TensorFlow Quick Reference Sheet

by

www.TensorFlow.eu

Import TensorFlow:

import tensorflow as tf

Basic math operations:

tf.add()	sum
tf.subtract()	substraction
tf.multiply()	multiplication
tf.div()	division
tf.mod()	module
tf.abs()	absolute value
tf.negative()	negative value
tf.sign()	return sign
tf.reciprocal()	reciprocal
tf.square()	square
tf.round()	nearest intiger
tf.sqrt()	square root
tf.pow()	power
tf.exp()	exponent
tf.log()	logarithm
tf.maximum()	maximum
tf.minimum()	minimum
tf.cos()	cosine
tf.sin()	sine

Basic operations on tensors:

tf.string_to_number()	converts string to numeric type
tf.cast()	casts to new type
tf.shape()	returns shape of tensor
tf.reshape()	reshapes tensor
tf.diag()	creates tensor with given diagonal values
tf.zeros()	creates tensor with all elements set to zero
tf.fill()	creates tensor with all elements set given value
tf.concat()	concatenates tensors
tf.slice()	extracts slice from tensor
tf.transpose()	transpose the argument
tf.matmul()	matrices multiplication
tf.matrix_determinant()	determinant of matrices
tf.matrix_inverse()	computes inverse of matrices

Control Flow:

tf.while_loop()	repeat body while condition true
tf.case()	case operator
tf.count_up_to()	incriments ref untill limit
tf.tuple()	groups tensors together

Logical/Comparison Operators:

tf.equal()	returns truth value element-wise
tf.not_equal()	returns truth value of X!=Y
tf.less()	returns truth value of X<Y
tf.less_equal()	returns truth value of X<=Y
tf.greater()	returns truth value of X>Y
tf.greater_equal()	returns truth value of X>=Y
tf.is_nan()	returns which elements are NaN
tf.logical_and()	returns truth value of 'AND' for given tensors
tf.logical_or()	returns truth value of 'OR' for given tensors
tf.logical_not()	returns truth value of 'NOT' for given tensors
tf.logical_xor()	returns truth value of 'XOR' for given tensors

Working with Images:

tf.image.decode_image()	converts image to tensor type uint8
tf.image.resize_images()	resize images
tf.image.resize_image_with_crop_	resize image by cropping or padding
tf.image.flip_up_down()	flip image horizontally
tf.image.rot90()	rotate image 90 degrees counter-clockwise
tf.image.rgb_to_grayscale()	converts image from RGB to grayscale
tf.image.per_image_standardizati	scales image to zero mean and unit norm

Neural Networks:

tf.nn.relu()	rectified linear activation function
tf.nn.softmax()	softmax activation function
tf.nn.sigmoid()	sigmoid activation function
tf.nn.tanh()	hyperbolic tangent activation function
tf.nn.dropout	dropout
tf.nn.bias_add	adds bias to value
tf.nn.all_candidate_sampler()	set of all classes
tf.nn.weighted_moments()	returns mean and variance
tf.nn.softmax_cross_entropy_with_logits()	softmax cross entropy
tf.nn.sigmoid_cross_entropy_with_logits()	sigmoid cross entropy
tf.nn.l2_normalize()	normalization using L2 Norm
tf.nn.l2_loss()	L2 loss
tf.nn.dynamic_rnn()	RNN specified by given cell
tf.nn.conv2d()	2D convolutions given 4D input
tf.nn.conv1d()	1D convolution given 3D input
tf.nn.batch_normalization()	batch normalization
tf.nn.xw_plus_b()	computes matmul(x,weights)+biases

High level Machine Learning:

tf.contrib.keras	Keras API as high level API for TensorFlow
tf.contrib.layers.one_hot_column()	one hot encoding
tf.contrib.learn.LogisticRegressor()	logistic regression
tf.contrib.learn.DNNClassifier()	DNN classifier
tf.contrib.learn.DynamicRnnEstimator()	Rnn Estimator
tf.contrib.learn.KMeansClustering()	K-Means Clusstering
tf.contrib.learn.LinearClassifier()	linear classifier
tf.contrib.learn.LinearRegressor()	linear regressor
tf.contrib.learn.extract_pandas_data()	extract data from Pandas dataframe
tf.contrib.metrics.accuracy()	accuracy
tf.contrib.metrics.auc_using_histogram()	AUC
tf.contrib.metrics.confusion_matrix()	confusion matrix
tf.contrib.metrics.streaming_mean_absolute_error()	mean absolute error
tf.contrib.rnn.BasicLSTMCell()	basic lstm cell
tf.contrib.rnn.BasicRNNCell()	basic rnn cell

Placeholders and Variables:

tf.placeholder()	defines placeholder
tf.Variable(tf.random_normal([3, 4], stddev=0.1)	defines variable
tf.Variable(tf.zeros([50]), name='x')	defines variable
tf.global_variables_initializer()	initialize global variables
tf.local_variables_initializer()	initialize local variables
with tf.device("/cpu:0"): **v = tf.Variable()**	pin variable to CPU
with tf.device("/gpu:0"): **v = tf.Variable()**	pin variable to GPU

sess = tf.Session()
sess.run()
sess.close()
	run session

with tf.Session() as session: **session.run()**	run session(2)

saver=tf.train.Saver()
saver.save(sess,'file_name')
saver.restore(sess,'file_name')
	Saving and restoring variables.

Working with Data:

tf.decode_csv()	converts csv to tensors
tf.read_file()	reads file
tf.write_file()	writes to file
tf.train.batch()	creates batches of tensors

If you need more detailed information please visit WWW.TENSORFLOW.ORG all above information have been sourced there.