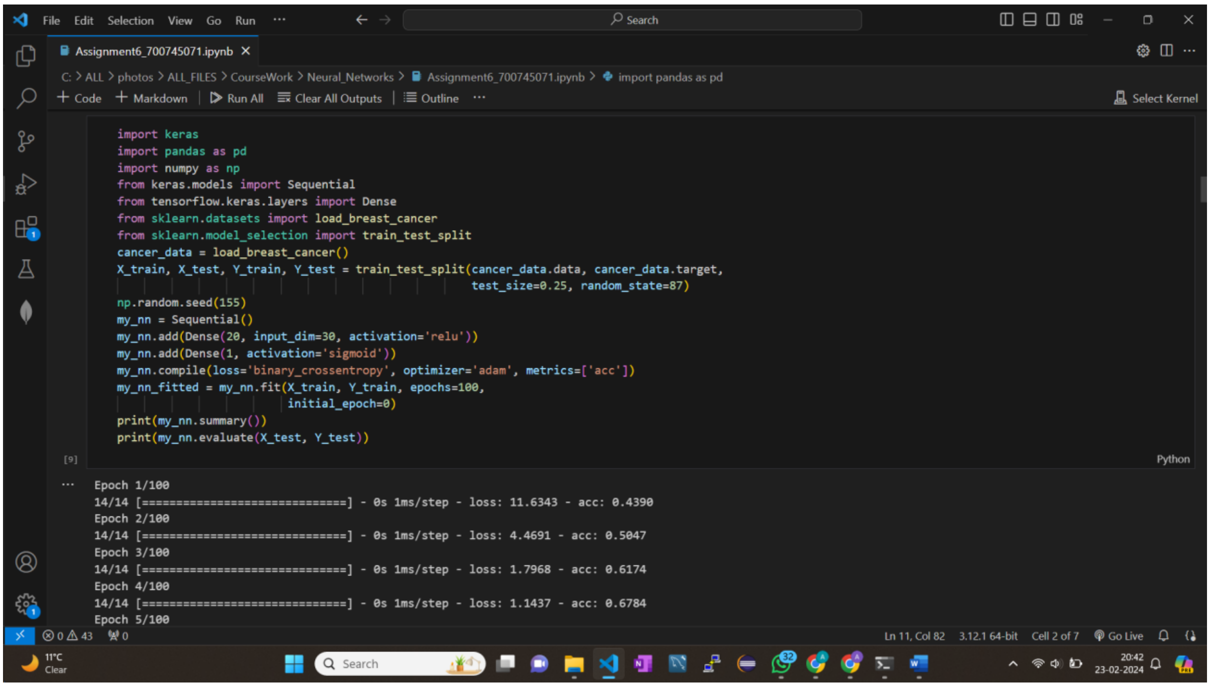
```
Epoch 1/25
```


File Edit Selection View Go Run ···
Search
Assignment6_700745071.ipynb
C: > ALL > photos > ALL_FILES > CourseWork > Neural_Networks > Assignment6_700745071.ipynb > import pandas as pd
+ Code + Markdown | Run All Clear All Outputs | Outline ···
Select Kernel
Ln 11, Col 82 3.12.1 64-bit Cell 2 of 7 Go Live
Temps to rise
Sunday
Search
20:40
23-02-2024

```
Epoch 1/25
18/18 [==============================] - 0s 1ms/step - loss: 7.3032 - accuracy: 0.3316
Epoch 2/25
18/18 [==============================] - 0s 1ms/step - loss: 2.8347 - accuracy: 0.3385
Epoch 3/25
18/18 [==============================] - 0s 1ms/step - loss: 0.9049 - accuracy: 0.5816
Epoch 4/25
18/18 [==============================] - 0s 2ms/step - loss: 0.7660 - accuracy: 0.6580
Epoch 5/25
18/18 [==============================] - 0s 2ms/step - loss: 0.7225 - accuracy: 0.6615
Epoch 6/25
18/18 [==============================] - 0s 2ms/step - loss: 0.6971 - accuracy: 0.6632
Epoch 7/25
18/18 [==============================] - 0s 806us/step - loss: 0.6821 - accuracy: 0.6632
Epoch 8/25
18/18 [==============================] - 0s 999us/step - loss: 0.6797 - accuracy: 0.6649
Epoch 9/25
18/18 [==============================] - 0s 2ms/step - loss: 0.6780 - accuracy: 0.6667
Epoch 10/25
18/18 [==============================] - 0s 1ms/step - loss: 0.6764 - accuracy: 0.6667
Epoch 11/25
18/18 [==============================] - 0s 1ms/step - loss: 0.6753 - accuracy: 0.6667
Epoch 12/25
18/18 [==============================] - 0s 992us/step - loss: 0.6739 - accuracy: 0.6649
Epoch 13/25
...
_________________________________________________________________
None
6/6 [==============================] - 0s 3ms/step - loss: 0.6828 - accuracy: 0.5990
[0.6827899813652039, 0.5989583134651184]
```

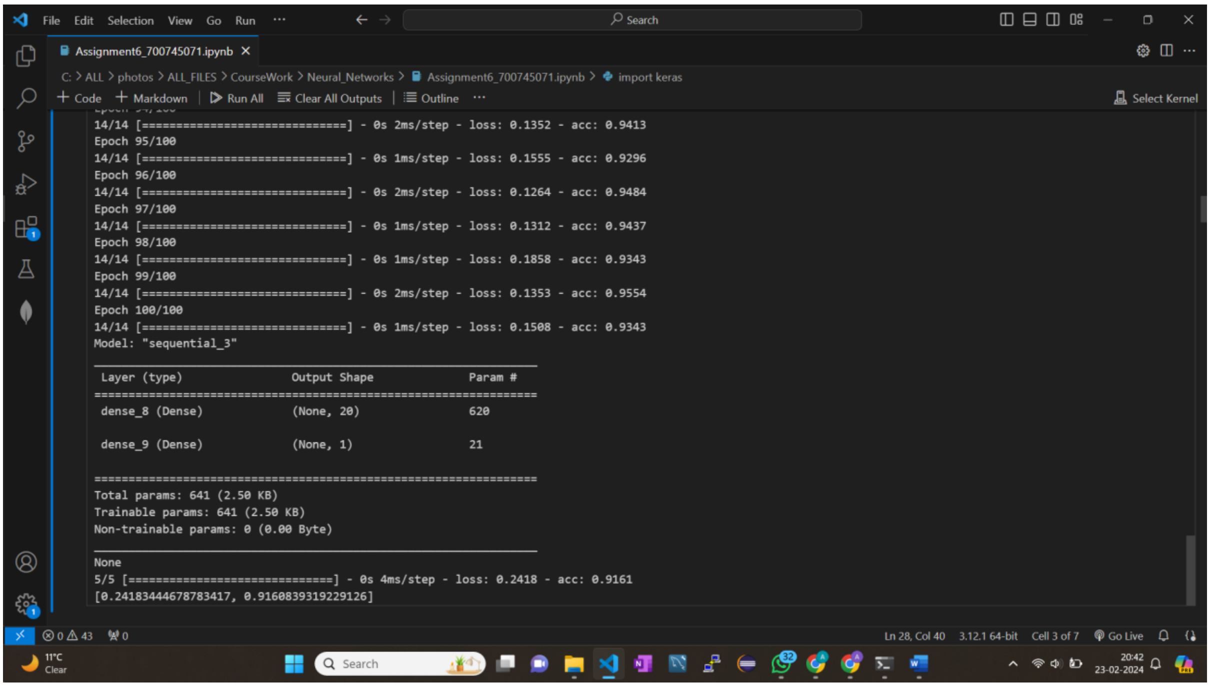

File Edit Selection View Go Run ···
Search
Assignment6_700745071.ipynb
C: > ALL > photos > ALL_FILES > CourseWork > Neural_Networks > Assignment6_700745071.ipynb > import pandas as pd
+ Code + Markdown | Run All Clear All Outputs | Outline ···
Select Kernel
Ln 11, Col 82 3.12.1 64-bit Cell 2 of 7 Go Live
11°C
Clear
Search
20:42
23-02-2024

```
Epoch 20/25
18/18 [==============================] - 0s 3ms/step - loss: 0.6640 - accuracy: 0.6667
Epoch 21/25
18/18 [==============================] - 0s 2ms/step - loss: 0.6627 - accuracy: 0.6667
Epoch 22/25
18/18 [==============================] - 0s 2ms/step - loss: 0.6624 - accuracy: 0.6667
Epoch 23/25
18/18 [==============================] - 0s 2ms/step - loss: 0.6602 - accuracy: 0.6667
Epoch 24/25
18/18 [==============================] - 0s 2ms/step - loss: 0.6595 - accuracy: 0.6632
Epoch 25/25
18/18 [==============================] - 0s 1ms/step - loss: 0.6583 - accuracy: 0.6667
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_3 (Dense)             (None, 20)                180

 dense_4 (Dense)             (None, 4)                 84

 dense_5 (Dense)             (None, 1)                 5

=================================================================
Total params: 269 (1.05 KB)
Trainable params: 269 (1.05 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
None
6/6 [==============================] - 0s 3ms/step - loss: 0.6828 - accuracy: 0.5990
[0.6827899813652039, 0.5989583134651184]
```

