

TensorFlow Cheat Sheet



About

TensorFlow

TensorFlow™ is an open source software library for numerical computation using data flow graphs. TensorFlow was originally developed for the purposes of conducting machine learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well.

Skflow

Skflow provides a set of high level model classes that you can use to easily integrate with your existing Sklearn pipeline code. Skflow is a simplified interface for TensorFlow, to get people started on predictive analytics and data mining. Skflow has been merged into TensorFlow since version 0.8 and now called TensorFlow Learn.

Keras

Keras is a minimalist, highly modular neural networks library, written in Python and capable of running on top of either TensorFlow or Theano

Installation

How to install new package in Python:

```
pip install <package-name>
```

Example: pip install requests

How to install tensorflow?

```
device = cpu/gpu
```

```
python_version = cp27/cp34
```

```
sudo pip install
```

```
https://storage.googleapis.com/tensorflow/linux/$device/tensorflow-0.8.0-$python_version-none-linux_x86_64.whl
```

How to install Skflow

```
pip install sklearn
```

How to install Keras

```
pip install keras
```

```
update ./keras/keras.json - replace "theano" by "tensorflow"
```

Helpers

Python helper

Important functions

`type(object)`

Get object type

`help(object)`

Get help for object (list of available methods, attributes, signatures and so on)

`dir(object)`

Get list of object attributes

(fields, functions)

`str(object)`

Transform an object to string

`object?`

Shows documentation about the object

`globals()`

Return the dictionary containing the current scope's global variables.

`locals()`

Update and return a dictionary containing the current scope's local variables.

`id(object)`

Return the identity of an object. This is guaranteed to be unique among simultaneously existing objects.

```
import __builtin__
```

```
dir(__builtin__)
```

Other built-in functions

TensorFlow

Main classes

```
tf.Graph()
```

```
tf.Operation()
```

```
tf.Tensor()
```

```
tf.Session()
```

Some useful functions

```
tf.get_default_session()
```

```
tf.get_default_graph()
```

```
tf.reset_default_graph()
```

```
ops.reset_default_graph()
```

```
tf.device('/cpu:0')
```

```
tf.name_scope(value)
```

```
tf.convert_to_tensor(value)
```

TensorFlow Optimizers

```
GradientDescentOptimizer
```

```
AdadeltaOptimizer
```

```
AdagradOptimizer
```

```
MomentumOptimizer
```

```
AdamOptimizer
```

```
FtrlOptimizer
```

```
RMSPropOptimizer
```

Reduction

```
reduce_sum
```

```
reduce_prod
```

```
reduce_min
```

```
reduce_max
```

```
reduce_mean
```

```
reduce_all
```

```
reduce_any
```

```
accumulate_n
```

Activation functions

```
tf.nn?
```

```
relu
```

```
relu6
```

```
elu
```

```
softplus
```

```
softsign
```

```
dropout
```

```
bias_add
```

```
sigmoid
```

```
tanh
```

```
sigmoid_cross_entropy_with_logits
```

```
softmax
```

```
log_softmax
```

```
softmax_cross_entropy_with_logits
```

```
sparse_softmax_cross_entropy_with_logits
```

```
weighted_cross_entropy_with_logits
```

etc.

Skflow

Main classes

```
TensorFlowClassifier
```

```
TensorFlowRegressor
```

```
TensorFlowDNNClassifier
```

```
TensorFlowDNNRegressor
```

```
TensorFlowLinearClassifier
```

```
TensorFlowLinearRegressor
```

```
TensorFlowRNNClassifier
```

```
TensorFlowRNNRegressor
```

TensorFlowEstimator

Each classifier and regressor have following fields

`n_classes=0` (Regressor), `n_classes` are expected to be input (Classifiers)

`batch_size=32`,

`steps=200`, // except

`TensorFlowRNNClassifier` - there is 50

`optimizer='Adagrad'`,

`learning_rate=0.1`,

`class_weight=None`,

`clip_gradients=5.0`,

`continue_training=False`,

`config=None`,

`verbose=1`.

Each class has a method fit

```
fit(X, y, monitor=None, logdir=None)
```

`X`: matrix or tensor of shape [n_samples, n_features...]. Can be iterator that returns arrays of features. The training input samples for fitting the model.

`y`: vector or matrix [n_samples] or

[n_samples, n_outputs]. Can be iterator that returns array of targets. The training target values (class labels in classification, real numbers in regression).

`monitor`: Monitor object to print training progress and invoke early stopping
`logdir`: the directory to save the log file that can be used for optional visualization.

`predict` (`X, axis=1, batch_size=None`)

`Args`:

`X`: array-like matrix, [n_samples,

n_features...] or iterator.

`axis`: Which axis to argmax for classification.

By default axis 1 (next after batch) is used. Use 2 for sequence predictions.

`batch_size`: If test set is too big, use batch size to split it into mini batches. By default the `batch_size` member variable is used.

Returns:

`y`: array of shape [n_samples]. The predicted classes or predicted value.

Useful links

TensorFlow API

https://www.tensorflow.org/versions/r0.9/api_docs/index.html

TensorFlow How-Tos

https://www.tensorflow.org/versions/r0.9/how_tos/index.html

Skflow github

<https://github.com/tensorflow/skflow>

Skflow API

<http://skflow.learn.org/stable/modules/classes.html#module-skflow.cluster>

Keras

<http://keras.io/>

Our github

<https://github.com/Altoros/TensorFlow-training>

TensorFlow github

<https://github.com/tensorflow/tensorflow>