Numbers

December 12, 2023

1 Model:

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
from keras.utils import to_categorical
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from keras.callbacks import EarlyStopping, ModelCheckpoint
from keras.datasets import mnist
(X_train, y_train), (X_test, y_test) = mnist.load_data()

print(X_train.shape, y_train.shape , X_test.shape , y_test.shape)

# Assuming y_train is integer labels
y_train_one_hot = to_categorical(y_train, num_classes=10)

# Reshape the input data to include the channel dimension
X_train = X_train.reshape((X_train.shape[0], X_train.shape[1], X_train.
 ↪shape[2], 1))

# Define the model
model = Sequential()

# Add layers to the model
model.add(Conv2D(32, kernel_size=(3, 3), input_shape=(28, 28, 1),␣
 ↪activation='relu'))
model.add(MaxPooling2D((2, 2)))

model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))

model.add(Flatten())

model.add(Dropout(0.25))

# Adjust the number of units in the output layer based on your classification␣
 ↪task
```

```
model.add(Dense(10, activation='softmax'))

# Compile the model
model.compile(optimizer='adam', loss='categorical_crossentropy',␣
 ↪metrics=['accuracy'])

# Define callbacks
es = EarlyStopping(monitor='val_accuracy', min_delta=0.01, patience=4,␣
 ↪verbose=1, restore_best_weights=True)
mc = ModelCheckpoint("model.h5", monitor="val_accuracy", verbose=1,␣
 ↪save_best_only=True)

# Train the model without callbacks
history = model.fit(X_train, y_train_one_hot, epochs=50, validation_split=0.3)
```

```
WARNING:tensorflow:From C:\Users\tejus\AppData\Roaming\Python\Python311\site-
packages\keras\src\losses.py:2976: The name
tf.losses.sparse_softmax_cross_entropy is deprecated. Please use
tf.compat.v1.losses.sparse_softmax_cross_entropy instead.

(60000, 28, 28) (60000,) (10000, 28, 28) (10000,)
WARNING:tensorflow:From C:\Users\tejus\AppData\Roaming\Python\Python311\site-
packages\keras\src\backend.py:873: The name tf.get_default_graph is deprecated.
Please use tf.compat.v1.get_default_graph instead.

WARNING:tensorflow:From C:\Users\tejus\AppData\Roaming\Python\Python311\site-
packages\keras\src\layers\pooling\max_pooling2d.py:161: The name tf.nn.max_pool
is deprecated. Please use tf.nn.max_pool2d instead.

WARNING:tensorflow:From C:\Users\tejus\AppData\Roaming\Python\Python311\site-
packages\keras\src\optimizers\__init__.py:309: The name tf.train.Optimizer is
deprecated. Please use tf.compat.v1.train.Optimizer instead.

Epoch 1/50
WARNING:tensorflow:From C:\Users\tejus\AppData\Roaming\Python\Python311\site-
packages\keras\src\utils\tf_utils.py:492: The name tf.ragged.RaggedTensorValue
is deprecated. Please use tf.compat.v1.ragged.RaggedTensorValue instead.

WARNING:tensorflow:From C:\Users\tejus\AppData\Roaming\Python\Python311\site-
packages\keras\src\engine\base_layer_utils.py:384: The name
tf.executing_eagerly_outside_functions is deprecated. Please use
tf.compat.v1.executing_eagerly_outside_functions instead.

1313/1313 [==============================] - 11s 7ms/step - loss: 0.6987 -
accuracy: 0.9050 - val_loss: 0.1065 - val_accuracy: 0.9687
Epoch 2/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.1198 -
```

