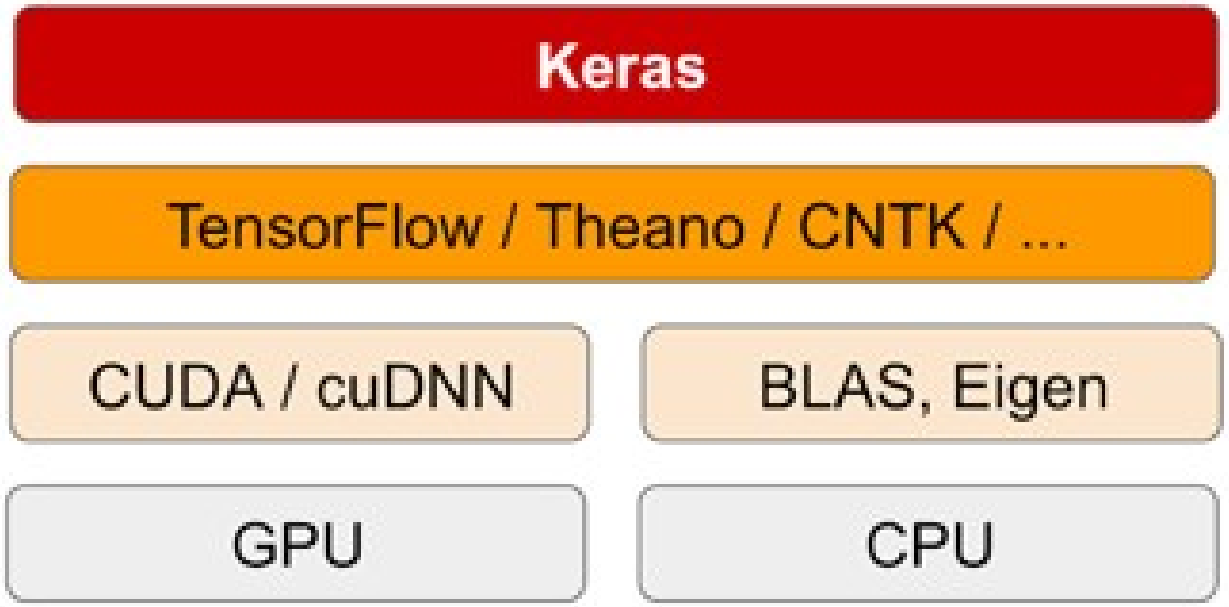
Introduction to Keras

Prof. Kuan-Ting Lai
TA Alan Tian
2020/3/17

Keras (keras.io)

- Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano
- Developed by Francois Chollet
- Officially s

