

Mì "Python"

Mì AI Training

Bài số 04



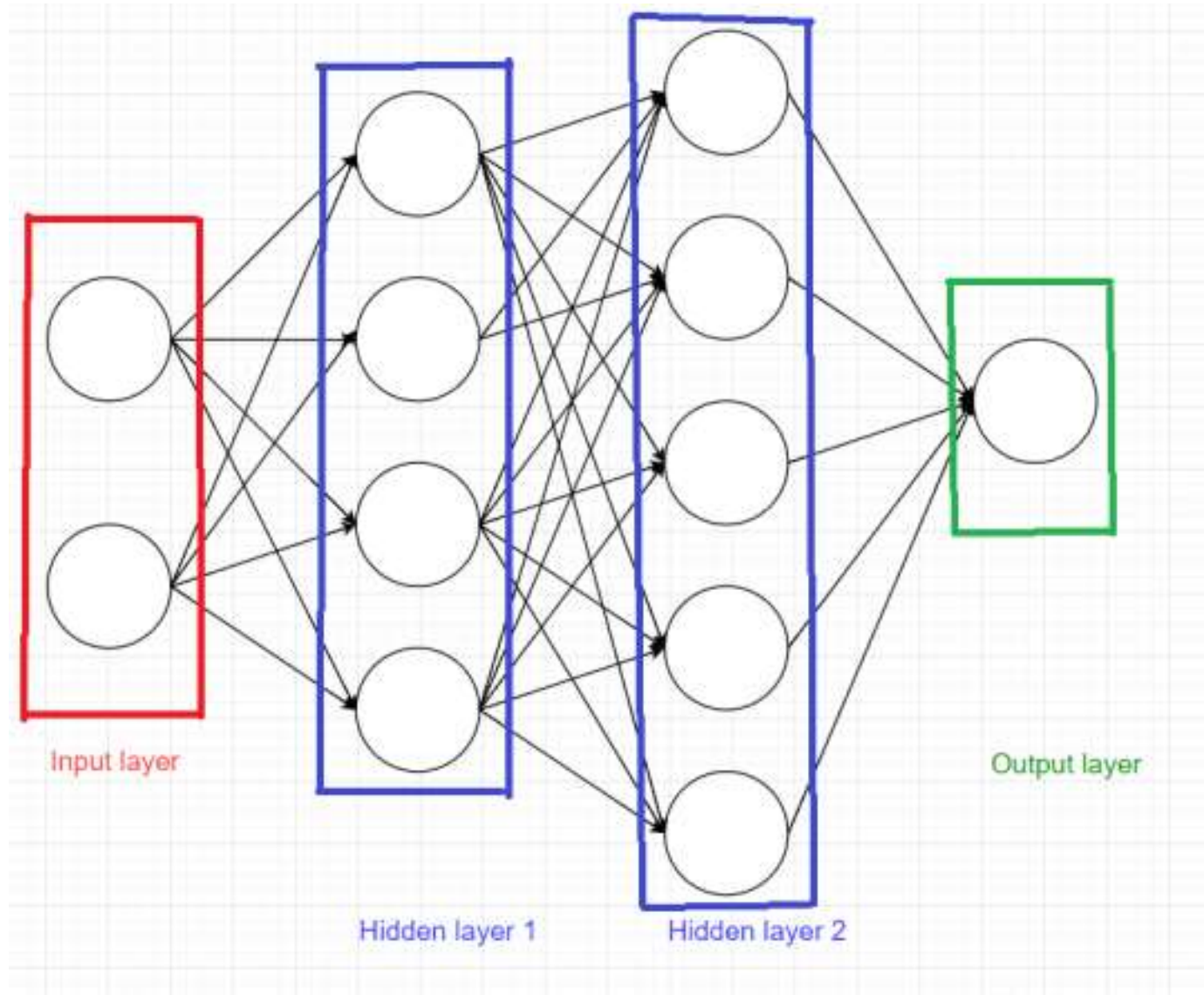
Nội dung khóa học

- Bài 1. Python **cơ bản** A
- Bài 2. Python **cơ bản** B
- Bài 3. Python **với** OpenCV
- Bài 4. Python **với** Keras
- Bài 5. Python **với** Keras 2
- Bài 6. Python **với** Pandas
- Bài 7. Xây **dựng** Backend Server **với** Python

