Project: scikit-learn

2014-02-24T22:53:11Z adverley 97comments

GridSearchCV parallel execution with own scorer freezes

Web url: https://github.com/scikit-learn/scikit-learn/issues/2889

API url: https://api.github.com/repos/scikit-learn/scikit-learn/issues/2889

I have been searching hours on this problem and can consistently replicate it:

```

clf = GridSearchCV( sk.LogisticRegression(),

tuned\_parameters,

cv = N\_folds\_validation,

pre\_dispatch='6\*n\_jobs',

n\_jobs=4,

verbose = 1,

scoring=metrics.make\_scorer(metrics.scorer.f1\_score, average="macro")

)

```

This snippet crashes because of scoring=metrics.make\_scorer(metrics.scorer.f1\_score, average="macro") where metrics refers to sklearn.metrics module. If I cancel out the scoring=... line, the parallel execution works. If I want to use the f1 score as evaluation method, I have to cancel out the parallel execution by setting n\_jobs = 1.

Is there a way I can define another score method without losing the parallel execution possibility?

Thanks

-------------------------------------------------------------------------

2014-02-24T23:41:49Z jnothman

This is surprising, so we'll have to work out what the problem is and make sure it works!

Can you please provide a little more detail:

- What do you mean by "crashes"?

- What version of scikit-learn is this? If it's 0.14, does it still happen in the current development version?

- Multiprocessing has platform-specific issues. What platform are you on? (e.g. `import platform; platform.platform()`)

- Have you tried it on different datasets?

FWIW, my machine has no problem fitting iris with this snippet on the development version of sklearn.

-------------------------------------------------------------------------

2014-02-25T09:40:56Z adverley

Thank you for your fast reply.

With crashing I actually mean freezing. It doesn't continue anymore and there is also no more activity to be monitored in the python process of task manager of windows. The processes are still there and consume a constant amount of RAM but require no processing time.

This is scikit-learn version 0.14, last updated and run using Enthought Canopy.

I am on platform "Windows-7-6.1.7601-SP1".

I will go more into depth by providing a generic example of the problem. I think it has to do with the GridSearchCV being placed in a for loop. (To not waste too much of your time, you should probably start at the run\_tune\_process() method which is being called at the bottom of the code and calls the method containing GridSearchCV() in a for loop)

# Code:

```

import sklearn.metrics as metrics

from sklearn.grid\_search import GridSearchCV

import numpy as np

import os

from sklearn import datasets

from sklearn import svm as sk

def tune\_hyperparameters(trainingData, period):

allDataTrain = trainingData

# Define hyperparameters and construct a dictionary of them

amount\_kernels = 2

kernels = ['rbf','linear']

gamma\_range = 10. \*\* np.arange(-5, 5)

C\_range = 10. \*\* np.arange(-5, 5)

tuned\_parameters = [

{'kernel': ['rbf'], 'gamma': gamma\_range , 'C': C\_range},

{'kernel': ['linear'], 'C': C\_range}

]

print("Tuning hyper-parameters on period = " + str(period) + "\n")

clf = GridSearchCV( sk.SVC(),

tuned\_parameters,

cv=5,

pre\_dispatch='4\*n\_jobs',

n\_jobs=2,

verbose = 1,

scoring=metrics.make\_scorer(metrics.scorer.f1\_score, average="macro")

)

clf.fit(allDataTrain[:,1:], allDataTrain[:,0:1].ravel())

# other code will output some data to files, graphs and will save the optimal model with joblib package

# Eventually we will return the optimal model

return clf

def run\_tune\_process(hyperparam\_tuning\_method,trainingData, testData):

for period in np.arange(0,100,10):

clf = hyperparam\_tuning\_method(trainingData,period)

y\_real = testData[:,0:1].ravel()

y\_pred = clf.predict(testData[:,1:])

# import some data to play with

iris = datasets.load\_iris()

X\_training = iris.data[0:100,:]

Y\_training = (iris.target[0:100]).reshape(100,1)

trainingset = np.hstack((Y\_training, X\_training))

X\_test = iris.data[100:150,:]

Y\_test = (iris.target[100:150]).reshape(50,1)

testset = np.hstack((Y\_test, X\_test))

run\_tune\_process(tune\_hyperparameters,trainingset,testset)

```

Once again, this code works on my computer only when I change n\_jobs to 1 or when I don't define a scoring= argument.

-------------------------------------------------------------------------

2014-02-25T10:52:55Z jnothman

Generally multiprocessing in Windows encounters a lot of problems. But I

don't know why this should be correlated with a custom metric. There's

nothing about the average=macro option in 0.14 that suggests it should be

more likely to hang than the default average (weighted). At the development

head, this completes in 11s on my macbook, and in 7s at version 0.14

(that's something to look into!)

Are you able to try this out in the current development version, to see if

it's still an issue?

On 25 February 2014 20:40, adverley notifications@github.com wrote:

> Thank you for your fast reply.

>

> With crashing I actually mean freezing. It doesn't continue anymore and

> there is also no more activity to be monitored in the python process of

> task manager of windows. The processes are still there and consume a

> constant amount of RAM but require no processing time.

>

> This is scikit-learn version 0.14, last updated and run using Enthought

> Canopy.

>

> I am on platform "Windows-7-6.1.7601-SP1".

>

> I will go more into depth by providing a generic example of the problem. I

> think it has to do with the GridSearchCV being placed in a for loop. (To

> not waste too much of your time, you should probably start at the

> run\_tune\_process() method which is being called at the bottom of the code

> and calls the method containing GridSearchCV() in a for loop)

> Code:

>

> import sklearn.metrics as metrics

> from sklearn.grid\_search import GridSearchCV

> import numpy as np

> import os

> from sklearn import datasets

> from sklearn import svm as sk

>

> def tune\_hyperparameters(trainingData, period):

> allDataTrain = trainingData

>

> ```

> # Define hyperparameters and construct a dictionary of them

> amount\_kernels = 2

> kernels = ['rbf','linear']

> gamma\_range = 10. \*\* np.arange(-5, 5)

> C\_range = 10. \*\* np.arange(-5, 5)

> tuned\_parameters = [

> {'kernel': ['rbf'], 'gamma': gamma\_range , 'C': C\_range},

> {'kernel': ['linear'], 'C': C\_range}

> ]

>

> print("Tuning hyper-parameters on period = " + str(period) + "\n")

>

> clf = GridSearchCV( sk.SVC(),

> tuned\_parameters,

> cv=5,

> pre\_dispatch='4\*n\_jobs',

> n\_jobs=2,

> verbose = 1,

> scoring=metrics.make\_scorer(metrics.scorer.f1\_score, average="macro")

> )

> clf.fit(allDataTrain[:,1:], allDataTrain[:,0:1].ravel())

>

> # other code will output some data to files, graphs and will save the optimal model with joblib package

>

>

> # Eventually we will return the optimal model

> return clf

> ```

>

> def run\_tune\_process(hyperparam\_tuning\_method,trainingData, testData):

> for period in np.arange(0,100,10):

> clf = hyperparam\_tuning\_method(trainingData,period)

>

> ```

> y\_real = testData[:,0:1].ravel()

> y\_pred = clf.predict(testData[:,1:])

> ```

>

> # import some data to play with

>

> iris = datasets.load\_iris()

> X\_training = iris.data[0:100,:]

> Y\_training = (iris.target[0:100]).reshape(100,1)

> trainingset = np.hstack((Y\_training, X\_training))

>

> X\_test = iris.data[100:150,:]

> Y\_test = (iris.target[100:150]).reshape(50,1)

> testset = np.hstack((Y\_test, X\_test))

>

> run\_tune\_process(tune\_hyperparameters,trainingset,testset)

>

> ##

>

> Reply to this email directly or view it on GitHubhttps://github.com/scikit-learn/scikit-learn/issues/2889#issuecomment-35990430

> .