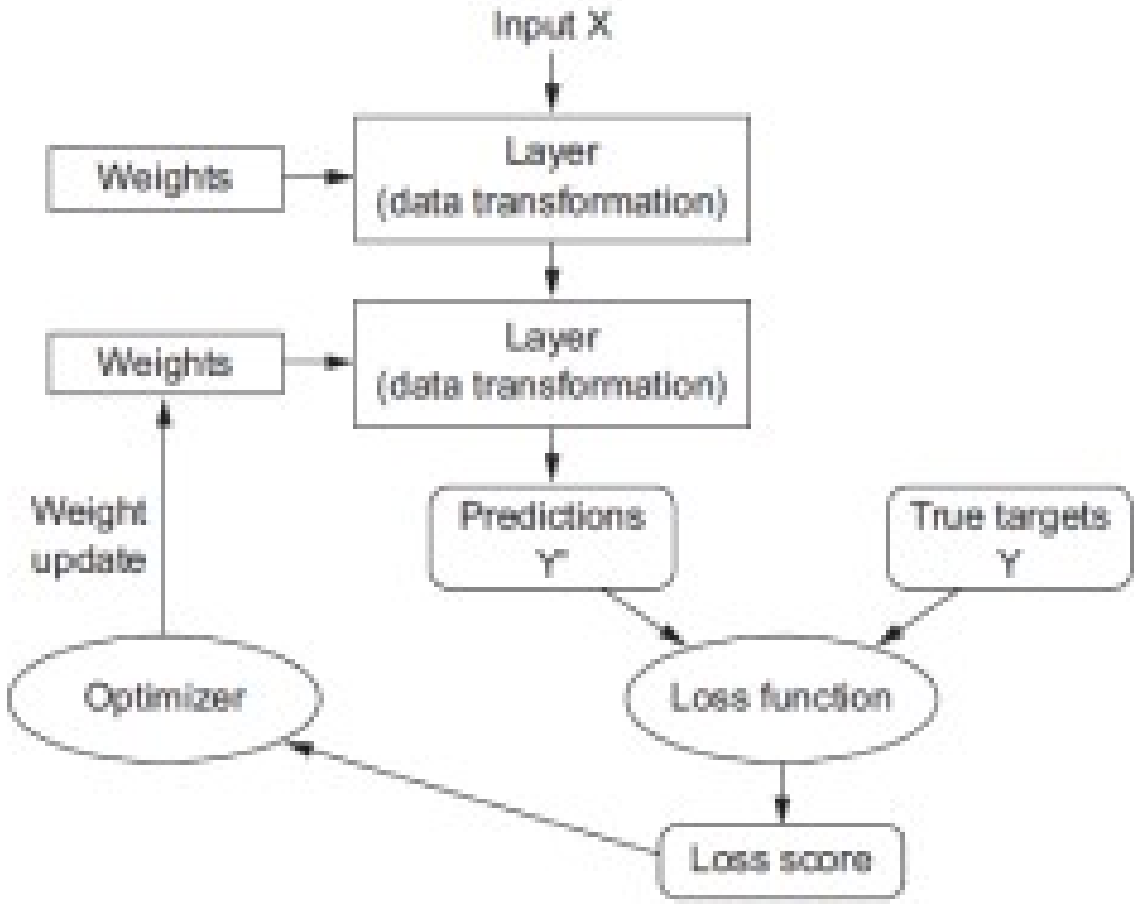


Migrating TensorFlow 1 code to TensorFlow 2

- https://www.tensorflow.org/guide/migrate
- Running 1.X unmodified
  - import tensorflow.compat.v1 as tf
  - tf.disable_v2_behavior()
- Running Keras code
  - Change package “keras” to “tensorflow.keras”
- On Colab
  - Add magic %tensorflow_version 1.x magic