```
accuracy: 0.9650 - val_loss: 0.0784 - val_accuracy: 0.9766
Epoch 3/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0910 -
accuracy: 0.9727 - val_loss: 0.0903 - val_accuracy: 0.9752
Epoch 4/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0829 -
accuracy: 0.9749 - val_loss: 0.0578 - val_accuracy: 0.9822
Epoch 5/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0746 -
accuracy: 0.9776 - val_loss: 0.0632 - val_accuracy: 0.9815
Epoch 6/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0677 -
accuracy: 0.9788 - val_loss: 0.0624 - val_accuracy: 0.9813
Epoch 7/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0619 -
accuracy: 0.9811 - val_loss: 0.0561 - val_accuracy: 0.9833
Epoch 8/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0572 -
accuracy: 0.9826 - val_loss: 0.0566 - val_accuracy: 0.9857
Epoch 9/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0518 -
accuracy: 0.9835 - val_loss: 0.0668 - val_accuracy: 0.9823
Epoch 10/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0502 -
accuracy: 0.9850 - val_loss: 0.0603 - val_accuracy: 0.9842
Epoch 11/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0477 -
accuracy: 0.9855 - val_loss: 0.0603 - val_accuracy: 0.9843
Epoch 12/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0428 -
accuracy: 0.9869 - val_loss: 0.0671 - val_accuracy: 0.9841
Epoch 13/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0424 -
accuracy: 0.9873 - val_loss: 0.0695 - val_accuracy: 0.9823
Epoch 14/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0403 -
accuracy: 0.9874 - val_loss: 0.0712 - val_accuracy: 0.9844
Epoch 15/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0430 -
accuracy: 0.9875 - val_loss: 0.0609 - val_accuracy: 0.9864
Epoch 16/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0404 -
accuracy: 0.9889 - val_loss: 0.0690 - val_accuracy: 0.9867
Epoch 17/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0376 -
accuracy: 0.9895 - val_loss: 0.0742 - val_accuracy: 0.9846
Epoch 18/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0401 -
```

```
accuracy: 0.9894 - val_loss: 0.0877 - val_accuracy: 0.9863
Epoch 19/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0361 -
accuracy: 0.9905 - val_loss: 0.0733 - val_accuracy: 0.9870
Epoch 20/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0363 -
accuracy: 0.9903 - val_loss: 0.0686 - val_accuracy: 0.9861
Epoch 21/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0381 -
accuracy: 0.9908 - val_loss: 0.0898 - val_accuracy: 0.9851
Epoch 22/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0419 -
accuracy: 0.9898 - val_loss: 0.0912 - val_accuracy: 0.9852
Epoch 23/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0402 -
accuracy: 0.9911 - val_loss: 0.0741 - val_accuracy: 0.9864
Epoch 24/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0384 -
accuracy: 0.9909 - val_loss: 0.0890 - val_accuracy: 0.9841
Epoch 25/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0420 -
accuracy: 0.9898 - val_loss: 0.0876 - val_accuracy: 0.9861
Epoch 26/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0293 -
accuracy: 0.9922 - val_loss: 0.0905 - val_accuracy: 0.9866
Epoch 27/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0345 -
accuracy: 0.9920 - val_loss: 0.0880 - val_accuracy: 0.9867
Epoch 28/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0346 -
accuracy: 0.9924 - val_loss: 0.1112 - val_accuracy: 0.9848
Epoch 29/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0431 -
accuracy: 0.9912 - val_loss: 0.1096 - val_accuracy: 0.9854
Epoch 30/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0368 -
accuracy: 0.9919 - val_loss: 0.0933 - val_accuracy: 0.9886
Epoch 31/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0378 -
accuracy: 0.9917 - val_loss: 0.1294 - val_accuracy: 0.9857
Epoch 32/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0369 -
accuracy: 0.9928 - val_loss: 0.1031 - val_accuracy: 0.9872
Epoch 33/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0352 -
accuracy: 0.9928 - val_loss: 0.1234 - val_accuracy: 0.9867
Epoch 34/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0414 -
```