-------------------------------------------------------------------------

2014-02-25T11:00:41Z jnothman

(As a side point, @ogrisel, I note there seems to be a lot more joblib

parallelisation overhead in master -- on OS X at least -- that wasn't there

in 0.14...)

On 25 February 2014 21:52, Joel Nothman jnothman@student.usyd.edu.auwrote:

> Generally multiprocessing in Windows encounters a lot of problems. But I

> don't know why this should be correlated with a custom metric. There's

> nothing about the average=macro option in 0.14 that suggests it should be

> more likely to hang than the default average (weighted). At the development

> head, this completes in 11s on my macbook, and in 7s at version 0.14

> (that's something to look into!)

>

> Are you able to try this out in the current development version, to see if

> it's still an issue?

>

> On 25 February 2014 20:40, adverley notifications@github.com wrote:

>

> > Thank you for your fast reply.

> >

> > With crashing I actually mean freezing. It doesn't continue anymore and

> > there is also no more activity to be monitored in the python process of

> > task manager of windows. The processes are still there and consume a

> > constant amount of RAM but require no processing time.

> >

> > This is scikit-learn version 0.14, last updated and run using Enthought

> > Canopy.

> >

> > I am on platform "Windows-7-6.1.7601-SP1".

> >

> > I will go more into depth by providing a generic example of the problem.

> > I think it has to do with the GridSearchCV being placed in a for loop. (To

> > not waste too much of your time, you should probably start at the

> > run\_tune\_process() method which is being called at the bottom of the code

> > and calls the method containing GridSearchCV() in a for loop)

> > Code:

> >

> > import sklearn.metrics as metrics

> > from sklearn.grid\_search import GridSearchCV

> > import numpy as np

> > import os

> > from sklearn import datasets

> > from sklearn import svm as sk

> >

> > def tune\_hyperparameters(trainingData, period):

> > allDataTrain = trainingData

> >

> > ```

> > # Define hyperparameters and construct a dictionary of them

> > amount\_kernels = 2

> > kernels = ['rbf','linear']

> > gamma\_range = 10. \*\* np.arange(-5, 5)

> > C\_range = 10. \*\* np.arange(-5, 5)

> > tuned\_parameters = [

> > {'kernel': ['rbf'], 'gamma': gamma\_range , 'C': C\_range},

> > {'kernel': ['linear'], 'C': C\_range}

> > ]

> >

> > print("Tuning hyper-parameters on period = " + str(period) + "\n")

> >

> > clf = GridSearchCV( sk.SVC(),

> > tuned\_parameters,

> > cv=5,

> > pre\_dispatch='4\*n\_jobs',

> > n\_jobs=2,

> > verbose = 1,

> > scoring=metrics.make\_scorer(metrics.scorer.f1\_score, average="macro")

> > )

> > clf.fit(allDataTrain[:,1:], allDataTrain[:,0:1].ravel())

> >

> > # other code will output some data to files, graphs and will save the optimal model with joblib package

> >

> >

> > # Eventually we will return the optimal model

> > return clf

> > ```

> >

> > def run\_tune\_process(hyperparam\_tuning\_method,trainingData, testData):

> > for period in np.arange(0,100,10):

> > clf = hyperparam\_tuning\_method(trainingData,period)

> >

> > ```

> > y\_real = testData[:,0:1].ravel()

> > y\_pred = clf.predict(testData[:,1:])

> > ```

> >

> > # import some data to play with

> >

> > iris = datasets.load\_iris()

> > X\_training = iris.data[0:100,:]

> > Y\_training = (iris.target[0:100]).reshape(100,1)

> > trainingset = np.hstack((Y\_training, X\_training))

> >

> > X\_test = iris.data[100:150,:]

> > Y\_test = (iris.target[100:150]).reshape(50,1)

> > testset = np.hstack((Y\_test, X\_test))

> >

> > run\_tune\_process(tune\_hyperparameters,trainingset,testset)

> >

> > ##

> >

> > Reply to this email directly or view it on GitHubhttps://github.com/scikit-learn/scikit-learn/issues/2889#issuecomment-35990430

> > .

-------------------------------------------------------------------------

2014-03-11T15:50:26Z larsmans

This has nothing to do with custom scorers. This is a [well-known feature](http://docs.python.org/2/library/multiprocessing.html#windows) of Python multiprocessing on Windows: you have to run everything that uses `n\_jobs=-1` in an `if \_\_name\_\_ == '\_\_main\_\_'` block or you'll get freezes/crashes. Maybe we should document this somewhere prominently, e.g. in the README?

-------------------------------------------------------------------------

2014-03-11T16:11:49Z GaelVaroquaux

> you have to run everything that uses n\_jobs= -1 in an if \*\*name\*\* ==

> '\*\*main\*\*' block or you'll get freezes/crashes.

Well, the good news is that nowadays joblib gives a meaningful error

message on such crash, rather than a fork bomb.

-------------------------------------------------------------------------

2014-03-11T16:15:43Z larsmans

@GaelVaroquaux does current scikit-learn give that error message? If so, the issue can be considered fixed, IMHO.

-------------------------------------------------------------------------

2014-03-11T16:17:27Z GaelVaroquaux

> @GaelVaroquaux does current scikit-learn give that error message? If so, the

> issue can be considered fixed, IMHO.

It should do. The only way to be sure is to check. I am on the move right

now, and I cannot boot up a Windows VM to do that.

-------------------------------------------------------------------------

2014-03-11T16:23:40Z larsmans

I'm not going to install a C compiler on Windows just for this. Sorry, but I really don't do Windows :)

-------------------------------------------------------------------------

2014-03-11T16:27:51Z GaelVaroquaux

> I'm not going to install a C compiler on Windows just for this. Sorry, but I

> really don't do Windows :)

I have a Windows VM. I can check. It's just a question of finding a

little be of time to do it.

-------------------------------------------------------------------------

2014-03-11T16:42:56Z adverley

@larsmans , you are completely right. The custom scorer object was a mistake of me, the problem lies indeed in the multiprocessing on windows. I tried this same code on a Linux and it runs well.

I don't get any error messages because it doesn't crash, it just stops doing any meaningful.

-------------------------------------------------------------------------

2014-03-15T15:38:38Z larsmans

@adverley Could you try the most recent version from GitHub on your Windows box?