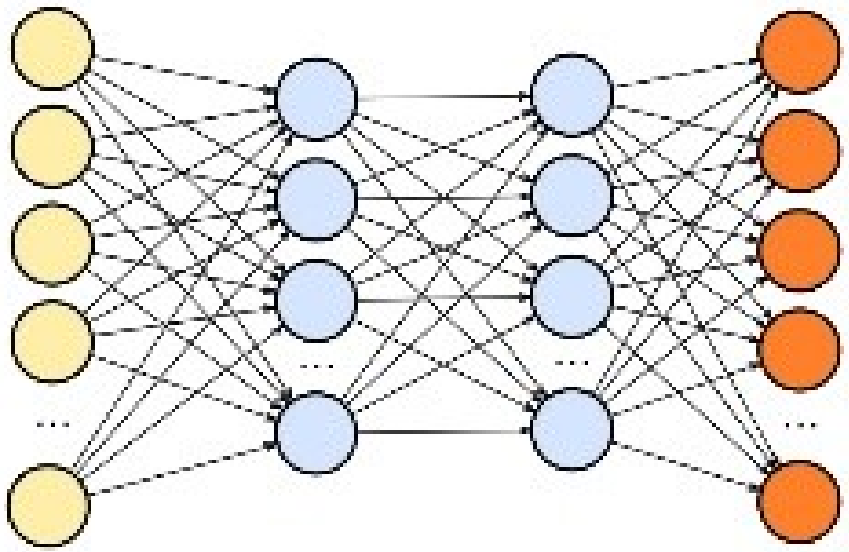
Terminologies of a Neural Network

- Weights
- Layers
- Loss function
- Optimizer



Build Your Own Networks with Keras

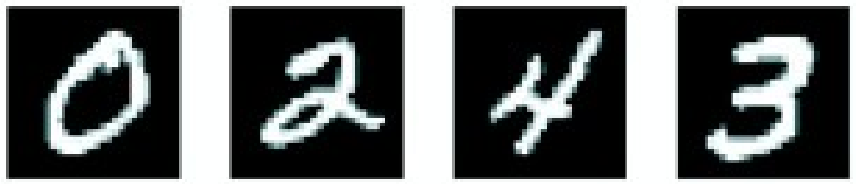
- Doing Deep learning with Keras is like playing LEGO



=

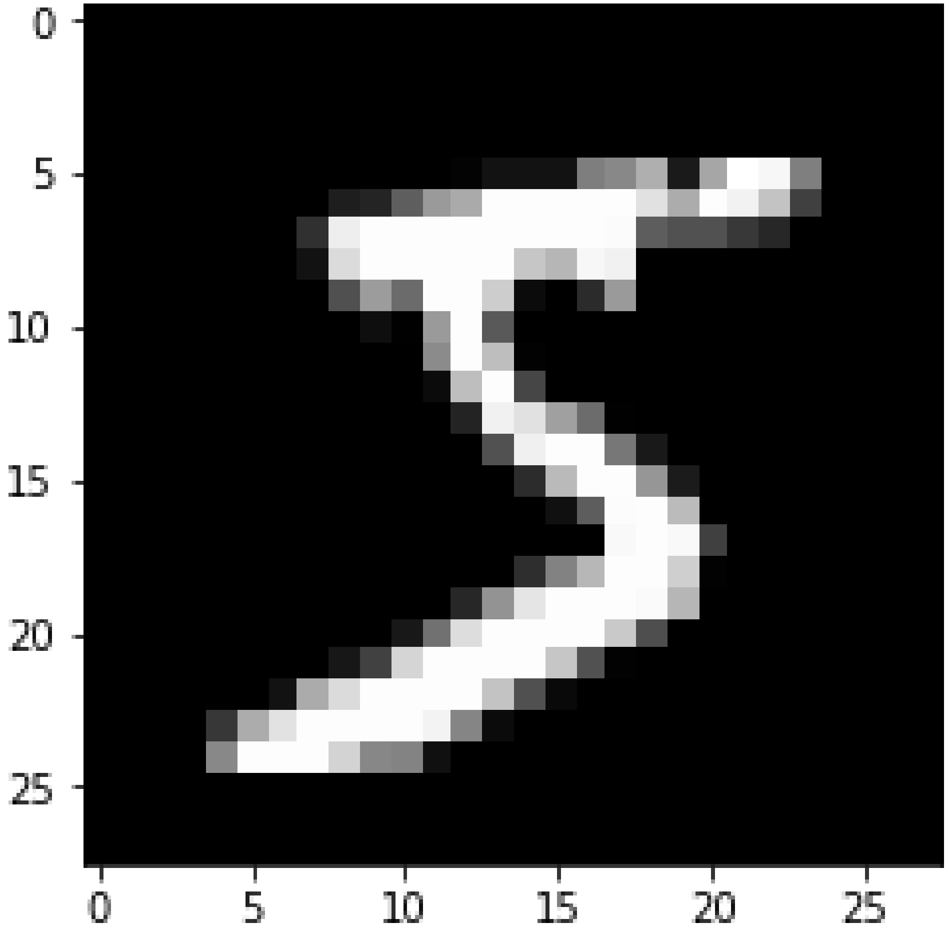
Hello Deep Learning

- Task: classify grayscale images of handwritten digits (28 × 28 pixels) into their 10 categories (0 ~ 9)
- Use the MNIST dataset created by Yann LeCun
- MNIST has 60,000 training and 10,000 test images