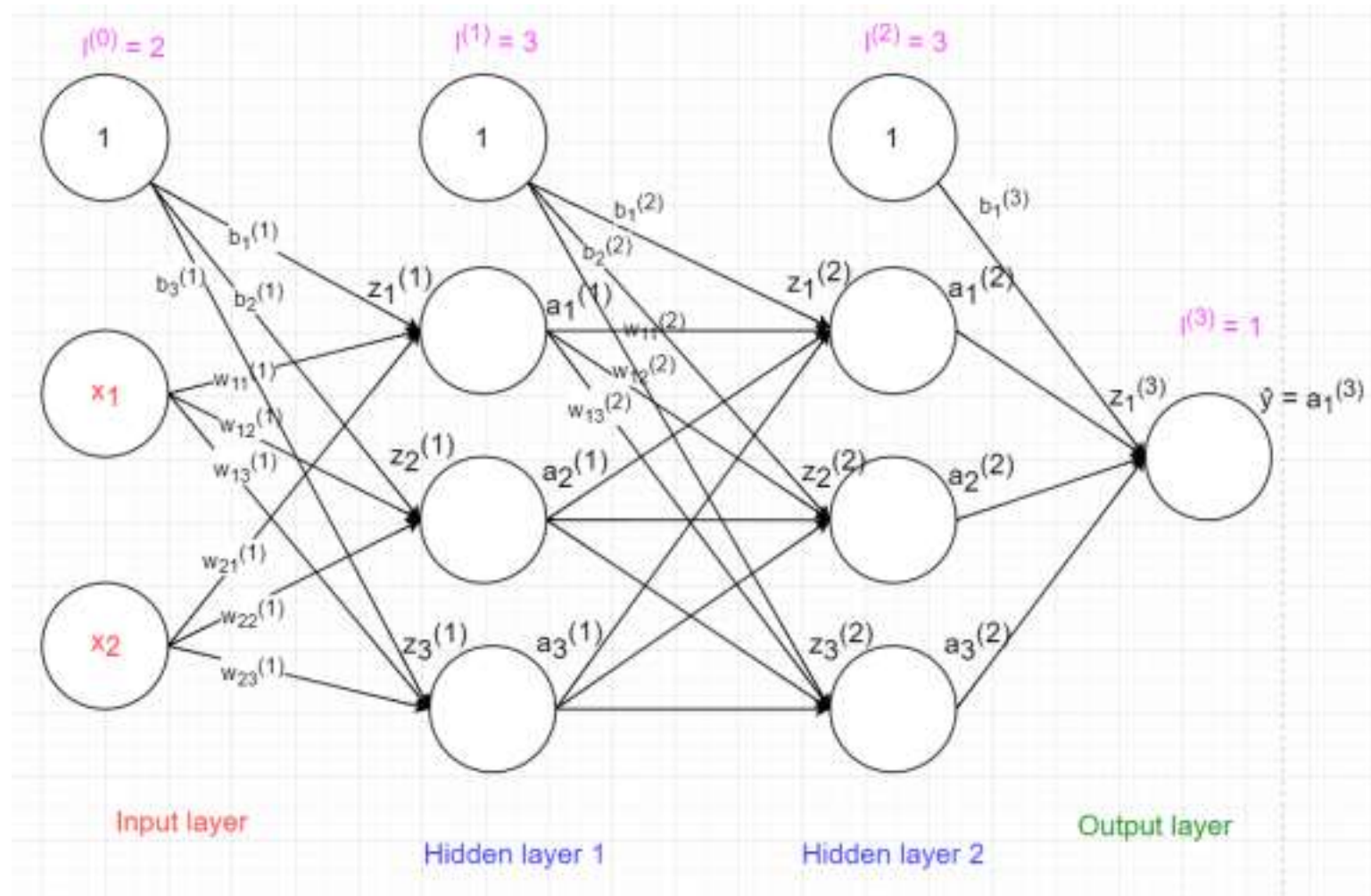
Bài 4

- Tensorflow và Keras là gì?
- Mạng Neural là gì?
- Các khái niệm trong Keras: model, layers, loss, optimizer..
- Các bước để xây dựng một mạng NN cơ bản bằng Keras

