NNDL_ASSIGNMENT-7

NAME:JYOSHNA YARRAGUNTLA

STUDENT ID:700758848

Image Classification with CNN:

1. Training the model

2. Evaluating the model

Programming elements:

1. About CNN

2. Hyperparameters of CNN

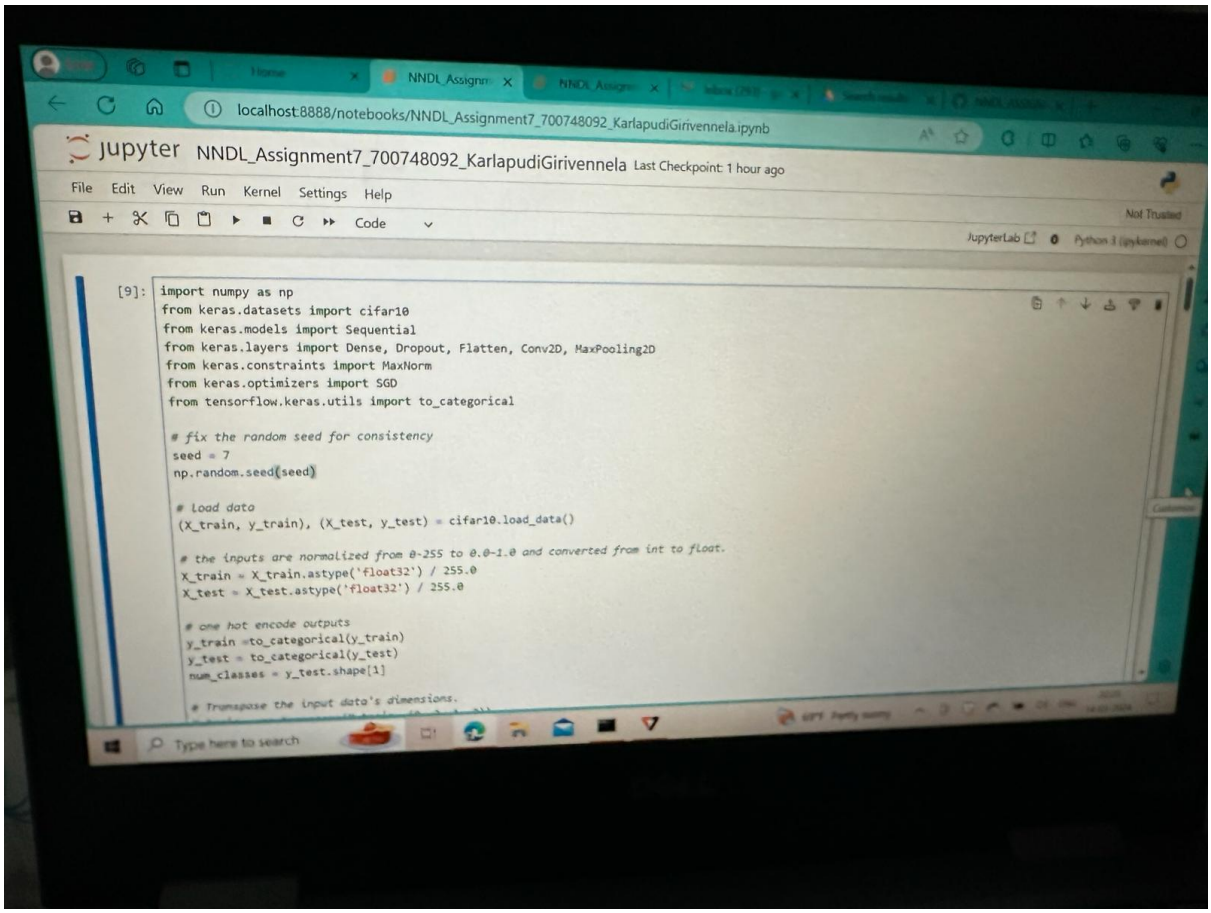
3. Image classification with CNN

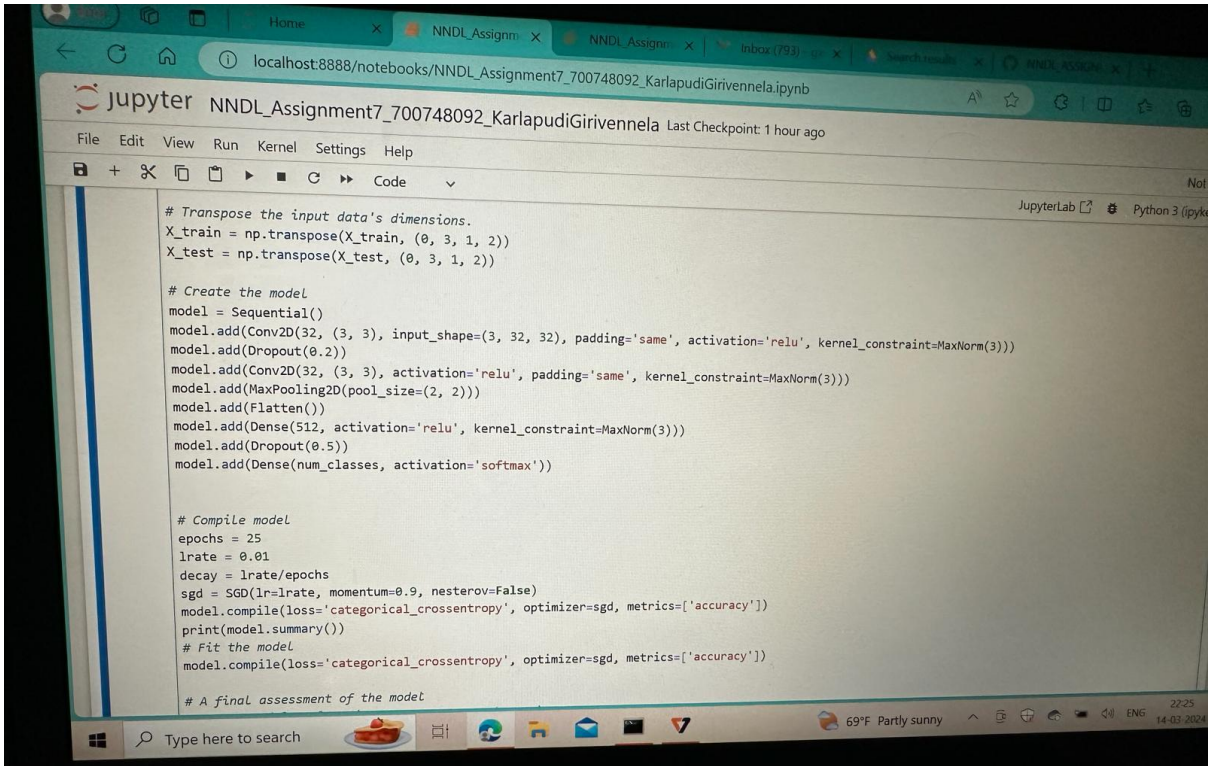
4. Follow the instruction below and then report how the performance changed.(apply all at once)
• Convolutional input layer, 32 feature maps with a size of 3×3 and a rectifier activation function. • Dropout layer at 20%. • Convolutional layer, 32 feature maps with a size of 3×3 and a rectifier activation function. • Max Pool layer with size 2×2. • Convolutional layer, 64 feature maps with a size of 3×3 and a rectifier activation function. •

