

## tf.nn.conv1d

```
conv1d(  
    value,  
    filters,  
    stride,  
    padding,  
    use_cudnn_on_gpu=None,  
    data_format=None,  
    name=None  
)
```

Defined in [tensorflow/python/ops/nn\\_ops.py](#).

See the guide: [Neural Network > Convolution](#)

Computes a 1-D convolution given 3-D input and filter tensors.

Given an input tensor of shape [batch, in\_width, in\_channels] if data\_format is "NHWC", or [batch, in\_channels, in\_width] if data\_format is "NCHW", and a filter / kernel tensor of shape [filter\_width, in\_channels, out\_channels], this op reshapes the arguments to pass them to conv2d to perform the equivalent convolution operation.

Internally, this op reshapes the input tensors and invokes `tf.nn.conv2d`. For example, if `data_format` does not start with "NC", a tensor of shape [batch, in\_width, in\_channels] is reshaped to [batch, 1, in\_width, in\_channels], and the filter is reshaped to [1, filter\_width, in\_channels, out\_channels]. The result is then reshaped back to [batch, out\_width, out\_channels] (where out\_width is a function of the stride and padding as in conv2d) and returned to the caller.

## Args:

- `value`: A 3D **Tensor**. Must be of type `float32` or `float64`.
- `filters`: A 3D **Tensor**. Must have the same type as `input`.
- `stride`: An **integer**. The number of entries by which the filter is moved right at each step.
- `padding`: 'SAME' or 'VALID'
- `use_cudnn_on_gpu`: An optional **bool**. Defaults to `True`.
- `data_format`: An optional **string** from "NHWC", "NCHW". Defaults to "NHWC", the data is stored in the order of [batch, in\_width, in\_channels]. The "NCHW" format stores data as [batch, in\_channels, in\_width].
- `name`: A name for the operation (optional).

## Returns:

A **Tensor**. Has the same type as input.

## Raises:

- `ValueError`: if `data_format` is invalid.

Stay Connected

- Blog
- GitHub
- Twitter

Support

- Issue Tracker
- Release Notes
- Stack Overflow