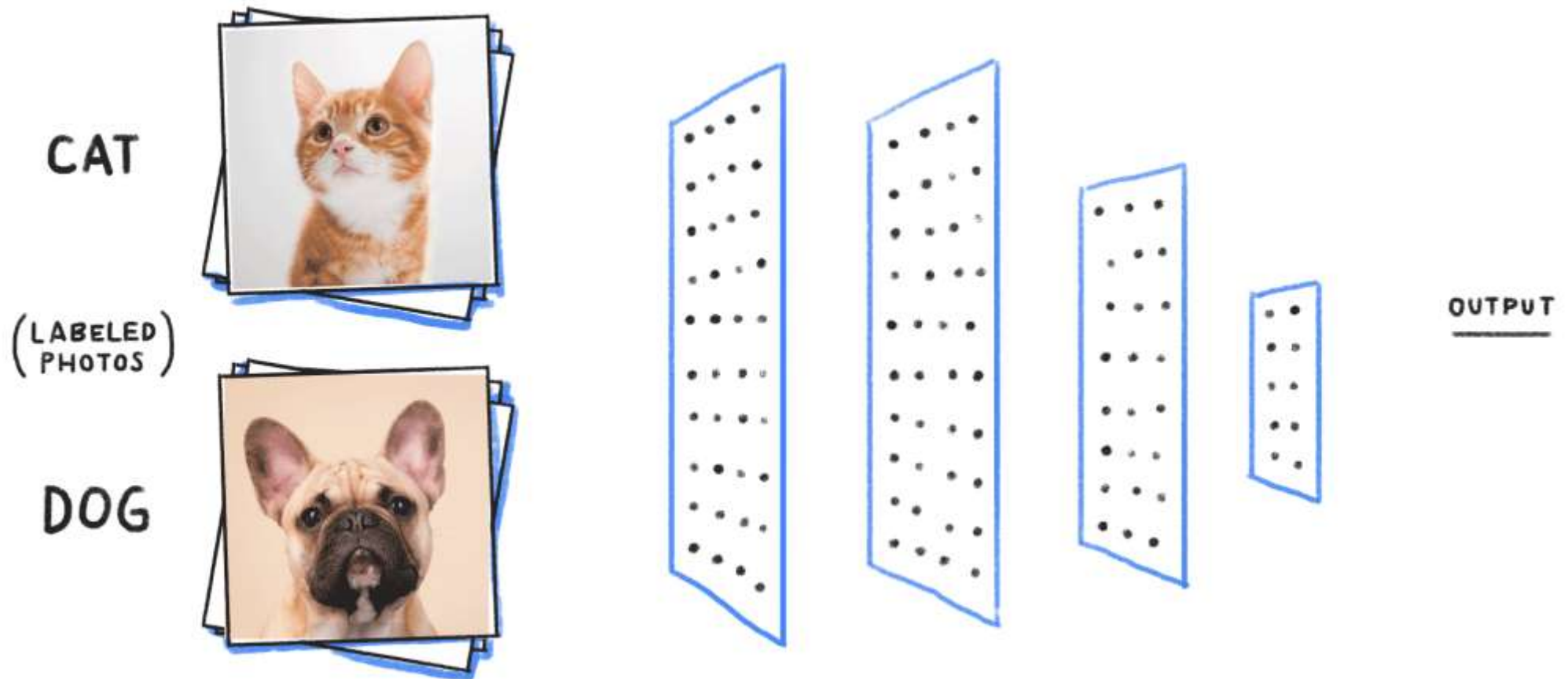
Mạng Neural là gì?



Mạng Neural là gì?



Mạng Neural là gì?



Tensorflow là gì?



- Tensorflow là một thư viện phần mềm mã nguồn mở hỗ trợ mạnh mẽ các phép toán học để tính toán trong machine learning và deep learning.
- Được phát triển bởi Google

Tensorflow là gì?