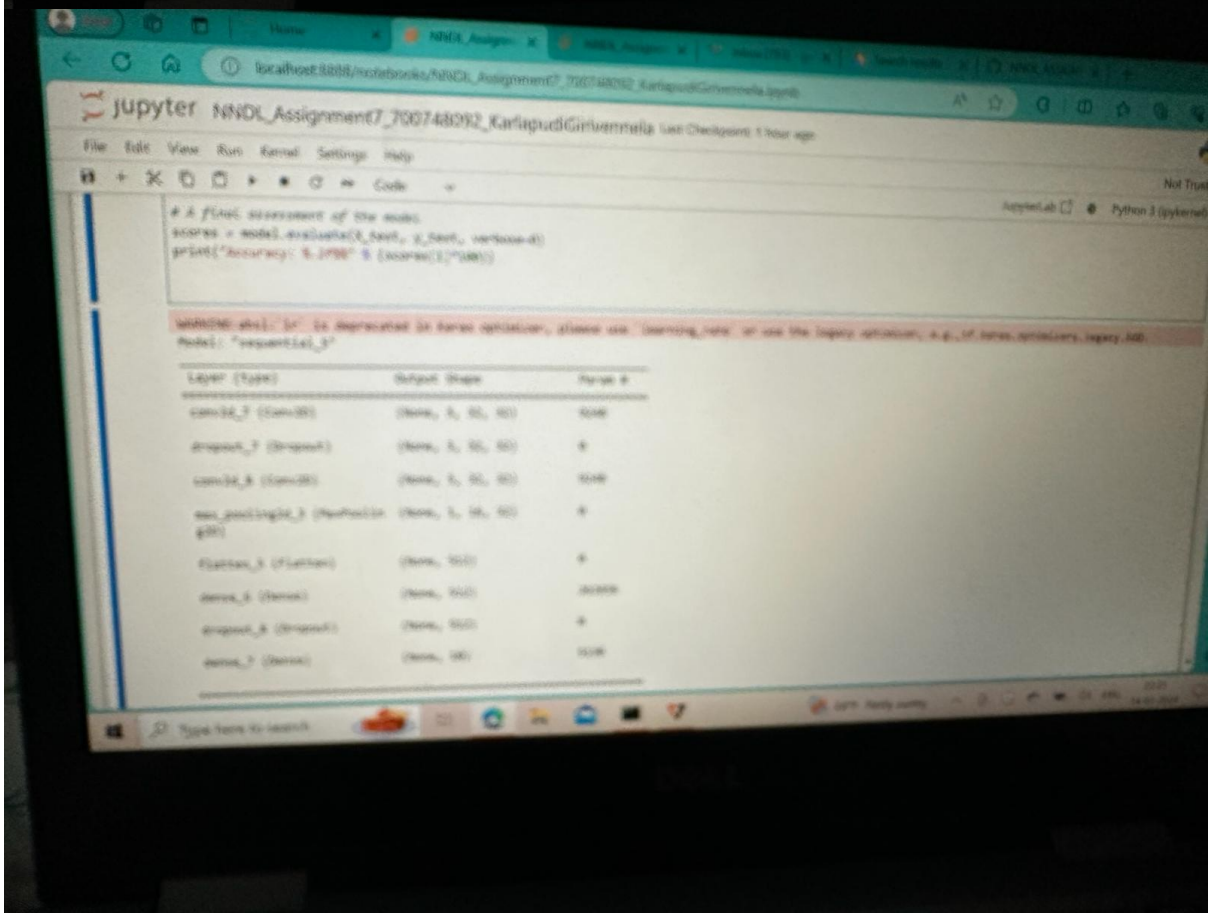
Dropout layer at 20%. • Convolutional layer, 64 feature maps with a size of 3×3 and a rectifier activation function. • Max Pool layer with size 2×2. • Convolutional layer, 128 feature maps with a size of 3×3 and a rectifier activation function. • Dropout layer at 20%. • Convolutional layer,128 feature maps with a size of 3×3 and a rectifier activation function. • Max Pool layer with size 2×2. • Flatten layer. • Dropout layer at 20%. • Fully connected layer with 1024 units and a rectifier activation function. • Dropout layer at 20%. • Fully connected layer with 512 units and a rectifier activation function. • Dropout layer at 20%. • Fully connected output layer with 10 units and a Softmax activation function