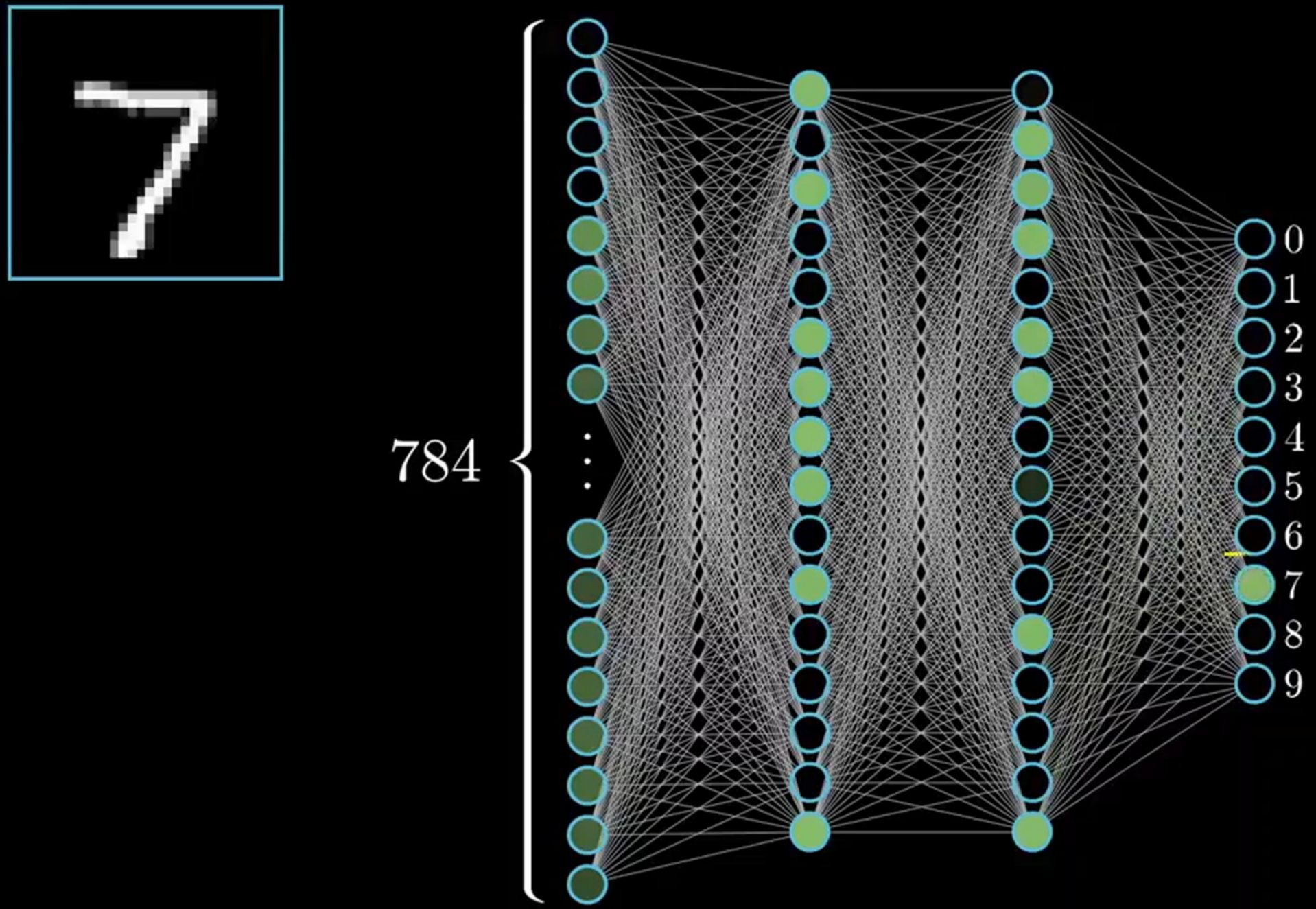


* Example code on Colab

Digital Images



```
array([
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 18, 18, 18, 126, 136, 175, 26, 166, 255, 247,
127, 0, 0, 0, 0], [ 0, 0, 0, 0, 0, 0, 0, 0, 30, 36, 94, 154, 170, 253, 253, 253, 253,
253, 225, 172, 253, 242, 195, 64, 0, 0, 0, 0], [ 0, 0, 0, 0, 0, 0, 0, 49, 238, 253,
253, 253, 253, 253, 253, 253, 253, 251, 93, 82, 82, 56, 39, 0, 0, 0, 0, 0], [ 0, 0, 0,
0, 0, 0, 0, 18, 219, 253, 253, 253, 253, 253, 198, 182, 247, 241, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0], [ 0, 0, 0, 0, 0, 0, 0, 0, 80, 156, 107, 253, 253, 205, 11, 0, 43, 154, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0], [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 14, 1, 154, 253, 90, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 139, 253, 190, 2, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 11, 190, 253,
70, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 35,
241, 225, 160, 108, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 81, 240, 253, 253, 119, 25, 0, 0, 0, 0, 0, 0, 0, 0, 0], [ 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 45, 186, 253, 253, 150, 27, 0, 0, 0, 0, 0, 0, 0, 0], [ 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 16, 93, 252, 253, 187, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 249, 253, 249, 64, 0, 0, 0, 0, 0,
0, 0], [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 46, 130, 183, 253, 253, 207, 2, 0,
0, 0, 0, 0, 0, 0], [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 39, 148, 229, 253, 253, 253,
250, 182, 0, 0, 0, 0, 0, 0, 0, 0], [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 24, 114, 221, 253,
253, 253, 253, 201, 78, 0, 0, 0, 0, 0, 0, 0, 0, 0], [ 0, 0, 0, 0, 0, 0, 0, 0, 23, 66,