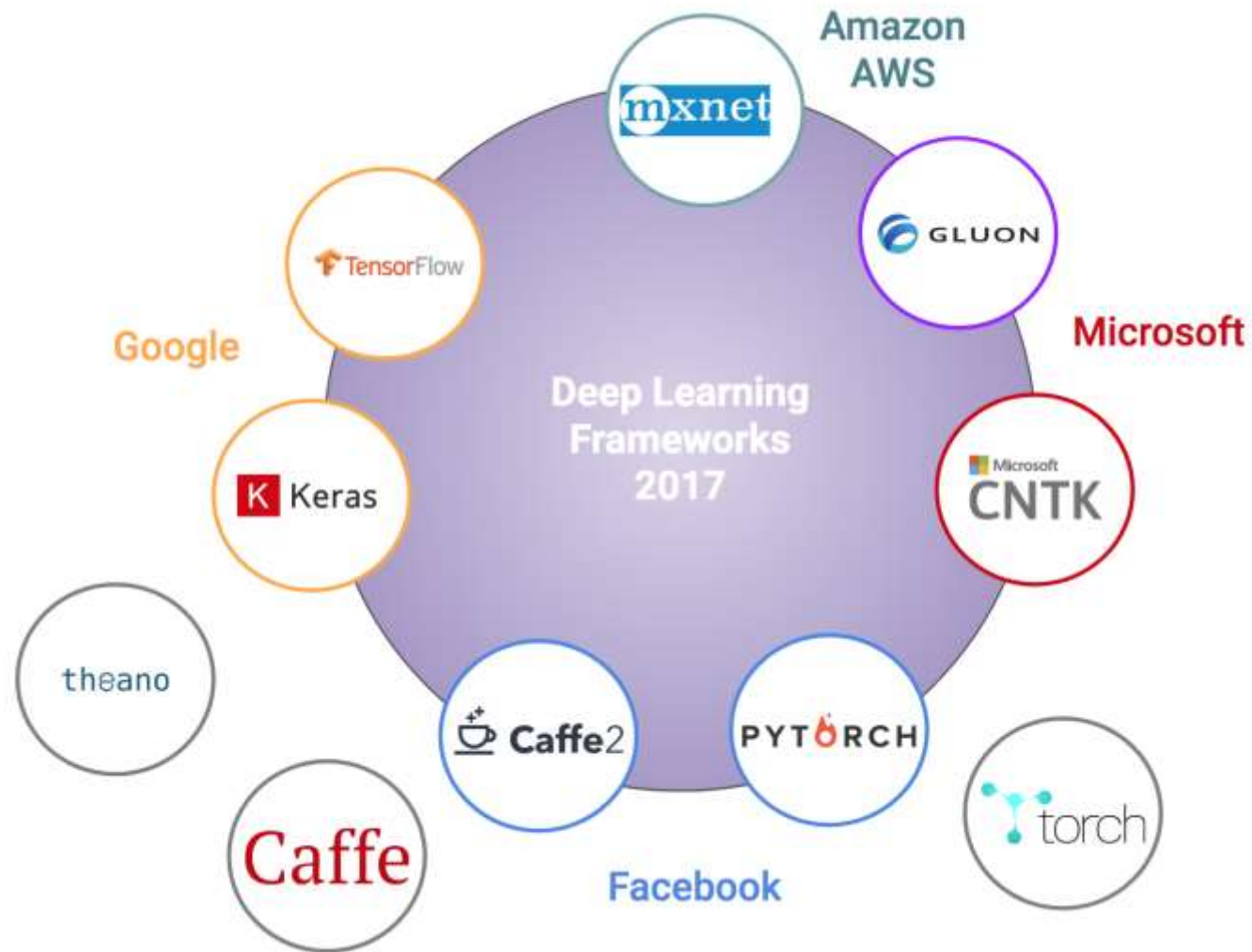
- Tensorflow không tích hợp sẵn và phải cài đặt bằng lệnh `pip install tensorflow`
- Sau khi cài đặt xong ta import bằng lệnh `import tensorflow`

Keras là gì?



- Keras là một thư viện 'high-level' với phần 'low-level' (còn được gọi là backend) là TensorFlow.
- Keras có cú pháp đơn giản hơn TensorFlow rất nhiều.
- Có thể chạy trên cả CPU và GPU.
- Tìm hiểu thêm lý do tại : <https://keras.io/why-use-keras/>

Keras là gì?



