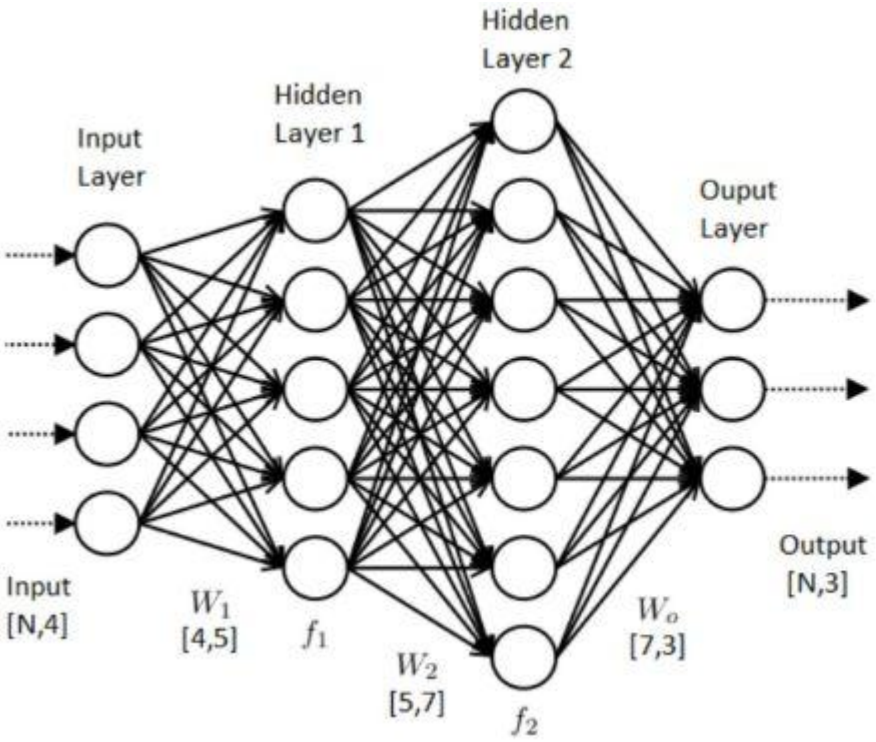
Neuron network with Keras library

minhtt@vietinbank.vn



Index

1. Neuron network library
2. Keras
   a. Overview
   b. Model
   c. Layer
   d. Optimizer
   e. Losses
   f. Callback
   g. Metric
3. Example
4. Lab
5. Reference

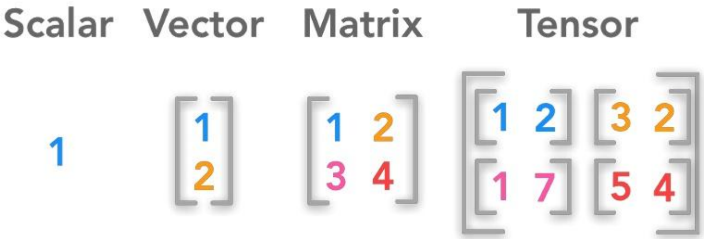
Neuron network libraries

TOP libraries

- ➢ Tensor flow
- ➢ Pytorch
- ➢ MXNet
- ➢ …

Why we need library?

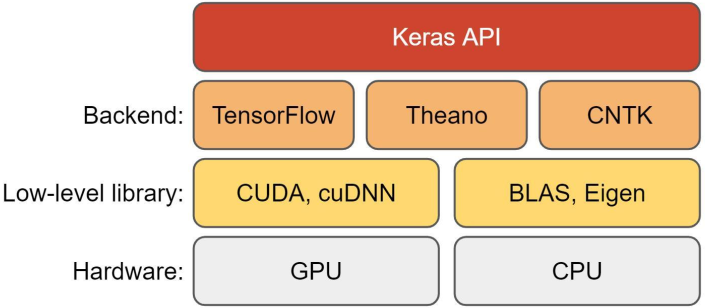
- Easy to build complex neuron network
- Auto gradient
- Utilities to control training and prediction process
- Easy to intergrate with GPU
- Processing unit is tensor



Keras

- Overview

- Simple
- Flexible
- Powerful



Keras API

Backend: TensorFlow | Theano | CNTK

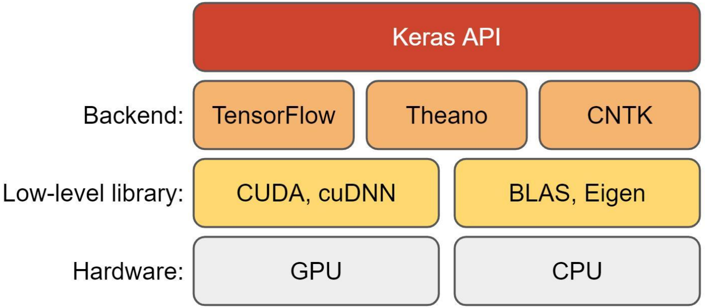
Low-level library: CUDA, cuDNN | BLAS, Eigen