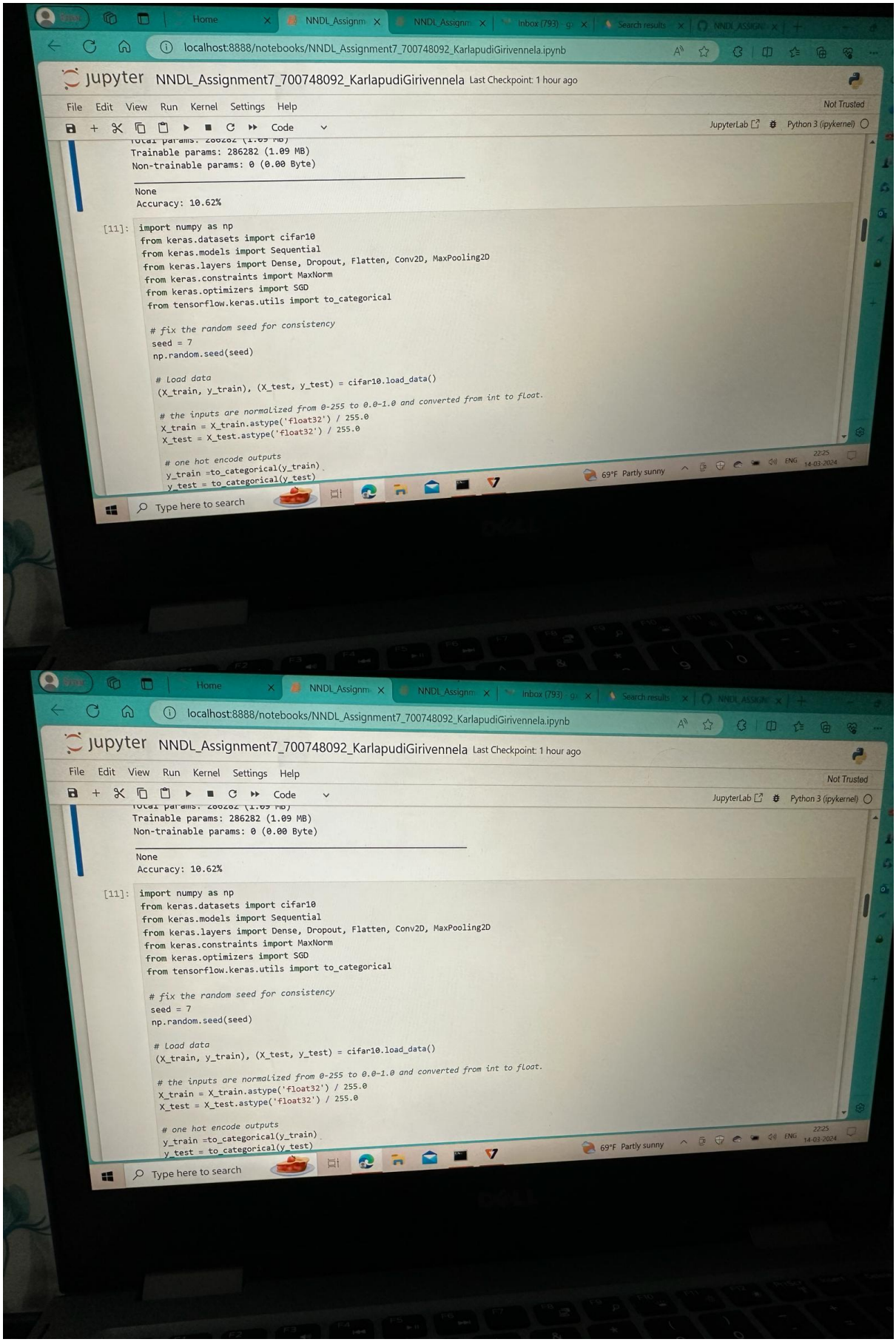
GITHUB LINK:

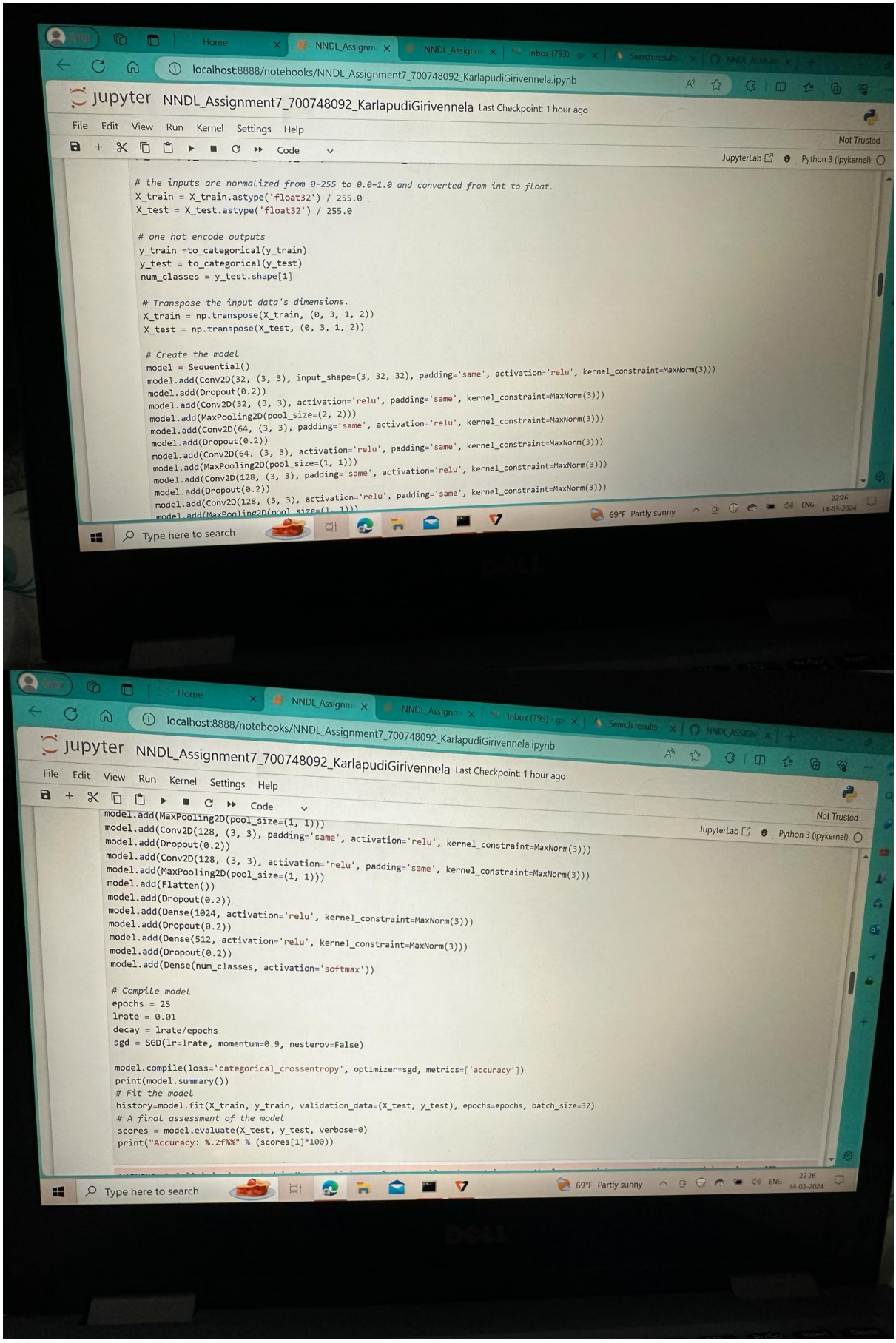
https://github.com/Jyoshna200/jyoshna-nueral-ass-7


