**Keras**



11.11.2019

I began reviewing Ian’s information on Keras in Moodle today.

|  |  |  |
| --- | --- | --- |
| 1 | Introduction to keras | How to install it - 10mins in length – 2.50 –  You use this to make predictions (5.30) – 6mins how to install it  **VIDEO**  DONE |
| 2 | Individual neurons in keras | 44mins  **VIDEO** |
| 3 | Neurons in keras notebook | <https://github.com/ianmcloughlin/jupyter-teaching-notebooks/blob/master/keras-neurons.ipynb>  DONE |
| 4 | Keras and iris | <https://nbviewer.jupyter.org/github/ianmcloughlin/jupyter-teaching-notebooks/blob/master/keras-and-iris.ipynb>  DONE |
| 5 | Keras website | <https://keras.io/>  DONE |
| 6 | Tensorflow website | <https://www.tensorflow.org/>  DONE |
| 7 | Look Ma, No For-Loops: Array Programming With NumPy | <https://realpython.com/numpy-array-programming/>  DONE |
| 8 | Pure Python vs NumPy vs TensorFlow Performance Comparison | <https://realpython.com/numpy-tensorflow-performance/>  DONE |

Etain’s notes from video ‘Individual neurons in keras’ (2)

Keras is a package for Python which can be used to make predictions.

*Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research.* (https://keras.io/)

Based on Ian’s video ‘Individual neurons in keras‘, it sounds like I will be able to use keras in this assignment to complete the below requirements;

*Within the repository, create a jupyter notebook that uses descriptive statistics and plots to describe the Boston House Prices dataset. Again using the same notebook, use keras to create a neural network that can predict the median house price based on the other variables in the dataset.*

Neurons are mentioned.

Ian demonstrates how to insert content to keras in order to have the exact same content expelled/output by keras. (1.45min)

Ian mentioned that the following 3 commands are the main commands I will use to create my neuron network;



# Create a new neural network.

m = kr.models.Sequential()

# Add a single neuron in a single layer, initialised with weight 1 and bias 0.

m.add(kr.layers.Dense(1, input\_dim=1, activation="linear", kernel\_initializer='ones', bias\_initializer='zeros'))

# Compile the model.

m.compile(loss="mean\_squared\_error", optimizer="sgd")

(Source: <https://github.com/ianmcloughlin/jupyter-teaching-notebooks/blob/master/keras-neurons.ipynb>)

Input – dataset. Output – you sometimes have an idea what this will look like already. E.g. the dataset could have several values, but you want the output to have only 2 potential outcomes such as ‘yes’ and ‘no’.

* Sequential – explained at 39mins
* Dense – explained at 41mins

\*\*\*

*TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.* (https://www.tensorflow.org/)