[[flatten]](https://keras.io/layers/core/" \l "flatten)

tf.keras.layers.Flatten()

[[fully connected]](https://keras.io/layers/core/#dense)

keras.layers.Dense(units,

activation=None,

use\_bias=True,

kernel\_initializer='glorot\_uniform',

bias\_initializer='zeros',

kernel\_regularizer=None,

bias\_regularizer=None,

activity\_regularizer=None,

kernel\_constraint=None,

bias\_constraint=None)

* **units**: Positive integer, dimensionality of the output space.
* **activation**: Activation function to use (see activations). If you don't specify anything, no activation is applied (ie. "linear" activation: a(x) = x).
* **use\_bias**: Boolean, whether the layer uses a bias vector.
* **kernel\_initializer**: Initializer for the kernel weights matrix (see initializers).
* **bias\_initializer**: Initializer for the bias vector (see initializers).
* **kernel\_regularizer**: Regularizer function applied to the kernel weights matrix (see regularizer).
* **bias\_regularizer**: Regularizer function applied to the bias vector (see regularizer).
* **activity\_regularizer**: Regularizer function applied to the output of the layer (its "activation"). (see regularizer).
* **kernel\_constraint**: Constraint function applied to the kernel weights matrix (see constraints).
* **bias\_constraint**: Constraint function applied to the bias vector (see constraints).

[[2D convolution]](https://keras.io/layers/convolutional/#conv2d)

tf.keras.layers.Conv2D(filters,

kernel\_size,

strides=(1, 1),

padding='valid',

dilation\_rate=(1, 1),

activation=None,

use\_bias=True,

kernel\_initializer='glorot\_uniform',

bias\_initializer='zeros',

kernel\_regularizer=None,

bias\_regularizer=None,

activity\_regularizer=None,

kernel\_constraint=None,

bias\_constraint=None)

* **filters**: Integer, the dimensionality of the output space (i.e. the number of output filters in the convolution).
* **kernel\_size**: An integer or tuple/list of 2 integers, specifying the height and width of the 2D convolution window. Can be a single integer to specify the same value for all spatial dimensions.
* **strides**: An integer or tuple/list of 2 integers, specifying the strides of the convolution along the height and width. Can be a single integer to specify the same value for all spatial dimensions. Specifying any stride value != 1 is incompatible with specifying any dilation\_rate value != 1.
* **padding**: one of "valid" or "same" (case-insensitive). Note that "same" is slightly inconsistent across backends with strides != 1, as described here
* **dilation\_rate**: an integer or tuple/list of 2 integers, specifying the dilation rate to use for dilated convolution. Can be a single integer to specify the same value for all spatial dimensions. Currently, specifying any dilation\_rate value != 1 is incompatible with specifying any stride value != 1.
* **activation**: Activation function to use (see activations). If you don't specify anything, no activation is applied (ie. "linear" activation: a(x) = x).
* **use\_bias**: Boolean, whether the layer uses a bias vector.
* **kernel\_initializer**: Initializer for the kernel weights matrix (see initializers).
* **bias\_initializer**: Initializer for the bias vector (see initializers).
* **kernel\_regularizer**: Regularizer function applied to the kernel weights matrix (see regularizer).
* **bias\_regularizer**: Regularizer function applied to the bias vector (see regularizer).
* **activity\_regularizer**: Regularizer function applied to the output of the layer (its "activation"). (see regularizer).
* **kernel\_constraint**: Constraint function applied to the kernel matrix (see constraints).
* **bias\_constraint**: Constraint function applied to the bias vector (see constraints).

[[2D maxpooling]](https://keras.io/layers/pooling/#maxpooling2d)

keras.layers.MaxPooling2D(pool\_size=(2, 2),

strides=None,

padding='valid')

* **pool\_size**: integer or tuple of 2 integers, factors by which to downscale (vertical, horizontal). (2, 2) will halve the input in both spatial dimension. If only one integer is specified, the same window length will be used for both dimensions.
* **strides**: Integer, tuple of 2 integers, or None. Strides values. If None, it will default to pool\_size.
* **padding**: One of "valid" or "same" (case-insensitive).

[[batch normalization]](https://keras.io/layers/normalization/)

keras.layers.BatchNormalization(axis=-1,

momentum=0.99,

epsilon=0.001,

center=True,

scale=True,

beta\_initializer='zeros',

gamma\_initializer='ones',

moving\_mean\_initializer='zeros',

moving\_variance\_initializer='ones',

beta\_regularizer=None,

gamma\_regularizer=None,

beta\_constraint=None,

gamma\_constraint=None)

* **axis:** Integer, the axis that should be normalized (typically the features axis). For instance, after a Conv2D layer with data\_format="channels\_first", set axis=1 in BatchNormalization.
* **momentum**: Momentum for the moving mean and the moving variance.
* **epsilon**: Small float added to variance to avoid dividing by zero.
* **center**: If True, add offset of beta to normalized tensor. If False, beta is ignored.
* **scale**: If True, multiply by gamma. If False, gamma is not used. When the next layer is linear (also e.g. nn.relu), this can be disabled since the scaling will be done by the next layer.
* **beta\_initializer**: Initializer for the beta weight.
* **gamma\_initializer**: Initializer for the gamma weight.
* **moving\_mean\_initializer**: Initializer for the moving mean.
* **moving\_variance\_initializer**: Initializer for the moving variance.
* **beta\_regularizer**: Optional regularizer for the beta weight.
* **gamma\_regularizer**: Optional regularizer for the gamma weight.
* **beta\_constraint**: Optional constraint for the beta weight.
* **gamma\_constraint**: Optional constraint for the gamma weight.

[[add]](https://keras.io/layers/merge/#add)

keras.layers.Add(<list\_of\_tensor>)

* **<list\_of\_tensor>**: the list of tensor that need to be added up