ICP11

CNN:

import numpy as np

import tensorflow as tf

from keras.datasets import mnist

from keras.models import Sequential

from keras.layers import Conv2D, MaxPooling2D, Dense, Flatten

from keras.utils import to\_categorical

(train\_images, train\_labels), (test\_images, test\_labels) = mnist.load\_data()

# Normalize the images.

train\_images = (train\_images / 255) - 0.5

test\_images = (test\_images / 255) - 0.5

# Reshape the images.

train\_images = np.expand\_dims(train\_images, axis=3)

test\_images = np.expand\_dims(test\_images, axis=3)

num\_filters = 8

filter\_size = 3

pool\_size = 2

# Build the model.

model = Sequential([

  Conv2D(num\_filters, filter\_size, input\_shape=(28, 28, 1)),

  MaxPooling2D(pool\_size=pool\_size),

  Flatten(),

  Dense(10, activation='softmax'),

])

# Compile the model.

model.compile(

  'adam',

  loss='categorical\_crossentropy',

  metrics=['accuracy'],

)

# Train the model.

model.fit(

  train\_images,

  to\_categorical(train\_labels),

  epochs=3,

  validation\_data=(test\_images, to\_categorical(test\_labels)),

)

# Predict on the first 5 test images.

predictions = model.predict(test\_images[:5])

# Print our model's predictions.

print("These are the model predictions :")

print(np.argmax(predictions, axis=1)) # [7, 2, 1, 0, 4]

print("\n")

# Check our predictions against the ground truths.

print("These are the corresponding labels :")

print(test\_labels[:5]) # [7, 2, 1, 0, 4]

Using TensorFlow backend.

Downloading data from <https://s3.amazonaws.com/img-datasets/mnist.npz>

11493376/11490434 [==============================] - 1s 0us/step

Train on 60000 samples, validate on 10000 samples

Epoch 1/3

60000/60000 [==============================] - 16s 273us/step - loss: 0.3487 - accuracy: 0.8998 - val\_loss: 0.1923 - val\_accuracy: 0.9443

Epoch 2/3

60000/60000 [==============================] - 16s 269us/step - loss: 0.1686 - accuracy: 0.9517 - val\_loss: 0.1401 - val\_accuracy: 0.9603

Epoch 3/3

60000/60000 [==============================] - 16s 263us/step - loss: 0.1257 - accuracy: 0.9631 - val\_loss: 0.1164 - val\_accuracy: 0.9645

These are the model predictions :

[7 2 1 0 4]

These are the corresponding labels :

[7 2 1 0 4]