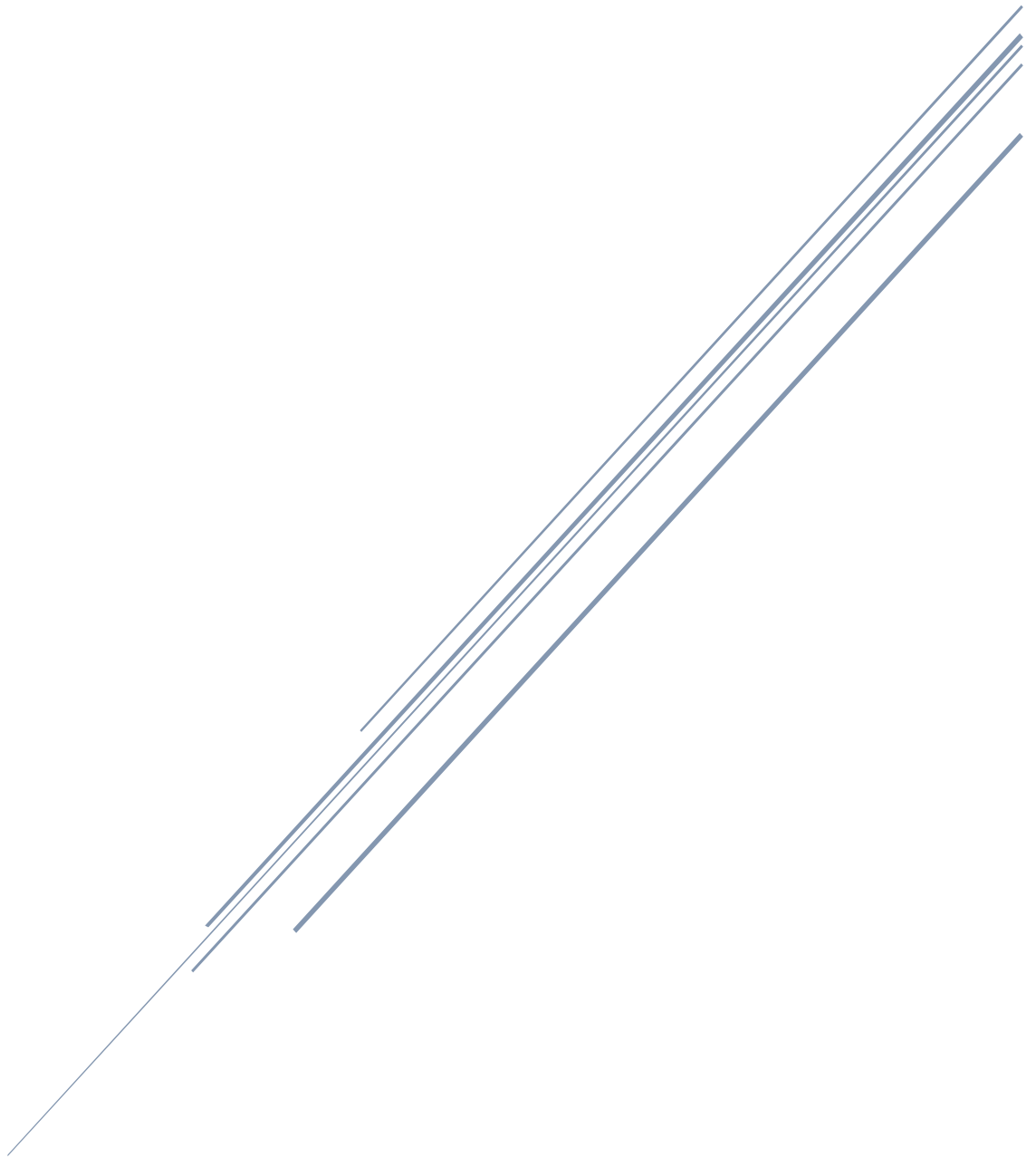


KERAS AND DEEP LEARNING LIBRARIES

Module 3



v-cardona
Deep Learning Fundamentals with Keras

Contenido

Deep learning libraries	2
Regression models with Keras.....	2
Classification models with Keras	2

Deep learning libraries

Keras vs PyTorch vs TensorFlow

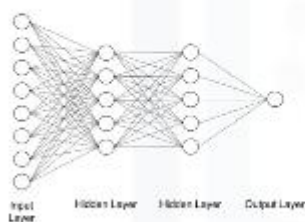
- TensorFlow is the most popular deep learning library, developed by Google.
- PyTorch is a cousin of the Lua-based Torch framework, and is a strong competitor to TensorFlow.
- Keras is the easiest API to use and the go-to library for quick prototyping and fast development.

Regression models with Keras

A dense network is that all the nodes in one layer are connected to all the other nodes in the next layer.

There are two models in the Keras library. One of them is the Sequential model and the other one is the model class used with the functional API. To create your model, you simply call the Sequential constructor. We specify the number of neurons in each layer and the activation function that we want to use. For the first hidden layer we need to pass in the "input_shape" parameter, which is the number of columns or predictors in our dataset. For training, we need to define an optimizer and the error metric. One of them is "adam". One of the main advantages of the "adam" optimizer is that you do not need to specify the learning rate. This saves us the task of optimizing the learning rate for our model. Then we use the fit method to train our model.

Keras Code



```
import keras
from keras.models import Sequential
from keras.layers import Dense

model = Sequential()

n_cols = concrete_data.shape[1]

model.add(Dense(5, activation='relu', input_shape=(n_cols,)))
model.add(Dense(5, activation='relu'))
model.add(Dense(1))

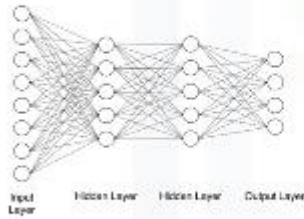
model.compile(optimizer='adam', loss='mean_squared_error')
model.fit(predictors, target)

predictions = model.predict(test_data)
```

Classification models with Keras

We will specify the evaluation metric to be "accuracy". "accuracy" is a built-in evaluation metric in Keras but you can actually define your own evaluation metric and pass it in the metrics parameter.

Keras Code



CRM DEVELOPER
SKILLS NETWORK

```
import keras
from keras.models import Sequential
from keras.layers import Dense
from keras.utils import to_categorical

model = Sequential()

n_cols = car_data.shape[1]

target = to_categorical(target)

model.add(Dense(5, activation='relu', input_shape=(n_cols,)))
model.add(Dense(5, activation='relu'))
model.add(Dense(4, activation='softmax'))

model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
model.fit(predictors, target, epochs=10)

model.predict(test_data)
```