

Usage of activations

Activations can either be used through an `Activation` layer, or through the `activation` argument supported by all forward layers:

```
from keras.layers.core import Activation, Dense

model.add(Dense(64))
model.add(Activation('tanh'))
```

is equivalent to:

```
model.add(Dense(64, activation='tanh'))
```

You can also pass an element-wise Theano/TensorFlow function as an activation:

```
from keras import backend as K

def tanh(x):
    return K.tanh(x)

model.add(Dense(64, activation=tanh))
model.add(Activation(tanh))
```

Available activations

- **softmax**: Softmax applied across inputs last dimension. Expects shape either `(nb_samples, nb_timesteps, nb_dims)` or `(nb_samples, nb_dims)`.
- **softplus**
- **softsign**
- **relu**
- **tanh**
- **sigmoid**
- **hard_sigmoid**
- **linear**

On Advanced Activations

Activations that are more complex than a simple Theano/TensorFlow function (eg. learnable activations, configurable activations, etc.) are available as **Advanced Activation layers**, and can be found in the module `keras.layers.advanced_activations`. These include PReLU and LeakyReLU.