```
accuracy: 0.9918 - val_loss: 0.1101 - val_accuracy: 0.9861
Epoch 35/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0356 -
accuracy: 0.9927 - val_loss: 0.1284 - val_accuracy: 0.9849
Epoch 36/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0357 -
accuracy: 0.9930 - val_loss: 0.1383 - val_accuracy: 0.9851
Epoch 37/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0414 -
accuracy: 0.9923 - val_loss: 0.1168 - val_accuracy: 0.9854
Epoch 38/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0345 -
accuracy: 0.9926 - val_loss: 0.1247 - val_accuracy: 0.9861
Epoch 39/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0356 -
accuracy: 0.9932 - val_loss: 0.1267 - val_accuracy: 0.9867
Epoch 40/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0356 -
accuracy: 0.9937 - val_loss: 0.1160 - val_accuracy: 0.9874
Epoch 41/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0401 -
accuracy: 0.9929 - val_loss: 0.1575 - val_accuracy: 0.9866
Epoch 42/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0485 -
accuracy: 0.9921 - val_loss: 0.1235 - val_accuracy: 0.9874
Epoch 43/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0402 -
accuracy: 0.9931 - val_loss: 0.1574 - val_accuracy: 0.9864
Epoch 44/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0389 -
accuracy: 0.9934 - val_loss: 0.1369 - val_accuracy: 0.9866
Epoch 45/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0405 -
accuracy: 0.9934 - val_loss: 0.1558 - val_accuracy: 0.9854
Epoch 46/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0379 -
accuracy: 0.9940 - val_loss: 0.1405 - val_accuracy: 0.9878
Epoch 47/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0395 -
accuracy: 0.9937 - val_loss: 0.1702 - val_accuracy: 0.9872
Epoch 48/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0431 -
accuracy: 0.9938 - val_loss: 0.1542 - val_accuracy: 0.9868
Epoch 49/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0461 -
accuracy: 0.9937 - val_loss: 0.1770 - val_accuracy: 0.9867
Epoch 50/50
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0369 -
```

```
accuracy: 0.9948 - val_loss: 0.1356 - val_accuracy: 0.9876
```

[2]:
```
# Train the model with callbacks
history = model.fit(X_train, y_train_one_hot, epochs=50, validation_split=0.3,␣
 ↪callbacks=[es, mc])
```

```
Epoch 1/50
1310/1313 [============================>.] - ETA: 0s - loss: 0.0392 - accuracy:
0.9948
Epoch 1: val_accuracy improved from -inf to 0.98539, saving model to model.h5
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0392 -
accuracy: 0.9949 - val_loss: 0.2018 - val_accuracy: 0.9854
Epoch 2/50
  25/1313 […] - ETA: 8s - loss: 0.0177 - accuracy:
0.9962

C:\Users\tejus\AppData\Roaming\Python\Python311\site-
packages\keras\src\engine\training.py:3103: UserWarning: You are saving your
model as an HDF5 file via `model.save()`. This file format is considered legacy.
We recommend using instead the native Keras format, e.g.
`model.save('my_model.keras')`.
  saving_api.save_model(

1305/1313 [============================>.] - ETA: 0s - loss: 0.0491 - accuracy:
0.9931
Epoch 2: val_accuracy improved from 0.98539 to 0.98683, saving model to model.h5
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0488 -
accuracy: 0.9931 - val_loss: 0.1695 - val_accuracy: 0.9868
Epoch 3/50
1306/1313 [============================>.] - ETA: 0s - loss: 0.0420 - accuracy:
0.9941
Epoch 3: val_accuracy improved from 0.98683 to 0.98883, saving model to model.h5
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0423 -
accuracy: 0.9941 - val_loss: 0.1483 - val_accuracy: 0.9888
Epoch 4/50
1305/1313 [============================>.] - ETA: 0s - loss: 0.0317 - accuracy:
0.9945
Epoch 4: val_accuracy did not improve from 0.98883
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0318 -
accuracy: 0.9945 - val_loss: 0.1386 - val_accuracy: 0.9885
Epoch 5/50
1311/1313 [============================>.] - ETA: 0s - loss: 0.0367 - accuracy:
0.9946Restoring model weights from the end of the best epoch: 1.