Hardware: GPU | CPU

Keras

- Model

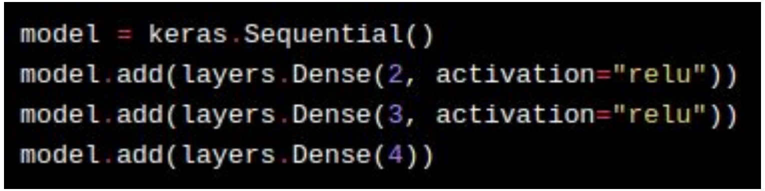
- **Sequential model**: single input, single output
- **Functional API**: support most cases
- **Model subclassing**: out of the box research use cases



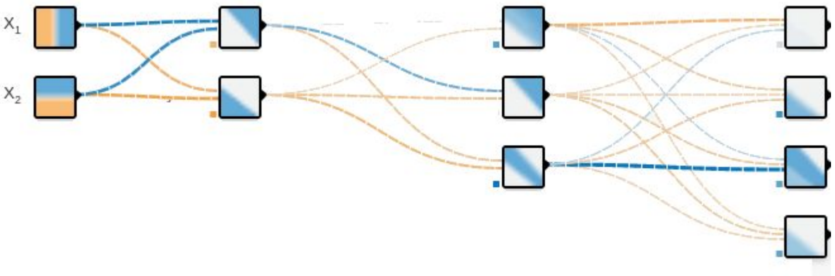
Keras

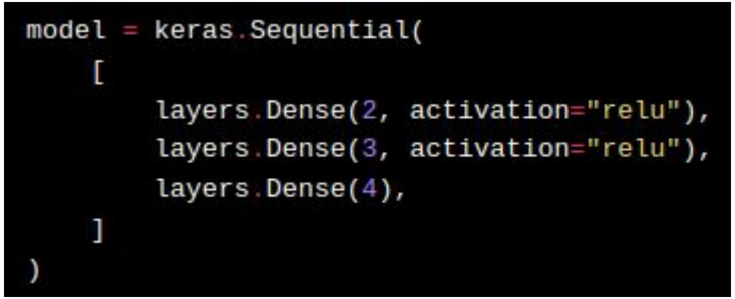
- Model

A Sequential model is appropriate for a plain stack of layers where each layer has exactly one input tensor and one output tensor.



```
model = keras.Sequential()
model.add(layers.Dense(2, activation="relu"))
model.add(layers.Dense(3, activation="relu"))
model.add(layers.Dense(4))
```



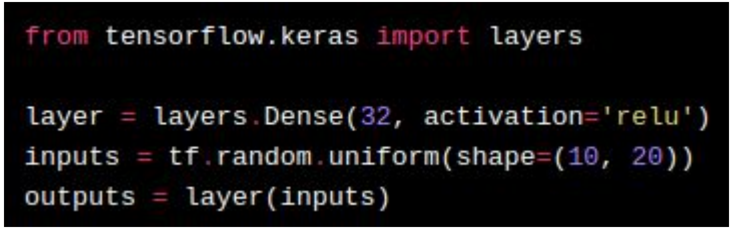


```
model = keras.Sequential(
    [
        layers.Dense(2, activation="relu"),
        layers.Dense(3, activation="relu"),
        layers.Dense(4),
    ]
)
```

Keras

- Layer

- A Layer instance is callable, much like a function
- Layers maintain a state, updated when the layer receives data during training, and stored in layer.weights
- Types of layers:
    - Core layer
    - Convolution layer
    - Pooling layer
    - Recurrent layer
    - Preprocessing layer
    - Normalization layer
    - Attention layer
    - Reshaping layer
    - Locally connected layer
    - Activation layer

