#_ important Tensorflow Operations [+100]

Basic Operations:

- tf.add(): Addition of tensors.
- tf.subtract(): Subtraction of tensors.
- tf.multiply(): Multiplication of tensors.
- tf.divide(): Division of tensors.
- tf.square(): Square of tensors.
- tf.sqrt(): Square root of tensors.

Tensor Manipulations:

- tf.reshape(): Reshape a tensor.
- tf.transpose(): Transpose a tensor.
- tf.squeeze(): Remove dimensions of size 1.
- tf.expand_dims(): Add a new dimension.
- tf.concat(): Concatenate tensors.
- tf.split(): Split a tensor into sub-tensors.

Variable and Initialization:

- tf.Variable(): Create a variable.
- tf.global_variables_initializer(): Initialize all global variables.

Math Operations:

- tf.reduce_sum(): Compute the sum of elements.
- tf.reduce_mean(): Compute the mean of elements.
- tf.reduce_max(): Compute the maximum of elements.
- tf.math.log(): Natural logarithm.
- tf.exp(): Exponential.

Matrix Operations:

- tf.linalg.matmul(): Matrix multiplication.

- tf.linalg.inv(): Matrix inverse.
- tf.linalg.det(): Matrix determinant.

Neural Network Building Blocks:

- tf.nn.relu(): Rectified linear unit 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 regularization.

Loss Functions:

- tf.losses.mean_squared_error(): Mean squared error loss.
- tf.losses.binary_crossentropy(): Binary cross-entropy loss.
- tf.losses.categorical_crossentropy(): Categorical cross-entropy loss.

Optimizers:

- tf.train.GradientDescentOptimizer(): Gradient descent optimizer.
- tf.train.AdamOptimizer(): Adam optimizer.
- tf.train.RMSPropOptimizer(): RMSProp optimizer.

Data Pipeline:

- tf.data.Dataset.from_tensor_slices(): Create a dataset.
- tf.data.Dataset.batch(): Batch elements of a dataset.
- tf.data.Dataset.shuffle(): Shuffle elements of a dataset.

Computational **Graph and** Sessions:

- tf.Graph(): Represents a collection of tf.Operations.
- tf.Session(): Encapsulates the environment in which Operations are executed.

Image **Operations**:

- tf.image.resize(): Resize images.

By: Waleed Mousa

- tf.image.flip_left_right(): Flip an image horizontally.
- tf.image.flip_up_down(): Flip an image vertically.
- tf.image.rgb_to_grayscale(): Convert RGB to grayscale.

I/O Operations:

- tf.read_file(): Read a file.
- tf.write_file(): Write to a file.
- tf.train.Saver(): Save and restore variables.

Control Flow:

- tf.cond(): Conditional operation.
- tf.while_loop(): While loop operation.

TensorBoard Visualization:

- tf.summary.FileWriter(): Write summaries for TensorBoard.
- tf.summary.scalar(): Log a scalar for TensorBoard.

Advanced Gradient Techniques:

- tf.gradients(): Compute the gradient of a tensor.
- tf.stop_gradient(): Stop gradient computation.
- tf.hessians(): Compute the Hessian matrix.

Regularization:

- tf.nn.l2_loss(): L2 regularization loss.

TensorFlow Extended (TFX):

- tfx.v1.TFRecordDataset(): Read data from TFRecord.
- tfx.v1.encode_csv(): Encode data as CSV.

TensorFlow Hub:

- hub.Module(): Use a reusable module.

Transfer Learning with Keras Applications:

- `tf.keras.applications.VGG16()`: VGG16 model.
- `tf.keras.applications.ResNet50()`: ResNet50 model.

TensorFlow Lite:

- `tf.lite.TFLiteConverter()`: Convert a model to TFLite format.

TensorFlow Estimator:

- `tf.estimator.Estimator()`: High-level API for TensorFlow.
- `tf.estimator.inputs.numpy_input_fn()`: Input function using numpy.

Custom Layers with Keras:

- `tf.keras.layers.Layer()`: Base layer class.

TensorFlow Functions and AutoGraph:

- `tf.function()`: Convert Python functions to TensorFlow computation.

Debugging:

- `tf.debugging.assert_equal()`: Assert tensors are equal.
- `tf.debugging.check_numerics()`: Check for NaN and Inf values.

Random Operations:

- `tf.random.normal()`: Outputs random values from a normal distribution.
- `tf.random.uniform()`: Outputs random values from a uniform distribution.

Training Utilities:

- `tf.train.get_global_step()`: Get the global step value.
- `tf.train.exponential_decay()`: Apply exponential decay to learning rate.

By: Waleed Mousa

Advanced Neural Network Components:

- tf.nn.batch_normalization(): Batch normalization.
- tf.nn.conv2d(): 2D convolution.

Model Saving and Loading:

- tf.saved_model.save(): Save a TensorFlow model.
- tf.saved_model.load(): Load a TensorFlow model.

Quantization:

- tf.quantization.quantize(): Quantize tensor.

Transforms and Filters:

- tf.signal.fft(): Compute fast Fourier transform.

TensorFlow Text:

- tf.text.WhitespaceTokenizer(): Tokenizer based on whitespace.

Sparse Tensor Operations:

- tf.sparse.SparseTensor(): Sparse tensor representation.
- tf.sparse.add(): Add sparse tensors.

Keras Callbacks:

- tf.keras.callbacks.EarlyStopping(): Stop training when a monitored metric has stopped improving.

Custom Training Loops:

- tf.GradientTape(): Record operations for automatic differentiation.

Distributed Training:

- tf.distribute.MirroredStrategy(): Synchronous training on multiple GPUs.

Utilities:

- tf.version.VERSION: Get TensorFlow version.

Custom Gradients:

- tf.custom_gradient(): Define custom gradient for an operation.

Model Pruning:

- tf.contrib.model_pruning.get_masked_weights(): Get masked weights for pruning.

Advanced Optimizers:

- tf.train.FtrlOptimizer(): FTRL optimizer.

TensorFlow Agents:

- tf_agents.environments.TFPyEnvironment(): TensorFlow environment for RL agents.

TensorFlow Graphics:

- tf_graphics.geometry.transformation.rotation_matrix_3d.from_euler(): Convert Euler angles to rotation matrix.

Advanced Activations:

- tf.keras.layers.LeakyReLU(): Leaky version of a Rectified Linear Unit.

Metrics and Evaluations:

- tf.metrics.Accuracy(): Calculates how often predictions equal labels.
- tf.metrics.Precision(): Precision metric.

TensorFlow Federated:

- tff.federated_computation(): Federated computation decorator.

TensorFlow Serving:

- tf_serving.apis.PredictionService(): Serving API for predictions.

Recurrent Neural Network Layers:

- tf.keras.layers.LSTM(): Long Short-Term Memory layer.

TensorFlow Privacy:

- privacy.optimizers.dp_optimizer.DPGradientDescentGaussianOptimizer(): Differential privacy optimizer.

Utilities:

- tf.test.is_gpu_available(): Check if TensorFlow can access a GPU.
- tf.test.is_built_with_cuda(): Check if TensorFlow was built with CUDA (GPU) support.

Convolutions:

- tf.nn.depthwise_conv2d(): Depthwise 2D convolution.

TensorFlow Decision Forests:

- tfdf.keras.RandomForestModel(): Random forest model in TF Decision Forests.

Regular Expressions:

- `tf.strings.regex_replace()`: Replace elements of input matching regex pattern.

Math Utilities:

- `tf.math.segment_sum()`: Compute sum of segments of a tensor.
- `tf.math.unsorted_segment_mean()`: Compute unsorted segment mean of a tensor.