213, 253, 253, 253, 253, 198, 81, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [ 0, 0, 0, 0, 0,
0, 18, 171, 219, 253, 253, 253, 253, 195, 80, 9, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
[ 0, 0, 0, 0, 55, 172, 226, 253, 253, 253, 253, 244, 133, 11, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0], [ 0, 0, 0, 0, 136, 253, 253, 253, 212, 135, 132, 16, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]], dtype=uint8)
```


784
0
1
2
3
4
5
6
7
8
9

Preparing the data & Labels

- Download the dataset from keras API and unpack the return data

```
[54] from tensorflow import keras

     (x_train, y_train), (x_test, y_test) = keras.datasets.mnist.load_data()
```

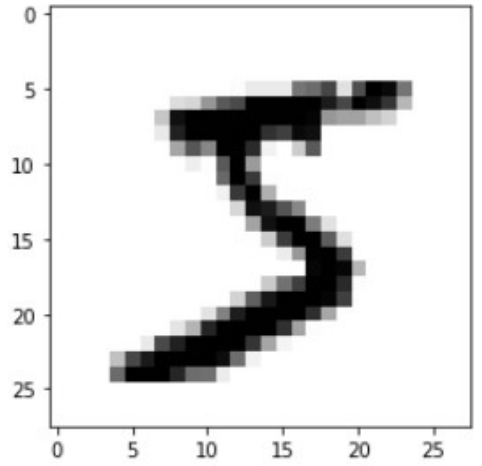
Preparing the data & Labels

- Check how the data looks like
- Print out the label of first imag

```
import numpy as np
import matplotlib.pyplot as plt
print(f"The first digit image of training data")
plt.imshow(x_train[0], cmap=plt.cm.binary)
plt.show()

print(f"The first label of training data is >>> {y_train[0]}")
```

```
The first digit image of training data
```




```
The first label of training data is >>> 5
```

The Network Architecture

- Layer: a layer in the deep network for processing data, like a filter
- Dense layer: fully connected neural layer
- Softmax layer: Output probabilities of 10 digits (0 ~ 9)

```
from tensorflow.keras import layers

# Build a simple model
inputs = keras.Input(shape=(28, 28))
x = layers.experimental.preprocessing.Rescaling(1.0 / 255)(inputs)
x = layers.Flatten()(x)
x = layers.Dense(128, activation="relu")(x)
x = layers.Dense(128, activation="relu")(x)
outputs = layers.Dense(10, activation="softmax")(x)
model = keras.Model(inputs, outputs)
model.summary()
```