localhost:8888/notebooks/NNDL_Assignment7_700748092_KarlapudiGirivennela.ipynb
jupyter NNDL_Assignment7_700748092_KarlapudiGirivennela Last Checkpoint: 1 hour ago
File Edit View Run Kernel Settings Help
Code
Not Trusted
JupyterLab
Python 3 (ipykernel)
[9]:
import numpy as np
from keras.datasets import cifar10
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPooling2D
from keras.constraints import MaxNorm
from keras.optimizers import SGD
from tensorflow.keras.utils import to_categorical
# fix the random seed for consistency
seed = 7
np.random.seed(seed)
# Load data
(X_train, y_train), (X_test, y_test) = cifar10.load_data()
# the inputs are normalized from 0-255 to 0.0-1.0 and converted from int to float.
X_train = X_train.astype('float32') / 255.0
X_test = X_test.astype('float32') / 255.0
# one hot encode outputs
y_train =to_categorical(y_train)
y_test = to_categorical(y_test)
num_classes = y_test.shape[1]
# Transpose the input data's dimensions.
Type here to search

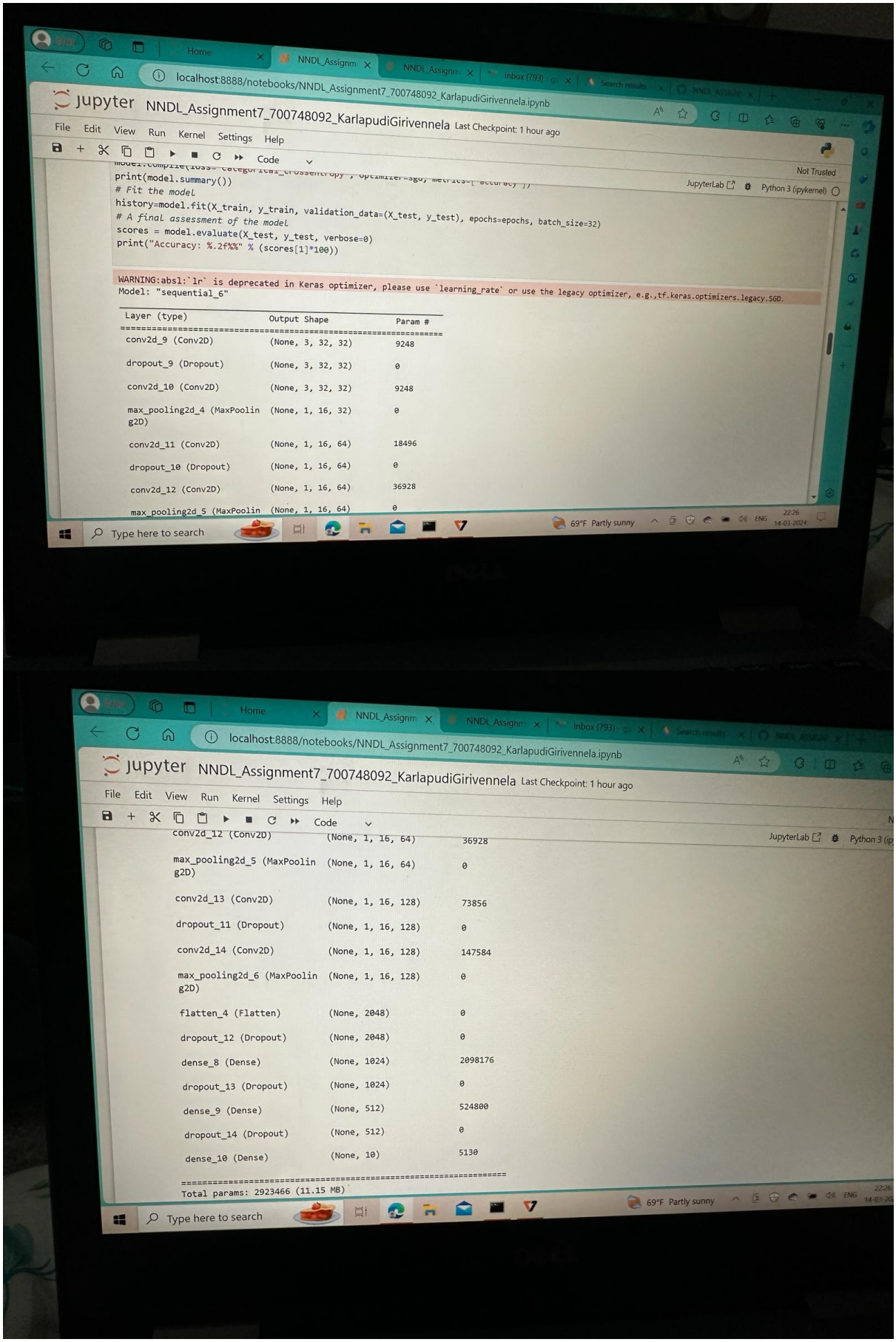

localhost:8888/notebooks/NNDL_Assignment7_700748092_KarlapudiGirivennela.ipynb
jupyter NNDL_Assignment7_700748092_KarlapudiGirivennela Last Checkpoint: 1 hour ago
File Edit View Run Kernel Settings Help
Code
# Transpose the input data's dimensions.
X_train = np.transpose(X_train, (0, 3, 1, 2))
X_test = np.transpose(X_test, (0, 3, 1, 2))
# Create the model
model = Sequential()
model.add(Conv2D(32, (3, 3), input_shape=(3, 32, 32), padding='same', activation='relu', kernel_constraint=MaxNorm(3)))
model.add(Dropout(0.2))
model.add(Conv2D(32, (3, 3), activation='relu', padding='same', kernel_constraint=MaxNorm(3)))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(512, activation='relu', kernel_constraint=MaxNorm(3)))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))
# Compile model
epochs = 25
lrate = 0.01
decay = lrate/epochs
sgd = SGD(lr=lrate, momentum=0.9, nesterov=False)
model.compile(loss='categorical_crossentropy', optimizer=sgd, metrics=['accuracy'])
print(model.summary())
# Fit the model
model.compile(loss='categorical_crossentropy', optimizer=sgd, metrics=['accuracy'])
# A final assessment of the model
Type here to search
69°F Partly sunny
ENG
14-03-2024