Các khái niệm trong Keras



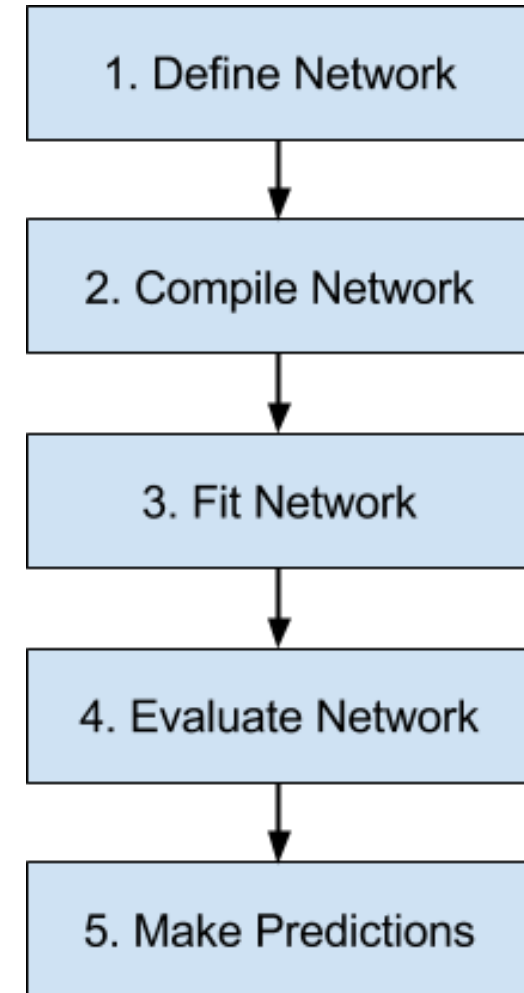
- Sequential
- Layers
- Activation Function
- Fit
- Loss
- Optimizer: SGD,
Adam
- Evaluation

Các bước xây dựng NN đơn giản bằng Keras

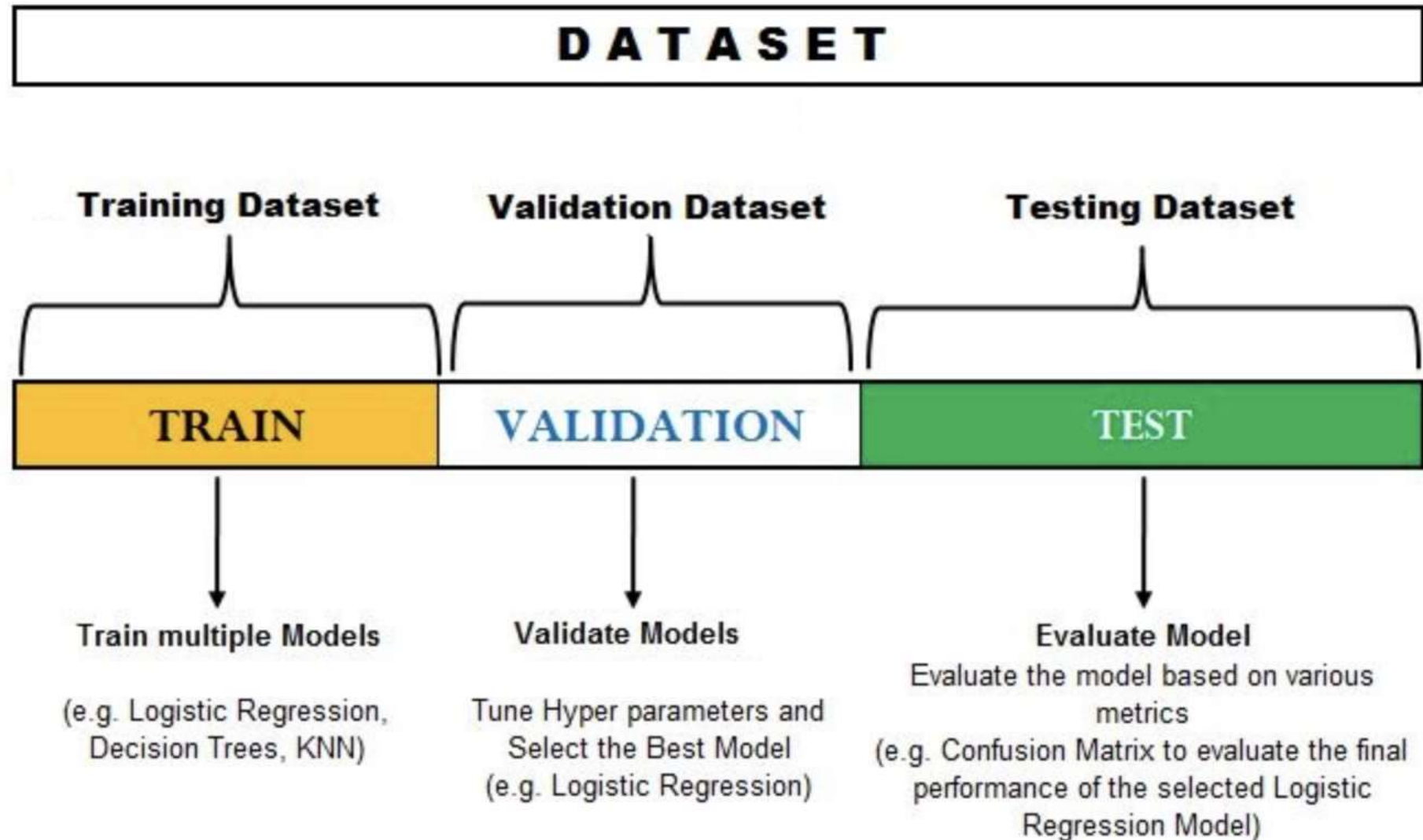
- Xây dựng bài toán dự đoán tiểu đường dựa vào dữ liệu Pima Indians Diabetes Database
 - Input: Thông số đo của bệnh nhân
 - Output: Dự đoán bệnh nhân bị tiểu đường hay không?
 - Nguồn: <https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-diabetes.names>

