Usage of initializations

Initializations define the way to set the initial random weights of Keras layers.

The keyword arguments used for passing initializations to layers will depend on the layer. Usually it is simply `init`:

```
model.add(Dense(64, init='uniform'))
```

Available initializations

- **uniform**
- **lecun_uniform**: Uniform initialization scaled by the square root of the number of inputs (LeCun 98).
- **normal**
- **identity**: Use with square 2D layers (`shape[0] == shape[1]`).
- **orthogonal**: Use with square 2D layers (`shape[0] == shape[1]`).
- **zero**
- **glorot_normal**: Gaussian initialization scaled by fan_in + fan_out (Glorot 2010)
- **glorot_uniform**
- **he_normal**: Gaussian initialization scaled by fan_in (He et al., 2014)
- **he_uniform**

An initialization may be passed as a string (must match one of the available initializations above), or as a callable. If a callable, then it must take two arguments: `shape` (shape of the variable to initialize) and `name` (name of the variable), and it must return a variable (e.g. output of `K.variable()`):

```
from keras import backend as K
import numpy as np

def my_init(shape, name=None):
    value = np.random.random(shape)
    return K.variable(value, name=name)

model.add(Dense(64, init=my_init))
```

You could also use functions from `keras.initializations` in this way:

```
from keras import initializations

def my_init(shape, name=None):
    return initializations.normal(shape, scale=0.01, name=name)

model.add(Dense(64, init=my_init))
```