Epoch 5: val_accuracy did not improve from 0.98883
1313/1313 [==============================] - 10s 7ms/step - loss: 0.0367 -
accuracy: 0.9946 - val_loss: 0.1857 - val_accuracy: 0.9842
Epoch 5: early stopping
```

2 Application:

[3]:
```
import pygame
import sys
import numpy as np
from tensorflow.keras.models import load_model
import cv2
# Press "N" to clear screen
WINDOWSIZEX = 640
WINDOWSIZEY = 480

WHITE = (255, 255, 255)
BLACK = (0, 0, 0)
RED = (255, 0, 0)

MODEL = load_model("model.h5")

LABELS = {0: "Zero", 1: "One",
          2: "Two", 3: "Three",
          4: "Four", 5: "Five",
          6: "Six", 7: "Seven",
          8: "Eight", 9: "Nine"}

pygame.init()

FONT = pygame.font.Font("freesansbold.ttf", 18)
DISPLAYSURF = pygame.display.set_mode((WINDOWSIZEX, WINDOWSIZEY))

pygame.display.set_caption("Digit Board")

iswriting = False
image_cnt = 1
BOUNDARYINC = 5
PREDICT = True
IMAGESAVE = False

number_xcord = []
number_ycord = []
rect_min_x, rect_min_y, rect_max_x, rect_max_y = 0, 0, 0, 0  # Initialize␣
 ↪rectangle coordinates

while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            sys.exit()
```

```
        if event.type == pygame.MOUSEMOTION and iswriting:
            xcord, ycord = event.pos
            pygame.draw.circle(DISPLAYSURF, WHITE, (xcord, ycord), 4, 0)

            number_xcord.append(xcord)
            number_ycord.append(ycord)

        if event.type == pygame.MOUSEBUTTONDOWN:
            iswriting = True

        if event.type == pygame.MOUSEBUTTONUP:
            iswriting = False
            if number_xcord and number_ycord:
                rect_min_x, rect_max_x = min(number_xcord) - BOUNDARYINC,␣
↪max(number_xcord) + BOUNDARYINC
                rect_min_y, rect_max_y = min(number_ycord) - BOUNDARYINC,␣
↪max(number_ycord) + BOUNDARYINC

                number_xcord = []
                number_ycord = []

                image_arr = np.array(pygame.PixelArray(DISPLAYSURF))[rect_min_x:
↪rect_max_x, rect_min_y:rect_max_y].T.astype(np.float32)

                if IMAGESAVE:
                    cv2.imwrite(f"image_{image_cnt}.png", image_arr)
                    image_cnt += 1

                if PREDICT:
                    image = cv2.resize(image_arr, (28, 28))
                    image = np.pad(image, ((10, 10), (10, 10)), 'constant',␣
↪constant_values=0)
                    image = cv2.resize(image, (28, 28)) / 255.0

                    label = str(LABELS[np.argmax(MODEL.predict(image.reshape(1,␣
↪28, 28, 1)))])
                    text_surface = FONT.render(label, True, RED, WHITE)
                    text_rect_obj = text_surface.get_rect()
                    text_rect_obj.left, text_rect_obj.bottom = rect_max_x + 5,␣
↪rect_min_y

                    DISPLAYSURF.blit(text_surface, text_rect_obj)

                # Draw a rectangular box around the written number
                pygame.draw.rect(DISPLAYSURF, RED, (rect_min_x, rect_min_y,␣
↪rect_max_x - rect_min_x, rect_max_y - rect_min_y), 2)
```

```
        if event.type == pygame.KEYDOWN:
            if event.unicode == 'n':
                DISPLAYSURF.fill(BLACK)

    pygame.display.update()
```

```
pygame 2.5.2 (SDL 2.28.3, Python 3.11.5)
Hello from the pygame community. https://www.pygame.org/contribute.html
1/1 [==============================] - 0s 62ms/step
1/1 [==============================] - 0s 17ms/step
1/1 [==============================] - 0s 16ms/step
1/1 [==============================] - 0s 16ms/step
```

```
An exception has occurred, use %tb to see the full traceback.

SystemExit
```

```
C:\Users\tejus\AppData\Roaming\Python\Python311\site-
packages\IPython\core\interactiveshell.py:3558: UserWarning: To exit: use
'exit', 'quit', or Ctrl-D.
  warn("To exit: use 'exit', 'quit', or Ctrl-D.", stacklevel=1)
```

[ ]: