

正則化の利用方法

正則化によって、最適化中にレイヤーパラメータあるいはレイヤーの出力に制約を課すことができます。これらの正則化はネットワークが最適化する損失関数に組み込まれます。

この正則化はレイヤー毎に適用されます。厳密なAPIはレイヤーに依存しますが、`Dense`、`Conv1D`、`Conv2D`、`Conv3D` レイヤーは統一的なAPIを持っています。

これらのレイヤーは3つの引数を取ります:

- `kernel_regularizer`: `keras.regularizers.Regularizer` のインスタンス
- `bias_regularizer`: `keras.regularizers.Regularizer` のインスタンス
- `activity_regularizer`: `keras.regularizers.Regularizer` のインスタンス

例

```
from keras import regularizers
model.add(Dense(64, input_dim=64,
                kernel_regularizer=regularizers.l2(0.01),
                activity_regularizer=regularizers.l1(0.01)))
```

利用可能な正則化

```
keras.regularizers.l1(0.)
keras.regularizers.l2(0.)
keras.regularizers.l1_l2(l1=0.01, l2=0.01)
```

新しい正則化の定義

重み行列から損失関数に寄与するテンソルを返す任意の関数は、正則化として利用可能です、例:

```
from keras import backend as K

def l1_reg(weight_matrix):
    return 0.01 * K.sum(K.abs(weight_matrix))

model.add(Dense(64, input_dim=64,
                kernel_regularizer=l1_reg))
```

また、オブジェクト指向的に正則化を定義できます。`keras/regularizers.py`モジュールの例を見てください。