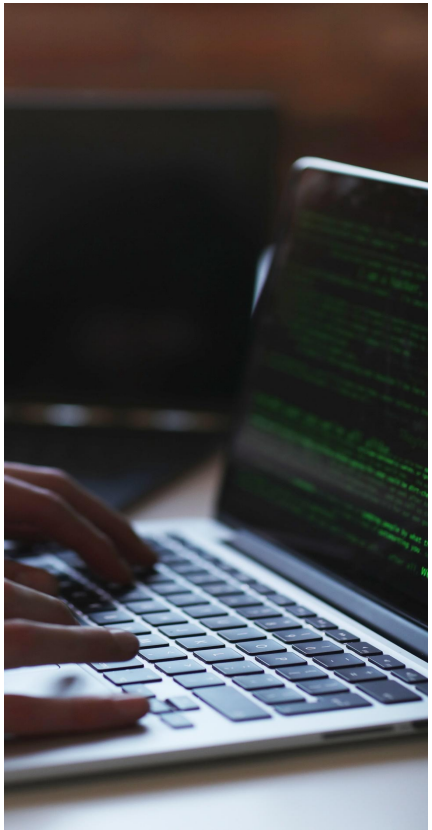
Course-Net
Empowering IT People

Introduction to Deep Learning and TensorFlow

Course-Net
Empowering IT People

#EmpoweringITPeople



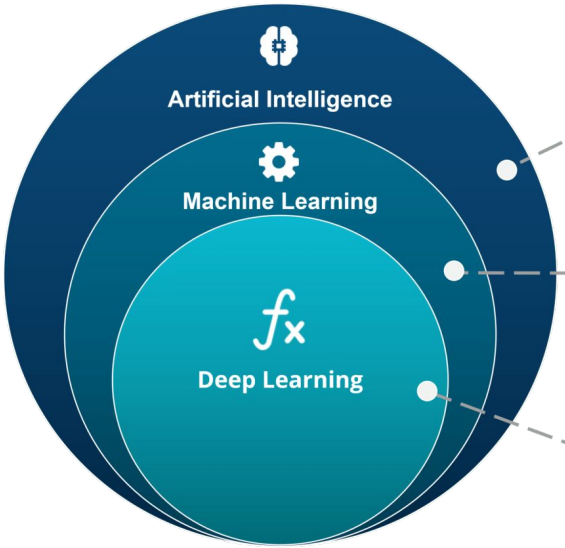
APA YANG AKAN KITA PELAJARI

Introduction to Deep Learning and TensorFlow

- What is Deep Learning?
- What is TensorFlow?
- Deep Learning with TensorFlow

What is Deep Learning?

Artificial Intelligence Taxonomy

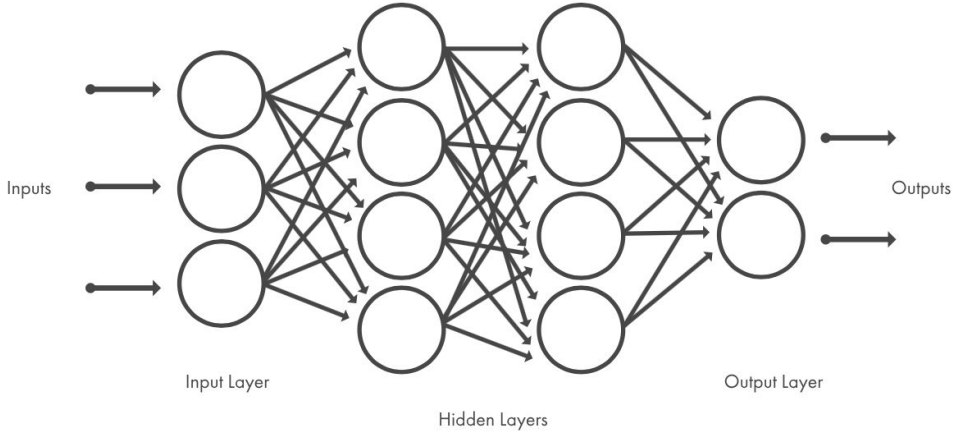


ARTIFICIAL INTELLIGENCE
A technique which enables machines to mimic human behaviour