-------------------------------------------------------------------------

2015-01-23T03:45:36Z amueller

Closing because of lack of feeback and it is probably a known issue that is fixed in newer joblib.

-------------------------------------------------------------------------

2015-03-27T00:53:28Z hirak99

Not sure if related, does seem to be.

In windows, custom scorer still freezes. I encountered this thread on google - removed the scorer, and the grid search works.

When it freezes, it shows no error message. There are 3 python processes spawned too (because I set n\_jobs=3). However, the CPU utilization remains 0 for all python processes. I am using IPython Notebook.

-------------------------------------------------------------------------

2015-04-01T00:45:39Z amueller

Can you share the code of the scorer? It seems a bit unlikely.

-------------------------------------------------------------------------

2015-04-01T00:46:21Z amueller

Does your scorer use joblib / n\_jobs anywhere? It shouldn't, and that could maybe cause problems (though I think joblib should detect that).

-------------------------------------------------------------------------

2015-04-01T03:12:13Z hirak99

Sure - here's the full code - http://pastebin.com/yUE26SNs

The scorer function is "score\_model", it doesn't use joblib.

This runs from command prompt, but not from IPython Notebook. The error message is -

`AttributeError: Can't get attribute 'score\_model' on <module '\_\_main\_\_' (built-in)>;`

Then the IPython and all the spawned python instances become idle - silently - and don't respond to any python code anymore till I restart it.

-------------------------------------------------------------------------

2015-04-01T15:51:38Z amueller

Fix the attribute error, then it'll work.

Do you do pylab imports in IPython notebook? Otherwise everything should be the same.

-------------------------------------------------------------------------

2015-04-01T23:34:09Z hirak99

Well I do not know what causes the AttributeError... Though it is most likely related to joblibs, since \_it happens only when n\_jobs is more than 1\_, runs fine with `n\_jobs=1`.

The error talks about attribute `score\_model` missing from `\_\_main\_\_`, whether or not I have a `if \_\_name\_\_ == '\_\_main\_\_'` in the IPython Notebook or not.

(I realized that the error line was pasted incorrectly above - I edited in the post above.)

I don't use pylab.

Here's the full extended error message - http://pastebin.com/23y5uHT2

-------------------------------------------------------------------------

2015-04-02T14:43:20Z amueller

Hum, that is likely related to issues of multiprocessing on windows. Maybe @GaelVaroquaux or @ogrisel can help.

I don't know what the notebook makes of the `\_\_name\_\_ == "\_\_main\_\_"`.

Try not defining the metric in the notebook, but in a separate file and import it. I'd think that would fix it.

This is not really related to GridSearchCV, but some interesting interaction between windows multiprocessing, IPython notebook and joblib.

-------------------------------------------------------------------------

2016-11-05T05:18:50Z alwaysandeep

guys...thanks for the thread. Anyway i should have checked this thread before, wasted 5 hours of my time on this. Trying to run in parallel processing. Thanks a lot :)

TO ADD A FEEDBACK: its still freezing. I faced the same issue when in presence of my own make\_Score cost function..my system starts freezing. When i did not use custom cost function, i did not face these freezes in parallel processing

-------------------------------------------------------------------------

2016-11-08T10:07:57Z lesteve

The best way of turning these 5 hours into something useful for the project, would be to provide us with a stand-alone example reproducing the problem.

-------------------------------------------------------------------------

2016-12-21T00:02:32Z vosilov

I was experiencing the same issue on Windows 10 working in Jupyter notebook trying to use a custom scorer within a nested cross-validation and n\_jobs=-1. I was getting the `AttributeError: Can't get attribute 'custom\_scorer' on <module '\_\_main\_\_' (built-in)>;` message.

As @amueller suggested, importing the custom scorer instead of defining it in the notebook works.

-------------------------------------------------------------------------

2017-08-02T08:29:41Z martinxtm

I have the exact same problem on OSX 10.10.5

-------------------------------------------------------------------------

2017-08-04T20:25:42Z boazsh

Same here.

OSX 10.12.5

-------------------------------------------------------------------------

2017-08-06T03:38:14Z jnothman

Please give a reproducible code snippet. We'd love to get to the bottom of this. It is hard to understand without code, including data, that shows us the issue.

-------------------------------------------------------------------------

2017-08-08T08:15:06Z boazsh

Just run these lines in a python shell

```python

import numpy as np

from sklearn.decomposition import PCA

from sklearn.svm import SVC

from sklearn.preprocessing import RobustScaler

from sklearn.metrics import classification\_report

from sklearn.pipeline import Pipeline

from sklearn.model\_selection import cross\_val\_predict

np.random.seed(1234)

X = np.random.sample((1000, 100))

Y = np.random.sample((1000)) > 0.5

svc\_pipeline = Pipeline([('pca', PCA(n\_components=95)), ('svc', SVC())])

predictions = cross\_val\_predict(svc\_pipeline, X, Y, cv=30, n\_jobs=-1)

print classification\_report(Y, predictions)

```

Note that removing the PCA step from the pipeline solves the issue.

More info:

Darwin-16.6.0-x86\_64-i386-64bit

('Python', '2.7.13 (default, Apr 4 2017, 08:47:57) \n[GCC 4.2.1 Compatible Apple LLVM 8.1.0 (clang-802.0.38)]')

('NumPy', '1.12.1')

('SciPy', '0.19.1')

('Scikit-Learn', '0.18.2')

-------------------------------------------------------------------------

2017-08-08T11:09:12Z jnothman

seeing as you don't use a custom scorer, should we assume that is a

separate issue?

On 8 Aug 2017 6:15 pm, "boazsh" <notifications@github.com> wrote:

> Just run these lines in a python shell

>

> from sklearn.decomposition import PCAfrom sklearn.svm import SVCfrom sklearn.preprocessing import RobustScalerfrom sklearn.metrics import classification\_reportfrom sklearn.pipeline import Pipelinefrom sklearn.model\_selection import cross\_val\_predict

>

> X = np.random.sample((1000, 100))

> Y = np.random.sample((1000)) > 0.5

> svc\_pipeline = Pipeline([('pca', PCA(n\_components=95)), ('svc', SVC())])

> predictions = cross\_val\_predict(svc\_pipeline, X, Y, cv=30, n\_jobs=-1)print classification\_report(Y, predictions)

>

> Note that removing the PCA step from the pipeline solves the issue.

>

> More info:

>

> scikit-learn==0.18.2

> scipy==0.19.1

> numpy==1.12.1

>

> ‚Äî

> You are receiving this because you commented.

> Reply to this email directly, view it on GitHub

> <https://github.com/scikit-learn/scikit-learn/issues/2889#issuecomment-320885103>,

> or mute the thread

> <https://github.com/notifications/unsubscribe-auth/AAEz6-6Klhc67b5kZ17fFTxc8RfZQ\_BWks5sWBkLgaJpZM4BkiD9>

> .

>

-------------------------------------------------------------------------

2017-08-08T11:19:44Z boazsh

When I first faced this issue I was using custom scorer, but while trying to simplify the example code as much as possible, I found that it is not necessarily have to contain custom scorer. At least on my machine. Importing the scorer also didn't help in my case. Anyway, the symptoms looks similar. The script hangs forever and the CPU utilization is low.