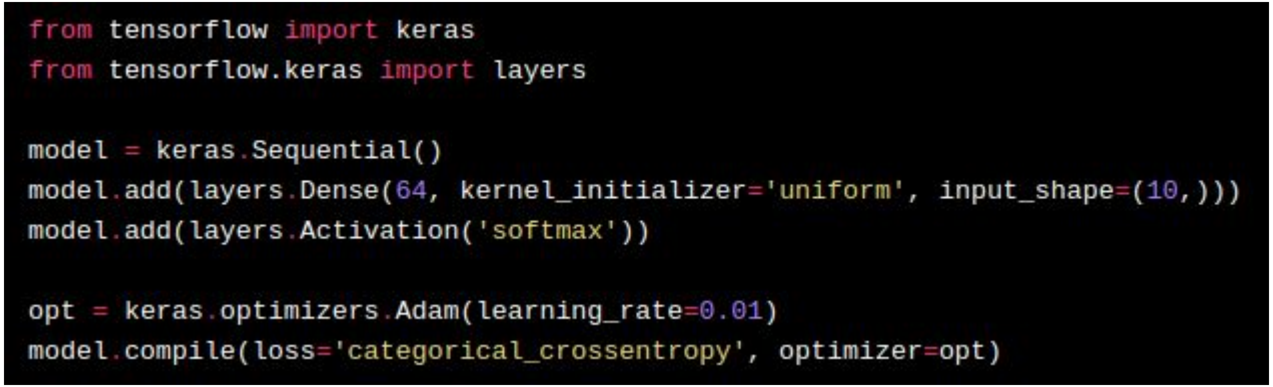


```
from tensorflow.keras import layers

layer = layers.Dense(32, activation='relu')
inputs = tf.random.uniform(shape=(10, 20))
outputs = layer(inputs)
```

Keras

- Optimizer

- SGD
- RMSprop
- Adam
- Adadelta
- Adagrad
- Adamax
- Nadam
- Ftrl

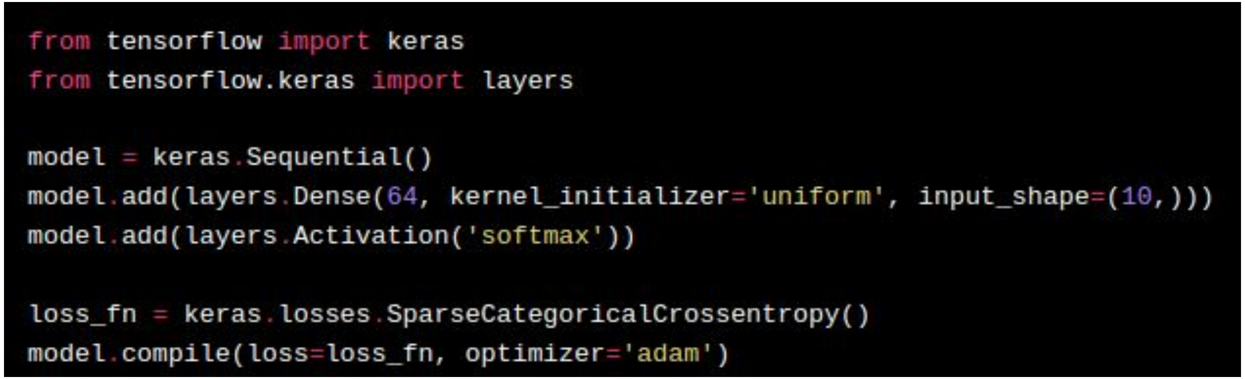


```python
from tensorflow import keras
from tensorflow.keras import layers

model = keras.Sequential()
model.add(layers.Dense(64, kernel_initializer='uniform', input_shape=(10,)))
model.add(layers.Activation('softmax'))

opt = keras.optimizers.Adam(learning_rate=0.01)
model.compile(loss='categorical_crossentropy', optimizer=opt)
```

Keras

- Losses

- The purpose of loss functions is to compute the quantity that a model should seek to minimize during training
- Reference in https://keras.io/api/losses/

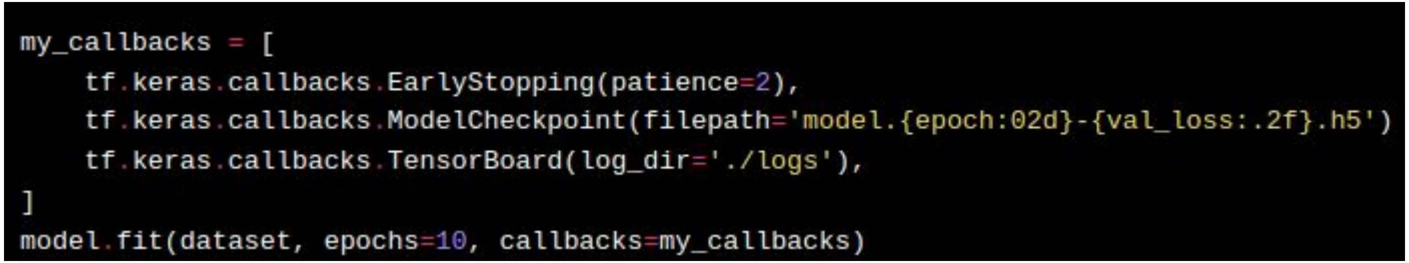


```
from tensorflow import keras
from tensorflow.keras import layers

model = keras.Sequential()
model.add(layers.Dense(64, kernel_initializer='uniform', input_shape=(10,)))
model.add(layers.Activation('softmax'))

loss_fn = keras.losses.SparseCategoricalCrossentropy()
model.compile(loss=loss_fn, optimizer='adam')
```