MACHINE LEARNING
Subset of AI technique which use statistical methods to enable machines to improve with experience

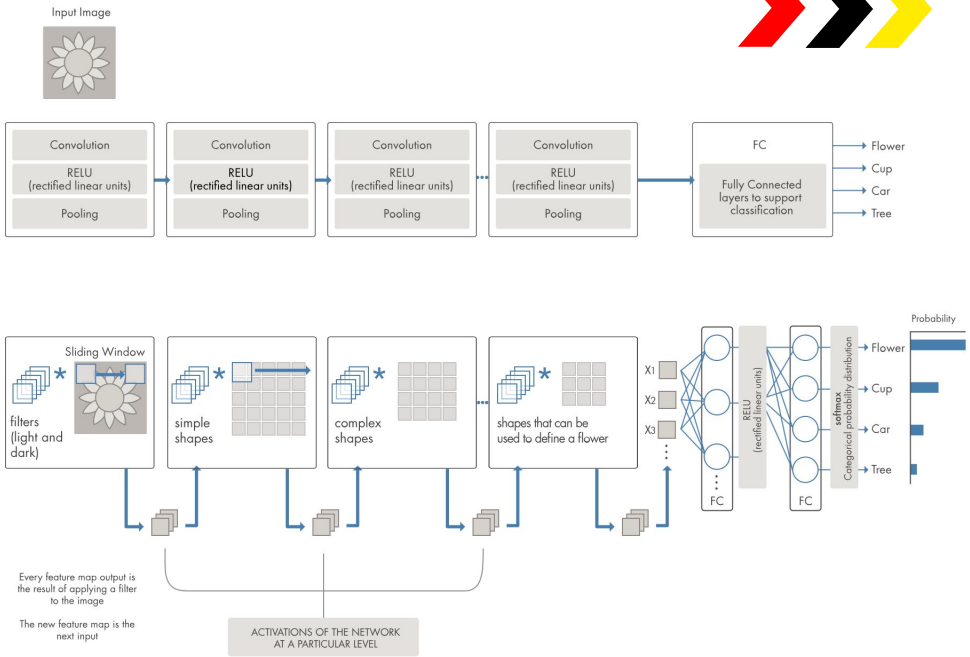
DEEP LEARNING
Subset of ML which make the computation of multi-layer neural network feasible

Neural Networks Architecture



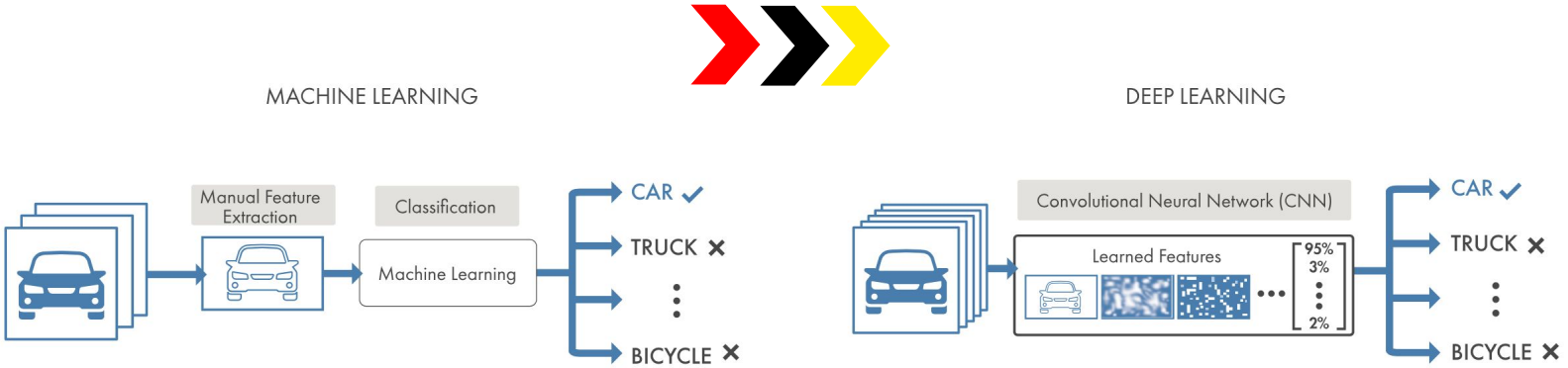
Neural Networks architecture consists of **input** layer, **hidden** layers, and **output** layer.