-------------------------------------------------------------------------

2017-08-08T11:52:44Z lesteve

@boazsh thanks a lot for the snippet, it is not deterministic though, can you edit it and use a `np.random.RandomState` to make sure the random numbers are always the same on each run.

Also there is a work-around if you are using Python 3 suggested for example in https://github.com/scikit-learn/scikit-learn/issues/5115#issuecomment-187683383.

I don't have a way to test this on OSX at the moment but I may be able to try in the upcoming days.

-------------------------------------------------------------------------

2017-08-08T11:57:22Z lesteve

Some piece of information useful to have (just add what is missing to your earlier comment https://github.com/scikit-learn/scikit-learn/issues/2889#issuecomment-320885103):

```py

import platform; print(platform.platform())

import sys; print("Python", sys.version)

import numpy; print("NumPy", numpy.\_\_version\_\_)

import scipy; print("SciPy", scipy.\_\_version\_\_)

import sklearn; print("Scikit-Learn", sklearn.\_\_version\_\_)

```

Also how did you install scikit-learn, with pip, with conda, with one of the OSX package managers (brew, etc ...) ?

-------------------------------------------------------------------------

2017-08-08T12:10:38Z boazsh

Updated the snippet (used np.random.seed)

Darwin-16.6.0-x86\_64-i386-64bit

('Python', '2.7.13 (default, Apr 4 2017, 08:47:57) \n[GCC 4.2.1 Compatible Apple LLVM 8.1.0 (clang-802.0.38)]')

('NumPy', '1.12.1')

('SciPy', '0.19.1')

('Scikit-Learn', '0.18.2')

-------------------------------------------------------------------------

2017-08-08T12:57:41Z lesteve

> Updated the snippet (used np.random.seed)

Great thanks a lot!

> Also how did you install scikit-learn, with pip, with conda, with one of the OSX package managers (brew, etc ...) ?

Have you answered this one, I can't find your answer ...

-------------------------------------------------------------------------

2017-08-08T12:59:30Z boazsh

Sorry, missed it - pip.

-------------------------------------------------------------------------

2017-08-08T13:21:40Z jnothman

FWIW, I have no problem running that snippet with:

>>> import platform; print(platform.platform())

Darwin-16.7.0-x86\_64-i386-64bit

>>> import sys; print("Python", sys.version)

Python 2.7.12 |Continuum Analytics, Inc.| (default, Jul 2 2016, 17:43:17)

[GCC 4.2.1 (Based on Apple Inc. build 5658) (LLVM build 2336.11.00)]

>>> import numpy; print("NumPy", numpy.\_\_version\_\_)

NumPy 1.13.1

>>> import scipy; print("SciPy", scipy.\_\_version\_\_)

SciPy 0.19.1

>>> import sklearn; print("Scikit-Learn", sklearn.\_\_version\_\_)

Scikit-Learn 0.18.2

Could you put verbose=10 in cross\_val\_predict, too, so that we can perhaps

see where it breaks for you?

On 8 August 2017 at 22:59, boazsh <notifications@github.com> wrote:

> Sorry, missed it - pip.

>

> ‚Äî

> You are receiving this because you commented.

> Reply to this email directly, view it on GitHub

> <https://github.com/scikit-learn/scikit-learn/issues/2889#issuecomment-320948362>,

> or mute the thread

> <https://github.com/notifications/unsubscribe-auth/AAEz67S64KIXUGvARGjvxBOw\_4aCAdqhks5sWFu0gaJpZM4BkiD9>

> .

>

-------------------------------------------------------------------------

2017-08-08T13:44:08Z lesteve

@jnothman I am guessing that your conda environment uses MKL and not Accelerate. This freezing problem is specific to Accelerate and Python multiprocessing. http://scikit-learn.org/stable/faq.html#why-do-i-sometime-get-a-crash-freeze-with-n-jobs-1-under-osx-or-linux for more details.

pip on the other hand will use wheels that are shipped with Accelerate (at the time of writing).

A work-around (other than the JOBLIB\_START\_METHOD) to avoid this particular bug is to use MKL (e.g. via conda) or OpenBLAS (e.g. via the conda-forge channel).

-------------------------------------------------------------------------

2017-08-08T13:45:43Z boazsh

Nothing is being printed...

![screen shot 2017-08-08 at 16 43 35](https://user-images.githubusercontent.com/8148263/29074692-e1a5f178-7c58-11e7-812f-6274bea026bc.png)

-------------------------------------------------------------------------

2017-08-08T13:48:15Z lesteve

> @jnothman I am guessing that your conda environment uses MKL and not Accelerate.

@jnothman in case you want to reproduce the problem, IIRC you can create an environment with Accelerate on OSX with something like:

```py

conda create -n test-env python=3 nomkl scikit-learn ipython

```

-------------------------------------------------------------------------

2017-08-09T14:24:55Z lesteve

FWIW I can not reproduce the problem on my OS X VM. I tried to mimic as close as possible @boazsh's versions:

```

Darwin-16.1.0-x86\_64-i386-64bit

('Python', '2.7.13 |Continuum Analytics, Inc.| (default, Dec 20 2016, 23:05:08) \n[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.57)]')

('NumPy', '1.12.1')

('SciPy', '0.19.1')

('Scikit-Learn', '0.18.2')

```

-------------------------------------------------------------------------

2017-08-09T14:42:55Z lesteve

Hmm actually I can reproduce but your snippet was not a complete reproducer. Here is an updated snippet:

```py

import numpy as np

from sklearn.decomposition import PCA

from sklearn.svm import SVC

from sklearn.preprocessing import RobustScaler

from sklearn.metrics import classification\_report

from sklearn.pipeline import Pipeline

from sklearn.model\_selection import cross\_val\_predict

np.random.seed(1234)

X = np.random.sample((1000, 100))

Y = np.random.sample((1000)) > 0.5

svc\_pipeline = Pipeline([('pca', PCA(n\_components=95)), ('svc', SVC())])

PCA(n\_components=95).fit(X, Y) # this line is required to reproduce the freeze

predictions = cross\_val\_predict(svc\_pipeline, X, Y, cv=30, n\_jobs=-1)

print(classification\_report(Y, predictions))

```

In any case, this is a known problem with Accelerate and Python multiprocessing. Work-arounds exist and have been listed in earlier posts. The easiest one is probably to use conda and make sure that you use MKL and not Accelerate.

-------------------------------------------------------------------------

2017-08-10T09:02:38Z ogrisel

On the longer term (probably scikit-learn 0.20) this problem will be universally solved by the new loky backend for joblib: https://github.com/scikit-learn/scikit-learn/issues/7650

-------------------------------------------------------------------------

2017-08-11T22:54:07Z amueller

Having a fix to multiprocessing be dependent on the scikit-learn version is symptomatic of the problems of vendoring....

-------------------------------------------------------------------------

2017-08-12T05:14:23Z GaelVaroquaux

> Having a fix to multiprocessing be dependent on the scikit-learn version is symptomatic of the problems of vendoring....

