



Deep Learning for ECE

EECE-580G

Introduction to TensorFlow 2.x

Deep Learning Frameworks

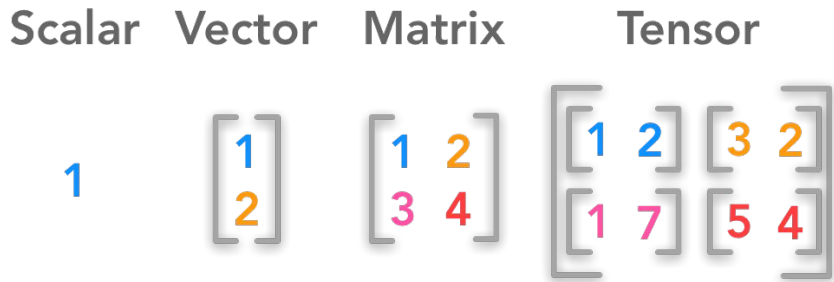
— — —

- This class will use **Tensorflow 2.x** (tf 2.x)
- Many other deep learning libraries exist (pytorch, MXNet, chainer, Theano, CNTK etc.)
- Tensorflow and Pytorch are the two most popular libraries of the past few years
- PyTorch is the preferred deep-learning library for researchers, while Tensorflow is more widely used in production
- It's easy to learn a new library once you are familiar with the mechanics
- **Important note:** We will specifically use tf 2.x, many tutorials and web resources use previous versions of tf (1.x). If you copy something that is not tf 2.x, it will (most probably) break your code
- tf 2.x is designed to be much simpler than tf 1.x

TF Basics

— — —

A tensor is a generalization of vectors and matrices to potentially higher dimensions. Remember, tensors are objects of the `tf.Tensor` class, they have their own methods and attributes, but mostly similar to usual numpy or python methods



[Image source](#)

TF Basics

— — —

>> Colab

High level APIs

— — —

High level APIs allow building, training, saving, using neural networks without going too deep in the code:

- **tf.estimator** = Popular in tf 1.x versions
- **tf.keras** = The new design standard in tf 2.x = **Our focus**

Tf.Keras

— — —

Keras is an API standard for deep learning (<https://keras.io/>) developed by [François Chollet](#)

- The tf.keras package is separate from the keras package (don't install keras)
- But as of Keras release 2.3.0, both are now in sync
- Tf.keras implements the keras API

Tf.Keras

— — —

Models

(functional, sequential,
subclassed)

Optimizers

Metrics

Layers

(tf.keras.layers, custom)

Losses

Callbacks

Tf.Keras

— — —

Tf.keras allows 3 levels of complexity

- **Sequential** = For simple models and use-cases
- **Functional** = Allows more complex models
- **Model subclassing** = Fully-customizable model

The 3 syntaxes can interact in one program

https://www.tensorflow.org/api_docs/python/tf/keras

Tf.Keras

— — —

>> Colab

Tf.Keras

— — —

A callback is a set of behaviors to be applied at given stages of the training procedure.

You can use callbacks to get a view on internal states and statistics of the model during training, or stop training if some condition is met, etc

https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/Callback

Tf.Keras

— — —

>> Colab

End