Compile Your Model

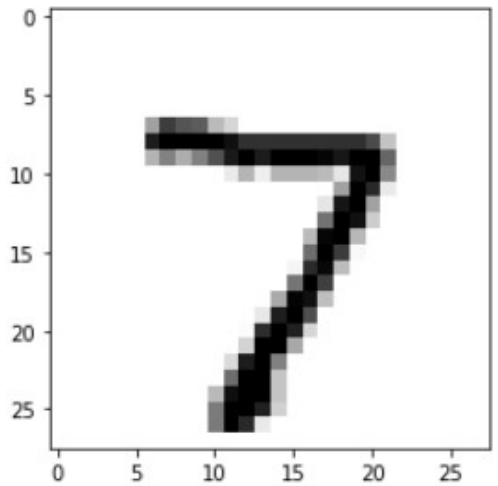
- Loss function: measure performance on training data
- Optimizer: the mechanism for updating parameters
- Metrics to evaluate the performance on test data (accuracy)

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

Predict the digit

```
# prediction
import matplotlib.pyplot as plt
plt.imshow(x_test[0], cmap=plt.cm.binary)
plt.show()
print(f"Ground truth : {y_test[0]}")

res = model.predict(x_test[0:1])
predict_class = np.argmax(res,axis=1)
print(f"Prediction : {predict_class[0]}")
```