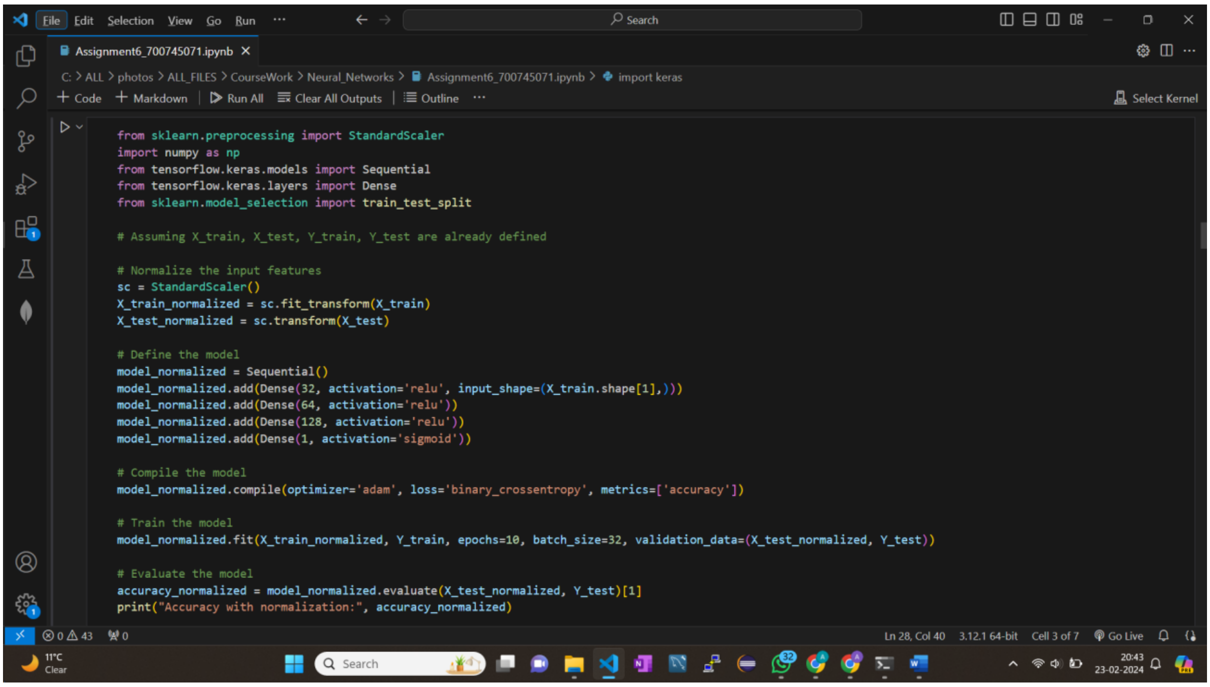

File Edit Selection View Go Run ···
Search
Assignment6_700745071.ipynb ×
C: > ALL > photos > ALL_FILES > CourseWork > Neural_Networks > Assignment6_700745071.ipynb > import pandas as pd
+ Code + Markdown | Run All Clear All Outputs | Outline ···
Select Kernel
[9]
Python
Ln 11, Col 82 3.12.1 64-bit Cell 2 of 7 Go Live
11°C Clear
Search
20:42 23-02-2024

```
import keras
import pandas as pd
import numpy as np
from keras.models import Sequential
from tensorflow.keras.layers import Dense
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
cancer_data = load_breast_cancer()
X_train, X_test, Y_train, Y_test = train_test_split(cancer_data.data, cancer_data.target,
                                                    test_size=0.25, random_state=87)
np.random.seed(155)
my_nn = Sequential()
my_nn.add(Dense(20, input_dim=30, activation='relu'))
my_nn.add(Dense(1, activation='sigmoid'))
my_nn.compile(loss='binary_crossentropy', optimizer='adam', metrics=['acc'])
my_nn_fitted = my_nn.fit(X_train, Y_train, epochs=100,
                         initial_epoch=0)
print(my_nn.summary())
print(my_nn.evaluate(X_test, Y_test))
```

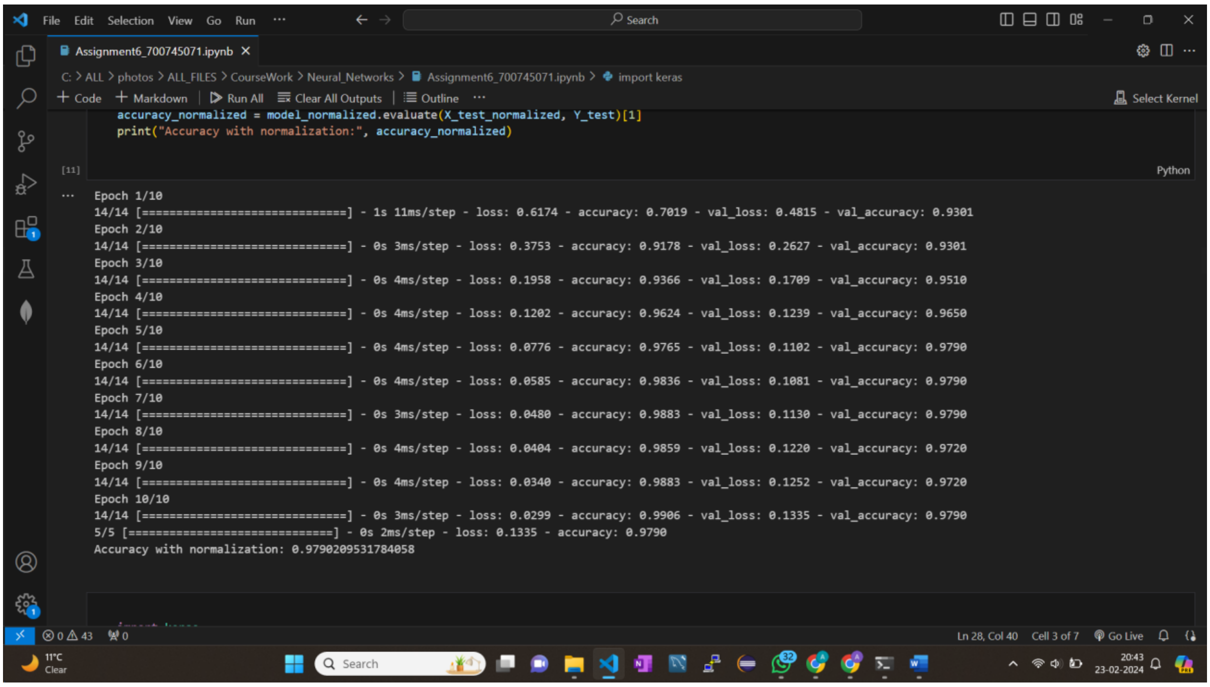
```
Epoch 1/100
14/14 [==============================] - 0s 1ms/step - loss: 11.6343 - acc: 0.4390
Epoch 2/100
14/14 [==============================] - 0s 1ms/step - loss: 4.4691 - acc: 0.5047
Epoch 3/100
14/14 [==============================] - 0s 1ms/step - loss: 1.7968 - acc: 0.6174
Epoch 4/100
14/14 [==============================] - 0s 1ms/step - loss: 1.1437 - acc: 0.6784
Epoch 5/100
```


File Edit Selection View Go Run ···
Search
Assignment6_700745071.ipynb ×
C: > ALL > photos > ALL_FILES > CourseWork > Neural_Networks > Assignment6_700745071.ipynb > import keras
+ Code + Markdown | Run All Clear All Outputs | Outline ···
Select Kernel
Ln 28, Col 40 3.12.1 64-bit Cell 3 of 7 Go Live
11°C Clear
Search
20:42 23-02-2024

```
14/14 [==============================] - 0s 2ms/step - loss: 0.1352 - acc: 0.9413
Epoch 95/100
14/14 [==============================] - 0s 1ms/step - loss: 0.1555 - acc: 0.9296
Epoch 96/100
14/14 [==============================] - 0s 2ms/step - loss: 0.1264 - acc: 0.9484
Epoch 97/100
14/14 [==============================] - 0s 1ms/step - loss: 0.1312 - acc: 0.9437
Epoch 98/100
14/14 [==============================] - 0s 1ms/step - loss: 0.1858 - acc: 0.9343
Epoch 99/100
14/14 [==============================] - 0s 2ms/step - loss: 0.1353 - acc: 0.9554
Epoch 100/100
14/14 [==============================] - 0s 1ms/step - loss: 0.1508 - acc: 0.9343
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_8 (Dense)             (None, 20)                620

 dense_9 (Dense)             (None, 1)                 21

=================================================================
Total params: 641 (2.50 KB)
Trainable params: 641 (2.50 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
None
5/5 [==============================] - 0s 4ms/step - loss: 0.2418 - acc: 0.9161
[0.24183444678783417, 0.9160839319229126]
```

