

Régularisation par Dropout

```
def create_model(dropout_rate=0.0, weight_constraint=0):
    # creer le modele
    # passer la valeur de l'argument 'weight_constraint' à l'attribut
    'kernel_constraint' de Dense
    model.add(Dense(12, input_dim=8, kernel_initializer='uniform',
activation='linear', kernel_constraint=maxnorm(weight_constraint)))
    # ajouter une couche Dropout avec la valeur de l'argument 'dropout_rate' pour
    déterminer le taux de dropout
    model.add(Dropout(dropout_rate))
    # ajouter une couche de sortie
    # compiler le modele
    return model

# charger la base de donnees

from keras.wrappers.scikit_learn import KerasClassifier
from sklearn.model_selection import GridSearchCV
from keras.layers import Dropout
from keras.constraints import maxnorm

# creer le modele
model = KerasClassifier(build_fn=define_model, epochs=100, batch_size=10, ver-
bose=0)
# parametres de la grille de recherche
weight_constraint = [1, 2, 3, 4, 5]
dropout_rate = [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]
param_grid = dict(dropout_rate=dropout_rate, weight_constraint=weight_constraint)
grid = GridSearchCV(estimator=model, param_grid= param_grid, n_jobs=-1, cv=3)
grid_result = grid.fit(X, Y)
# afficher les resultats
print("Best: %f with %s" % (grid_result.best_score_, grid_result.best_params_))
# afficher les resultats detaillés
means = grid_result.cv_results_['mean_test_score']
stds = grid_result.cv_results_['std_test_score']
params = grid_result.cv_results_['params']
for mean, stdev, param in zip(means, stds, params):
    print("mean (+/- std) = %f (%f) with: %r" % (mean, stdev, param))
```