

tf.keras.backend.batch_dot

```
batch_dot(
    x,
    y,
    axes=None
)
```

Defined in [tensorflow/python/keras/_impl/keras/backend.py](#).

Batchwise dot product.

batch_dot is used to compute dot product of **x** and **y** when **x** and **y** are data in batch, i.e. in a shape of **(batch_size, :)**. **batch_dot** results in a tensor or variable with less dimensions than the input. If the number of dimensions is reduced to 1, we use **expand_dims** to make sure that ndim is at least 2.

Arguments:

- **x**: Keras tensor or variable with **ndim** ≥ 2 .
- **y**: Keras tensor or variable with **ndim** ≥ 2 .
- **axes**: list of (or single) int with target dimensions. The lengths of **axes[0]** and **axes[1]** should be the same.

Returns:

A tensor with shape equal to the concatenation of **x**'s shape (less the dimension that was summed over) and **y**'s shape (less the batch dimension and the dimension that was summed over). If the final rank is 1, we reshape it to **(batch_size, 1)**.

Examples: Assume **x** = **[[[1, 2], [3, 4]]]** and **y** = **[[[5, 6], [7, 8]]]** **batch_dot(x, y, axes=1)** = **[[17, 53]]** which is the main diagonal of **x.dot(y.T)**, although we never have to calculate the off-diagonal elements.

Shape inference:

Let **x**'s shape be **(100, 20)** and **y**'s shape be **(100, 30, 20)**.
If **axes** is (1, 2), to find the output shape of resultant tensor,
loop through each dimension in **x**'s shape and **y**'s shape:

```
* `x.shape[0]` : 100 : append to output shape
* `x.shape[1]` : 20 : do not append to output shape,
    dimension 1 of `x` has been summed over. (`dot_axes[0]` = 1)
* `y.shape[0]` : 100 : do not append to output shape,
    always ignore first dimension of `y`
* `y.shape[1]` : 30 : append to output shape
* `y.shape[2]` : 20 : do not append to output shape,
    dimension 2 of `y` has been summed over. (`dot_axes[1]` = 2)
`output_shape` = `(100, 30)`
```

```
>>> x_batch = K.ones(shape=(32, 20, 1))
>>> y_batch = K.ones(shape=(32, 30, 20))
>>> xy_batch_dot = K.batch_dot(x_batch, y_batch, axes=[1, 2])
>>> K.int_shape(xy_batch_dot)
(32, 1, 30)
```

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.

Last updated November 2, 2017.

Stay Connected

[Blog](#)

[GitHub](#)

[Twitter](#)

Support

[Issue Tracker](#)

[Release Notes](#)

[Stack Overflow](#)

English

[Terms](#) | [Privacy](#)