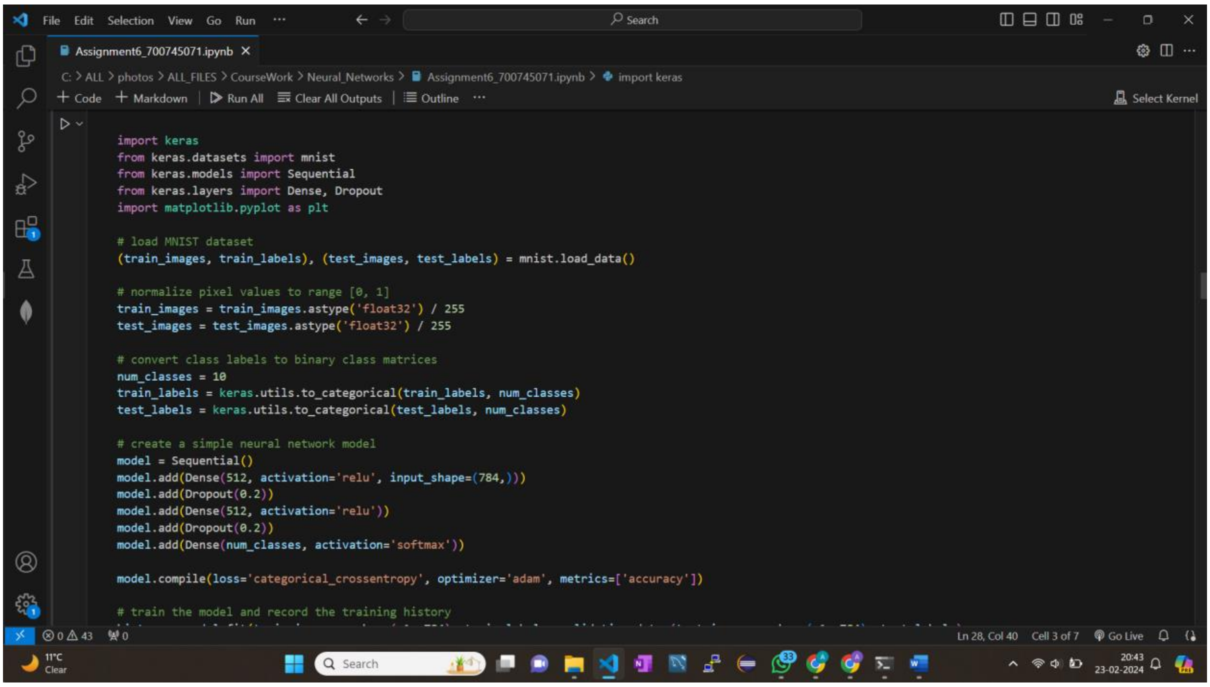

Assignment6_700745071.ipynb
C: > ALL > photos > ALL_FILES > CourseWork > Neural_Networks > Assignment6_700745071.ipynb > import keras
+ Code + Markdown | Run All Clear All Outputs | Outline
Select Kernel
Ln 28, Col 40 3.12.1 64-bit Cell 3 of 7 Go Live

```python
from sklearn.preprocessing import StandardScaler
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from sklearn.model_selection import train_test_split

# Assuming X_train, X_test, Y_train, Y_test are already defined

# Normalize the input features
sc = StandardScaler()
X_train_normalized = sc.fit_transform(X_train)
X_test_normalized = sc.transform(X_test)

# Define the model
model_normalized = Sequential()
model_normalized.add(Dense(32, activation='relu', input_shape=(X_train.shape[1],)))
model_normalized.add(Dense(64, activation='relu'))
model_normalized.add(Dense(128, activation='relu'))
model_normalized.add(Dense(1, activation='sigmoid'))

# Compile the model
model_normalized.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Train the model
model_normalized.fit(X_train_normalized, Y_train, epochs=10, batch_size=32, validation_data=(X_test_normalized, Y_test))

# Evaluate the model
accuracy_normalized = model_normalized.evaluate(X_test_normalized, Y_test)[1]
print("Accuracy with normalization:", accuracy_normalized)
```

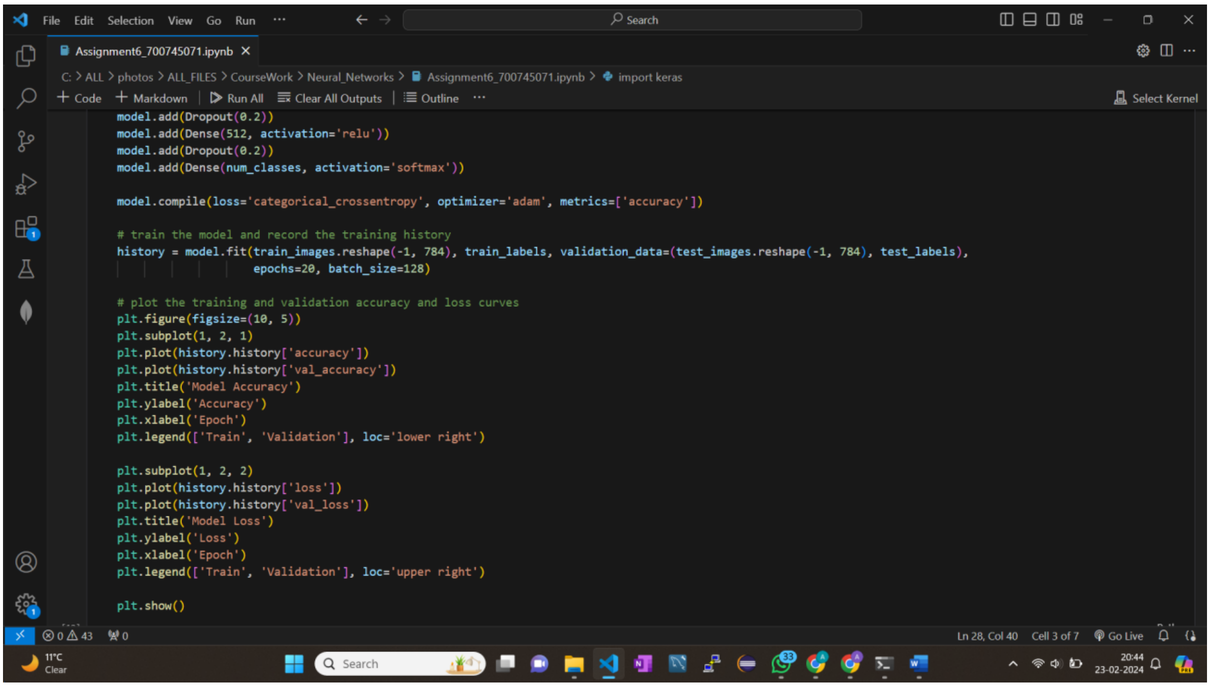

Assignment6_700745071.ipynb
C: > ALL > photos > ALL_FILES > CourseWork > Neural_Networks > Assignment6_700745071.ipynb > import keras
+ Code + Markdown | Run All Clear All Outputs | Outline
Select Kernel
[11]
Python
Ln 28, Col 40 Cell 3 of 7 Go Live

```python
accuracy_normalized = model_normalized.evaluate(X_test_normalized, Y_test)[1]
print("Accuracy with normalization:", accuracy_normalized)
```

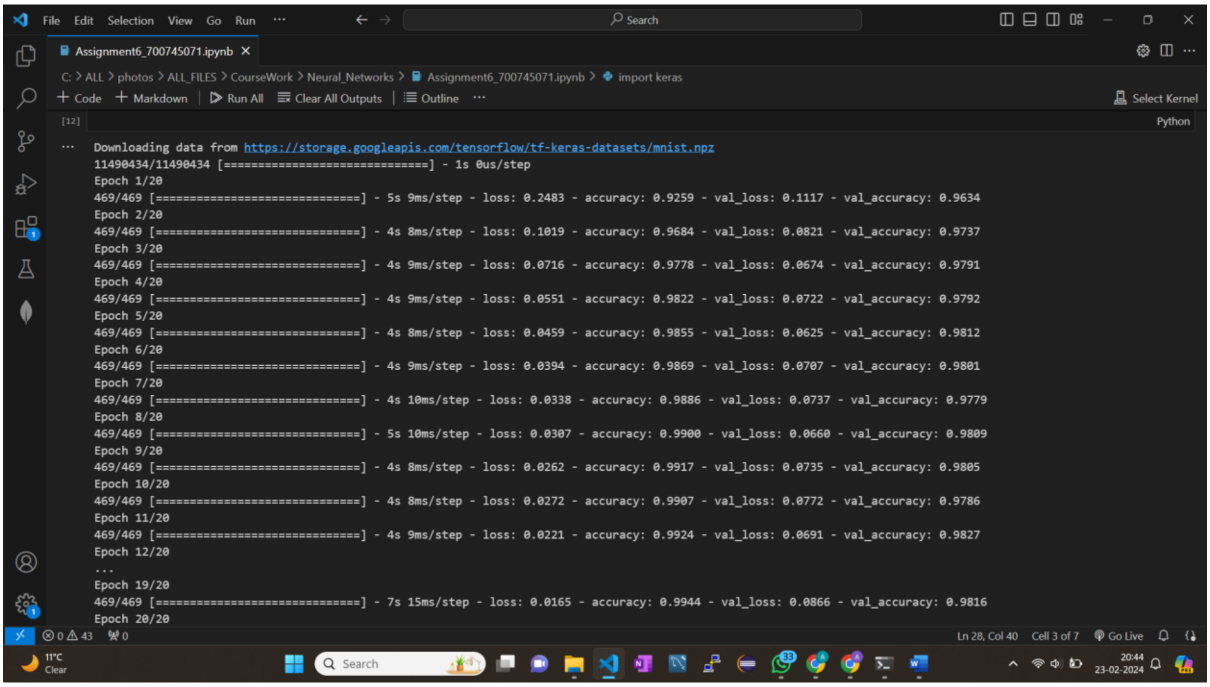
```
Epoch 1/10
14/14 [==============================] - 1s 11ms/step - loss: 0.6174 - accuracy: 0.7019 - val_loss: 0.4815 - val_accuracy: 0.9301
Epoch 2/10
14/14 [==============================] - 0s 3ms/step - loss: 0.3753 - accuracy: 0.9178 - val_loss: 0.2627 - val_accuracy: 0.9301
Epoch 3/10
14/14 [==============================] - 0s 4ms/step - loss: 0.1958 - accuracy: 0.9366 - val_loss: 0.1709 - val_accuracy: 0.9510
Epoch 4/10
14/14 [==============================] - 0s 4ms/step - loss: 0.1202 - accuracy: 0.9624 - val_loss: 0.1239 - val_accuracy: 0.9650
Epoch 5/10
14/14 [==============================] - 0s 4ms/step - loss: 0.0776 - accuracy: 0.9765 - val_loss: 0.1102 - val_accuracy: 0.9790
Epoch 6/10
14/14 [==============================] - 0s 4ms/step - loss: 0.0585 - accuracy: 0.9836 - val_loss: 0.1081 - val_accuracy: 0.9790
Epoch 7/10
14/14 [==============================] - 0s 3ms/step - loss: 0.0480 - accuracy: 0.9883 - val_loss: 0.1130 - val_accuracy: 0.9790
Epoch 8/10
14/14 [==============================] - 0s 4ms/step - loss: 0.0404 - accuracy: 0.9859 - val_loss: 0.1220 - val_accuracy: 0.9720
Epoch 9/10
14/14 [==============================] - 0s 4ms/step - loss: 0.0340 - accuracy: 0.9883 - val_loss: 0.1252 - val_accuracy: 0.9720
Epoch 10/10
14/14 [==============================] - 0s 3ms/step - loss: 0.0299 - accuracy: 0.9906 - val_loss: 0.1335 - val_accuracy: 0.9790
5/5 [==============================] - 0s 2ms/step - loss: 0.1335 - accuracy: 0.9790
Accuracy with normalization: 0.9790209531784058
```


