

Guided Project

Machine Learning with ChatGPT: Image Classification Model

Estimated Time
45 minutes



Instructor:
Jousef Murad

Coursera Instructor

Mechanical Engineer

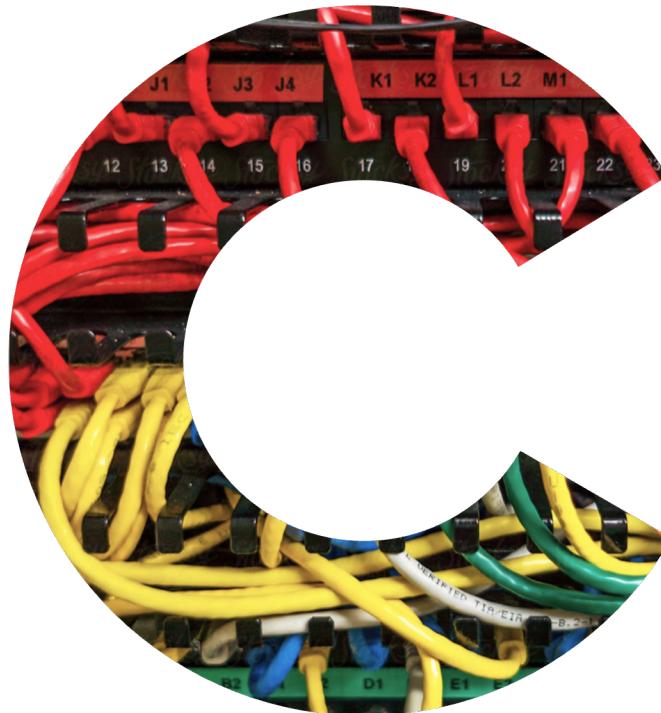
🎬 YouTuber

🌱 Founder of APEX

🚀 Mechanical Engineer

🎙️ Host of the Engineered-Mind Podcast

💻 Teaching Science at courses.jousefmurad.com

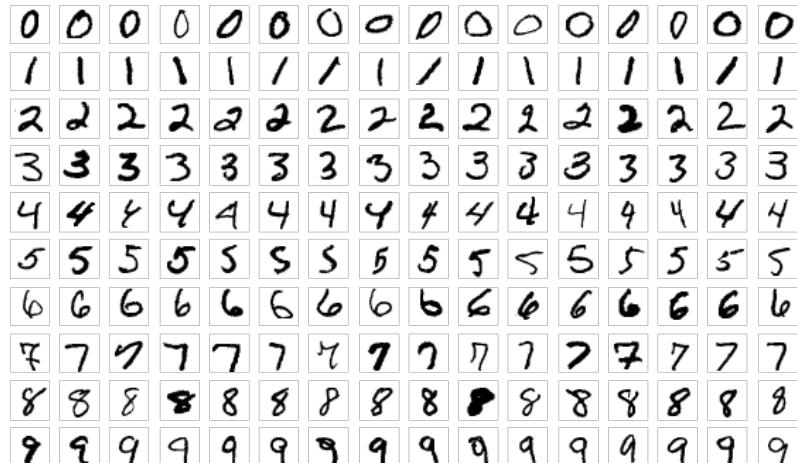




Project Goal

Use ChatGPT to generate Machine Learning Code with the open source library Keras using prompts.

MNIST



Final Example

Completed Example



What will you learn?

Build your own Machine Learning with the help of ChatGPT using the **MNIST** dataset.

Using ChatGPT, you will learn how to **improve** your model and make **better predictions** for recognizing handwritten digits.



Sure! Here's a simple neural network in Keras that solves the MNIST problem:

python

```
import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Flatten

# Load the MNIST dataset
(x_train, y_train), (x_test, y_test) = mnist.load_data()

# Normalize the input data
x_train = x_train.astype('float32') / 255
x_test = x_test.astype('float32') / 255

# Convert labels to one-hot encoding
num_classes = 10
y_train = keras.utils.to_categorical(y_train, num_classes)
y_test = keras.utils.to_categorical(y_test, num_classes)

# Create a sequential model
model = Sequential()

# Add layers to the model
model.add(Flatten(input_shape=x_train.shape[1:]))
model.add(Dense(128, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))

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

# Train the model
model.fit(x_train, y_train, epochs=10, batch_size=32, validation_data=(x_test, y_test))

# Evaluate the model
loss, accuracy = model.evaluate(x_test, y_test)
print('Test loss:', loss)
print('Test accuracy:', accuracy)
```