Deep Learning Architecture



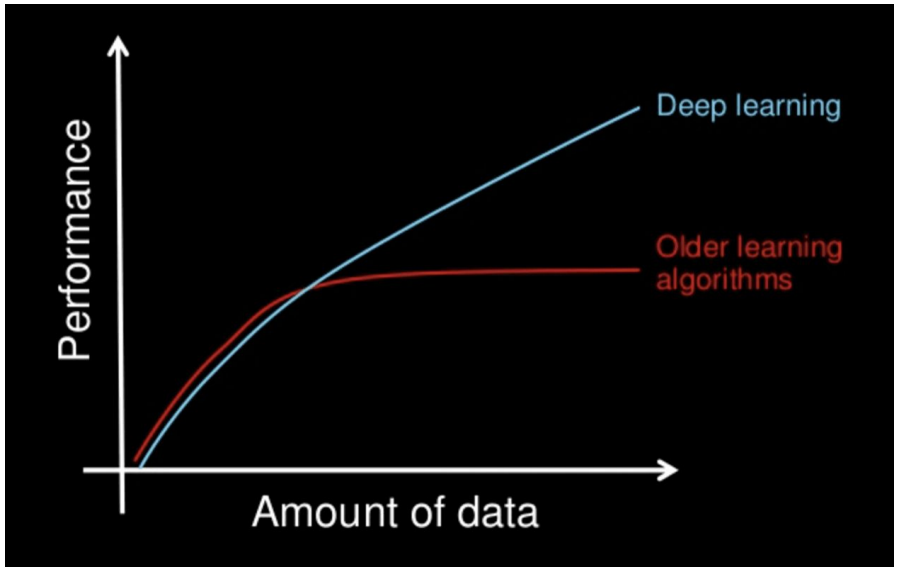
The availability of **huge dataset** and novel **learning algorithms** **are** enabling neural networks to have large number of hidden layers. This architecture is called **deep learning**.

Advantages of Deep Learning (1)



Traditionally, we need to do **features extraction** using machine learning algorithms. One advantage of using deep learning is the algorithms can learn the features (**features leaning**) for any given task.

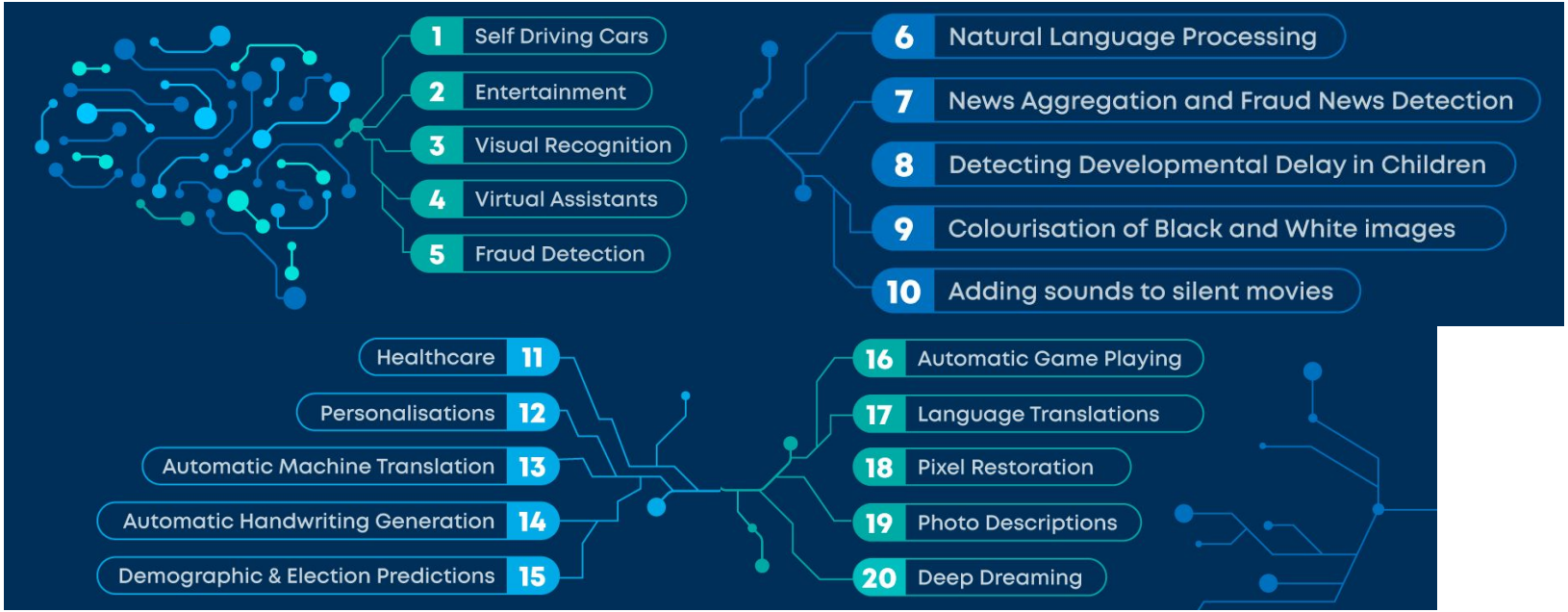
Advantages of Deep Learning (2)