Keras

- Call back

- Write TensorBoard logs after every batch of training to monitor your metrics
- Periodically save your model to disk
- Do early stopping
- Get a view on internal states and statistics of a model during training
- …

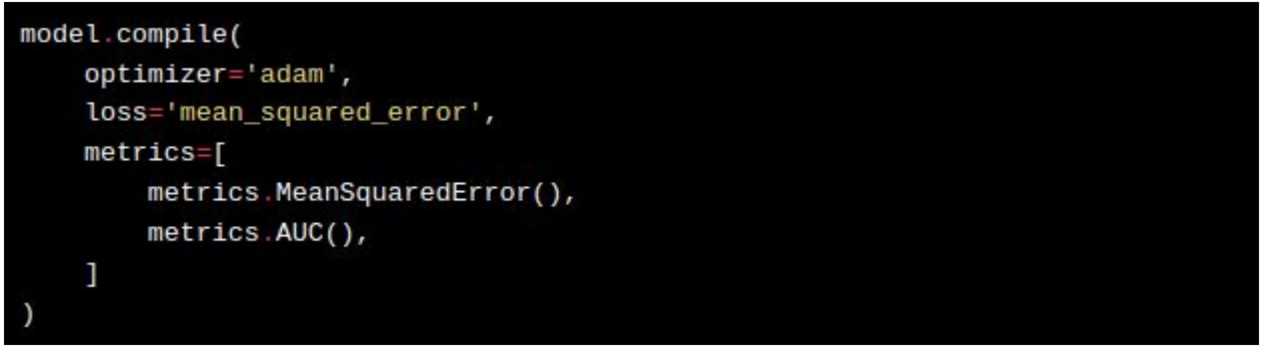


```
my_callbacks = [
    tf.keras.callbacks.EarlyStopping(patience=2),
    tf.keras.callbacks.ModelCheckpoint(filepath='model.{epoch:02d}-{val_loss:.2f}.h5')
    tf.keras.callbacks.TensorBoard(log_dir='./logs'),
]
model.fit(dataset, epochs=10, callbacks=my_callbacks)
```

Keras

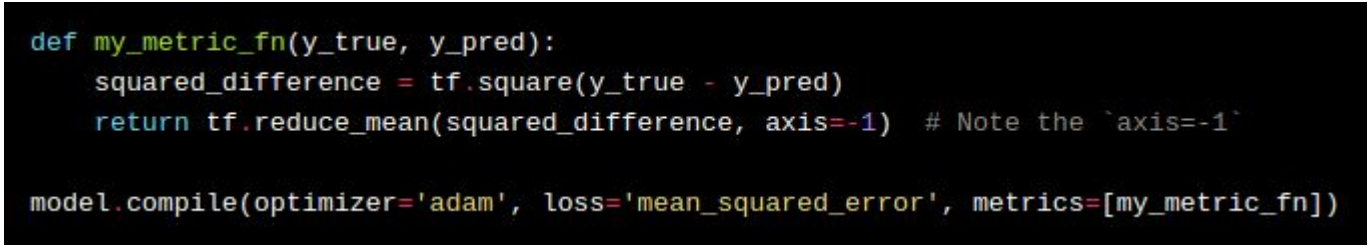
- Metric

Predefine metric



```
model.compile(
    optimizer='adam',
    loss='mean_squared_error',
    metrics=[
        metrics.MeanSquaredError(),
        metrics.AUC(),
    ]
)
```

Custom metric



```
def my_metric_fn(y_true, y_pred):
    squared_difference = tf.square(y_true - y_pred)
    return tf.reduce_mean(squared_difference, axis=-1)  # Note the `axis=-1`

model.compile(optimizer='adam', loss='mean_squared_error', metrics=[my_metric_fn])
```

References

- https://analyticsindiamag.com/top-7-python-neural-network-libraries-for-developers/
- https://keras.io/

Thank You!