I recently read the following, which I found interesting:

https://lwn.net/Articles/730630/rss

-------------------------------------------------------------------------

2017-10-03T12:41:52Z KaisJM

I have a similar issue with RandomizedSearchCV; it hangs indefinitely. I am using a 3 year old macbook pro, 16GB ram and core i7 and my scikit-learn version is 0.19.

Puzzling part is that it was working last Friday!!! Monday morning, I go back and try to run and it just freezes. I know from previous runs that it take about 60 min to finish, but I waited a lot longer than that and nothing happens, it just hangs, no error msgs, nothing and my computer heats up and sucks power like there's no tomorrow. Code below. I tried changing n\_iter to 2 and n\_jobs=1 after reading some comments here and that worked. So it may have something to do with n\_jobs=-1. Still, this code worked fine last Friday! it just hates Mondays. My dataset size is less that 20k examples with dimensionality < 100..

```

from sklearn.metrics import make\_scorer

from sklearn.cross\_validation import cross\_val\_score

from sklearn.grid\_search import RandomizedSearchCV

import sklearn\_crfsuite

crf = sklearn\_crfsuite.CRF(

algorithm='lbfgs',

max\_iterations=100,

all\_possible\_transitions=True

)

params\_space = {

'c1': scipy.stats.expon(scale=0.5),

'c2': scipy.stats.expon(scale=0.05),

}

f1\_scorer = make\_scorer(metrics.flat\_f1\_score,

average='weighted', labels=labels)

rs = RandomizedSearchCV(crf, params\_space,

cv=3,

verbose=1,

n\_jobs=-1,

n\_iter=50,

scoring=f1\_scorer)

rs.fit(X\_train, y\_train) # THIS IS WHERE IT FREEZES

```

-------------------------------------------------------------------------

2017-10-03T13:04:44Z jnothman

what is crf? just to eliminate the possibility, could you try using

return\_train\_score=False?

-------------------------------------------------------------------------

2017-10-03T13:17:29Z lesteve

It is very likely that this @KaisJM's problem is due to the well known limitation on Accelerate with multiprocessing, see our [faq](http://scikit-learn.org/stable/faq.html#why-do-i-sometime-get-a-crash-freeze-with-n-jobs-1-under-osx-or-linux).

How did you install scikit-learn?

Also for future reference, can you paste the output of:

```py

import platform; print(platform.platform())

import sys; print("Python", sys.version)

import numpy; print("NumPy", numpy.\_\_version\_\_)

import scipy; print("SciPy", scipy.\_\_version\_\_)

import sklearn; print("Scikit-Learn", sklearn.\_\_version\_\_)

```

-------------------------------------------------------------------------

2017-10-03T13:27:25Z KaisJM

this was working last Friday!! I done nothing since. I think scikit learn is part of anaconda, but I did upgrade with pip (pip install --upgrade sklearn), but thats before I got this problem.. I ran the code fine after upgrading to 0.19.

here's the output of the above prints:

```

Darwin-15.6.0-x86\_64-i386-64bit

('Python', '2.7.12 |Continuum Analytics, Inc.| (default, Jul 2 2016, 17:43:17) \n[GCC 4.2.1 (Based on Apple Inc. build 5658) (LLVM build 2336.11.00)]')

('NumPy', '1.13.1')

('SciPy', '0.19.1')

('Scikit-Learn', '0.19.0')

```

-------------------------------------------------------------------------

2017-10-03T13:50:23Z KaisJM

@jnothman : I am using RandomizedSearchCV from sklearn.grid\_search which does not have the return\_train\_score parameter. I know sklearn.grid\_search is depricated.. I will try the one from sklearn.model\_selection, but something tells me I will have the same exact issue). Updated original comment with more info and code.

-------------------------------------------------------------------------

2017-10-03T14:06:41Z lesteve

Can you post the output of `conda list | grep numpy`. I would wild guess that by updating scikit-learn with pip you updated numpy with pip too and you got the numpy wheels which uses Accelerate and has the limitation mentioned above.

Small word of advice:

\* post a fully stand-alone snippet (for your next issue). That means anyone can copy and paste it in a IPython session and easily try to reproduce. This will give you the best chance of getting good feed-back.

\* if you are using conda, stick to conda to manage packages that are available through conda. Only use pip when you have to.

\* If you insist you want to use `pip install --update`, I would strongly recommend you use `pip install --update --no-deps`. Otherwise if a package dependends, say on numpy, and you happen not to have the latest numpy, numpy will be upgraded with pip, which you do not want.

-------------------------------------------------------------------------

2017-10-03T14:07:21Z lesteve

Oh yeah and BTW, sklearn.grid\_search is deprecated you probably want to use sklearn.model\_selection at one point not too far down the road.

-------------------------------------------------------------------------

2017-10-03T14:12:20Z KaisJM

Good advice, thank you. So is the workaround to downgrade numpy? what limitation are you referring to? the FAQ link above? I did read it, but I do not understand this stuff (i'm just an algo guy :) ).

output of `conda list | grep numpy`

numpy 1.12.0 <pip>

numpy 1.12.0 py27\_0

numpy 1.13.1 <pip>

numpydoc 0.7.0 <pip>

-------------------------------------------------------------------------

2017-10-03T14:29:05Z lesteve

Wow three numpy installed I saw two before but never three ... anyway this seems indicative of the problem I was mentioning, i.e. that you have mixed pip and conda which is a bad idea for a given package.

```

pip uninstall -y # maybe a few times to make sure you have removed pip installed packages

conda install numpy -f

```

Hopefully after that you will have a single numpy that uses MKL.

If I were you I would double-check that you don't have the same problem for other core scientific packages, e.g. scipy, etc ...

-------------------------------------------------------------------------

2017-10-03T15:30:19Z KaisJM

the reason I resort to pip for some packages is that conda does not have some packages, which actually is very frustrating because I know mixing pip with conda is a bad idea. Next time that happens I'll use the --no-deps option.

-------------------------------------------------------------------------

2017-10-03T17:43:21Z KaisJM

one thing I should've mentioned is that I installed Spyder within the python env I was working in. However, I was able to run the code after installing Spyder, both in Spyder and in Jupyter.

I did uninstall Spyder and the numpys above, re-installed bumpy with conda (which updated scikit to 0.19) and still get the same error. Something may have happened because of the Spyder install, but then why would it work for a day and then suddenly stop??

-------------------------------------------------------------------------

2017-10-03T18:37:12Z KaisJM

ok, nothing is working!! should I just create a new environment (using conda) and re-install everything there? will that solve it or make it worse?

-------------------------------------------------------------------------

2017-10-03T23:25:39Z jnothman

Sounds worth a try!

-------------------------------------------------------------------------

2017-10-04T03:20:27Z KaisJM

created a new env and installed everything with conda, still freezes indefinitely. only one copy of each package etc.

n\_jobs=1 works,but takes forever of course (it worked in the previous env as well). n\_jobs=-1 is what freezes indefinitely.

```

conda list | grep numpy

numpy 1.13.1 py27hd567e90\_2

Darwin-15.6.0-x86\_64-i386-64bit

('Python', '2.7.13 |Continuum Analytics, Inc.| (default, Dec 20 2016, 23:05:08) \n[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.57)]')

('NumPy', '1.13.1')

('SciPy', '0.19.1')

('Scikit-Learn', '0.19.0')

```

-------------------------------------------------------------------------

2017-10-04T11:24:30Z lesteve

Then I don't know. The only way we can investigate, is that you post a fully standalone snippet which we can just copy and paste in an IPython sesion and see if we can reproduce the problem.