For some problems with limited number of data, usually other machine learning algorithms can perform better than deep learning.

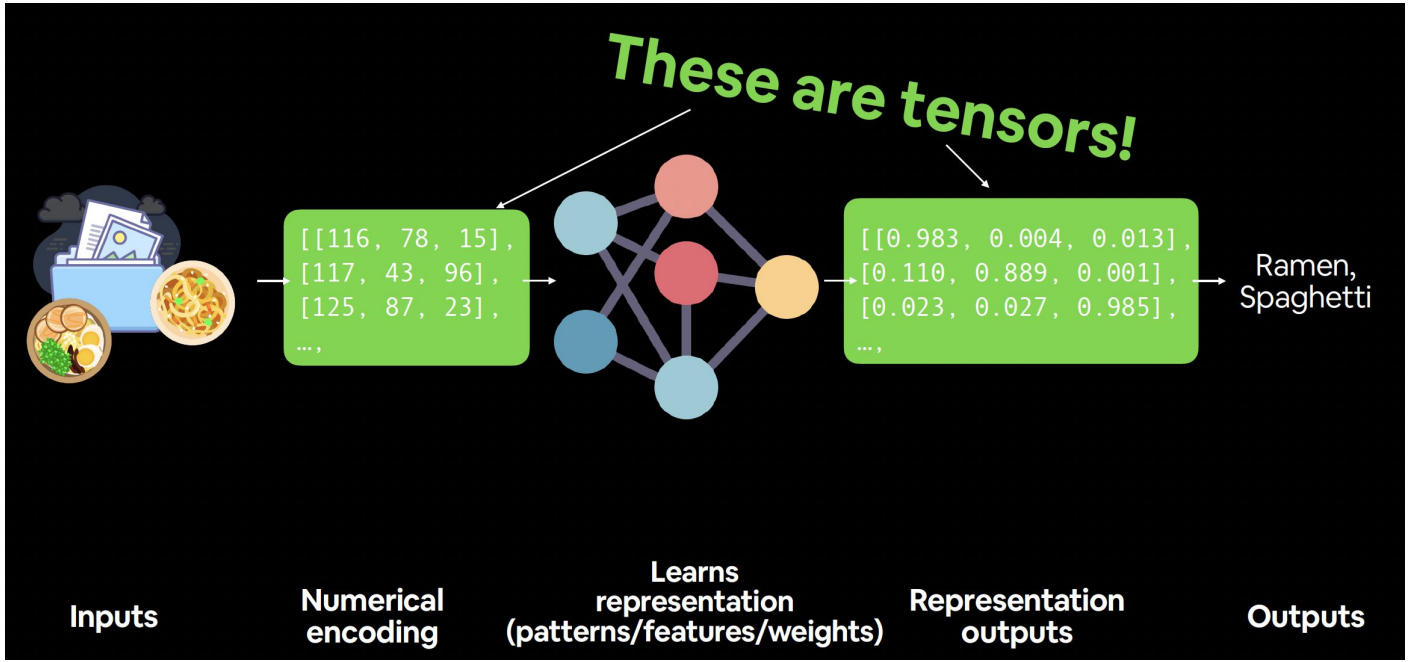
For problem with **huge amount of data** (internet scale, generated by IoT, millions of records), deep learning can be superior to others.

