Usage of regularizers

Regularizers allow to apply penalties on layer parameters or layer activity during optimization. These penalties are incorporated in the loss function that the network optimizes.

The penalties are applied on a per-layer basis. The exact API will depend on the layer, but the layers `Dense`, `TimeDistributedDense`, `MaxoutDense`, `Convolution1D` and `Convolution2D` have a unified API.

These layers expose 3 keyword arguments:

- `W_regularizer`: instance of `keras.regularizers.WeightRegularizer`
- `b_regularizer`: instance of `keras.regularizers.WeightRegularizer`
- `activity_regularizer`: instance of `keras.regularizers.ActivityRegularizer`

Example

```
from keras.regularizers import l2, activity_l2
model.add(Dense(64, input_dim=64, W_regularizer=l2(0.01), activity_regularizer=activity_l2(0.01
```

Available penalties

```
keras.regularizers.WeightRegularizer(l1=0., l2=0.)
```

```
keras.regularizers.ActivityRegularizer(l1=0., l2=0.)
```

Shortcuts

These are shortcut functions available in `keras.regularizers`.

- **l1**(l=0.01): L1 weight regularization penalty, also known as LASSO
- **l2**(l=0.01): L2 weight regularization penalty, also known as weight decay, or Ridge
- **l1l2**(l1=0.01, l2=0.01): L1-L2 weight regularization penalty, also known as ElasticNet
- **activity_l1**(l=0.01): L1 activity regularization
- **activity_l2**(l=0.01): L2 activity regularization
- **activity_l1l2**(l1=0.01, l2=0.01): L1+L2 activity regularization