File Edit Selection View Go Run ···
Search
Assignment6_700745071.ipynb
C: > ALL > photos > ALL_FILES > CourseWork > Neural_Networks > Assignment6_700745071.ipynb > import keras
+ Code + Markdown | Run All Clear All Outputs | Outline ···
Select Kernel
⊗ 0 ⚠ 43 0
Ln 28, Col 40 Cell 3 of 7 Go Live
11°C Clear
Search
20:43 23-02-2024

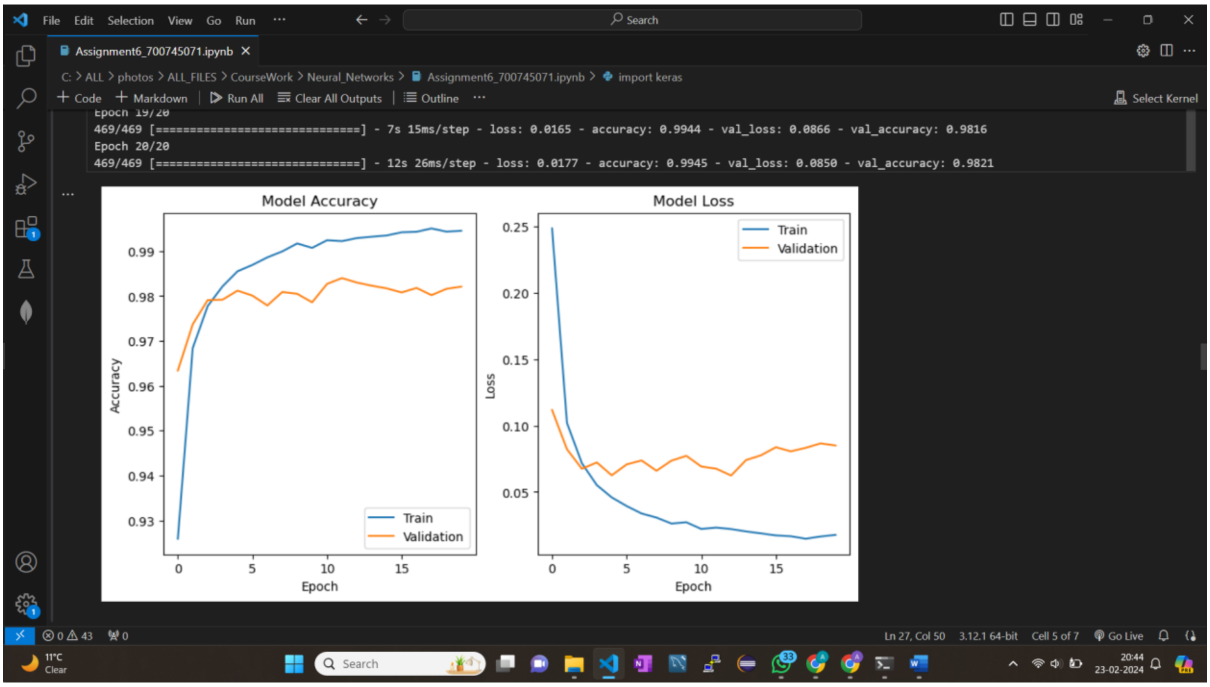
```python
import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout
import matplotlib.pyplot as plt

# load MNIST dataset
(train_images, train_labels), (test_images, test_labels) = mnist.load_data()

# normalize pixel values to range [0, 1]
train_images = train_images.astype('float32') / 255
test_images = test_images.astype('float32') / 255

# convert class labels to binary class matrices
num_classes = 10
train_labels = keras.utils.to_categorical(train_labels, num_classes)
test_labels = keras.utils.to_categorical(test_labels, num_classes)

# create a simple neural network model
model = Sequential()
model.add(Dense(512, activation='relu', input_shape=(784,)))
model.add(Dropout(0.2))
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(num_classes, activation='softmax'))

model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

# train the model and record the training history
```


File Edit Selection View Go Run ···
Search
Assignment6_700745071.ipynb
C: > ALL > photos > ALL_FILES > CourseWork > Neural_Networks > Assignment6_700745071.ipynb > import keras
+ Code + Markdown | Run All Clear All Outputs | Outline ···
Select Kernel
⊗ 0 ⚠ 43 0
Ln 28, Col 40 Cell 3 of 7 Go Live
11°C Clear
Search
20:44 23-02-2024

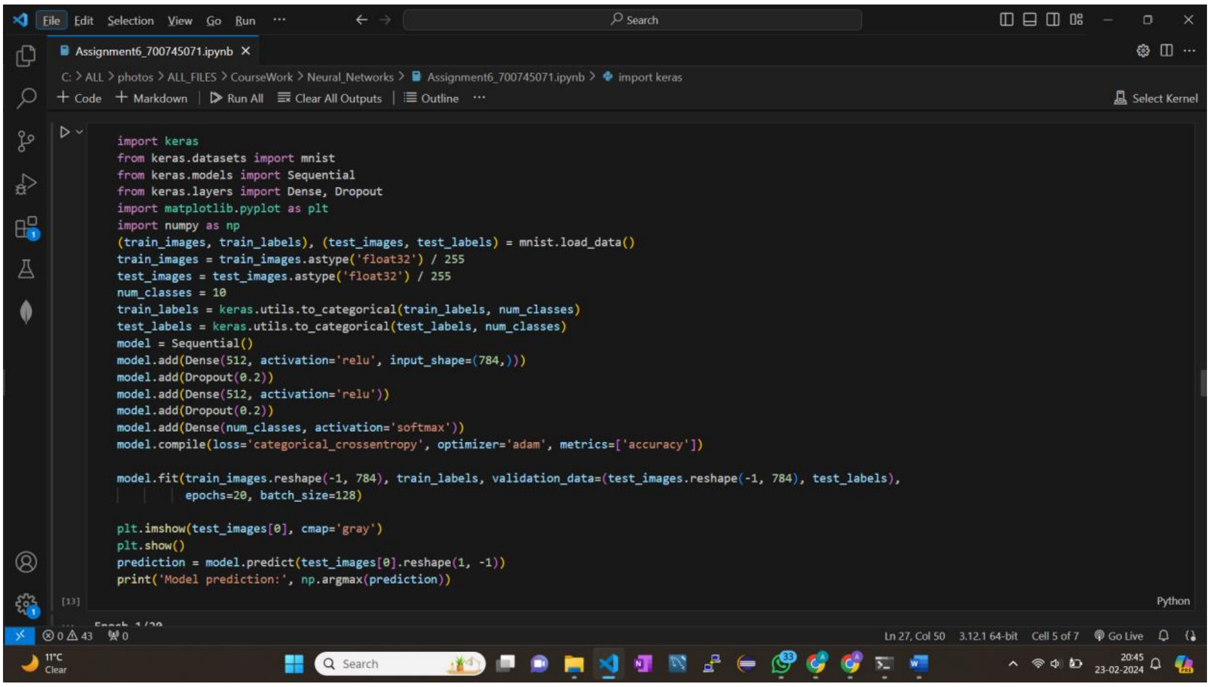
```python
model.add(Dropout(0.2))
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(num_classes, activation='softmax'))

model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

# train the model and record the training history
history = model.fit(train_images.reshape(-1, 784), train_labels, validation_data=(test_images.reshape(-1, 784), test_labels),
                    epochs=20, batch_size=128)

# plot the training and validation accuracy and loss curves
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model Accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='lower right')

plt.subplot(1, 2, 2)
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model Loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper right')

plt.show()
```



```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz
11490434/11490434 [==============================] - 1s 0us/step
Epoch 1/20
469/469 [==============================] - 5s 9ms/step - loss: 0.2483 - accuracy: 0.9259 - val_loss: 0.1117 - val_accuracy: 0.9634
Epoch 2/20
469/469 [==============================] - 4s 8ms/step - loss: 0.1019 - accuracy: 0.9684 - val_loss: 0.0821 - val_accuracy: 0.9737
Epoch 3/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0716 - accuracy: 0.9778 - val_loss: 0.0674 - val_accuracy: 0.9791
Epoch 4/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0551 - accuracy: 0.9822 - val_loss: 0.0722 - val_accuracy: 0.9792
Epoch 5/20
469/469 [==============================] - 4s 8ms/step - loss: 0.0459 - accuracy: 0.9855 - val_loss: 0.0625 - val_accuracy: 0.9812
Epoch 6/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0394 - accuracy: 0.9869 - val_loss: 0.0707 - val_accuracy: 0.9801
Epoch 7/20
469/469 [==============================] - 4s 10ms/step - loss: 0.0338 - accuracy: 0.9886 - val_loss: 0.0737 - val_accuracy: 0.9779
Epoch 8/20
469/469 [==============================] - 5s 10ms/step - loss: 0.0307 - accuracy: 0.9900 - val_loss: 0.0660 - val_accuracy: 0.9809
Epoch 9/20
469/469 [==============================] - 4s 8ms/step - loss: 0.0262 - accuracy: 0.9917 - val_loss: 0.0735 - val_accuracy: 0.9805
Epoch 10/20
469/469 [==============================] - 4s 8ms/step - loss: 0.0272 - accuracy: 0.9907 - val_loss: 0.0772 - val_accuracy: 0.9786
Epoch 11/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0221 - accuracy: 0.9924 - val_loss: 0.0691 - val_accuracy: 0.9827
Epoch 12/20
...
Epoch 19/20
469/469 [==============================] - 7s 15ms/step - loss: 0.0165 - accuracy: 0.9944 - val_loss: 0.0866 - val_accuracy: 0.9816
Epoch 20/20
```