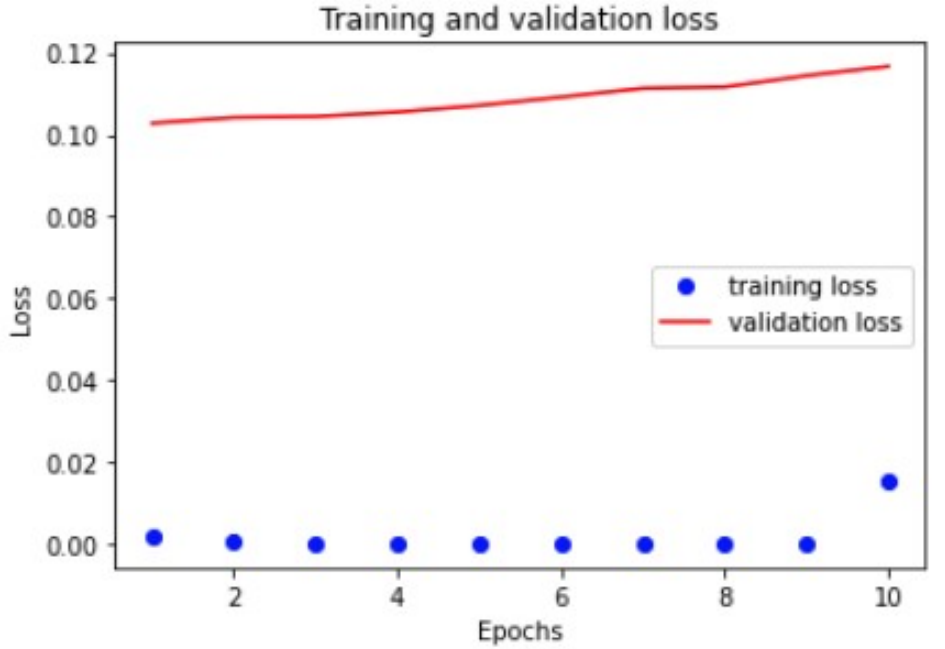


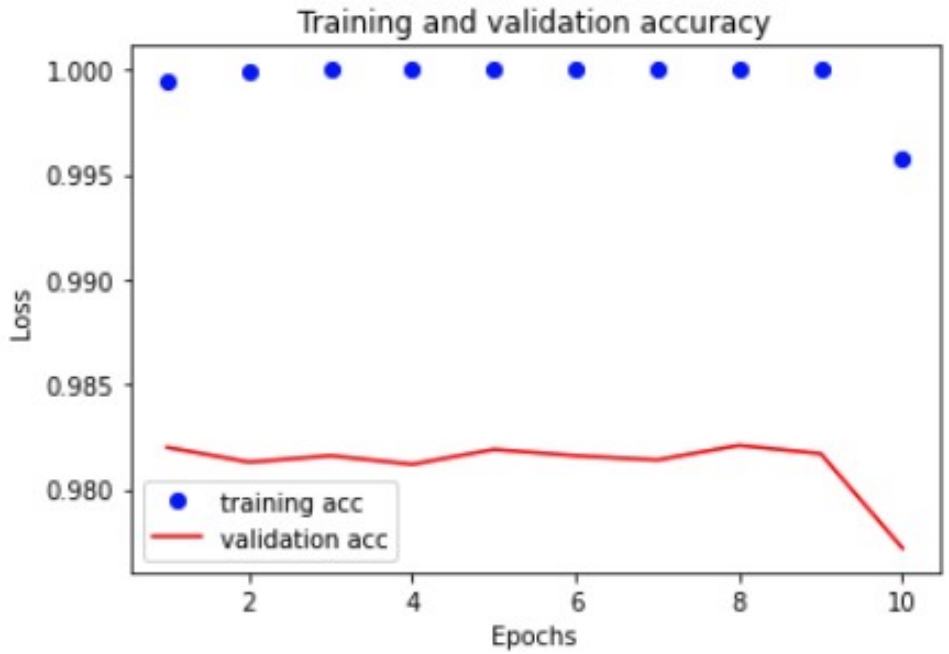

```
Ground truth : 7
Prediction : 7
```

- Plot the first image of testing data
- Use the model we trained to predict the digit

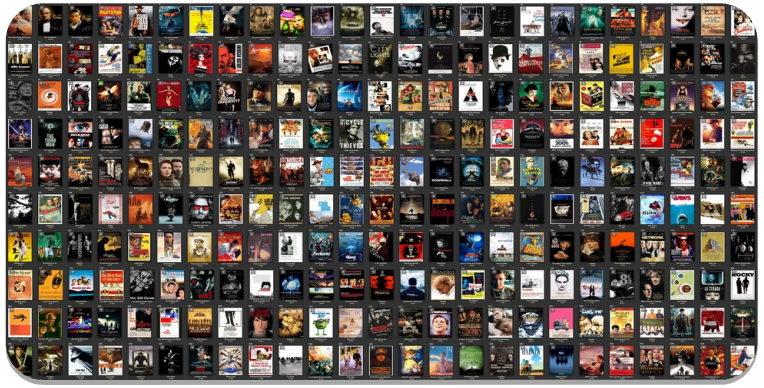
Training Loss & Accuracy

- Plot the training process of loss and accuracy
- Check if the model is overfiting





Keras Training Examples



Is the Movie Review Positive?

- Binary Classification
- 50,000 polarized reviews from IMDB

Classify Financial News

- Multi-class Classification
- 46 exclusive topics including earn, grain, crude, trade,…