-------------------------------------------------------------------------

2017-10-04T12:57:19Z KaisJM

will try to create a minimal example that reproduces the problem. I need to do that to debug more efficiently.

I read the FAQ entry you refer to about "Accelerate".. its not much help for me. What I took from it is that fork() NOT followed by exec() call is bad. I've done some googling on this and nothing so far even hints at a workaround. Can you point to some more information, more detail about what the problem is? thanks,

-------------------------------------------------------------------------

2017-10-04T13:29:26Z lesteve

Try this snippet (taken from https://github.com/numpy/numpy/issues/4776):

```py

import multiprocessing as mp

import numpy as np

def compute(n):

print('Enter')

np.dot(np.eye(n), np.eye(n))

print('Exit')

print('\nWithout multiprocessing:')

compute(1000)

print('\nWith multiprocessing:')

workers = mp.Pool(1)

results = workers.map(compute, (1000, 1000))

```

\* If this freezes (i.e. it does not finish within one second) that means you are using Accelerate and the freeze is a known limitation with Python multiprocessing.The work-around is to not use Accelerate. On OSX you can do that with conda which uses MKL by default. You can also use OpenBLAS using conda-forge.

\* If it does not freeze then you are not using Accelerate, and we would need a stand-alone snippet to investigate.

-------------------------------------------------------------------------

2017-10-04T13:58:52Z KaisJM

will try to reproduce with minimal code.

```

Without multiprocessing:

Enter

Exit

With multiprocessing:

Enter

Exit

Enter

Exit

```

-------------------------------------------------------------------------

2017-10-04T15:30:05Z amueller

@GaelVaroquaux scikit-learn is not an app but a library in a rich ecosystem. If everybody did what we do, everything would come crashing down. That's a pretty clear signal that we need to change. And there are many environments where the opposite is true from that comment.

-------------------------------------------------------------------------

2017-10-05T15:33:33Z KaisJM

I used a ubuntu virtual instance in google cloud compute engine (bumpy, spicy, scikit etc were not the most up to date). The code ran fine. Then I installed Gensim. This updated numpy and scipy to the latest versions and installed few other things it needs (boto, bz2file and smart\_open). After that the code freezes. I hope this gives a useful clue as to what causes this freeze.

\*\*after installing Gensim\*\*

numpy (1.10.4) updated to numpy (1.13.3)

scipy (0.16.1) updated to scipy (0.19.1)

\*\*more info:\*\*

Doing some research I found that libblas, liblapack and liblapack\_atlas were missing from my /usr/lib/, also I did not see the directory /usr/lib/atlas-base/. I don't know if they were there and installing gensim removed them since it updated numpy etc, but this is likely since the code worked before installing gensim. I installed them using `sudo apt-get --yes install libatlas-base-dev` and "\_update-alternatives\_" according to the advanced scikit [installation instructions](http://scikit-learn.org/stable/developers/advanced\_installation.html) , but it did not help, the code still freezes with n\_jobs=-1.

-------------------------------------------------------------------------

2017-10-05T17:52:36Z KaisJM

I think the problem is that numpy is using OpenBlas. Will switch it to ATLAS and see what happens.

```

>>> import numpy as np

>>> np.\_\_config\_\_.show()

lapack\_opt\_info:

libraries = ['openblas']

library\_dirs = ['/usr/local/lib']

define\_macros = [('HAVE\_CBLAS', None)]

language = c

blas\_opt\_info:

libraries = ['openblas']

library\_dirs = ['/usr/local/lib']

define\_macros = [('HAVE\_CBLAS', None)]

language = c

openblas\_info:

libraries = ['openblas']

library\_dirs = ['/usr/local/lib']

define\_macros = [('HAVE\_CBLAS', None)]

language = c

openblas\_lapack\_info:

libraries = ['openblas']

library\_dirs = ['/usr/local/lib']

define\_macros = [('HAVE\_CBLAS', None)]

language = c

blas\_mkl\_info:

NOT AVAILABLE

```

-------------------------------------------------------------------------

2017-10-05T20:06:36Z paulaceccon

Still the same problem. The following runs fine, unless I insert n\_jobs=-1.

```

from sklearn.metrics import fbeta\_score

def f2\_score(y\_true, y\_pred):

y\_true, y\_pred, = np.array(y\_true), np.array(y\_pred)

return fbeta\_score(y\_true, y\_pred, beta=2, average='binary')

clf\_rf = RandomForestClassifier()

grid\_search = GridSearchCV(clf\_rf, param\_grid=param\_grid, scoring=make\_scorer(f2\_score), cv=5)

grid\_search.fit(X\_train, y\_train)

```

-------------------------------------------------------------------------

2017-10-06T12:15:16Z KaisJM

@paulaceccon are your Numpy and Scipy installations using ATLAS or OpenBLAS?

-------------------------------------------------------------------------

2017-10-06T13:17:02Z lesteve

It is a bit hard to follow what you have done @KaisJM. From a maintainer's point of view what we need is a fully stand-alone python snippet to see if we can reproduce. If we can reproduce, only then can we investigate and try to understand what is happening. If that only happens when you install gensim and you manage to reproduce this behaviour consistently, then we would need full instructions how to create a Python environment that has the problem vs a Python environment that doesn't have the problem.

This requires a non negligible amount of time and effort, I completely agree, but without it, I am afraid that there is not much we can do to investigate the problem you are facing.

-------------------------------------------------------------------------

2017-10-06T13:31:33Z lesteve

> according to the advanced [installation instructions](http://scikit-learn.org/stable/developers/advanced\_installation.html)

@KaisJM by the way, this page is out-of-date, since nowadays wheels are available on Linux and contain their own OpenBLAS. If you install a released scikit-learn with pip you will be using OpenBLAS.

-------------------------------------------------------------------------

2017-10-06T13:50:35Z KaisJM

@lesteve are you saying that Openblas does not cause a freeze anymore?

-------------------------------------------------------------------------

2017-10-06T13:57:20Z KaisJM

@lesteve paula has posted a snippet that also has the same problem. I can see it's not complete code, but I hope it gives some clue. I can make here snippet "complete" and post for you. However, it is clear that the "out-of-date" -as you call it- instructions page may not be so out of date. The highest likelihood is that OpenBLAS is causing the fees they are talking about in that page.

-------------------------------------------------------------------------

2017-10-06T14:12:51Z lesteve

These instructions are outdated believe me. If you read in details, it says "but can freeze joblib/multiprocessing prior to OpenBLAS version 0.2.8-4". I checked a recent numpy wheel and it contains OpenBLAS 0.2.8.18. The freeze they are referring to is the one in https://github.com/scikit-learn/scikit-learn/issues/2889#issuecomment-334155175, which you don't seem to have.