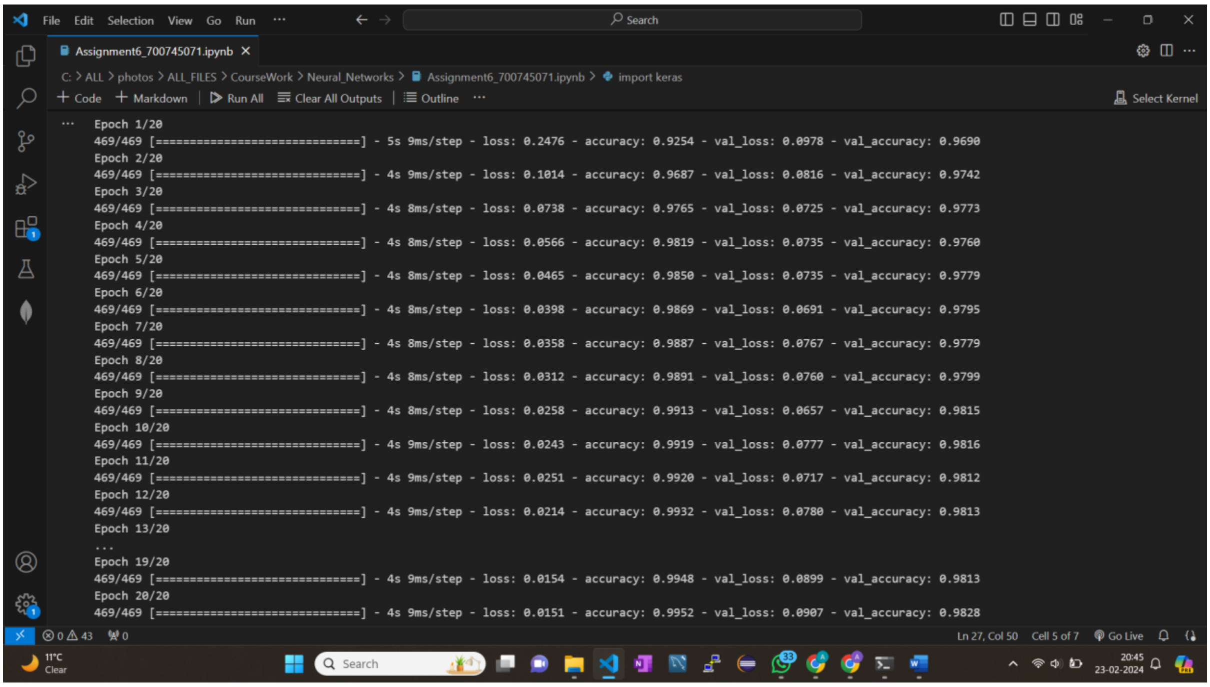


```
Epoch 19/20
469/469 [==============================] - 7s 15ms/step - loss: 0.0165 - accuracy: 0.9944 - val_loss: 0.0866 - val_accuracy: 0.9816
Epoch 20/20
469/469 [==============================] - 12s 26ms/step - loss: 0.0177 - accuracy: 0.9945 - val_loss: 0.0850 - val_accuracy: 0.9821
```



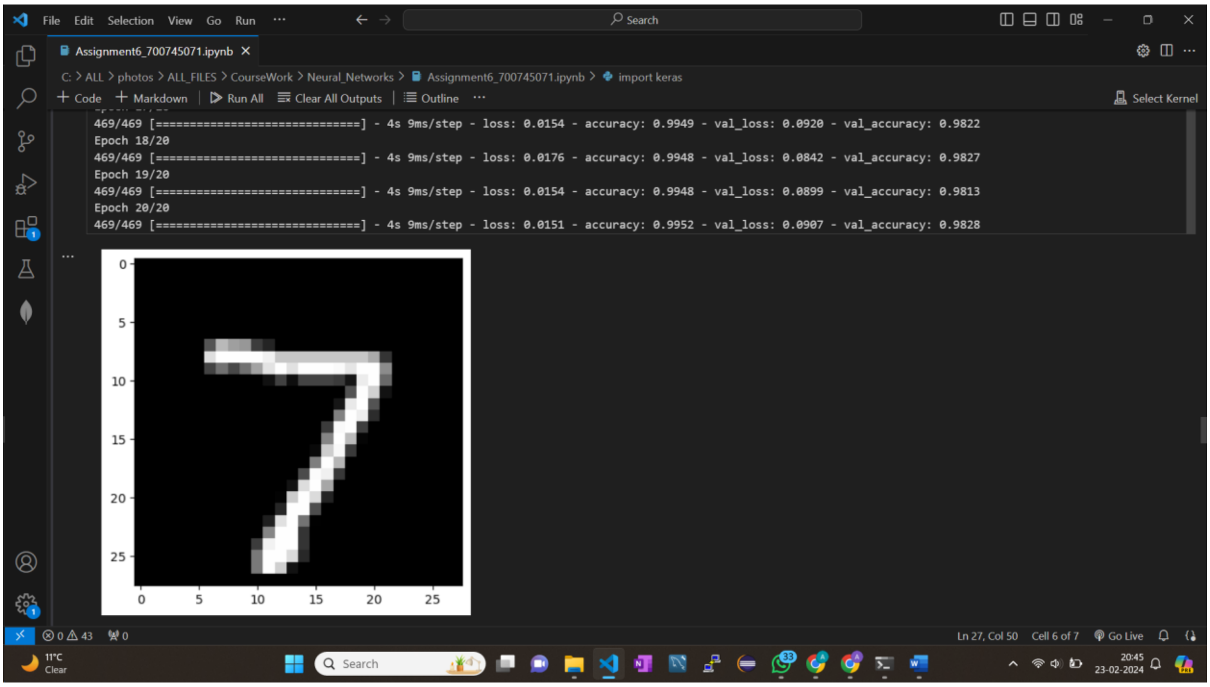
File Edit Selection View Go Run ···
Assignment6_700745071.ipynb
C: > ALL > photos > ALL_FILES > CourseWork > Neural_Networks > Assignment6_700745071.ipynb > import keras
+ Code + Markdown | Run All Clear All Outputs | Outline ···
Select Kernel
[13]
Python
Ln 27, Col 50 3.12.1 64-bit Cell 5 of 7 Go Live
11°C Clear
Search
20:45 23-02-2024

```python
import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout
import matplotlib.pyplot as plt
import numpy as np
(train_images, train_labels), (test_images, test_labels) = mnist.load_data()
train_images = train_images.astype('float32') / 255
test_images = test_images.astype('float32') / 255
num_classes = 10
train_labels = keras.utils.to_categorical(train_labels, num_classes)
test_labels = keras.utils.to_categorical(test_labels, num_classes)
model = Sequential()
model.add(Dense(512, activation='relu', input_shape=(784,)))
model.add(Dropout(0.2))
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(num_classes, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

model.fit(train_images.reshape(-1, 784), train_labels, validation_data=(test_images.reshape(-1, 784), test_labels),
          epochs=20, batch_size=128)

plt.imshow(test_images[0], cmap='gray')
plt.show()
prediction = model.predict(test_images[0].reshape(1, -1))
print('Model prediction:', np.argmax(prediction))
```

