

MaxPooling1D

[\[source\]](#)

```
keras.layers.pooling.MaxPooling1D(pool_length=2, stride=None, border_mode='valid')
```

Max pooling operation for temporal data.

Input shape

3D tensor with shape: `(samples, steps, features)`.

Output shape

3D tensor with shape: `(samples, downsampled_steps, features)`.

Arguments

- **pool_length**: factor by which to downscale. 2 will halve the input.
- **stride**: integer, or None. Stride value. If None, it will default to `pool_length`.
- **border_mode**: 'valid' or 'same'.
 - **Note**: 'same' will only work with TensorFlow for the time being.

MaxPooling2D

[\[source\]](#)

```
keras.layers.pooling.MaxPooling2D(pool_size=(2, 2), strides=None, border_mode='valid', dim_ordering='th')
```

Max pooling operation for spatial data.

Arguments

- **pool_size**: tuple of 2 integers, factors by which to downscale (vertical, horizontal). (2, 2) will halve the image in each dimension.
- **strides**: tuple of 2 integers, or None. Strides values. If None, it will default to `pool_size`.
- **border_mode**: 'valid' or 'same'.
 - **Note**: 'same' will only work with TensorFlow for the time being.
- **dim_ordering**: 'th' or 'tf'. In 'th' mode, the channels dimension (the depth) is at index 1, in 'tf' mode is it at index 3. It defaults to the `image_dim_ordering` value found in your Keras config file at `~/.keras/keras.json`. If you never set it, then it will be "th".

Input shape

4D tensor with shape: `(samples, channels, rows, cols)` if `dim_ordering='th'` or 4D tensor with shape: `(samples, rows, cols, channels)` if `dim_ordering='tf'`.

Output shape

4D tensor with shape: `(nb_samples, channels, pooled_rows, pooled_cols)` if `dim_ordering='th'` or 4D tensor with shape: `(samples, pooled_rows, pooled_cols, channels)` if `dim_ordering='tf'`.

MaxPooling3D

[\[source\]](#)

```
keras.layers.pooling.MaxPooling3D(pool_size=(2, 2, 2), strides=None, border_mode='valid', dim_o
```

Max pooling operation for 3D data (spatial or spatio-temporal).

Arguments

- **pool_size**: tuple of 3 integers, factors by which to downscale (dim1, dim2, dim3). (2, 2, 2) will halve the size of the 3D input in each dimension.
- **strides**: tuple of 3 integers, or None. Strides values.
- **border_mode**: 'valid' or 'same'.
- **dim_ordering**: 'th' or 'tf'. In 'th' mode, the channels dimension (the depth) is at index 1, in 'tf' mode is it at index 4. It defaults to the `image_dim_ordering` value found in your Keras config file at `~/.keras/keras.json`. If you never set it, then it will be "th".

Input shape

5D tensor with shape: `(samples, channels, len_pool_dim1, len_pool_dim2, len_pool_dim3)` if `dim_ordering='th'` or 5D tensor with shape: `(samples, len_pool_dim1, len_pool_dim2, len_pool_dim3, channels)` if `dim_ordering='tf'`.

Output shape

5D tensor with shape: `(nb_samples, channels, pooled_dim1, pooled_dim2, pooled_dim3)` if `dim_ordering='th'` or 5D tensor with shape: `(samples, pooled_dim1, pooled_dim2, pooled_dim3, channels)` if `dim_ordering='tf'`.

AveragePooling1D

[\[source\]](#)

```
keras.layers.pooling.AveragePooling1D(pool_length=2, stride=None, border_mode='valid')
```

Average pooling for temporal data.

Arguments

- **pool_length**: factor by which to downscale. 2 will halve the input.
- **stride**: integer, or None. Stride value. If None, it will default to `pool_length`.
- **border_mode**: 'valid' or 'same'.
 - **Note**: 'same' will only work with TensorFlow for the time being.

Input shape

3D tensor with shape: `(samples, steps, features)`.

Output shape

3D tensor with shape: `(samples, downsampled_steps, features)`.

AveragePooling2D

[\[source\]](#)

```
keras.layers.pooling.AveragePooling2D(pool_size=(2, 2), strides=None, border_mode='valid', dim_
```

Average pooling operation for spatial data.

Arguments

- **pool_size**: tuple of 2 integers, factors by which to downscale (vertical, horizontal). (2, 2) will halve the image in each dimension.
- **strides**: tuple of 2 integers, or None. Strides values. If None, it will default to `pool_size`.
- **border_mode**: 'valid' or 'same'.
 - **Note**: 'same' will only work with TensorFlow for the time being.
- **dim_ordering**: 'th' or 'tf'. In 'th' mode, the channels dimension (the depth) is at index 1, in 'tf' mode is it at index 3. It defaults to the `image_dim_ordering` value found in your Keras config file at `~/.keras/keras.json`. If you never set it, then it will be "th".

Input shape

4D tensor with shape: `(samples, channels, rows, cols)` if `dim_ordering='th'` or 4D tensor with shape: `(samples, rows, cols, channels)` if `dim_ordering='tf'`.

Output shape

4D tensor with shape: `(nb_samples, channels, pooled_rows, pooled_cols)` if `dim_ordering='th'` or 4D tensor with shape: `(samples, pooled_rows, pooled_cols, channels)` if `dim_ordering='tf'`.

AveragePooling3D

[\[source\]](#)

```
keras.layers.pooling.AveragePooling3D(pool_size=(2, 2, 2), strides=None, border_mode='valid', d
```

Average pooling operation for 3D data (spatial or spatio-temporal).

Arguments

- **pool_size**: tuple of 3 integers, factors by which to downscale (dim1, dim2, dim3). (2, 2, 2) will halve the size of the 3D input in each dimension.
- **strides**: tuple of 3 integers, or None. Strides values.
- **border_mode**: 'valid' or 'same'.
- **dim_ordering**: 'th' or 'tf'. In 'th' mode, the channels dimension (the depth) is at index 1, in 'tf' mode is it at index 4. It defaults to the `image_dim_ordering` value found in your Keras config file at `~/.keras/keras.json`. If you never set it, then it will be "th".

Input shape

5D tensor with shape: `(samples, channels, len_pool_dim1, len_pool_dim2, len_pool_dim3)` if `dim_ordering='th'` or 5D tensor with shape: `(samples, len_pool_dim1, len_pool_dim2, len_pool_dim3, channels)` if `dim_ordering='tf'`.

Output shape

5D tensor with shape: `(nb_samples, channels, pooled_dim1, pooled_dim2, pooled_dim3)` if `dim_ordering='th'` or 5D tensor with shape: `(samples, pooled_dim1, pooled_dim2, pooled_dim3, channels)` if `dim_ordering='tf'`.