

EVML

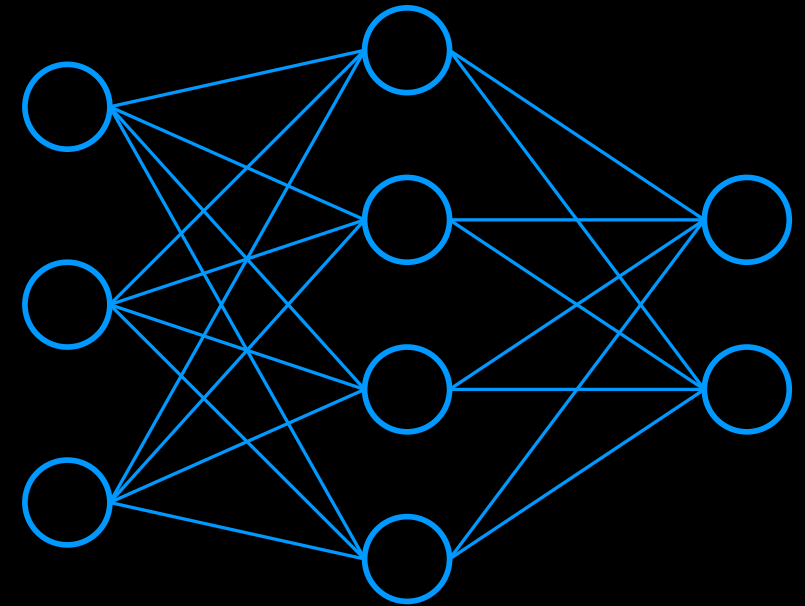
TENSORFLOW & KERAS

HANDS-ON

JEROEN VEEN

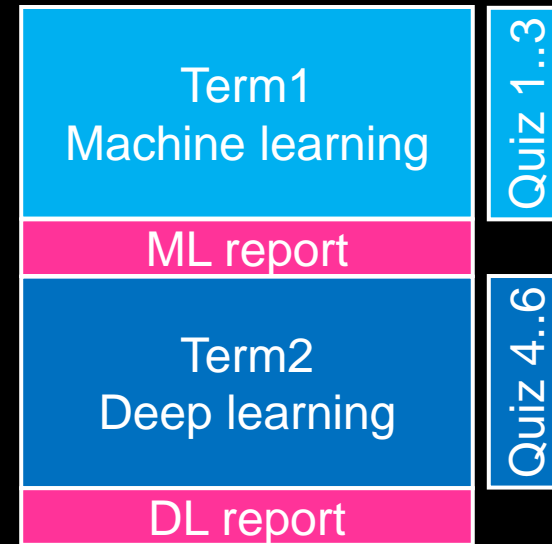
AGENDA

- Deep learning workshop organization
- Deep learning
- Deep learning frameworks
- Installing TensorFlow 2
- Image classification exercise



ORGANIZATION OF THE WORKSHOP

- Theory with integrated quizzes.
- Hands-on with 2 mini-projects
- Final mark:
80% ML + DL report,
20% quiz results