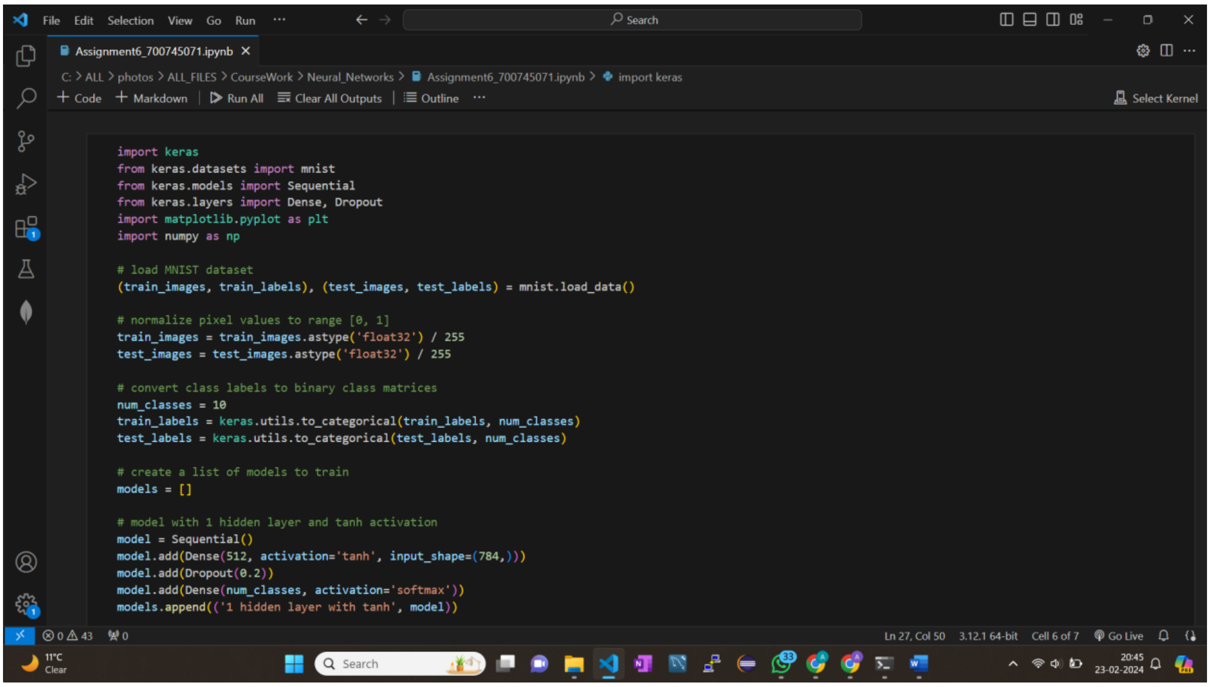


File Edit Selection View Go Run ···
Assignment6_700745071.ipynb
C: > ALL > photos > ALL_FILES > CourseWork > Neural_Networks > Assignment6_700745071.ipynb > import keras
+ Code + Markdown | Run All Clear All Outputs | Outline ···
Select Kernel
Ln 27, Col 50 Cell 5 of 7 Go Live
11°C Clear
Search
20:45 23-02-2024

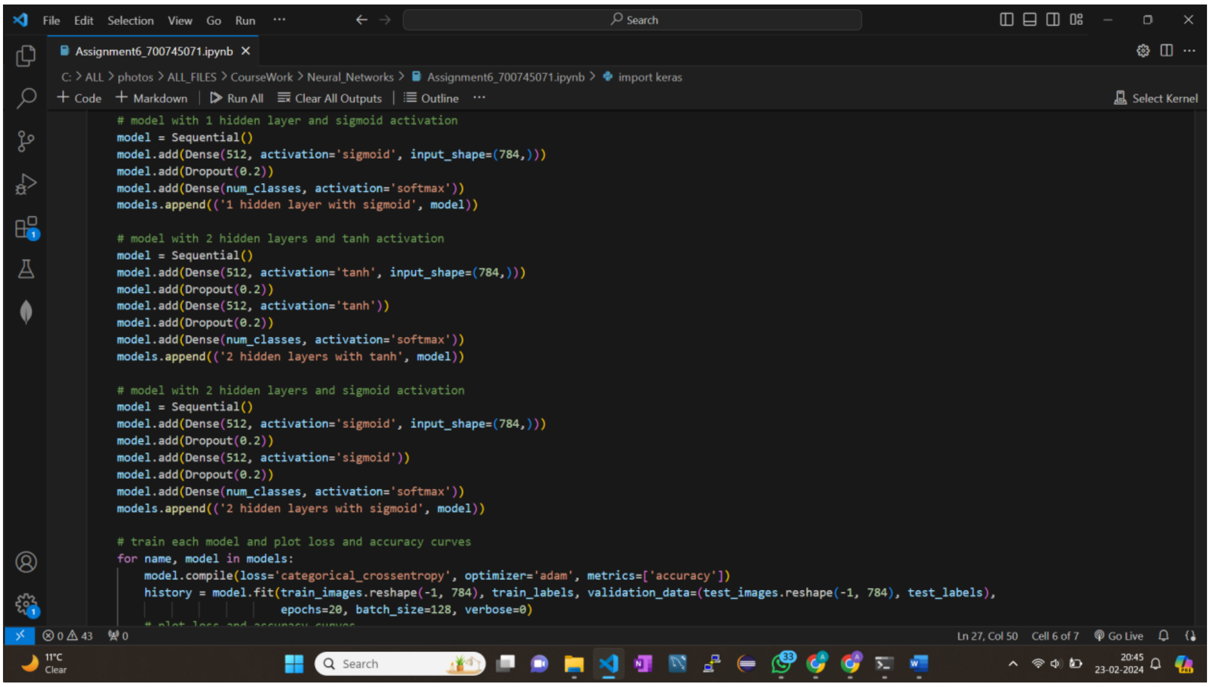
```
Epoch 1/20
469/469 [==============================] - 5s 9ms/step - loss: 0.2476 - accuracy: 0.9254 - val_loss: 0.0978 - val_accuracy: 0.9690
Epoch 2/20
469/469 [==============================] - 4s 9ms/step - loss: 0.1014 - accuracy: 0.9687 - val_loss: 0.0816 - val_accuracy: 0.9742
Epoch 3/20
469/469 [==============================] - 4s 8ms/step - loss: 0.0738 - accuracy: 0.9765 - val_loss: 0.0725 - val_accuracy: 0.9773
Epoch 4/20
469/469 [==============================] - 4s 8ms/step - loss: 0.0566 - accuracy: 0.9819 - val_loss: 0.0735 - val_accuracy: 0.9760
Epoch 5/20
469/469 [==============================] - 4s 8ms/step - loss: 0.0465 - accuracy: 0.9850 - val_loss: 0.0735 - val_accuracy: 0.9779
Epoch 6/20
469/469 [==============================] - 4s 8ms/step - loss: 0.0398 - accuracy: 0.9869 - val_loss: 0.0691 - val_accuracy: 0.9795
Epoch 7/20
469/469 [==============================] - 4s 8ms/step - loss: 0.0358 - accuracy: 0.9887 - val_loss: 0.0767 - val_accuracy: 0.9779
Epoch 8/20
469/469 [==============================] - 4s 8ms/step - loss: 0.0312 - accuracy: 0.9891 - val_loss: 0.0760 - val_accuracy: 0.9799
Epoch 9/20
469/469 [==============================] - 4s 8ms/step - loss: 0.0258 - accuracy: 0.9913 - val_loss: 0.0657 - val_accuracy: 0.9815
Epoch 10/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0243 - accuracy: 0.9919 - val_loss: 0.0777 - val_accuracy: 0.9816
Epoch 11/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0251 - accuracy: 0.9920 - val_loss: 0.0717 - val_accuracy: 0.9812
Epoch 12/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0214 - accuracy: 0.9932 - val_loss: 0.0780 - val_accuracy: 0.9813
Epoch 13/20
...
Epoch 19/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0154 - accuracy: 0.9948 - val_loss: 0.0899 - val_accuracy: 0.9813
Epoch 20/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0151 - accuracy: 0.9952 - val_loss: 0.0907 - val_accuracy: 0.9828
```


File Edit Selection View Go Run
Search
Assignment6_700745071.ipynb
C: > ALL > photos > ALL_FILES > CourseWork > Neural_Networks > Assignment6_700745071.ipynb > import keras
+ Code + Markdown | Run All Clear All Outputs | Outline
Select Kernel
0
5
10
15
20
25
0
5
10
15
20
25
Ln 27, Col 50 Cell 6 of 7 Go Live
11°C
Clear
20:45
23-02-2024

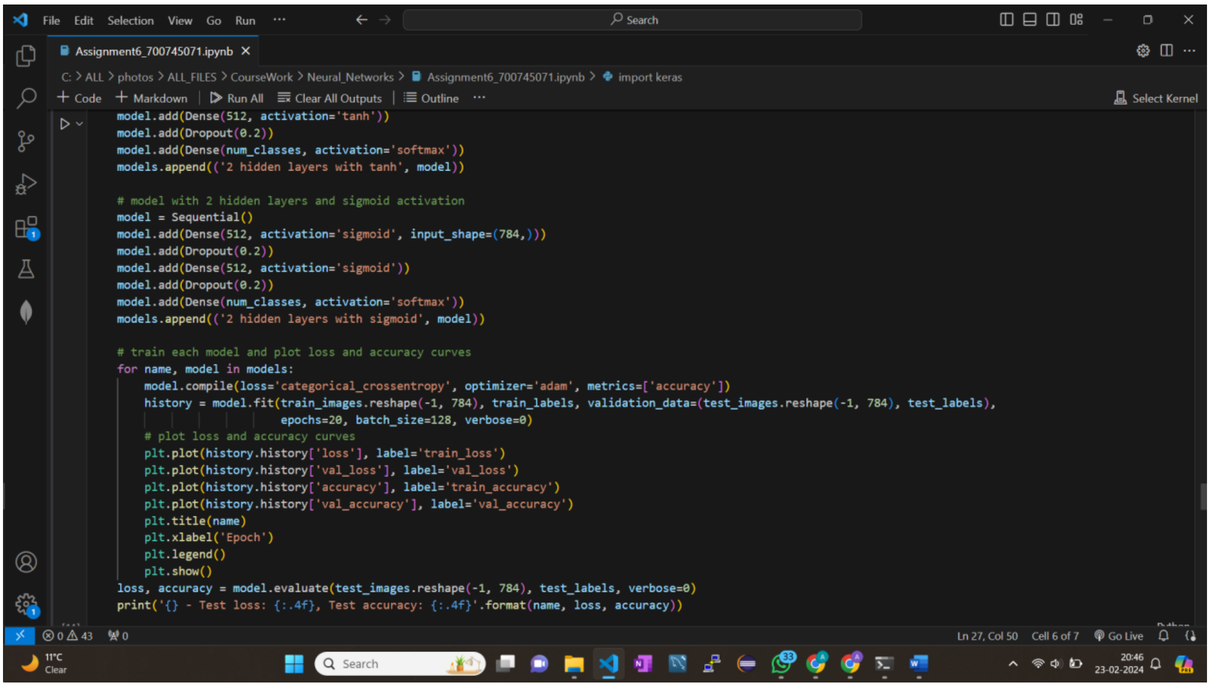
```
469/469 [==============================] - 4s 9ms/step - loss: 0.0154 - accuracy: 0.9949 - val_loss: 0.0920 - val_accuracy: 0.9822
Epoch 18/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0176 - accuracy: 0.9948 - val_loss: 0.0842 - val_accuracy: 0.9827
Epoch 19/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0154 - accuracy: 0.9948 - val_loss: 0.0899 - val_accuracy: 0.9813
Epoch 20/20
469/469 [==============================] - 4s 9ms/step - loss: 0.0151 - accuracy: 0.9952 - val_loss: 0.0907 - val_accuracy: 0.9828
```


