tf.contrib.layers.fully_connected

```
fully_connected(
    inputs,
    num_outputs,
    activation_fn=tf.nn.relu,
    normalizer_fn=None,
    normalizer_params=None,
    weights_initializer=initializers.xavier_initializer(),
    weights_regularizer=None,
    biases_initializer=tf.zeros_initializer(),
    biases_regularizer=None,
    reuse=None,
    variables_collections=None,
    outputs_collections=None,
    trainable=True,
    scope=None
)
```

Defined in `tensorflow/contrib/layers/python/layers/layers.py`.

See the guide: Layers (contrib) > Higher level ops for building neural network layers

Adds a fully connected layer.

`fully_connected` creates a variable called `weights`, representing a fully connected weight matrix, which is multiplied by the `inputs` to produce a `Tensor` of hidden units. If a `normalizer_fn` is provided (such as `batch_norm`), it is then applied. Otherwise, if `normalizer_fn` is None and a `biases_initializer` is provided then a `biases` variable would be created and added the hidden units. Finally, if `activation_fn` is not `None`, it is applied to the hidden units as well.

> **Note:** that if `inputs` have a rank greater than 2, then `inputs` is flattened prior to the initial matrix multiply by `weights`.

Args:

- `inputs`: A tensor of at least rank 2 and static value for the last dimension; i.e. `[batch_size, depth]`, `[None, None, None, channels]`.
- `num_outputs`: Integer or long, the number of output units in the layer.
- `activation_fn`: Activation function. The default value is a ReLU function. Explicitly set it to None to skip it and maintain a linear activation.
- `normalizer_fn`: Normalization function to use instead of `biases`. If `normalizer_fn` is provided then `biases_initializer` and `biases_regularizer` are ignored and `biases` are not created nor added. default set to None for no normalizer function
- `normalizer_params`: Normalization function parameters.
- `weights_initializer`: An initializer for the weights.
- `weights_regularizer`: Optional regularizer for the weights.
- `biases_initializer`: An initializer for the biases. If None skip biases.
- `biases_regularizer`: Optional regularizer for the biases.
- `reuse`: Whether or not the layer and its variables should be reused. To be able to reuse the layer scope must be

given.

- `variables_collections` : Optional list of collections for all the variables or a dictionary containing a different list of collections per variable.
- `outputs_collections` : Collection to add the outputs.
- `trainable` : If `True` also add variables to the graph collection `GraphKeys.TRAINABLE_VARIABLES` (see tf.Variable).
- `scope` : Optional scope for variable_scope.

Returns:

The tensor variable representing the result of the series of operations.

Raises:

- `ValueError` : If x has rank less than 2 or if its last dimension is not set.

Last updated November 2, 2017.

Stay Connected

Blog

GitHub

Twitter

Support

Issue Tracker

Release Notes

Stack Overflow

English

Terms | **Privacy**