- Short video clip showing your deployed model

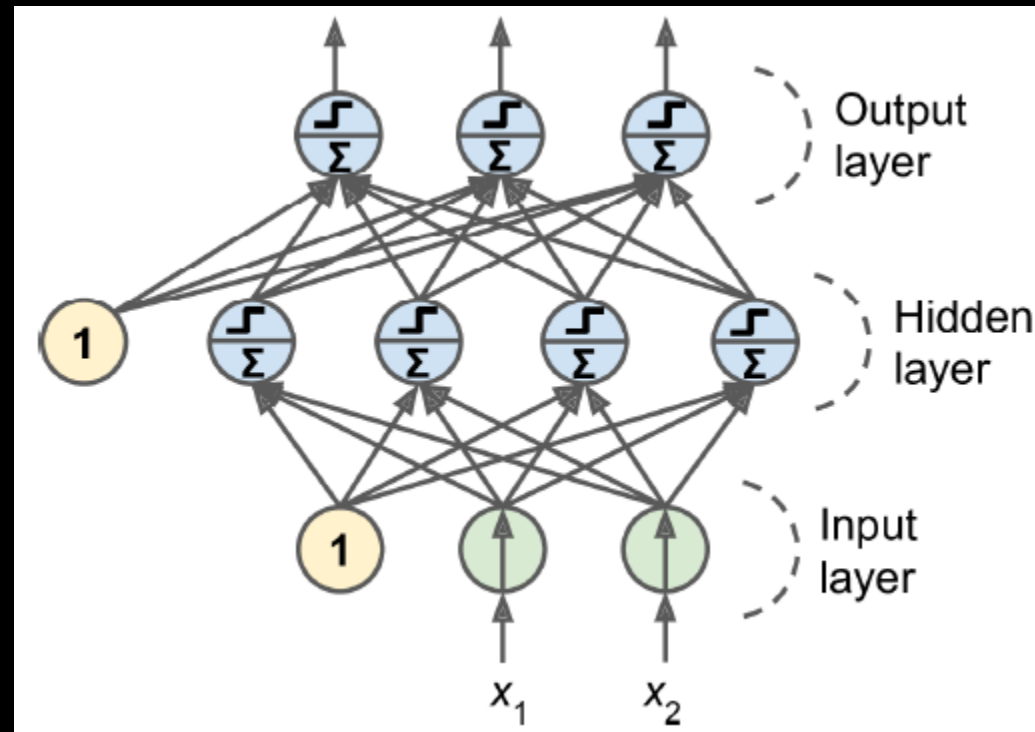
DL PORTFOLIO TEMPLATE

TABLE OF CONTENTS

1	INTRODUCTION.....	3
2	PROBLEM STATEMENT	4
3	DATA AUGMENTATION AND PREPROCESSING.....	5
4	CNN ARCHITECTURE AND TRAINING.....	6
5	DEPLOY AND TEST	7
6	CONCLUSION	8
7	REFERENCES.....	9
	CODE APPENDICES.....	10

NEURAL NETWORKS

- Classification



Source: Géron, ISBN: 9781492032632

DEEP LEARNING FRAMEWORKS

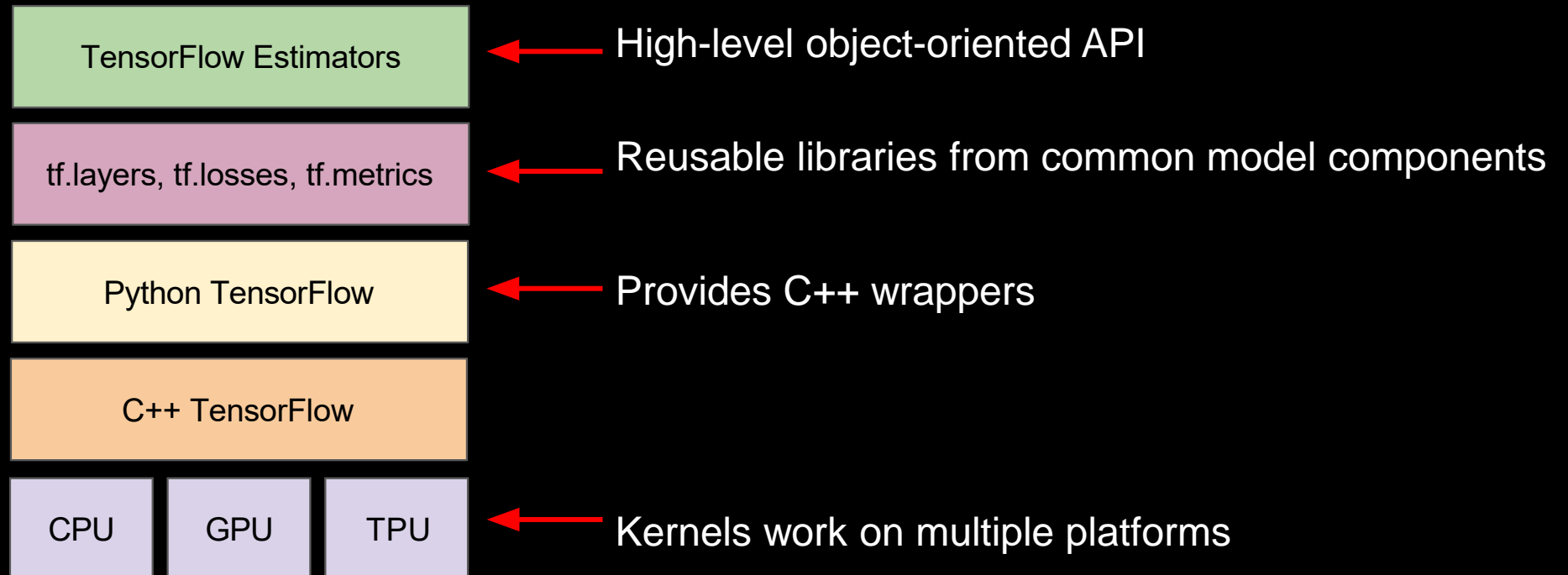
Pro's

- Standardized environment
- Reduce time, costs, complexity of development
- Problem domain specific, e.g. decision support, financial, end-user application, web