Predicting Housing Price

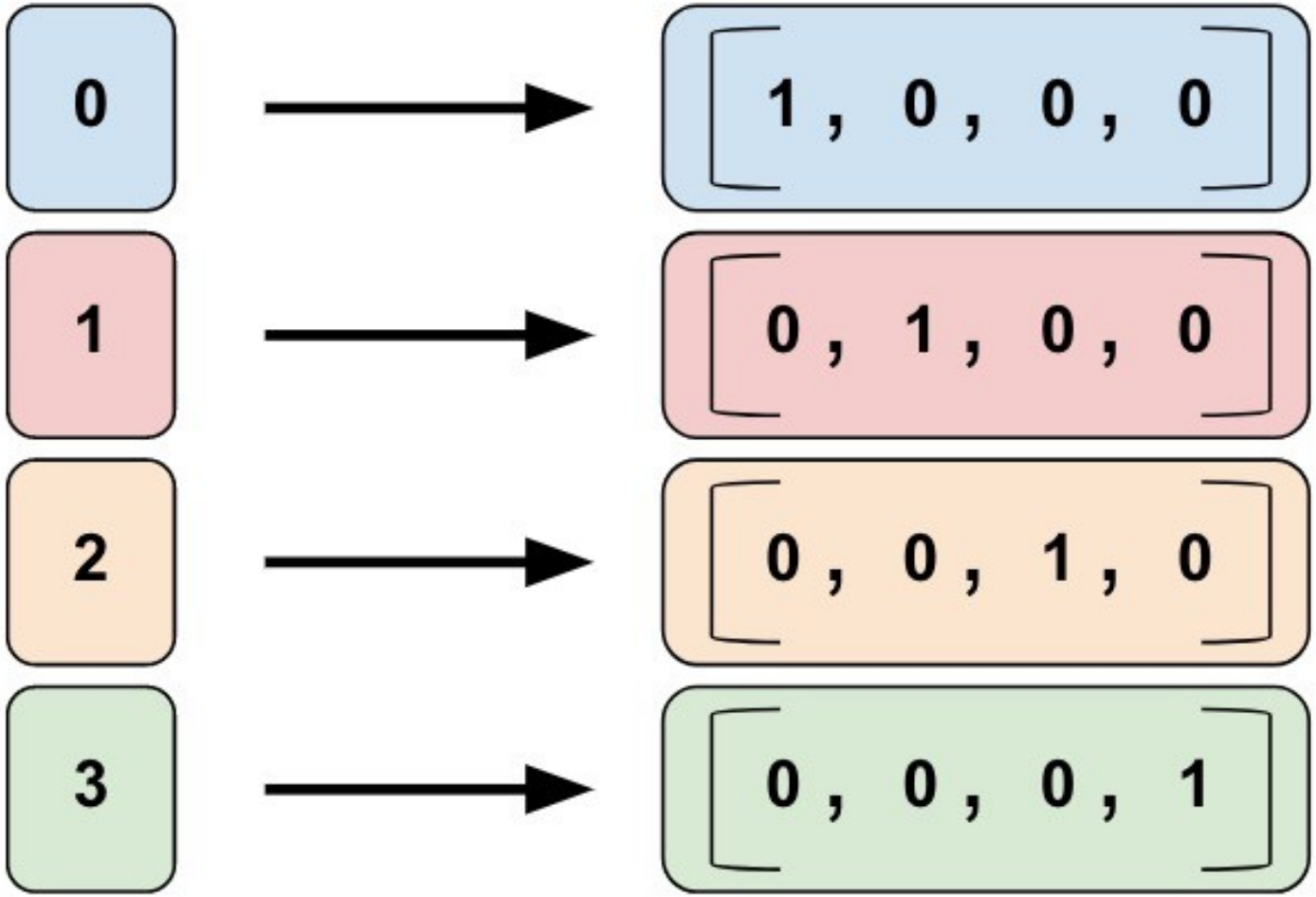
- Regression
- Use Boston housing price dataset with 506 samples and 13 features (crime rate, rooms, age, …)


Binary Classification: Check if a Movie Review is Positive or Negative
IMDB Movie reviews dataset (50,000 reviews)

IMDb Movie Review Datasets

- Internet Movie Database
- 50,000 polarized reviews (50% positive and 50% negative reviews)
- Goal
  - Predict if a review is positive or negative (binary classification)
- Practice 2 on colab
  - Use keras API to create model
  - Train the model for 20 epochs
  - plot the training history (Loss and Accuracy)
- Reference
  - Keras Official Document

One-hot Encoding



https://tensorflow.rstudio.com/tools/tfdatasets/articles/feature_columns.html

Review One-hot Encoding

i am a great fan of david lynch and have everything that he's made on dvd except for hotel room the 2 hour twin peaks movie so when i found out about this i immediately grabbed it

Dictionary Mapping with get_word_index() function