Deep Learning Applications



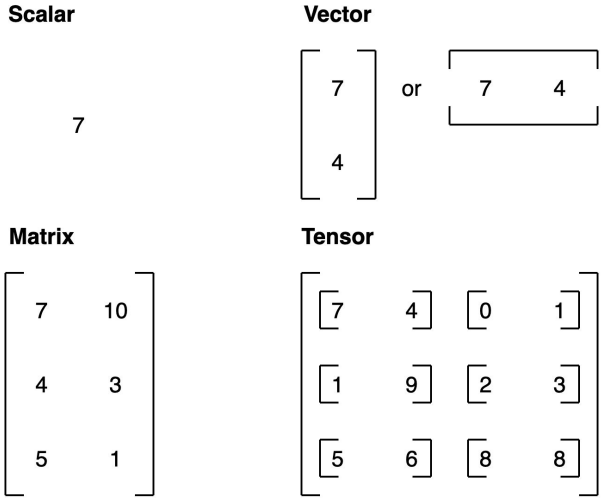
What is TensorFlow?

What is Tensors?



Tensors vs Scalar, Vector, Matrix

- **scalar**: a single number.
- **vector**: a number with direction (e.g. wind speed with direction).
- **matrix**: a 2-dimensional array of numbers.
- **tensor**: an n-dimensional array of numbers (where n can be any number, a 0-dimension tensor is a scalar, a 1-dimension tensor is a vector).



TensorFlow

- **End-to-end** platform for machine learning
- Write **fast deep learning code** in Python/other accessible languages (able to run on a GPU/TPU)
- Able to access many **pre-built deep learning models** (TensorFlow Hub)
- **Whole stack**: preprocess data, model data, deploy model in your application
- Originally designed and used in-house by Google (now **open-source**)

Why TensorFlow?

Easy model building

Build and train ML models easily using intuitive high-level APIs like Keras with eager execution, which makes for immediate model iteration and easy debugging.

Robust ML production anywhere

Easily train and deploy models in the cloud, on-prem, in the browser, or on-device no matter what language you use.

Powerful experimentation for research

A simple and flexible architecture to take new ideas from concept to code, to state-of-the-art models, and to publication faster.