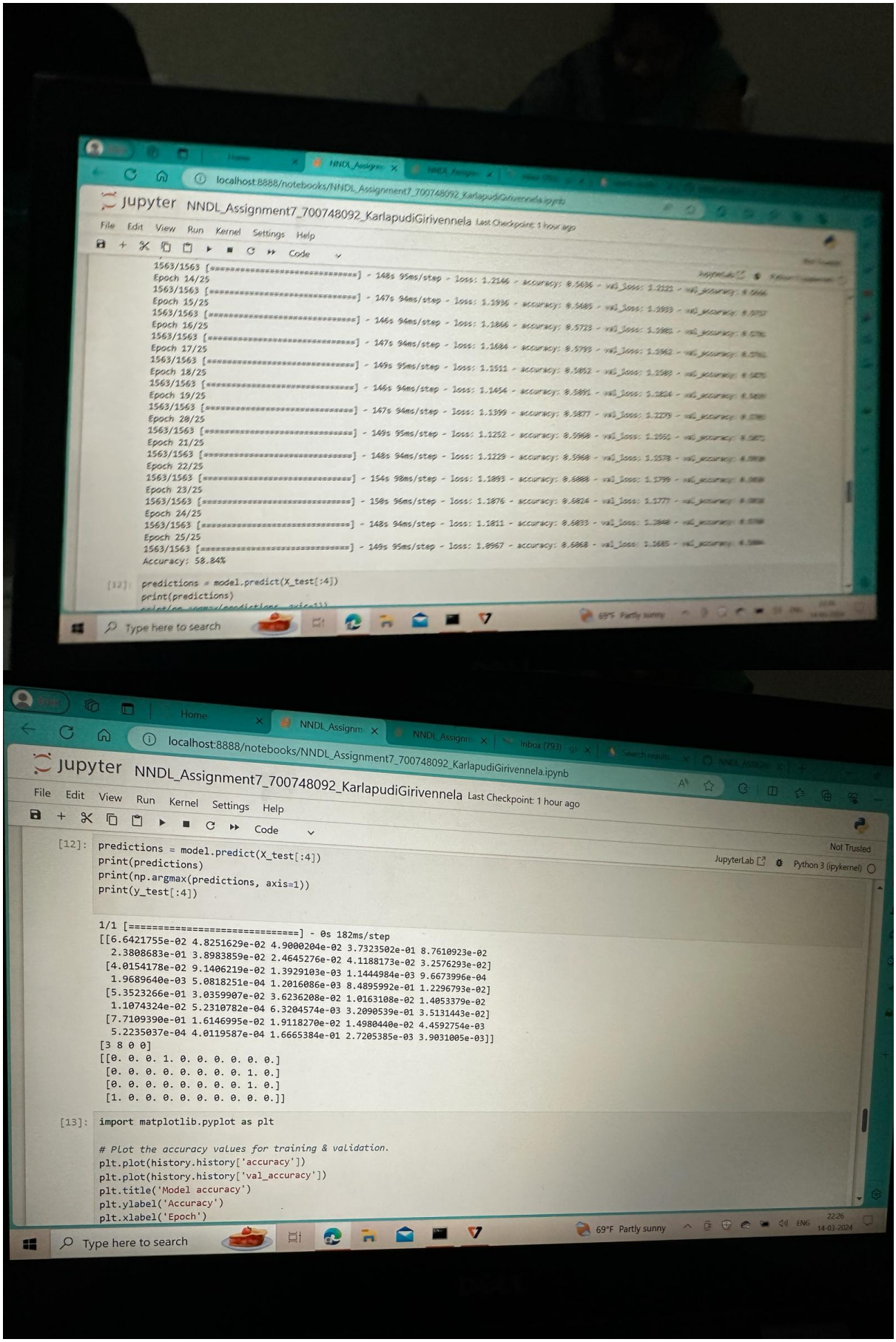

localhost:8888/notebooks/NNDL_Assignment7_700748092_KarlapudiGirivennela.ipynb
jupyter NNDL_Assignment7_700748092_KarlapudiGirivennela Last Checkpoint: 1 hour ago
File Edit View Run Kernel Settings Help
Not Trusted
Code
JupyterLab Python 3 (ipykernel)
Trainable params: 286282 (1.09 MB)
Non-trainable params: 0 (0.00 Byte)
None
Accuracy: 10.62%
[11]:
import numpy as np
from keras.datasets import cifar10
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPooling2D
from keras.constraints import MaxNorm
from keras.optimizers import SGD
from tensorflow.keras.utils import to_categorical
# fix the random seed for consistency
seed = 7
np.random.seed(seed)
# load data
(X_train, y_train), (X_test, y_test) = cifar10.load_data()
# the inputs are normalized from 0-255 to 0.0-1.0 and converted from int to float.
X_train = X_train.astype('float32') / 255.0
X_test = X_test.astype('float32') / 255.0
# one hot encode outputs
y_train =to_categorical(y_train)
y_test = to_categorical(y_test)
Type here to search
69°F Partly sunny
ENG
22:25
14-03-2024



```
# the inputs are normalized from 0-255 to 0.0-1.0 and converted from int to float.
X_train = X_train.astype('float32') / 255.0
X_test = X_test.astype('float32') / 255.0

# one hot encode outputs
y_train =to_categorical(y_train)
y_test = to_categorical(y_test)
num_classes = y_test.shape[1]

# Transpose the input data's dimensions.
X_train = np.transpose(X_train, (0, 3, 1, 2))
X_test = np.transpose(X_test, (0, 3, 1, 2))

# Create the model
model = Sequential()
model.add(Conv2D(32, (3, 3), input_shape=(3, 32, 32), padding='same', activation='relu', kernel_constraint=MaxNorm(3)))
model.add(Dropout(0.2))
model.add(Conv2D(32, (3, 3), activation='relu', padding='same', kernel_constraint=MaxNorm(3)))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(64, (3, 3), padding='same', activation='relu', kernel_constraint=MaxNorm(3)))
model.add(Dropout(0.2))
model.add(Conv2D(64, (3, 3), activation='relu', padding='same', kernel_constraint=MaxNorm(3)))
model.add(MaxPooling2D(pool_size=(1, 1)))
