Introduction to Keras

February 09 , 2022

Outline

- ❑ Introduction
- ❑ Architecture of Keras
- ❑ Building a Simple Deep Learning Network using Keras

Introduction

Complete Neural Network Pipeline include :

- Problem (Application)
- Dataset
- Preprocessing
- Training and Testing Dataset
- Type of Model
- No of layers
- No of Nodes
- Activation Function
- Batch size
- Epoch
- Optimization Function
- Initialization of Weights and Bias
- Evaluation Metrics

Introduction

- **Implementing complete pipeline and Experimenting with it is a complex task**

- High-level deep learning API
- Written in python
- Use TensorFlow or Theano for its backend
- Support almost all deep learning models
- Runs smoothly on CPU and GPU

Introduction

Why Keras

- Easy to use and enable fast experimentation
- Support distributed training
- Modular in nature
- Models are described in Python, which make it easy to debug and explore.

Introduction

Installation

Important libraries :

- Python
- Numpy
- Scipy
- h5py
- Matplotlib
- TensorFlow

Tools:

- Google Colab
- Anaconda

- Visit the Keras page to install and explore the API : https://keras.io

Architecture of Keras

Models

- Sequential API
- Functional API

Layers

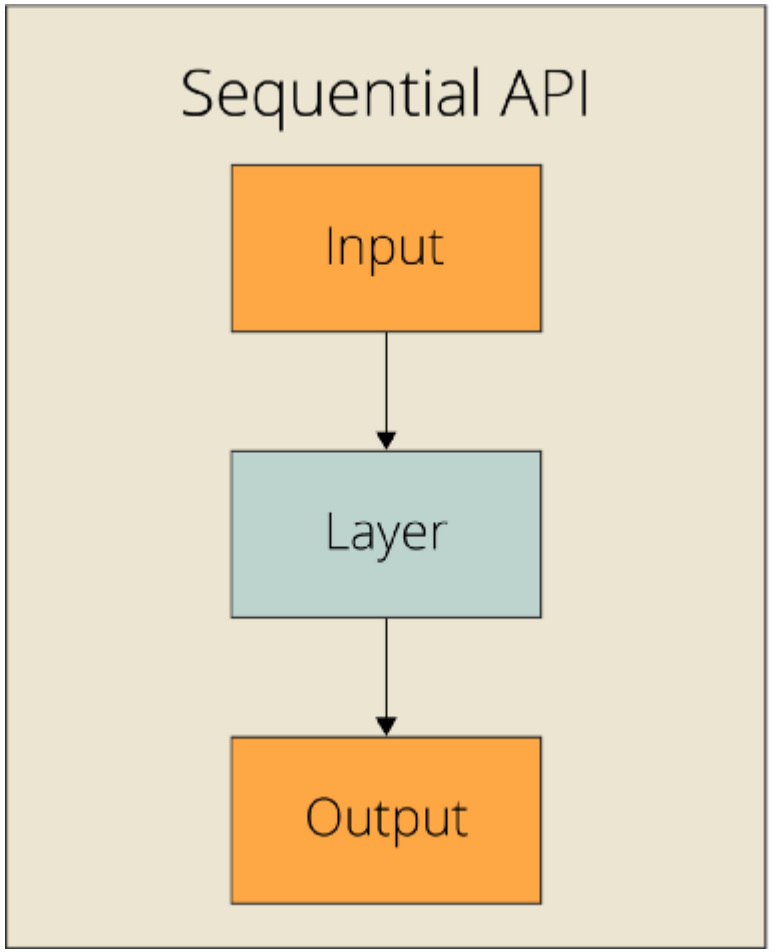
- Core Layers
  - Dense
  - Activation
  - Flatten
  - Reshape
  - many more..
- Convolution Layers
- Pooling Layers
- Recurrent Layers
- Reshaping Layer
- many more….

Modules

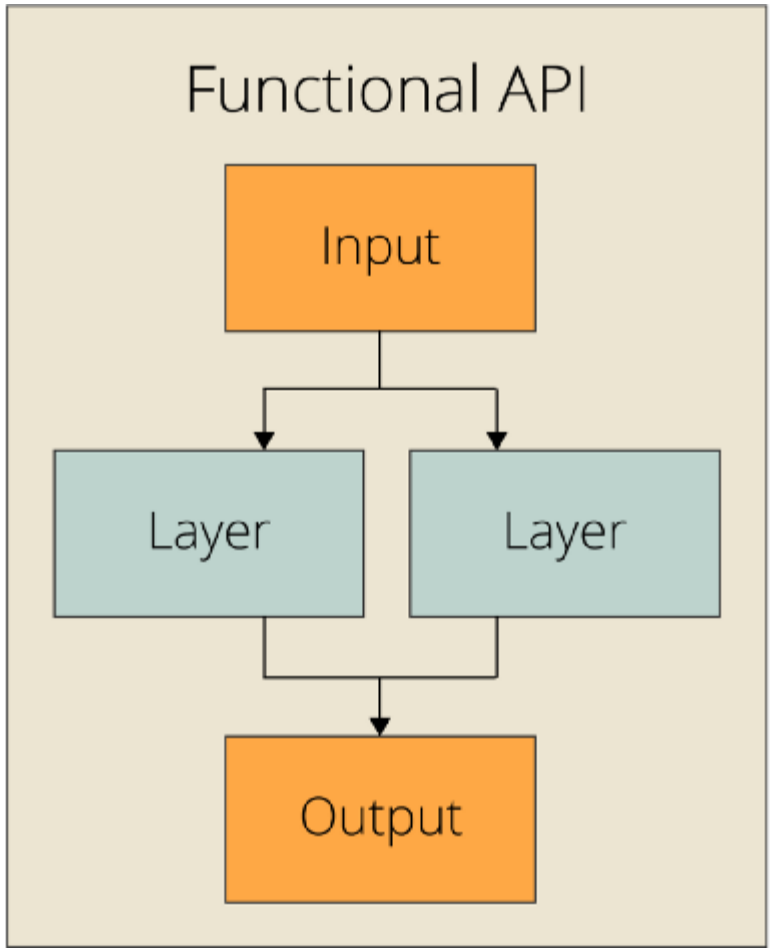
- Datasets
- Applications
- Utilities
- Keras Tuner