Sure! Here's a simple neural network in Keras that solves the MNIST problem:

```
python Copy code  
  
import keras  
from keras.datasets import mnist  
from keras.models import Sequential  
from keras.layers import Dense, Flatten  
  
# Load the MNIST dataset  
(x_train, y_train), (x_test, y_test) = mnist.load_data()  
  
# Normalize the input data  
x_train = x_train.astype('float32') / 255  
x_test = x_test.astype('float32') / 255  
  
# Convert labels to one-hot encoding  
num_classes = 10  
y_train = keras.utils.to_categorical(y_train, num_classes)  
y_test = keras.utils.to_categorical(y_test, num_classes)  
  
# Create a sequential model  
model = Sequential()  
  
# Add layers to the model  
model.add(Flatten(input_shape=x_train.shape[1:]))  
model.add(Dense(128, activation='relu'))  
model.add(Dense(num_classes, activation='softmax'))  
  
# Compile the model  
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])  
  
# Train the model  
model.fit(x_train, y_train, epochs=10, batch_size=32, validation_data=(x_test, y_test))  
  
# Evaluate the model  
loss, accuracy = model.evaluate(x_test, y_test)  
print('Test loss:', loss)  
print('Test accuracy:', accuracy)
```



Google Colab

```
ChatGPT_ML_Coursera.ipynb star  
Datei Bearbeiten Anzeige Einfügen Laufzeit Tools Hilfe Zuletzt bearbeitet am 26. Februar  
+ Code + Text  
  
(x)  
↳ import numpy as np  
↳ import tensorflow as tf  
↳ from tensorflow import keras  
↳ from tensorflow.keras import layers  
  
# Load the MNIST dataset  
(x_train, y_train), (x_test, y_test) = keras.datasets.mnist.load_data()  
  
# Normalize pixel values to be between 0 and 1  
x_train = x_train.astype('float32') / 255.0  
x_test = x_test.astype('float32') / 255.0  
  
# Convert labels to one-hot encoding  
y_train = keras.utils.to_categorical(y_train)  
y_test = keras.utils.to_categorical(y_test)  
  
# Define the model architecture  
model = keras.Sequential([  
    layers.Flatten(input_shape=(28, 28)),  
    layers.Dense(128, activation='relu'),  
    layers.Dense(10, activation='softmax')  
])  
  
# Compile the model  
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])  
  
# Train the model  
model.fit(x_train, y_train, epochs=5, validation_data=(x_test, y_test))  
  
# Evaluate the model on test data  
test_loss, test_acc = model.evaluate(x_test, y_test, verbose=2)  
print('Test accuracy:', test_acc)  
  
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz  
11490434/11490434 [=====] - 1s 0us/step  
Epoch 1/5  
1035/1035 [=====] - 19s 8ms/step - loss: 0.2597 - accuracy: 0.9268 - val_loss: 0.1371 - val_accuracy: 0.9570  
Epoch 2/5  
1875/1875 [=====] - 8s 4ms/step - loss: 0.1133 - accuracy: 0.9672 - val_loss: 0.0967 - val_accuracy: 0.9704  
Epoch 3/5  
1875/1875 [=====] - 9s 5ms/step - loss: 0.0766 - accuracy: 0.9773 - val_loss: 0.0823 - val_accuracy: 0.9741  
Epoch 4/5  
1875/1875 [=====] - 9s 5ms/step - loss: 0.0586 - accuracy: 0.9821 - val_loss: 0.0729 - val_accuracy: 0.9776  
Epoch 5/5  
1875/1875 [=====] - 8s 4ms/step - loss: 0.0448 - accuracy: 0.9863 - val_loss: 0.0698 - val_accuracy: 0.9795  
313/313 - 1s - loss: 0.0698 - accuracy: 0.9795 - 517ms/epoch - 2ms/step  
Test accuracy: 0.9794999957084656
```

An Introduction to ChatGPT - Opportunities & Limitations

ChatGPT & How to Log Into ChatGPT



ChatGPT – Generative Pre-Trained Transformer



Things to Note

- ✓ Powerful tool to enhance your abilities!
 - ✗ Outdated Data (ask ChatGPT!)
 - ✗ Cannot make Qualitative Judgements
 - ✗ Data I/O limited (model purely text based)
 - ✗ Inherently biased
 - (✗ Bad Jokes)
- Not a magic pill, but step in the right direction



Linas Beliūnas [in](#)
@linas.beliunas

Time it took to reach **1 million users**:

Netflix - 3.5 years
Airbnb - 2.5 years
Facebook - 10 months
Spotify - 5 months
Instagram - 2.5 months
iPhone - 74 days
ChatGPT - 5 days



ChatGPT: Optimizing Language Models for Dialogue

We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests. ChatGPT is a sibling model to InstructGPT, which is trained to follow an instruction in a prompt and provide a detailed response.

[TRY CHATGPT](#) ↗

Let's Jump Straight Into It!