File Edit Selection View Go Run
Search
Assignment6_700745071.ipynb
C: > ALL > photos > ALL_FILES > CourseWork > Neural_Networks > Assignment6_700745071.ipynb > import keras
+ Code + Markdown | Run All Clear All Outputs | Outline
Select Kernel
Ln 27, Col 50 3.12.1 64-bit Cell 6 of 7 Go Live
11°C
Clear
20:45
23-02-2024

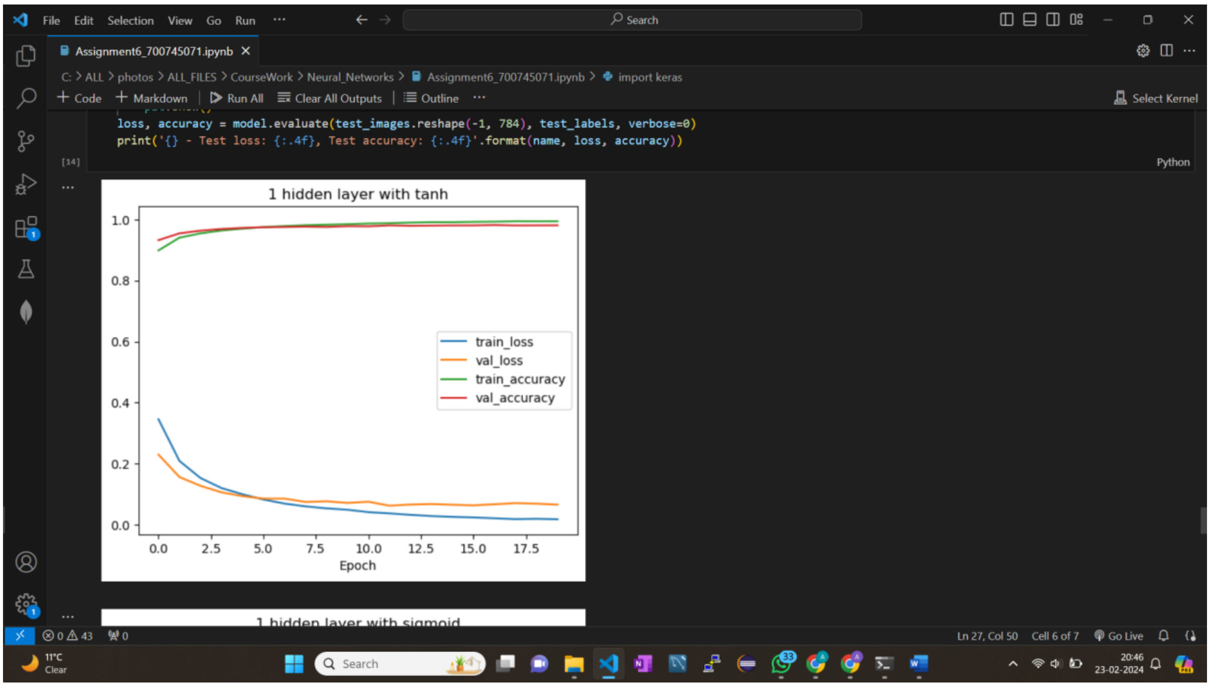
```python
import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout
import matplotlib.pyplot as plt
import numpy as np

# load MNIST dataset
(train_images, train_labels), (test_images, test_labels) = mnist.load_data()

# normalize pixel values to range [0, 1]
train_images = train_images.astype('float32') / 255
test_images = test_images.astype('float32') / 255

# convert class labels to binary class matrices
num_classes = 10
train_labels = keras.utils.to_categorical(train_labels, num_classes)
test_labels = keras.utils.to_categorical(test_labels, num_classes)

# create a list of models to train
models = []

# model with 1 hidden layer and tanh activation
model = Sequential()
model.add(Dense(512, activation='tanh', input_shape=(784,)))
model.add(Dropout(0.2))
model.add(Dense(num_classes, activation='softmax'))
models.append(('1 hidden layer with tanh', model))
```


File Edit Selection View Go Run ···
Search
Assignment6_700745071.ipynb
C: > ALL > photos > ALL_FILES > CourseWork > Neural_Networks > Assignment6_700745071.ipynb > import keras
+ Code + Markdown ▷ Run All Clear All Outputs Outline ···
Select Kernel
Ln 27, Col 50 Cell 6 of 7 Go Live
11°C Clear
Search
20:45 23-02-2024

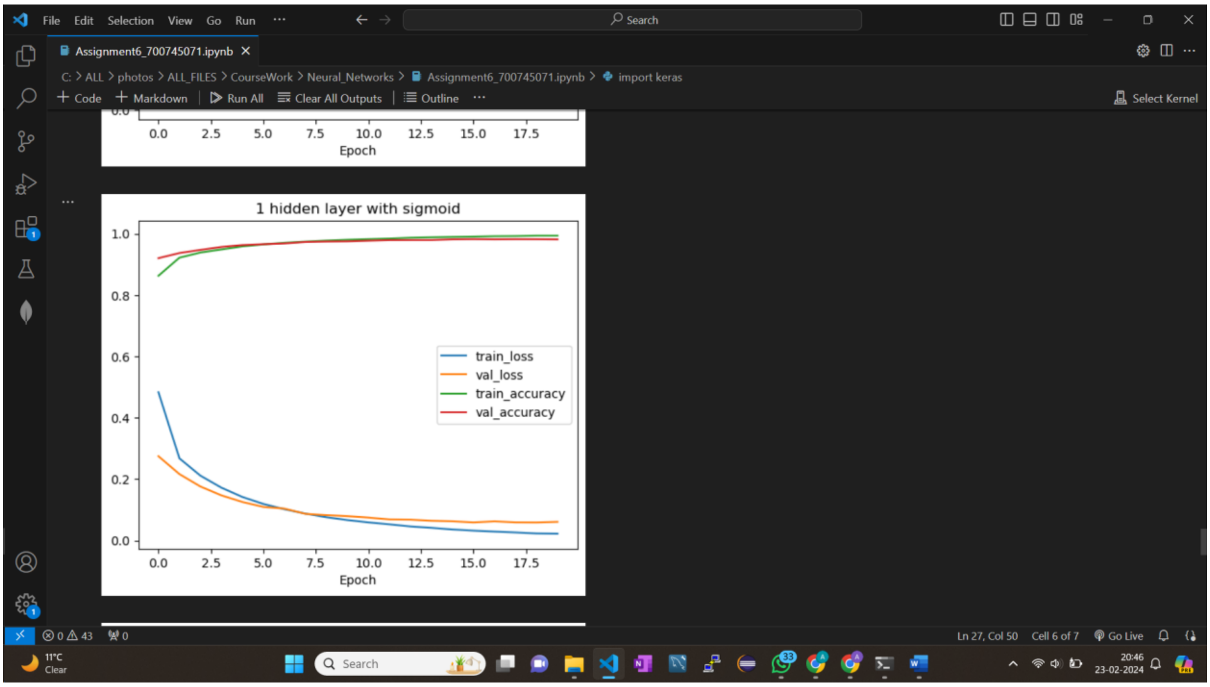
```
# model with 1 hidden layer and sigmoid activation
model = Sequential()
model.add(Dense(512, activation='sigmoid', input_shape=(784,)))
model.add(Dropout(0.2))
model.add(Dense(num_classes, activation='softmax'))
models.append(('1 hidden layer with sigmoid', model))

# model with 2 hidden layers and tanh activation
model = Sequential()
model.add(Dense(512, activation='tanh', input_shape=(784,)))
model.add(Dropout(0.2))
model.add(Dense(512, activation='tanh'))
model.add(Dropout(0.2))
model.add(Dense(num_classes, activation='softmax'))
models.append(('2 hidden layers with tanh', model))

# model with 2 hidden layers and sigmoid activation
model = Sequential()
model.add(Dense(512, activation='sigmoid', input_shape=(784,)))
model.add(Dropout(0.2))
model.add(Dense(512, activation='sigmoid'))
model.add(Dropout(0.2))
model.add(Dense(num_classes, activation='softmax'))
models.append(('2 hidden layers with sigmoid', model))

# train each model and plot loss and accuracy curves
for name, model in models:
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    history = model.fit(train_images.reshape(-1, 784), train_labels, validation_data=(test_images.reshape(-1, 784), test_labels),
                        epochs=20, batch_size=128, verbose=0)
    # plot loss and accuracy curves
```