> I can see it's not complete code, but I hope it gives some clue

Not really no. We have reports of users that seems to indicate that freezing can still happen, none of which we have managed to reproduce AFAIK. That seems to indicate, that this problem happens in some very specific combination of factors. Unless someone that has the problem spends some time and figures out how to reproduce in a controlled way and we manage to reproduce, there is just no way we can do anything about it.

> I can make here snippet "complete" and post for you

That would be great. That would be great if you could check if such a snippet still cause the freeze in a separate conda environment (or virtualenv depending on what you use).

-------------------------------------------------------------------------

2017-10-06T14:50:37Z KaisJM

@lesteve @paulaceccon :I took Paula's excerpt code and made a complete run-able code snippet. Just paste it into a Jupyter cell and run it. Paula: I could not get this snippet to freeze. Notice that n\_jobs=-1 and runs fine. Would be great if you can take a look and post a version of it that freezes. Notice that you can switch between grid\_search module and model\_selection module, both ran fine for me.

```py

import platform; print(platform.platform())

import sys; print("Python", sys.version)

import numpy as np; print("NumPy", np.\_\_version\_\_)

import scipy; print("SciPy", scipy.\_\_version\_\_)

import sklearn; print("Scikit-Learn", sklearn.\_\_version\_\_)

from sklearn.ensemble import RandomForestClassifier

from sklearn.datasets import make\_classification

import scipy.stats

from sklearn.metrics import make\_scorer

from sklearn.grid\_search import RandomizedSearchCV

#from sklearn.model\_selection import RandomizedSearchCV

#from sklearn.model\_selection import GridSearchCV

from sklearn.metrics import fbeta\_score

X, y = make\_classification(n\_samples=1000, n\_features=4,

n\_informative=2, n\_redundant=0,

random\_state=0, shuffle=False)

clf\_rf = RandomForestClassifier(max\_depth=2, random\_state=0)

def f2\_score(y\_true, y\_pred):

y\_true, y\_pred, = np.array(y\_true), np.array(y\_pred)

return fbeta\_score(y\_true, y\_pred, beta=2, average='binary')

param\_grid = {'max\_depth':[2, 3, 4], 'random\_state':[0, 3, 7, 17]}

grid\_search = RandomizedSearchCV(clf\_rf, param\_grid, n\_jobs=-1, scoring=make\_scorer(f2\_score), cv=5)

grid\_search.fit(X, y)

```

-------------------------------------------------------------------------

2017-10-06T17:07:53Z lesteve

@KaisJM I think it is more useful if you start from your freezing script and manage to simplify and post a fully stand-alone that freezes for you.

-------------------------------------------------------------------------

2017-10-06T18:13:26Z KaisJM

@lesteve Agreed. I created a new python2 environment like the one I had before installing Gensim. Code ran fine, NO freeze with n\_jobs=-1. What's more, Numpy is using OpenBLAS and has the same config as the environment that exhibits the freeze (the one where Gensim was installed). So it seems that openblas is not the cause of this freeze.

```

bumpy.\_\_config\_\_.show()

lapack\_opt\_info:

libraries = ['openblas']

library\_dirs = ['/usr/local/lib']

define\_macros = [('HAVE\_CBLAS', None)]

language = c

blas\_opt\_info:

libraries = ['openblas']

library\_dirs = ['/usr/local/lib']

define\_macros = [('HAVE\_CBLAS', None)]

language = c

openblas\_info:

libraries = ['openblas']

library\_dirs = ['/usr/local/lib']

define\_macros = [('HAVE\_CBLAS', None)]

language = c

openblas\_lapack\_info:

libraries = ['openblas']

library\_dirs = ['/usr/local/lib']

define\_macros = [('HAVE\_CBLAS', None)]

language = c

blas\_mkl\_info:

NOT AVAILABLE

```

-------------------------------------------------------------------------

2017-10-10T02:00:05Z paulaceccon

@KaisJM I'm running the same snippet here (windows) and it freezes.

```

from sklearn.datasets import make\_classification

X, y = make\_classification()

from sklearn.ensemble import RandomForestClassifier

clf\_rf\_params = {

'n\_estimators': [400, 600, 800],

'min\_samples\_leaf' : [5, 10, 15],

'min\_samples\_split' : [10, 15, 20],

'criterion': ['gini', 'entropy'],

'class\_weight': [{0: 0.51891309, 1: 13.71835531}]

}

import numpy as np

def ginic(actual, pred):

actual = np.asarray(actual) # In case, someone passes Series or list

n = len(actual)

a\_s = actual[np.argsort(pred)]

a\_c = a\_s.cumsum()

giniSum = a\_c.sum() / a\_s.sum() - (n + 1) / 2.0

return giniSum / n

def gini\_normalizedc(a, p):

if p.ndim == 2: # Required for sklearn wrapper

p = p[:,1] # If proba array contains proba for both 0 and 1 classes, just pick class 1

return ginic(a, p) / ginic(a, a)

from sklearn import metrics

gini\_sklearn = metrics.make\_scorer(gini\_normalizedc, True, True)

from sklearn.model\_selection import GridSearchCV

clf\_rf = RandomForestClassifier()

grid = GridSearchCV(clf\_rf, clf\_rf\_params, scoring=gini\_sklearn, cv=3, verbose=1, n\_jobs=-1)

grid.fit(X, y)

print (grid.best\_params\_)

```

I know that it's awkward but it didn't froze when running with a \_custom\_ metric.

-------------------------------------------------------------------------

2017-10-16T09:34:15Z snovik75

I have a similar problem. I have been running the same code and simply wanted to update the model with the new month data and it stopped running. i believe sklearn got updated in the meantime to 0.19

-------------------------------------------------------------------------

2017-10-19T17:31:56Z thomberg1

Running GridSearchCV or RandomizedSearchCV in a loop and n\_jobs > 1 would hang silently in Jupiter & IntelliJ:

```

for trial in tqdm(range(NUM\_TRIALS)):

...

gscv = GridSearchCV(estimator=estimator, param\_grid=param\_grid,

scoring=scoring, cv=cv, verbose=1, n\_jobs=-1)

gscv.fit(X\_data, y\_data)

...

```

Followed @lesteve recommendation & checked environment & removed numpy installed with pip:

Darwin-16.6.0-x86\_64-i386-64bit

Python 3.6.1 |Anaconda custom (x86\_64)| (default, May 11 2017, 13:04:09)

[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.57)]

NumPy 1.13.1

SciPy 0.19.1

Scikit-Learn 0.19.0

$conda list | grep numpy

gnumpy 0.2 pip

numpy 1.13.1 py36\_0

numpy 1.13.3 pip

numpydoc 0.6.0 py36\_0

$pip uninstall numpy

$conda list | grep numpy

gnumpy 0.2 pip

numpy 1.13.1 py36\_0

numpydoc 0.6.0 py36\_0

$conda install numpy -f // most likely unnecessary

$conda list | grep numpy

gnumpy 0.2 pip

numpy 1.13.1 py36\_0

numpydoc 0.6.0 py36\_0

Fixed my problem.