model.add(Conv2D(128, (3, 3), padding='same', activation='relu', kernel_constraint=MaxNorm(3)))
model.add(Dropout(0.2))
model.add(Conv2D(128, (3, 3), activation='relu', padding='same', kernel_constraint=MaxNorm(3)))
model.add(MaxPooling2D(pool_size=(1, 1)))
model.add(Flatten())
model.add(Dropout(0.2))
model.add(Dense(1024, activation='relu', kernel_constraint=MaxNorm(3)))
model.add(Dropout(0.2))
model.add(Dense(512, activation='relu', kernel_constraint=MaxNorm(3)))
model.add(Dropout(0.2))
model.add(Dense(num_classes, activation='softmax'))

# Compile model
epochs = 25
lrate = 0.01
decay = lrate/epochs
sgd = SGD(lr=lrate, momentum=0.9, nesterov=False)

model.compile(loss='categorical_crossentropy', optimizer=sgd, metrics=['accuracy'])
print(model.summary())
# Fit the model
history=model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=epochs, batch_size=32)
# A final assessment of the model
scores = model.evaluate(X_test, y_test, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))
```


Home
NNDL_Assignm
Inbox (793)
localhost:8888/notebooks/NNDL_Assignment7_700748092_KarlapudiGirivennela.ipynb
jupyter NNDL_Assignment7_700748092_KarlapudiGirivennela Last Checkpoint: 1 hour ago
File Edit View Run Kernel Settings Help
Code
Not Trusted
JupyterLab
Python 3 (ipykernel)
print(model.summary())
# Fit the model
history=model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=epochs, batch_size=32)
# A final assessment of the model
scores = model.evaluate(X_test, y_test, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))
WARNING:absl:`lr` is deprecated in Keras optimizer, please use `learning_rate` or use the legacy optimizer, e.g.,tf.keras.optimizers.legacy.SGD.
Model: "sequential_6"
Layer (type) Output Shape Param #
conv2d_9 (Conv2D) (None, 3, 32, 32) 9248
dropout_9 (Dropout) (None, 3, 32, 32) 0
conv2d_10 (Conv2D) (None, 3, 32, 32) 9248
max_pooling2d_4 (MaxPoolin g2D) (None, 1, 16, 32) 0
conv2d_11 (Conv2D) (None, 1, 16, 64) 18496
dropout_10 (Dropout) (None, 1, 16, 64) 0
conv2d_12 (Conv2D) (None, 1, 16, 64) 36928
max_pooling2d_5 (MaxPoolin (None, 1, 16, 64) 0
Type here to search
69°F Partly sunny
ENG
22:26
14-03-2024