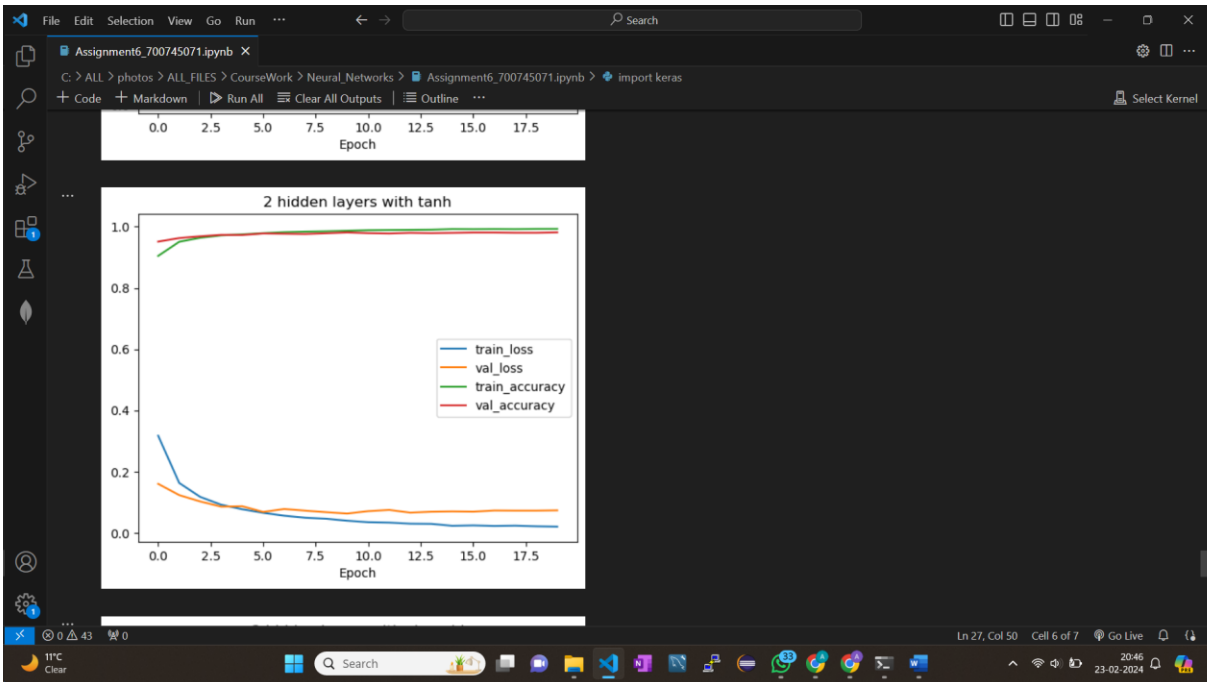

File Edit Selection View Go Run ···
Search
Assignment6_700745071.ipynb
C: > ALL > photos > ALL_FILES > CourseWork > Neural_Networks > Assignment6_700745071.ipynb > import keras
+ Code + Markdown ▷ Run All Clear All Outputs Outline ···
Select Kernel
Ln 27, Col 50 Cell 6 of 7 Go Live
11°C Clear
Search
20:46 23-02-2024

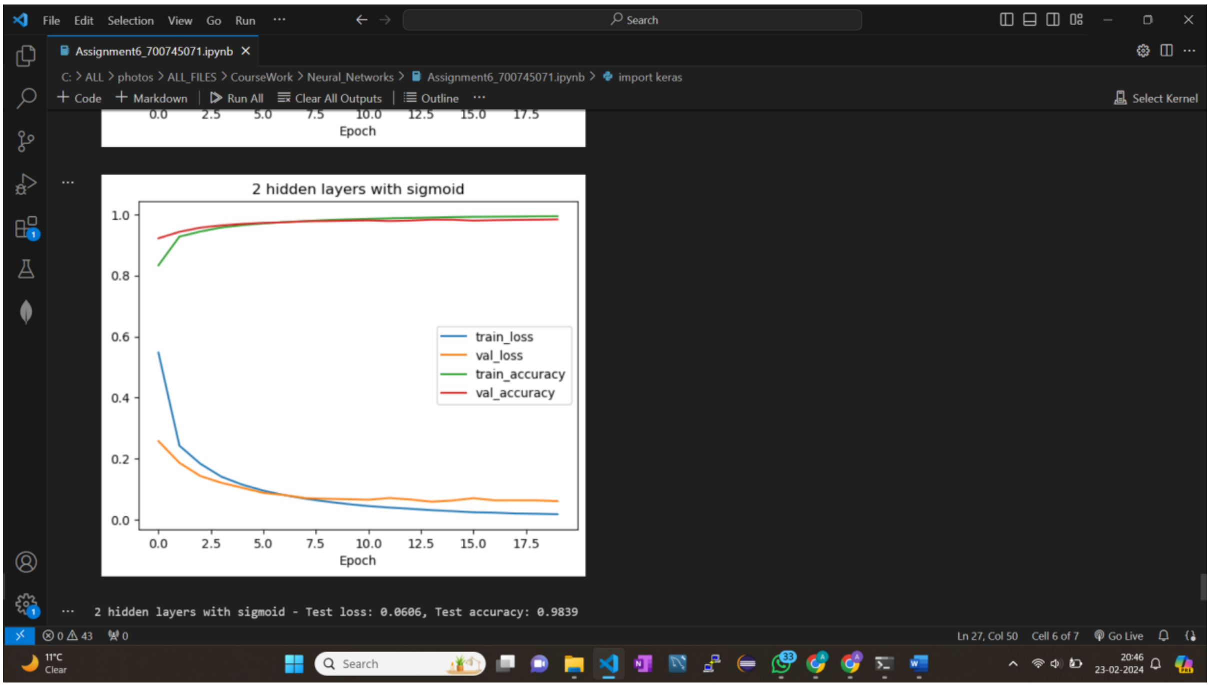
```
model.add(Dense(512, activation='tanh'))
model.add(Dropout(0.2))
model.add(Dense(num_classes, activation='softmax'))
models.append(('2 hidden layers with tanh', model))

# model with 2 hidden layers and sigmoid activation
model = Sequential()
model.add(Dense(512, activation='sigmoid', input_shape=(784,)))
model.add(Dropout(0.2))
model.add(Dense(512, activation='sigmoid'))
model.add(Dropout(0.2))
model.add(Dense(num_classes, activation='softmax'))
models.append(('2 hidden layers with sigmoid', model))

# train each model and plot loss and accuracy curves
for name, model in models:
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    history = model.fit(train_images.reshape(-1, 784), train_labels, validation_data=(test_images.reshape(-1, 784), test_labels),
                        epochs=20, batch_size=128, verbose=0)
    # plot loss and accuracy curves
    plt.plot(history.history['loss'], label='train_loss')
    plt.plot(history.history['val_loss'], label='val_loss')
    plt.plot(history.history['accuracy'], label='train_accuracy')
    plt.plot(history.history['val_accuracy'], label='val_accuracy')
    plt.title(name)
    plt.xlabel('Epoch')
    plt.legend()
    plt.show()
loss, accuracy = model.evaluate(test_images.reshape(-1, 784), test_labels, verbose=0)
print('{} - Test loss: {:.4f}, Test accuracy: {:.4f}'.format(name, loss, accuracy))
```


File Edit Selection View Go Run ···
Search
Assignment6_700745071.ipynb
C: > ALL > photos > ALL_FILES > CourseWork > Neural_Networks > Assignment6_700745071.ipynb > import keras
+ Code + Markdown | Run All Clear All Outputs | Outline ···
Select Kernel
[14]
Python
1 hidden layer with tanh
1.0
0.8
0.6
0.4
0.2
0.0
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5
Epoch
train_loss
val_loss
train_accuracy
val_accuracy
1 hidden layer with sigmoid
Ln 27, Col 50 Cell 6 of 7 Go Live
⊗ 0 ⚠ 43 0
11°C Clear
Search
20:46 23-02-2024

```
loss, accuracy = model.evaluate(test_images.reshape(-1, 784), test_labels, verbose=0)
print('{} - Test loss: {:.4f}, Test accuracy: {:.4f}'.format(name, loss, accuracy))
```


File Edit Selection View Go Run ···
Search
Assignment6_700745071.ipynb
C: > ALL > photos > ALL_FILES > CourseWork > Neural_Networks > Assignment6_700745071.ipynb > import keras
+ Code + Markdown | Run All Clear All Outputs | Outline ···
Select Kernel
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5
Epoch
1 hidden layer with sigmoid
1.0
0.8
0.6
0.4
0.2
0.0
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5
Epoch
train_loss
val_loss
train_accuracy
val_accuracy
Ln 27, Col 50 Cell 6 of 7 Go Live
⊗ 0 ⚠ 43 0
11°C Clear
Search
20:46 23-02-2024


File Edit Selection View Go Run ···
Search
Assignment6_700745071.ipynb
C: > ALL > photos > ALL_FILES > CourseWork > Neural_Networks > Assignment6_700745071.ipynb > import keras
+ Code + Markdown Run All Clear All Outputs Outline ···
Select Kernel
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5
Epoch
2 hidden layers with tanh
1.0
0.8
0.6
0.4
0.2
0.0
train_loss
val_loss
train_accuracy
val_accuracy
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5
Epoch
0 43 0
Ln 27, Col 50 Cell 6 of 7 Go Live
11°C Clear
Search
20:46 23-02-2024


File Edit Selection View Go Run ···
Search
Assignment6_700745071.ipynb
C: > ALL > photos > ALL_FILES > CourseWork > Neural_Networks > Assignment6_700745071.ipynb > import keras
+ Code + Markdown Run All Clear All Outputs Outline ···
Select Kernel
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5
Epoch
2 hidden layers with sigmoid
1.0
0.8
0.6
0.4
0.2
0.0
train_loss
val_loss
train_accuracy
val_accuracy
0.0 2.5 5.0 7.5 10.0 12.5 15.0 17.5
Epoch
0 43 0
Ln 27, Col 50 Cell 6 of 7 Go Live
11°C Clear
Search
20:46 23-02-2024

```
2 hidden layers with sigmoid - Test loss: 0.0606, Test accuracy: 0.9839
```