

BatchNormalization

[\[source\]](#)

```
keras.layers.normalization.BatchNormalization(epsilon=1e-06, mode=0, axis=-1, momentum=0.99, we
```

Normalize the activations of the previous layer at each batch, i.e. applies a transformation that maintains the mean activation close to 0 and the activation standard deviation close to 1.

Arguments

- **epsilon**: small float > 0. Fuzz parameter.
- **mode**: integer, 0, 1 or 2.
 - 0: feature-wise normalization. Each feature map in the input will be normalized separately. The axis on which to normalize is specified by the `axis` argument. Note that if the input is a 4D image tensor using Theano conventions (samples, channels, rows, cols) then you should set `axis` to `1` to normalize along the channels axis. During training we use per-batch statistics to normalize the data, and during testing we use running averages computed during the training phase.
 - 1: sample-wise normalization. This mode assumes a 2D input.
 - 2: feature-wise normalization, like mode 0, but using per-batch statistics to normalize the data during both testing and training.
- **axis**: integer, axis along which to normalize in mode 0. For instance, if your input tensor has shape (samples, channels, rows, cols), set axis to 1 to normalize per feature map (channels axis).
- **momentum**: momentum in the computation of the exponential average of the mean and standard deviation of the data, for feature-wise normalization.
- **weights**: Initialization weights. List of 2 Numpy arrays, with shapes: `[(input_shape,), (input_shape,)]`. Note that the order of this list is [gamma, beta, mean, std]
- **beta_init**: name of initialization function for shift parameter (see [initializations](#)), or alternatively, Theano/TensorFlow function to use for weights initialization. This parameter is only relevant if you don't pass a `weights` argument.
- **gamma_init**: name of initialization function for scale parameter (see [initializations](#)), or alternatively, Theano/TensorFlow function to use for weights initialization. This parameter is only relevant if you don't pass a `weights` argument.

Input shape

Arbitrary. Use the keyword argument `input_shape` (tuple of integers, does not include the samples axis) when using this layer as the first layer in a model.

Output shape

Same shape as input.

References

- [Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift](#)