Task Summary



Task Goal

An Introduction to ChatGPT - Opportunities & Limitations



Key Takeaways

- The GPT in ChatGPT stands for Generative Pre-Trained Transformer
- ChatGPT is incredibly powerful but also has its weaknesses
- Very easy to use but needs to be handled with care
- How to get started with ChatGPT



Next Task

- Build a first simple Neural Network with the MNIST data set

Build a Neural Network with MNIST Using Keras

**Use the ChatGPT prompt to build a simple
Neural Network with the MNIST data set.**

Prompt

“Write Python code of a neural network to solve the MNIST problem using Keras using fully connected layers and categorical crossentropy as the loss function without evaluating the model on the test set”

Task Summary



Task Goal

Build a Neural Network with MNIST Using Keras



Key Takeaways

- You know how you can use ChatGPT's prompt
- How to utilise ChatGPT's output
- Create specific prompts to create a Neural Network
- Open your first Google Colab Notebook



Next Task

- Fine-Tune Code to Change Loss Function

Fine-Tune Code to Change Loss Function

Substitute Loss Function



Prompt

“Change the loss function to be sparse categorial crossentropy”

Task Summary



Task Goal

Fine-Tune Code to Change Loss Function



Key Takeaways

- What is a loss function and how do they differ
- How to substitute the existing loss function by using the ChatGPT prompt



Next Task

- Update to Use Convolutional Layer

Update to Use Convolutional Layer

Substitute Fully Connected Layer with
Convolutional Layer



Prompt

“Use convolutional Layers”

```
34 epochs = 20
--> 35 history = model.fit(x_train, y_train,
36             batch_size=batch_size,
37             epochs=epochs,
38
39             ↓ 1 frames ↓
40 /usr/local/lib/python3.8/dist-packages/keras/engine/training.py in tf_train_function(iterator)
41     13         try:
42             14             do_return = True
43 --> 15             retval_ = ag__.converted_call(ag__.ld(step_function), (ag__.ld(self), ag__.ld(iterator)), None, fscope)
44             16         except:
45             17             do_return = False
46
47 ValueError: in user code:
48
49 File "/usr/local/lib/python3.8/dist-packages/keras/engine/training.py", line 1249, in train_function  *
50     return step_function(self, iterator)
51 File "/usr/local/lib/python3.8/dist-packages/keras/engine/training.py", line 1233, in step_function  **
52     outputs = model.distribute_strategy.run(run_step, args=(data,))
53 File "/usr/local/lib/python3.8/dist-packages/keras/engine/training.py", line 1222, in run_step  **
54     outputs = model.train_step(data)
55 File "/usr/local/lib/python3.8/dist-packages/keras/engine/training.py", line 1024, in train_step
56     loss = self.compute_loss(x, y, y_pred, sample_weight)
57 File "/usr/local/lib/python3.8/dist-packages/keras/engine/training.py", line 1082, in compute_loss
58     return self.compiled_loss(
59 File "/usr/local/lib/python3.8/dist-packages/keras/engine/compile_utils.py", line 265, in __call__
60     loss_value = loss_obj(y_t, y_p, sample_weight=sw)
61 File "/usr/local/lib/python3.8/dist-packages/keras/losses.py", line 152, in __call__
62     losses = call_fn(y_true, y_pred)
63 File "/usr/local/lib/python3.8/dist-packages/keras/losses.py", line 284, in call  **
64     return ag_fn(y_true, y_pred, **self._fn_kwargs)
65 File "/usr/local/lib/python3.8/dist-packages/keras/losses.py", line 2004, in categorical_crossentropy
66     return backend.categorical_crossentropy(
67 File "/usr/local/lib/python3.8/dist-packages/keras/backend.py", line 5532, in categorical_crossentropy
68     target.shape.assert_is_compatible_with(output.shape)
69
70 ValueError: Shapes (None, 1) and (None, 10) are incompatible
```

Prompt to Fix Error

“ValueError: Shapes (None, 1) and (None, 10) are incompatible - fix that”

Task Summary



Task Goal

Update to Use Convolutional Layer



Key Takeaways

- You learned how to change the type of neural network
- How to troubleshoot errors using ChatGPT
- You used ChatGPT and use fixed code
- ChatGPT is powerful but still might suggest outdated code snippets!



Next Task

- Split Training & Test Set

Evaluate Model & Create Validation Set

Evaluate model performance & accuracy and create validation set to keep track of overfitting.



