Keras - High-level neural networks API

PRESENTED BY -
Aayush Agrawal

Aayush Agrawal

Education -

- MS in Business Analytics, Carlson School of Management, University of Minnesota, 2017
- B.Tech in Electrical Engineering, Malaviya National Institute of Technology, India, 2013

Experience –

- >4 years in Data science, Currently working as Data scientist for Land O' Lakes, Inc.
- Moderator and rank 3rd at https://www.analyticsvidhya.com/
- Kaggle Expert - https://www.kaggle.com/aayushmnit

Reach me out at (@aayushmnit) –

Email - aayushmnit@gmail.com

LinkedIn - https://www.linkedin.com/in/aayushmnit/

Github - https://github.com/aayushmnit

Agenda

Introduction to Keras

Why Keras?

Modules

Demo

Keras is an open source Deep learning library

- Written in Python and capable of running over Theano, TensorFlow, CNTK, MXNet, Deeplearning4j
- Runs seamlessly on CPU and GPU
- Supports both CNNs and RNNs, as well as the combination of two
- Developed as a part of research project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System)
- François Chollet - Primary Author and maintainer, Google Engineer
- Documentation – https://keras.io/

Agenda

Introduction to Keras

Why Keras?

Modules

Demo

Keras is built for enabling fast experimentation

Four guiding principles of Keras –

User friendliness –

- Keras is an API designed for human beings
- Keras follows best practices for reducing cognitive load
- Offers consistent & simple APIs

Modularity –

- A model is understood as a sequence or a graph of standalone
- Fully-configurable modules that can be plugged together with as little restrictions as possible.

Easy extensibility –

- New modules are simple to add (as new classes and functions)
- Existing modules provide ample examples

Work with Python –

- No separate models configuration files in a declarative format. Models are described in Python code, which is compact, easier to debug, and allows for ease of extensibility.

Agenda

Introduction to Keras

Why Keras?

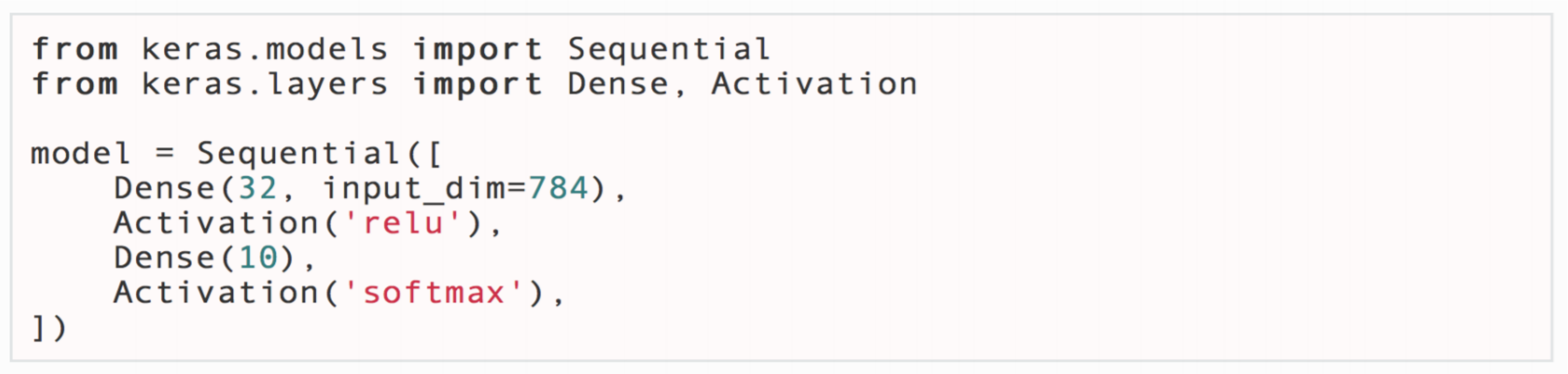
Modules

Demo

Introduction to Keras Sequential model

- The Sequential model is a linear stack of layers

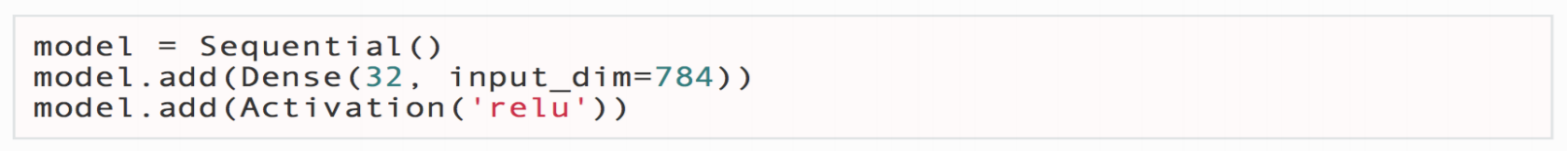
- We can create a Sequential model by passing a list of layer instances to the contructor



