# *Wrappers for the Scikit-Learn API*

Reference Link: <https://keras.io/scikit-learn-api/>

**Introduction:**

The Scikit-Learn API use is used to wrap your model and you can pass it has a parameter for various inbuilt function.

*There are two wrappers available:*

* keras.wrappers.scikit\_learn.KerasClassifier(build\_fn=None, \*\*sk\_params), which implements the Scikit-Learn classifier interface,
* keras.wrappers.scikit\_learn.KerasRegressor(build\_fn=None, \*\*sk\_params), which implements the Scikit-Learn regressor interface.

Arguments

**build\_fn**: callable function or class instance

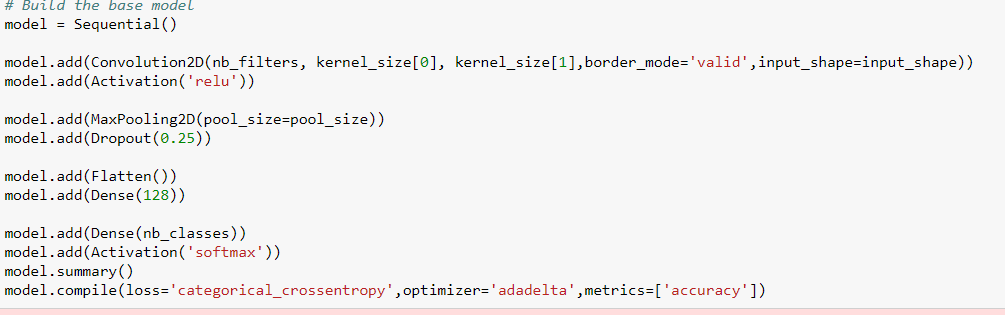
**sk\_params**: model parameters & fitting parameters

**Data Processing:**

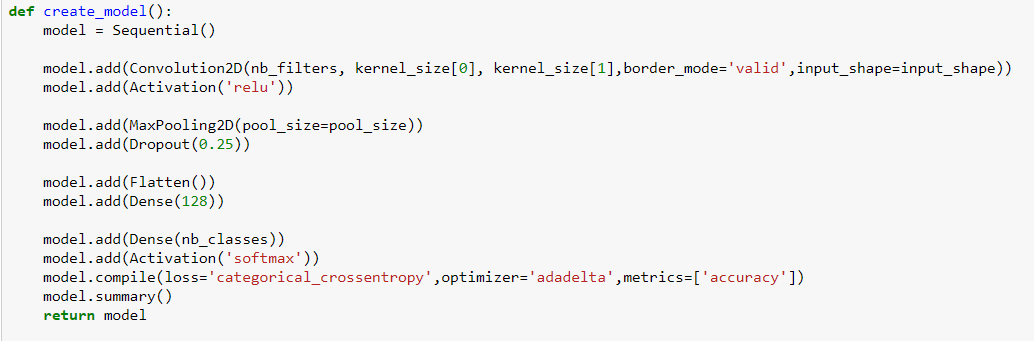
* Initially we are loading the data using cifar10.load\_data()
* Then we are splitting the data into TEST, TRAIN, VALIDATE
* We are taking only .1 % as your testing data. (It is just to explain the concept and to make your model run faster)

**Build the Base model:**

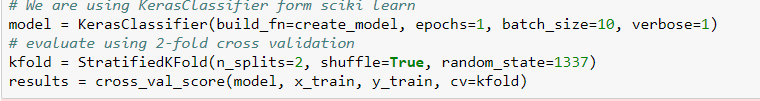
* initially we are building our own model



**Build the model inside the function to use sciki learn api effectively:**

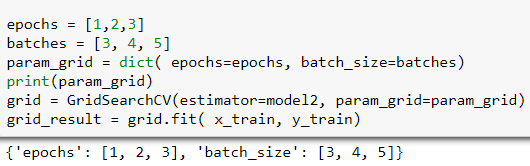


* Once we built the model we are passing as a parameter inside the kerasclassifier so that we can perform cross validation

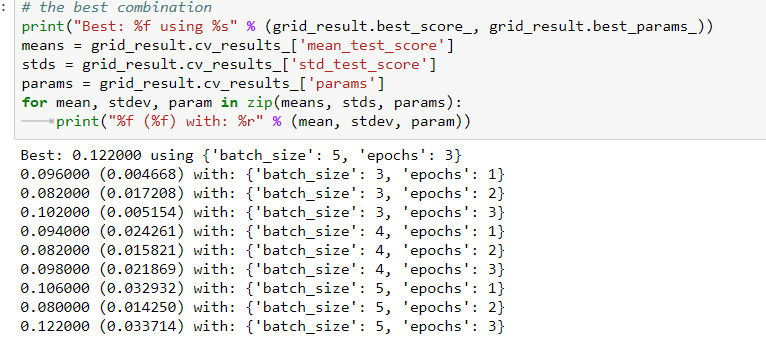


**Grid Search with multiple parameters**

* Here instead of giving a single variable as input, we can give multiple values at the same time. Gridsearchcv will do the grid search and give you the best combination.
* The options are specified into a dictionary and passed to the configuration of the [GridSearchCV](http://scikit-learn.org/stable/modules/generated/sklearn.grid_search.GridSearchCV.html) scikit-learn class. This class will evaluate a version of our neural network model for each combination of parameters (3 x 3 for the combinations of epochs and batches). Each combination is then evaluated using the default of 3-fold stratified cross validation.



* For our model there is 3\*3 = 9 combination in which gridserach will happen . At the end the model will give you the best of all the combination.



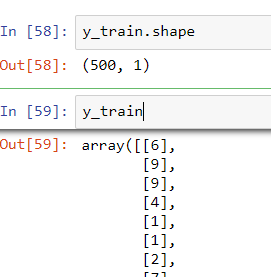
* In our case 'batch\_size': 5, 'epochs': 3 was the best of all
* Not only epoch , batch size , we can also give parameters like optimizer , pool six etc in grid search
* The problem with grid search is it will take lot of computation power

Note:

Stratified k-fold doesn't support multilabel format as it's would require to balance the proportion of positive for each label.

Multi-label emphasizes on mutually inclusive so that an observation could be members of multiple classes at the same time.

* Initial Y\_train variable



* After reshaping the y\_train variable , It looks like