Con's

- Steep learning curve
- Inefficient use of resources

TENSORFLOW TOOLKIT



<https://developers.google.com/machine-learning/crash-course/first-steps-with-tensorflow/toolkit>

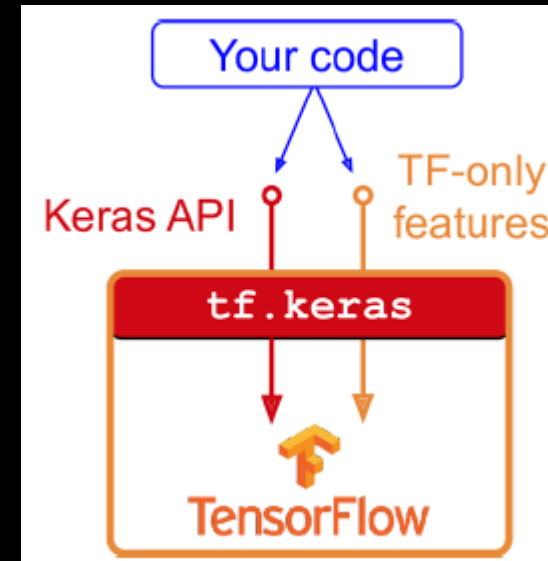
KERAS

- Using the TensorFlow framework directly is a lot of hard work.
- Extremely verbose and prone to subtle, hard-to-catch bugs
- Steep learning curve

Solution: use API like Keras (or TFLearn)

- Fast experimentation.
- Capable of running on top of TensorFlow, CNTK, or Theano.

<https://datascience.stackexchange.com/questions/25317/what-are-the-pros-and-cons-of-keras-and-tflearn>



MORE ON KERAS

- <https://www.youtube.com/watch?v=tDaGT4N4aCA&t=141s>

INSTALLING TENSORFLOW 2

- <https://www.tensorflow.org/install> for latest version

```
$ python3 -m pip install --upgrade tensorflow
```



For GPU support, at the time of this writing you need to install `tensorflow-gpu` instead of `tensorflow`, but the TensorFlow team is working on having a single library that will support both CPU-only and GPU-equipped systems. You will still need to install extra libraries for GPU support (see <https://tensorflow.org/install> for more details). We will look at GPUs in more depth in [Chapter 19](#).

To test your installation, open a Python shell or a Jupyter notebook, then import TensorFlow and `tf.keras` and print their versions:

```
>>> import tensorflow as tf
>>> from tensorflow import keras
>>> tf.__version__
'2.0.0'
>>> keras.__version__
'2.2.4-tf'
```

KERAS SEQUENTIAL MODEL

- <https://www.youtube.com/watch?v=qFJeN9V1Zsl&t=401s>
up to 10:30
- Terminology:
x = samples = input data
y = labels = target data = classes
- <https://www.youtube.com/watch?v=qFJeN9V1Zsl&t=1150s>
up to 23:27
- <https://www.youtube.com/watch?v=qFJeN9V1Zsl&t=1508s>
up to 29:33

EXERCISE: BASIC IMAGE CLASSIFICATION

- <https://www.tensorflow.org/tutorials/keras/classification>
- Train a shallow net, see also Géron pp. 297-307
- Trouble downloading the datasets? Let me know
- Training can take quite some time...
 - >> Use a small number of weights in the hidden layers
 - >> Work on colab.research.google.com
- Train and validate (see Géron p. 300)



Instead of passing a validation set using the `validation_data` argument, you could set `validation_split` to the ratio of the training set that you want Keras to use for validation. For example, `validation_split=0.1` tells Keras to use the last 10% of the data (before shuffling) for validation.

EXERCISE: CONFUSION MATRIX

- Hint 1:

The model returns probabilities, find the most probable class using `np.argmax(..., axis=1)`

- Hint 2:

`from sklearn.metrics import confusion_matrix`

REFERENCES

- <http://tflearn.org/>
- <https://www.tensorflow.org/>
- <https://keras.io/>
- <https://www.tensorflow.org/tensorboard/>
- <https://developers.google.com/machine-learning/crash-course/ml-intro>
- <https://developers.google.com/machine-learning/glossary/tensorflow>