Prompt

“Evaluate the model using its test set”

“Use 20% of the training set as a validation set”

Task Summary



Task Goal

Split Training & Test Set



Key Takeaways

- You added a function to evaluate the models performance
- A validation set has been created to keep track of overfitting problems



Next Task

- Plot Losses & Random Images

Plot Training & Validation Loss & Random Images

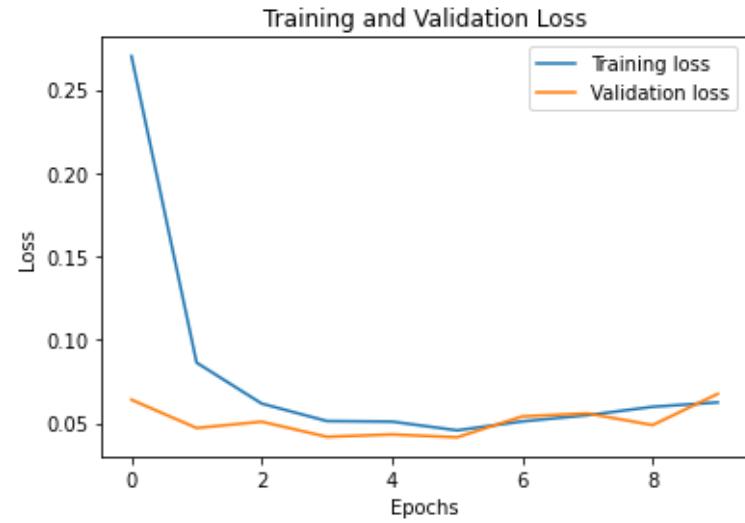
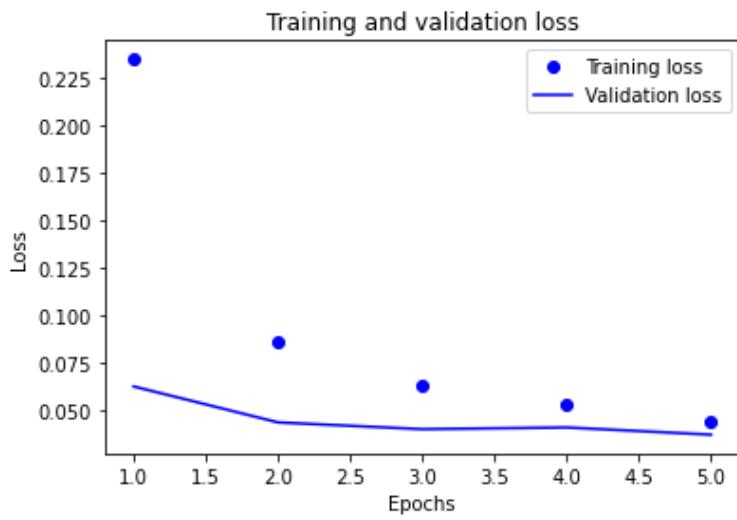
Plot Losses using Matplotlib and random images to check accuracy visually



Prompt

“plot the training and test loss using matplotlib”

“plot 20 random images with label from the training set and see if the model predicts the images correct by color coding wrong predictions with red and right predictions with green”



True: 2.0, Pred: 2



True: 0.0, Pred: 0



True: 2.0, Pred: 2



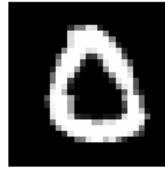
True: 6.0, Pred: 6



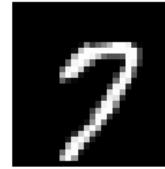
True: 8.0, Pred: 8



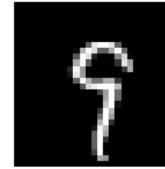
True: 0.0, Pred: 0



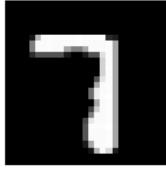
True: 7.0, Pred: 7



True: 9.0, Pred: 9



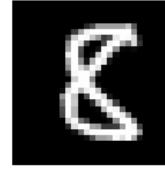
True: 7.0, Pred: 7



True: 6.0, Pred: 6



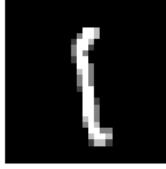
True: 8.0, Pred: 8



True: 4.0, Pred: 4



True: 1.0, Pred: 1



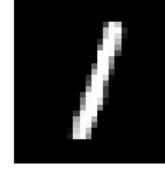
True: 5.0, Pred: 5



True: 7.0, Pred: 7



True: 1.0, Pred: 1



True: 7.0, Pred: 7



True: 5.0, Pred: 5



True: 0.0, Pred: 0



True: 6.0, Pred: 6



Task Summary



Task Goal

Plot Training & Test Loss & Random Images



Key Takeaways

- Create a plot using ChatGPT
- You learned how to write longer prompts to plot random numbers and visualise the accuracy of the network



Next Task

- Capstone Project

GUIDED PROJECT

Cumulative Challenge



**This challenge is optional and ungraded.
The goal is to build your confidence.**

Scenario/ Challenge

**Adapt the existing model with learned knowledge about ChatGPT
& share results with the community.**



Your Task

The goal of the Capstone project is to add/perform the following tasks:

1. Plot structure of the model
2. Change everything inside of Google Colab
3. Run code & share results with your community
4. Add a confusion matrix