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

model = Sequential([
    Dense(32, input_dim=784),
    Activation('relu'),
    Dense(10),
    Activation('softmax'),
])
```

- Similary the same layers can be added via the .add() method:



```
model = Sequential()
model.add(Dense(32, input_dim=784))
model.add(Activation('relu'))
```

Components of CNN and their Keras equivalents

Typical CNN contains primarily the following components –

Dense layer of neural network –

- Git link - https://github.com/fchollet/keras/blob/master/keras/layers/core.py
- Sample code - model.add(Dense(<number of units>))

Activation function –

- Git link - https://github.com/fchollet/keras/blob/master/keras/activations.py
- Usually all the universal activation functions like linear, sigmoid, tanh, relus etc. are present
- Sample code – model.add(Activation('relu'))

Advanced activation function-

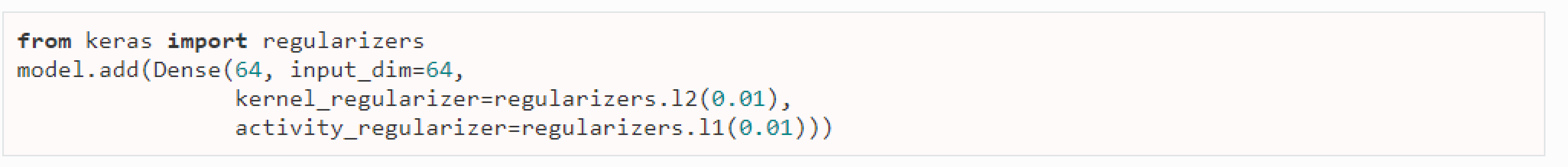
- Git link - https://github.com/fchollet/keras/blob/master/keras/layers/advanced_activations.py
- Activation functions like LeakyReLU, Parametric ReLU, SReLU

Components of CNN and their Keras equivalents(Cont..)

Typical CNN contains primarily the following components –

Regularizers –

- Git link - https://github.com/fchollet/keras/blob/master/keras/regularizers.py
- Regularizes like L1 and L2 weight penalty



