Module: tf.contrib.framework

Defined in `tensorflow/contrib/framework/__init__.py`.

Framework utilities.

See the Framework (contrib) guide.

Modules

`nest` module: ## Functions for working with arbitrarily nested sequences of elements.

Classes

`class VariableDeviceChooser`: Device chooser for variables.

Functions

`add_arg_scope(...)`: Decorates a function with args so it can be used within an arg_scope.

`add_model_variable(...)`: Adds a variable to the `GraphKeys.MODEL_VARIABLES` collection.

`arg_scope(...)`: Stores the default arguments for the given set of list_ops.

`arg_scoped_arguments(...)`: Returns the list kwargs that arg_scope can set for a func.

`assert_global_step(...)`: DEPRECATED FUNCTION

`assert_or_get_global_step(...)`: Verifies that a global step tensor is valid or gets one if None is given.

`assert_same_float_dtype(...)`: Validate and return float type based on `tensors` and `dtype`.

`assert_scalar(...)`

`assert_scalar_int(...)`: Assert `tensor` is 0-D, of type `tf.int32` or `tf.int64`.

`assign_from_checkpoint(...)`: Creates an operation to assign specific variables from a checkpoint.

`assign_from_checkpoint_fn(...)`: Returns a function that assigns specific variables from a checkpoint.

`assign_from_values(...)`: Creates an assignment operation from a given mapping.

`assign_from_values_fn(...)`: Returns a function that assigns specific variables from the given values.

`convert_to_tensor_or_sparse_tensor(...)`: Converts value to a `SparseTensor` or `Tensor`.

`create_global_step(...)` : Create global step tensor in graph. (deprecated)

`deprecated(...)` : Decorator for marking functions or methods deprecated.

`deprecated_arg_values(...)` : Decorator for marking specific function argument values as deprecated.

`deprecated_args(...)` : Decorator for marking specific function arguments as deprecated.

`filter_variables(...)` : Filter a list of variables using regular expressions.

`get_global_step(...)` : DEPRECATED FUNCTION

`get_graph_from_inputs(...)` : Returns the appropriate graph to use for the given inputs.

`get_local_variables(...)` : Gets the list of local variables, filtered by scope and/or suffix.

`get_model_variables(...)` : Gets the list of model variables, filtered by scope and/or suffix.

`get_name_scope(...)` : Returns the current name scope of the default graph.

`get_or_create_global_step(...)` : Returns and create (if necessary) the global step tensor. (deprecated)

`get_trainable_variables(...)` : Gets the list of trainable variables, filtered by scope and/or suffix.

`get_unique_variable(...)` : Gets the variable uniquely identified by that var_op_name.

`get_variable_full_name(...)` : Returns the full name of a variable.

`get_variables(...)` : Gets the list of variables, filtered by scope and/or suffix.

`get_variables_by_name(...)` : Gets the list of variables that were given that name.

`get_variables_by_suffix(...)` : Gets the list of variables that end with the given suffix.

`get_variables_to_restore(...)` : Gets the list of the variables to restore.

`has_arg_scope(...)` : Checks whether a func has been decorated with @add_arg_scope or not.

`init_from_checkpoint(...)` : Using assignment map initializes current variables with loaded tensors.

`is_tensor(...)` : Check whether `x` is of tensor type.

`list_variables(...)` : Returns list of all variables in the latest checkpoint.

`load_and_remap_matrix_initializer(...)` : Returns a var initializer for loading and remapping a 2-D (matrix) tensor.

`load_checkpoint(...)` : Returns CheckpointReader for latest checkpoint.

`load_embedding_initializer(...)` : Returns a variable initializer for loading pre-trained embeddings.

`load_linear_multiclass_bias_initializer(...)` : Loads pre-trained multi-class biases for linear models from checkpoint.

`load_variable(...)` : Returns a Tensor with the contents of the given variable in the checkpoint.

`load_variable_slot_initializer(...)` : Loads pre-trained multi-class slots for linear models from checkpoint.

`local_variable(...)` : Create variable and add it to `GraphKeys.LOCAL_VARIABLES` collection.

`model_variable(...)` : Gets an existing model variable with these parameters or creates a new one.

`prepend_name_scope(...)` : Prepends name scope to a name.

`reduce_sum_n(...)` : Reduce tensors to a scalar sum.

`remove_squeezable_dimensions(...)` : Squeeze last dim if ranks of `predictions` and `labels` differ by 1. (deprecated)

`strip_name_scope(...)` : Removes name scope from a name.

`variable(...)` : Gets an existing variable with these parameters or creates a new one.

`with_same_shape(...)` : Assert tensors are the same shape, from the same graph.

`with_shape(...)` : Asserts tensor has expected shape.

`zero_initializer(...)` : Initialize 'ref' with all zeros, ref tensor should be uninitialized.

Last updated November 2, 2017.