Tensor Attributes

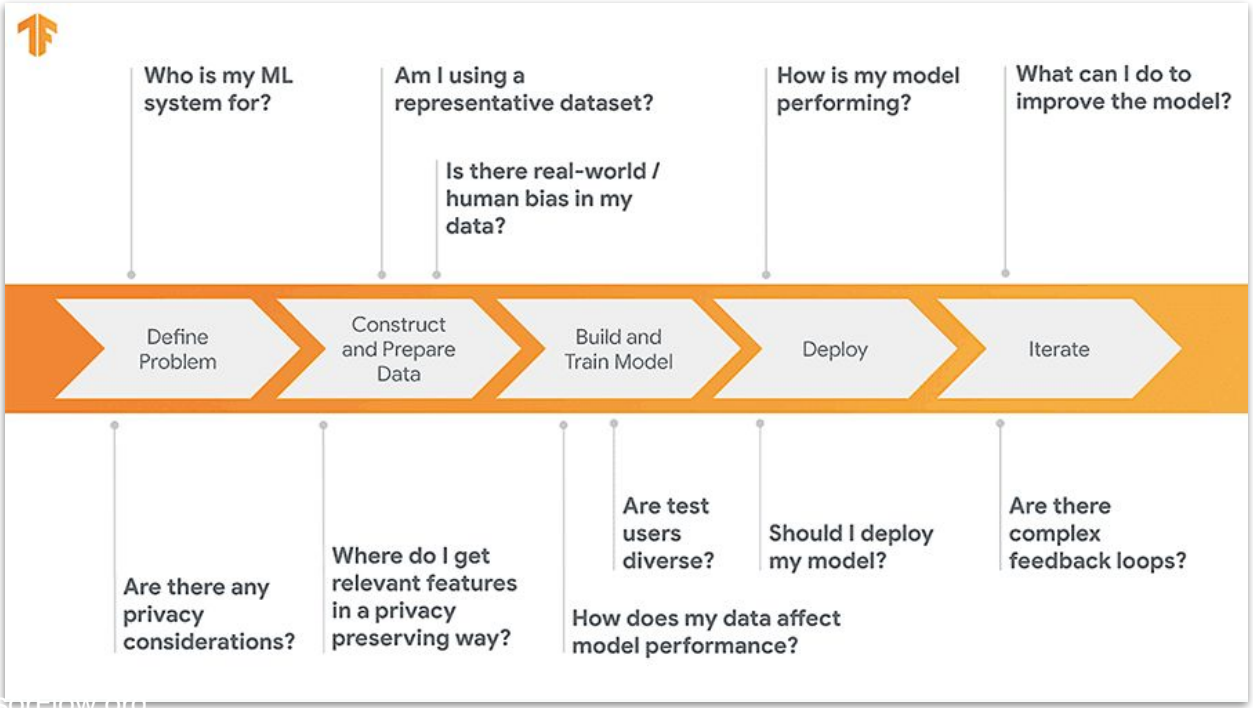
Attribute	Meaning	Code
Shape	The length (number of elements) of each of the dimensions of a tensor.	`tensor.shape`
Rank	The number of tensor dimensions. A scalar has rank 0, a vector has rank 1, a matrix is rank 2, a tensor has rank n.	`tensor.ndim`
Axis or dimension	A particular dimension of a tensor.	`tensor[0], tensor[:, 1]…`
Size	The total number of items in the tensor.	`tf.size(tensor)`

Course-Net
Empowering IT People

#EmpoweringITPeople

Deep Learning with TensorFlow

TensorFlow Workflow



Basic Steps Deep Learning with TensorFLow

- **Step 1: Import TensorFlow and related libraries**
- **Step 2: Import datasets**
- **Step 3: Preprocess data**
- **Step 4: Build a model**
- **Step 5: Compile the model**
- **Step 6: Train the model**
- **Step 7: Evaluate the model**
- **Step 8: Improve the model**

Lets try a running example!

Basic Classification: Classify images of clothing

Step 1: Import TensorFlow and related libraries

```
# TensorFlow and tf.keras
import tensorflow as tf

# Helper libraries
import numpy as np
import matplotlib.pyplot as plt

print(tf.__version__)
```

