tf.contrib.layers.conv2d_transpose

Contents

Aliases:

- `tf.contrib.layers.conv2d_transpose`
- `tf.contrib.layers.convolution2d_transpose`

```
conv2d_transpose(
    inputs,
    num_outputs,
    kernel_size,
    stride=1,
    padding='SAME',
    data_format=DATA_FORMAT_NHWC,
    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 convolution2d_transpose with an optional batch normalization layer.

The function creates a variable called `weights`, representing the kernel, that is convolved with the input. If `normalizer_fn` is `None`, a second variable called 'biases' is added to the result of the operation.

Args:

- `inputs`: A 4-D `Tensor` of type `float` and shape `[batch, height, width, in_channels]` for `NHWC` data format or `[batch, in_channels, height, width]` for `NCHW` data format.
- `num_outputs`: Integer, the number of output filters.
- `kernel_size`: A list of length 2 holding the [kernel_height, kernel_width] of of the filters. Can be an int if both values are the same.
- `stride`: A list of length 2: [stride_height, stride_width]. Can be an int if both strides are the same. Note that presently both strides must have the same value.
- `padding`: One of 'VALID' or 'SAME'.

- `data_format` : A string. `NHWC` (default) and `NCHW` are supported.
- `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 collection per variable.
- `outputs_collections` : Collection to add the outputs.
- `trainable` : Whether or not the variables should be trainable or not.
- `scope` : Optional scope for variable_scope.

Returns:

A tensor representing the output of the operation.

Raises:

- `ValueError` : If 'kernel_size' is not a list of length 2.
- `ValueError` : If `data_format` is neither `NHWC` nor `NCHW` .
- `ValueError` : If `C` dimension of `inputs` is None.

Last updated November 2, 2017.

English

Terms | Privacy