Models

Sequential API



Functional API



Keras Provides

Optimizer

- Algorithm used to update weights while we train our model such as sgd (Stochastic gradient descent optimizer)

Objective Function

- Used by the optimizer to navigate the space of weights such as mse (mean squared error)

Metrics

- Used to judge the performance of your model such as accuracy

Building a Simple Deep Learning Network using Keras

Steps

- Import libraries and modules
- Load data
- Pre-process data
- Define model architecture
- Compile model
- Fit and evaluate

Building a Simple Deep Learning Network using Keras

Problem :

- Digit recognition from image data

Dataset :

- Keras provides in-build many datasets such as MNIST , CIFAR10 and many more.

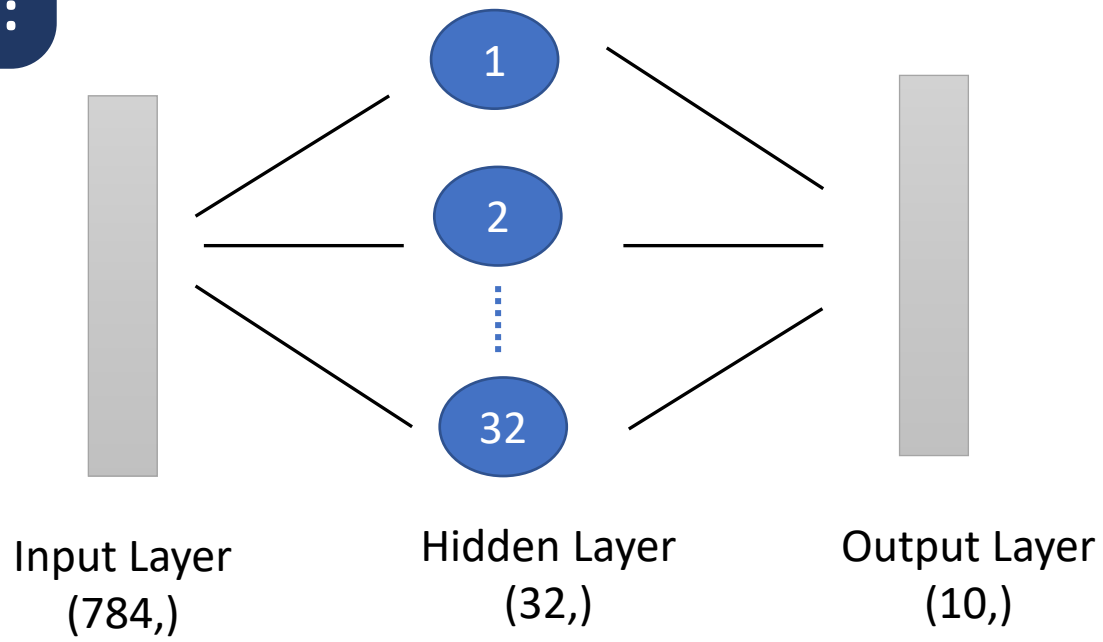
- **MNIST**
  - Dataset contains grayscale images of handwritten single digits between 0 and 9
  - 28×28 pixels
  - Training set of 60,000 examples, and a test set of 10,000 examples
  - Keras provides method to load MNIST data set

Building a Simple Deep Learning Network using Keras

Data Preprocessing :

- Reshaping
- Convert data type
- Change the labels from integer to categorical data

Model Architecture :



Building a Simple Deep Learning Network using Keras

Model Architecture :

- Use sequential model
- A sequential model is defined as

  model = Sequential()
- Add layers
  - First layer in a Sequential model needs to receive information about its input shape
  - Dense(32, input dim=784) specifies that
    - Input dimension is 784
    - It is first hidden layer
    - output dimension is 32
    - If no activation function specified, no activation is applied (i.e. ”linear” activation: a(x) = x).

Building a Simple Deep Learning Network using Keras

Model Architecture :

- There are many other initializations available in Keras
- Rectifier (ReLU) activation function is used for the neurons in the hidden layer
- Softmax activation function is used on the output layer

Building a Simple Deep Learning Network using Keras

Compile Model :

- Before training, use compile() method to build network. It uses three arguments:
  - Optimizer : Adam
  - Loss function : Logarithmic loss
  - list of metrics : Accuracy

Train Model :

- Use fit() function

Evaluate Model on test data:

- Use evaluate () function

Building a Simple Deep Learning Network using Keras

Options to explore:

- Different learning rate for optimizer
- Number of neurons in hidden layer
- Batch size
- Additional hidden layers
- with dropout
- Different optimizers
- Increases number of epochs

Questions?