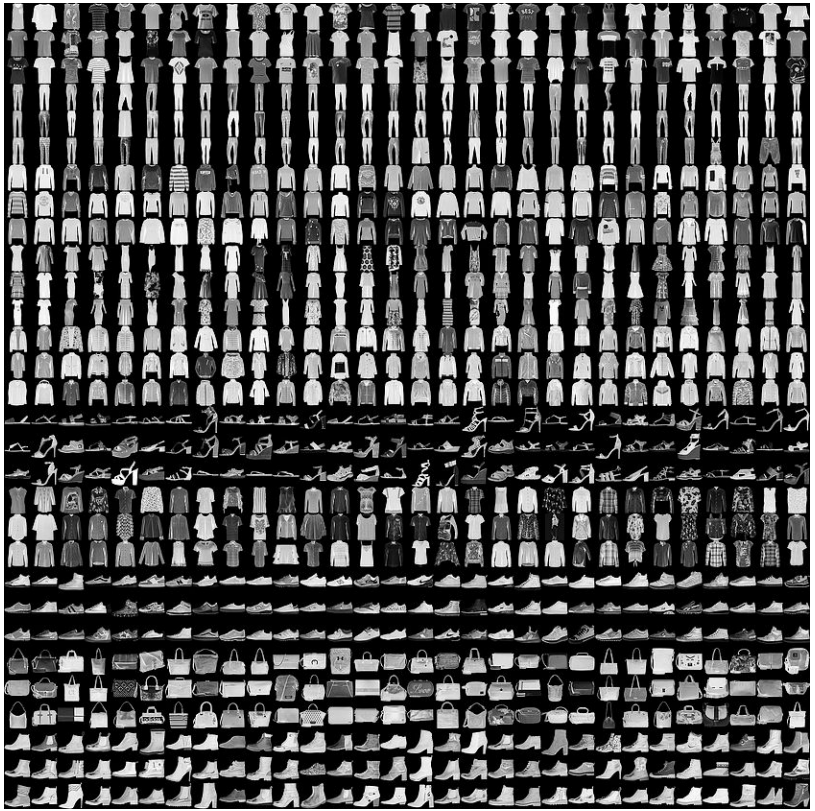
```
2.8.0
```

Step 2: Import datasets

```
fashion_mnist = tf.keras.datasets.fashion_mnist

(train_images, train_labels), (test_images, test_labels) = fashion_mnist.load_data()
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-labels-
32768/29515 [================================] - 0s 0us/step
40960/29515 [========================================] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/train-images-
26427392/26421880 [==============================] - 0s 0us/step
26435584/26421880 [==============================] - 0s 0us/step
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-labels-i
16384/5148 [===============================================================================
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/t10k-images-i
4423680/4422102 [==============================] - 0s 0us/step
4431872/4422102 [==============================] - 0s 0us/step
```



```
class_names = ['T-shirt/top', 'Trouser', 'Pullover', 'Dress', 'Coat',
               'Sandal', 'Shirt', 'Sneaker', 'Bag', 'Ankle boot']
```

Label	Class
0	T-shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot

Step 3: Preprocess data

```
train_images = train_images / 255.0

test_images = test_images / 255.0
```

- **Scale** these values to **a range of 0 to 1** before feeding them to the neural network model. To do so, **divide the values by 255**.
- It's important that the *training set* and the *testing set* be **preprocessed in the same way**.
- Different problems and datasets may have **different preprocessing methods**.

Step 4: Build a model

```
model = tf.keras.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(10)
])
```

Step 5: Compile the model

```
model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])
```

Step 6: Train the model

```
model.fit(train_images, train_labels, epochs=10)
```

```
1875/1875 [==============================] - 3s 2ms/step - loss: 0.3124 - accuracy: 0.8858
Epoch 5/10
1875/1875 [==============================] - 3s 2ms/step - loss: 0.2949 - accuracy: 0.8913
Epoch 6/10
1875/1875 [==============================] - 3s 2ms/step - loss: 0.2776 - accuracy: 0.8977
Epoch 7/10
1875/1875 [==============================] - 3s 2ms/step - loss: 0.2669 - accuracy: 0.9022
Epoch 8/10
1875/1875 [==============================] - 3s 2ms/step - loss: 0.2552 - accuracy: 0.9046
Epoch 9/10
1875/1875 [==============================] - 3s 2ms/step - loss: 0.2463 - accuracy: 0.9089
Epoch 10/10
1875/1875 [==============================] - 3s 2ms/step - loss: 0.2376 - accuracy: 0.9117
<keras.callbacks.History at 0x7f5f2c785110>
```

Step 7: Evaluate the model

```
test_loss, test_acc = model.evaluate(test_images,  test_labels, verbose=2)

print('\nTest accuracy:', test_acc)
```

```
313/313 - 1s - loss: 0.3176 - accuracy: 0.8895 - 553ms/epoch - 2ms/step

Test accuracy: 0.8895000219345093
```

Step 8: Improve the model (common ways)