Home
NNDL_Assignm
Inbox (793)
localhost:8888/notebooks/NNDL_Assignment7_700748092_KarlapudiGirivennela.ipynb
jupyter NNDL_Assignment7_700748092_KarlapudiGirivennela Last Checkpoint: 1 hour ago
File Edit View Run Kernel Settings Help
Code
JupyterLab
conv2d_12 (Conv2D) (None, 1, 16, 64) 36928
max_pooling2d_5 (MaxPoolin g2D) (None, 1, 16, 64) 0
conv2d_13 (Conv2D) (None, 1, 16, 128) 73856
dropout_11 (Dropout) (None, 1, 16, 128) 0
conv2d_14 (Conv2D) (None, 1, 16, 128) 147584
max_pooling2d_6 (MaxPoolin g2D) (None, 1, 16, 128) 0
flatten_4 (Flatten) (None, 2048) 0
dropout_12 (Dropout) (None, 2048) 0
dense_8 (Dense) (None, 1024) 2098176
dropout_13 (Dropout) (None, 1024) 0
dense_9 (Dense) (None, 512) 524800
dropout_14 (Dropout) (None, 512) 0
dense_10 (Dense) (None, 10) 5130
Total params: 2923466 (11.15 MB)
Type here to search
69°F Partly sunny
ENG


localhost:8888/notebooks/NNDL_Assignment7_700748092_KarlapudiGirivennela.ipynb
Jupyter NNDL_Assignment7_700748092_KarlapudiGirivennela Last Checkpoint: 1 hour ago
File Edit View Run Kernel Settings Help
Code
1563/1563 [==========] - 148s 95ms/step - loss: 1.2146 - accuracy: 0.5636 - val_loss: 1.2321 - val_accuracy: 0.5666
Epoch 14/25
1563/1563 [==========] - 147s 94ms/step - loss: 1.1936 - accuracy: 0.5685 - val_loss: 1.1893 - val_accuracy: 0.5757
Epoch 15/25
1563/1563 [==========] - 146s 94ms/step - loss: 1.1866 - accuracy: 0.5723 - val_loss: 1.1981 - val_accuracy: 0.5781
Epoch 16/25
1563/1563 [==========] - 147s 94ms/step - loss: 1.1684 - accuracy: 0.5799 - val_loss: 1.1963 - val_accuracy: 0.5791
Epoch 17/25
1563/1563 [==========] - 149s 95ms/step - loss: 1.1511 - accuracy: 0.5852 - val_loss: 1.1582 - val_accuracy: 0.5875
Epoch 18/25
1563/1563 [==========] - 146s 94ms/step - loss: 1.1454 - accuracy: 0.5891 - val_loss: 1.1824 - val_accuracy: 0.5830
Epoch 19/25
1563/1563 [==========] - 147s 94ms/step - loss: 1.1399 - accuracy: 0.5877 - val_loss: 1.2279 - val_accuracy: 0.5781
Epoch 20/25
1563/1563 [==========] - 149s 95ms/step - loss: 1.1252 - accuracy: 0.5968 - val_loss: 1.1551 - val_accuracy: 0.5871
Epoch 21/25
1563/1563 [==========] - 148s 94ms/step - loss: 1.1229 - accuracy: 0.5968 - val_loss: 1.1578 - val_accuracy: 0.5910
Epoch 22/25
1563/1563 [==========] - 154s 98ms/step - loss: 1.1093 - accuracy: 0.6003 - val_loss: 1.1799 - val_accuracy: 0.5868
Epoch 23/25
1563/1563 [==========] - 150s 96ms/step - loss: 1.1076 - accuracy: 0.6014 - val_loss: 1.1777 - val_accuracy: 0.5858
Epoch 24/25
1563/1563 [==========] - 148s 94ms/step - loss: 1.1011 - accuracy: 0.6033 - val_loss: 1.1848 - val_accuracy: 0.5798
Epoch 25/25
1563/1563 [==========] - 149s 95ms/step - loss: 1.0967 - accuracy: 0.6068 - val_loss: 1.1685 - val_accuracy: 0.5884
Accuracy: 58.84%
[12]: predictions = model.predict(X_test[:4])
print(predictions)
Type here to search
69°F Partly sunny
localhost:8888/notebooks/NNDL_Assignment7_700748092_KarlapudiGirivennela.ipynb
Jupyter NNDL_Assignment7_700748092_KarlapudiGirivennela Last Checkpoint: 1 hour ago
File Edit View Run Kernel Settings Help
Code
Not Trusted
JupyterLab Python 3 (ipykernel)
[12]: predictions = model.predict(X_test[:4])
print(predictions)
print(np.argmax(predictions, axis=1))
print(y_test[:4])
1/1 [==========] - 0s 182ms/step
[[6.6421755e-02 4.8251629e-02 4.9000204e-02 3.7323502e-01 8.7610923e-02
2.3808683e-01 3.8983859e-02 2.4645276e-02 4.1188173e-02 3.2576293e-02]
[4.0154178e-02 9.1406219e-02 1.3929103e-03 1.1444984e-03 9.6673996e-04
1.9689640e-03 5.0818251e-04 1.2016086e-03 8.4895992e-01 1.2296793e-02]
[5.3523266e-01 3.0359907e-02 3.6236208e-02 1.0163108e-02 1.4053379e-02
1.1074324e-02 5.2310782e-04 6.3204574e-03 3.2090539e-01 3.5131443e-02]
[7.7109390e-01 1.6146995e-02 1.9118270e-02 1.4980440e-02 4.4592754e-03
5.2235037e-04 4.0119587e-04 1.6665384e-01 2.7205385e-03 3.9031005e-03]]
[3 8 0 0]
[[0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]
[0. 0. 0. 0. 0. 0. 0. 0. 1. 0.]
[0. 0. 0. 0. 0. 0. 0. 0. 1. 0.]
[1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
[13]: import matplotlib.pyplot as plt
# Plot the accuracy values for training & validation.
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
Type here to search
69°F Partly sunny
ENG
22:26
14-03-2024

localhost:8888/notebooks/NNDL_Assignment7_700748092_KarlapudiGirivennela.ipynb

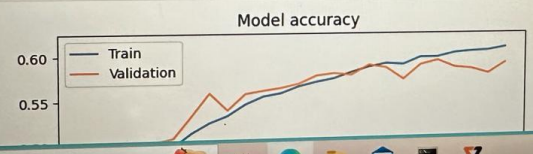
jupyter NNDL_Assignment7_700748092_KarlapudiGirivennela Last Checkpoint: 1 hour ago