Các bước xây dựng NN đơn giản bằng Keras

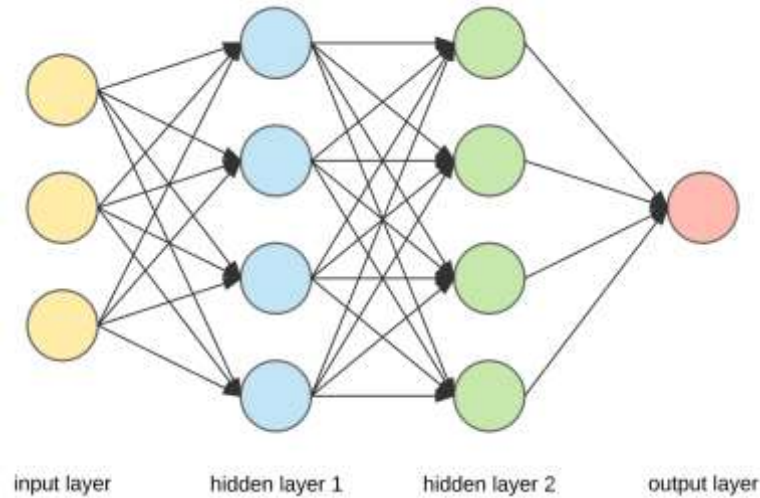
- Load dữ liệu
- Xây dựng model
- Compile model
- Train model
- Đánh giá model
- Dự đoán dữ liệu mới
- Lưu model
- Load model



Nạp dữ liệu



Xây dựng model



```
model = Sequential()  
model.add(Dense(16, input_dim=8, activation='relu'))  
model.add(Dense(8, activation='relu'))  
model.add(Dense(1, activation='sigmoid'))
```


Summary model



```
model.summary()
```

Compile model



- Loss
- Optimizer
- Metric

Train model



- Epoch
- Batch_size
- Validation

Đánh giá model



Một số vấn đề cần chú ý

- Các thông số Hyperparameters
- Chia data phải đảm bảo các class phân phối đều nhau
- Còn rất vấn đề kỹ thuật khác cần xử lý:
 - Train mãi không tăng accuracy
 - Accuracy tập train tốt, test xấu
 - Check point để lưu model...