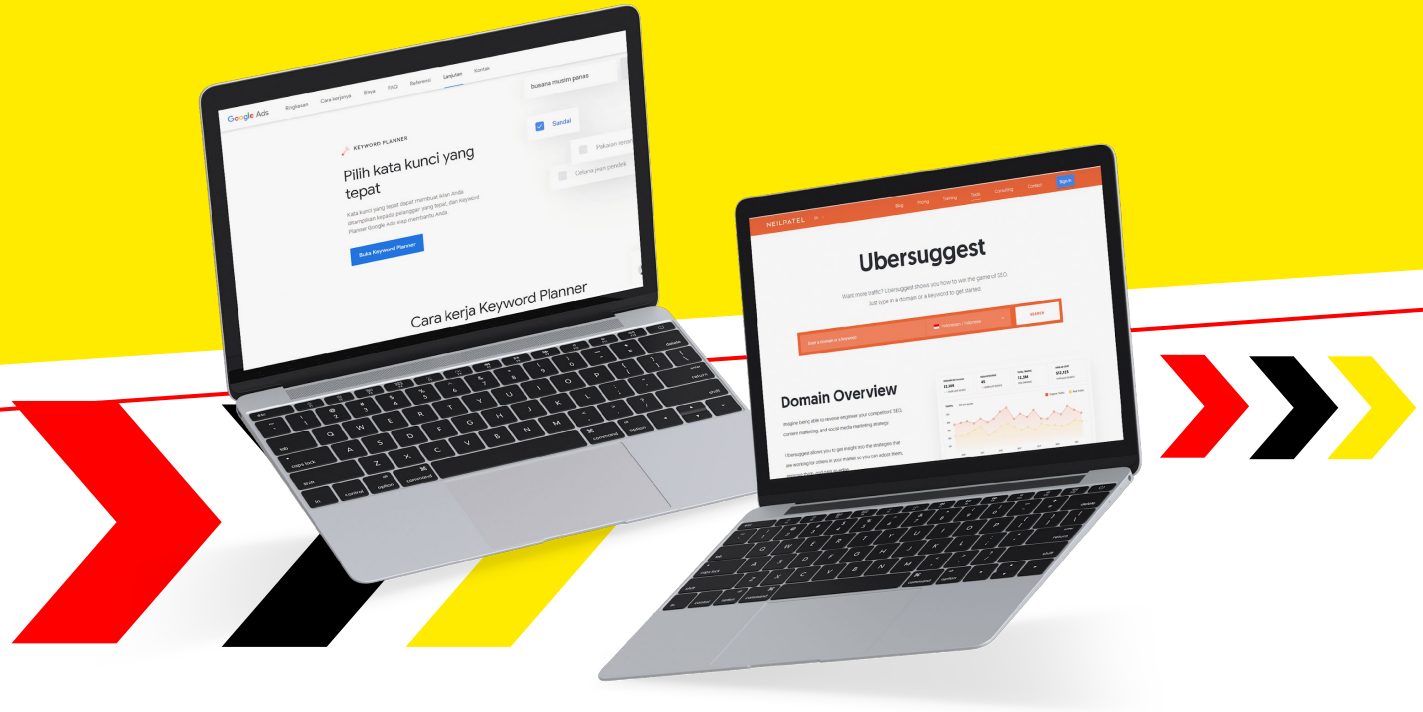
- Adding layers
- Increase the number of hidden units
- Change the activation functions
- Change the optimization function
- Change the learning rate
- Fitting on more data
- Fitting longer (more epochs)

References and Reading Materials

- AI vs Machine Learning vs Deep Learning by Atul. https://www.edureka.co/blog/ai-vs-machine-learning-vs-deep-learning/
- What is Deep Learning? by Jason Brownlee. https://machinelearningmastery.com/what-is-deep-learning/
- What Is Deep Learning? https://www.mathworks.com/discovery/deep-learning.html
- Top 20 Applications of Deep Learning in 2022 Across Industries by Marina Chatterjee. https://www.mygreatlearning.com/blog/deep-learning-applications/
- Basic classification: Classify images of clothing. https://www.tensorflow.org/tutorials/keras/classification
- Zero to Mastery Deep Learning with TensorFlow by Andrei Neagoie and Daniel Bourke. https://github.com/mrdbourke/tensorflow-deep-learning

Course-Net
Empowering IT People

#EmpoweringITPeople



YUK LATIHAN TENSORFLOW!