```
from keras import regularizers
model.add(Dense(64, input_dim=64,
                kernel_regularizer=regularizers.l2(0.01),
                activity_regularizer=regularizers.l1(0.01)))
```

Dropout –

- Git link - https://github.com/fchollet/keras/blob/master/keras/layers/core.py
- Code sample - model.add(Dropout(0.5))

Convolutional Layer –

- Git link - https://github.com/fchollet/keras/blob/master/keras/layers/convolutional.py
- Code sample – model.add(Convolutional2D(32,3,3,border_mode =‘same’, input_shape=<shape>)

Pooling

- Git link - https://github.com/fchollet/keras/blob/master/keras/layers/pooling.py
- Code sample - model.add(MaxPooling2D((2,2), strides=(2,2)))

Agenda

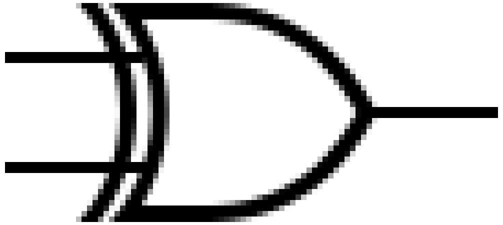
Introduction to Keras

Why Keras?

Modules

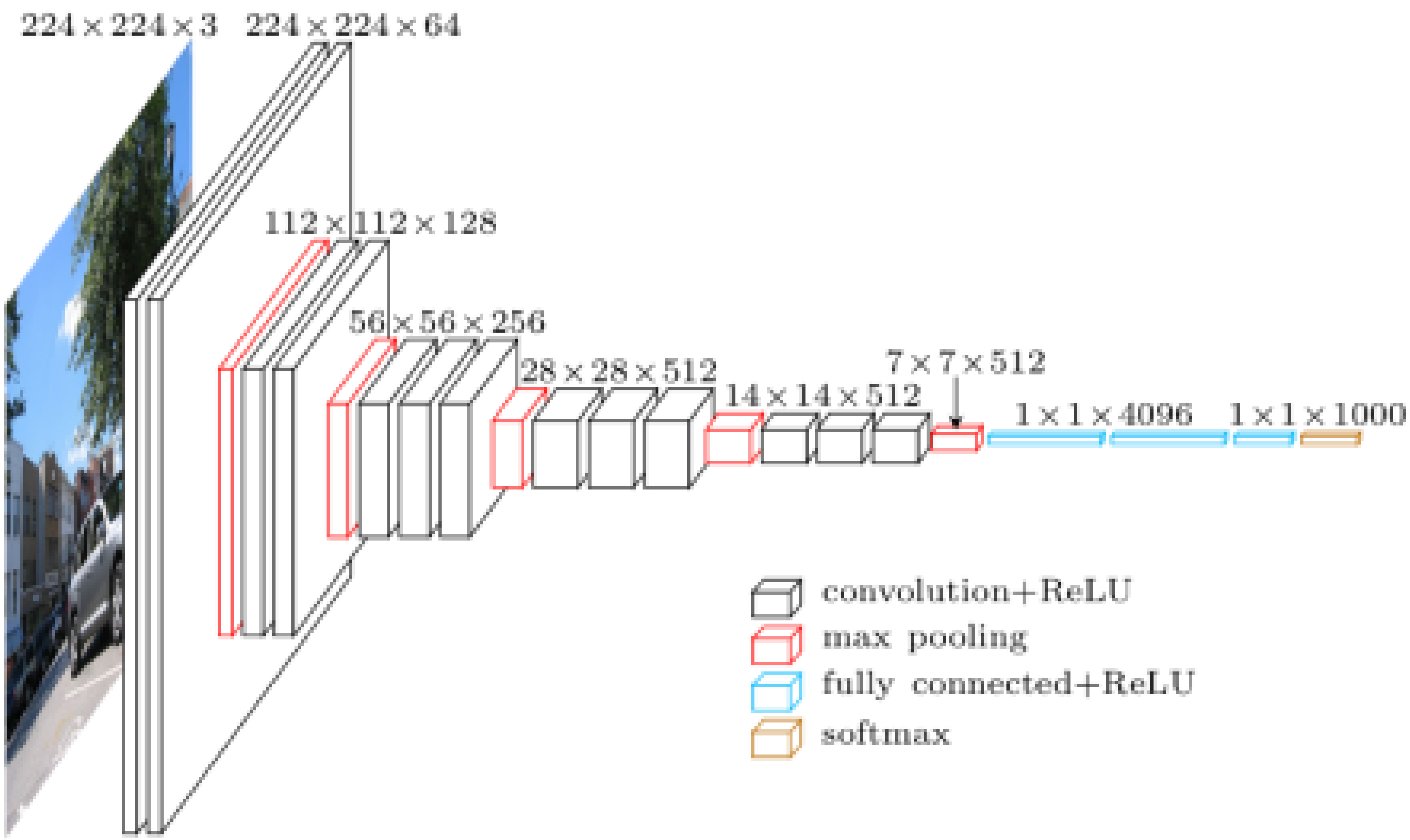
Demo

Lets train a simple neural net to learn XOR gate



Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	0

Lets look at a VGG16 model



Deep learning path Suggestion

Data science path - https://www.analyticsvidhya.com/blog/2017/01/the-most-comprehensive-data-science-learning-plan-for-2017/

My Deep learning track –

1) Machine learning by Andre NG(his first course and the most popular course in MOOC history) -> https://www.coursera.org/learn/machine-learning (Low difficulty)

2) Deep learning by Google on udacity - https://www.udacity.com/course/deep-learning--ud730 (Hard)

3) Practical deep learning for Coders by Jeremy Howard (Former Kaggle #1) - http://course.fast.ai/ (Medium/Hard)

4) A book on deep learning (Goodfellow) - http://www.deeplearningbook.org/ (If you need to understand deep math)

5) Andrew NGs deep learning track - https://www.coursera.org/specializations/deep-learning (easy/medium)

6) Just some collection of good blogs - http://colah.github.io/

Thank You!