File Edit View Run Kernel Settings Help

Not Trusted

JupyterLab Python 3 (ipykernel)

[13]:
```
import matplotlib.pyplot as plt

# Plot the accuracy values for training & validation.
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='upper left')
plt.show()

# Visualize the training and validation loss values.
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 left')
plt.show()
```



Type here to search 69°F Partly sunny ENG 22:26 14-03-2024

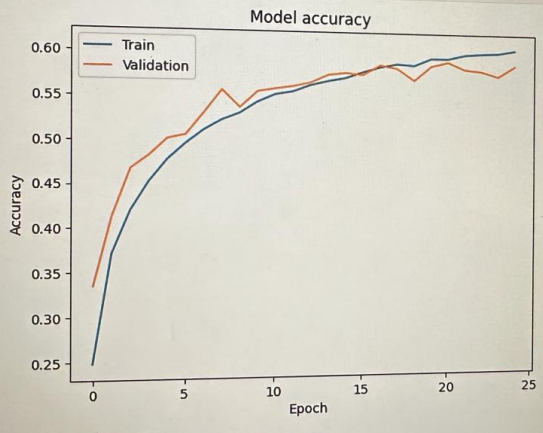
localhost:8888/notebooks/NNDL_Assignment7_700748092_KarlapudiGirivennela.ipynb

jupyter NNDL_Assignment7_700748092_KarlapudiGirivennela Last Checkpoint: 1 hour ago

File Edit View Run Kernel Settings Help

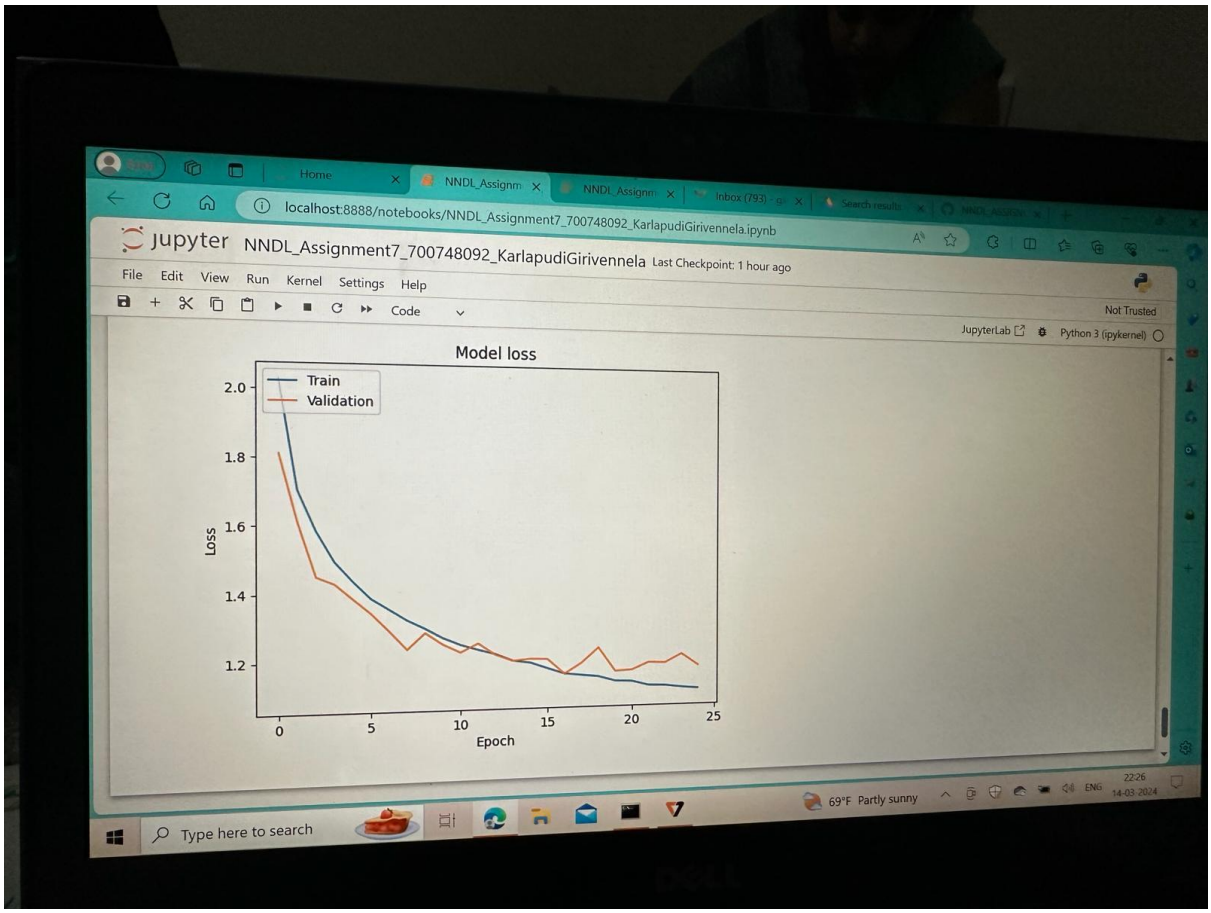
Not Trusted

JupyterLab Python 3 (ipykernel)



Model loss

Type here to search 69°F Partly sunny ENG 22:26 14-03-2024


Home
NNDL_Assignm
NNDL_Assignm
Inbox (793) - gi
localhost:8888/notebooks/NNDL_Assignment7_700748092_KarlapudiGirivennela.ipynb
jupyter NNDL_Assignment7_700748092_KarlapudiGirivennela Last Checkpoint: 1 hour ago
File Edit View Run Kernel Settings Help
Code
Not Trusted
JupyterLab
Python 3 (ipykernel)
Model loss
Train
Validation
Loss
Epoch
2.0
1.8
1.6
1.4
1.2
0
5
10
15
20
25
Type here to search
69°F Partly sunny
ENG
22:26
14-03-2024