-------------------------------------------------------------------------

2017-10-19T17:49:32Z thomberg1

@paulaceccon your problem is related to

> https://stackoverflow.com/questions/36533134/cant-get-attribute-abc-on-module-main-from-abc-h-py

> If you declare the pool prior to declaring the function you are trying to use in parallel it will throw this error. Reverse the order and it will no longer throw this error.

The following will run your code:

```

import multiprocessing

if \_\_name\_\_ == '\_\_main\_\_':

multiprocessing.set\_start\_method('spawn')

from external import \*

from sklearn.datasets import make\_classification

X, y = make\_classification()

from sklearn.ensemble import RandomForestClassifier

clf\_rf\_params = {

'n\_estimators': [400, 600, 800],

'min\_samples\_leaf' : [5, 10, 15],

'min\_samples\_split' : [10, 15, 20],

'criterion': ['gini', 'entropy'],

'class\_weight': [{0: 0.51891309, 1: 13.71835531}]

}

from sklearn.model\_selection import GridSearchCV

clf\_rf = RandomForestClassifier()

grid = GridSearchCV(clf\_rf, clf\_rf\_params, scoring=gini\_sklearn, cv=3, verbose=1, n\_jobs=-1)

grid.fit(X, y)

print (grid.best\_params\_)

```

with external.py

```

import numpy as np

def ginic(actual, pred):

actual = np.asarray(actual) # In case, someone passes Series or list

n = len(actual)

a\_s = actual[np.argsort(pred)]

a\_c = a\_s.cumsum()

giniSum = a\_c.sum() / a\_s.sum() - (n + 1) / 2.0

return giniSum / n

def gini\_normalizedc(a, p):

if p.ndim == 2: # Required for sklearn wrapper

p = p[:,1] # If proba array contains proba for both 0 and 1 classes, just pick class 1

return ginic(a, p) / ginic(a, a)

from sklearn import metrics

gini\_sklearn = metrics.make\_scorer(gini\_normalizedc, True, True)

```

Results running on 8 cores

Fitting 3 folds for each of 54 candidates, totalling 162 fits

[Parallel(n\_jobs=-1)]: Done 34 tasks | elapsed: 7.1s

[Parallel(n\_jobs=-1)]: Done 162 out of 162 | elapsed: 30.5s finished

{'class\_weight': {0: 0.51891309, 1: 13.71835531}, 'criterion': 'gini', 'min\_samples\_leaf': 10, 'min\_samples\_split': 20, 'n\_estimators': 400}

-------------------------------------------------------------------------

2018-02-12T08:08:20Z xtosis

Issue is still there guys. I am using a custom scorer and it keeps going on forever when I set n\_jobs to anything. When I don't specify n\_jobs at all it works fine but otherwise it freezes.

-------------------------------------------------------------------------

2018-02-12T10:01:36Z lesteve

Can you provide a stand-alone snippet to reproduce the problem ? Please read https://stackoverflow.com/help/mcve for more details.

-------------------------------------------------------------------------

2018-03-16T23:56:44Z paulaceccon

Still facing this problem with the same sample code.

Windows-10-10.0.15063-SP0

Python 3.6.4 |Anaconda custom (64-bit)| (default, Jan 16 2018, 10:22:32) [MSC v.1900 64 bit (AMD64)]

NumPy 1.14.1

SciPy 1.0.0

Scikit-Learn 0.19.1

-------------------------------------------------------------------------

2018-03-17T07:34:00Z glemaitre

Can you provide a stand-alone snippet to reproduce the problem ? Please read https://stackoverflow.com/help/mcve for more details.

-------------------------------------------------------------------------

2018-03-18T00:15:38Z jnothman

I suspect this is the same old multiprocessing in windows issue. see our FAQ

-------------------------------------------------------------------------

2018-04-11T01:35:45Z chi18000

I tested the code in thomberg1's https://github.com/scikit-learn/scikit-learn/issues/2889#issuecomment-337985212.

OS: Windows 10 x64 10.0.16299.309

Python package: WinPython-64bit-3.6.1

numpy (1.14.2)

scikit-learn (0.19.1)

scipy (1.0.0)

It worked fine in Jupyter Notebook and command-line.

-------------------------------------------------------------------------

2018-04-20T17:20:46Z siideffect

HI, i m having the same issue, so i did not want to open new one which could lead to almost identical thread.

-Macos

-Anaconda

-scikit-learn 0.19.1

-scipy 1.0.1

-numpy 1.14.2

```

# MLP for Pima Indians Dataset with grid search via sklearn

from keras.models import Sequential

from keras.layers import Dense

from keras.wrappers.scikit\_learn import KerasClassifier

from sklearn.model\_selection import GridSearchCV

import numpy

# Function to create model, required for KerasClassifier

def create\_model(optimizer='rmsprop', init='glorot\_uniform'):

# create model

model = Sequential()

model.add(Dense(12, input\_dim=8, kernel\_initializer=init, activation='relu'))

model.add(Dense(8, kernel\_initializer=init, activation='relu'))

model.add(Dense(1, kernel\_initializer=init, activation='sigmoid'))

# Compile model

model.compile(loss='binary\_crossentropy', optimizer=optimizer, metrics=['accuracy'])

return model

# fix random seed for reproducibility

seed = 7

numpy.random.seed(seed)

# load pima indians dataset

dataset = numpy.loadtxt("pima-indians-diabetes.csv", delimiter=",")

# split into input (X) and output (Y) variables

X = dataset[:,0:8]

Y = dataset[:,8]

# create model

model = KerasClassifier(build\_fn=create\_model, verbose=0)

# grid search epochs, batch size and optimizer

optimizers = ['rmsprop', 'adam']

init = ['glorot\_uniform', 'normal', 'uniform']

epochs = [50, 100, 150]

batches = [5, 10, 20]

param\_grid = dict(optimizer=optimizers, epochs=epochs, batch\_size=batches, init=init)

grid = GridSearchCV(estimator=model, param\_grid=param\_grid)

grid\_result = grid.fit(X, Y)

# summarize results

print("Best: %f using %s" % (grid\_result.best\_score\_, grid\_result.best\_params\_))

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("%f (%f) with: %r" % (mean, stdev, param))

```

Code is from a tutorial : https://machinelearningmastery.com/use-keras-deep-learning-models-scikit-learn-python/

I tried changing the n\_jobs parameter to 1, -1, but neither of these worked. Any hint?

-------------------------------------------------------------------------

2018-04-27T21:36:31Z thomberg1

it runs if I add the multiprocessing import and the if statement as show below - I don't work with keras so I don't have more insight

```

import multiprocessing

if \_\_name\_\_ == '\_\_main\_\_':

# MLP for Pima Indians Dataset with grid search via sklearn

from keras.models import Sequential

from keras.layers import Dense

from keras.wrappers.scikit\_learn import KerasClassifier

from sklearn.model\_selection import GridSearchCV

import numpy

# Function to create model, required for KerasClassifier

def create\_model(optimizer='rmsprop', init='glorot\_uniform'):

# create model

model = Sequential()

model.add(Dense(12, input\_dim=8, kernel\_initializer=init, activation='relu'))

model.add(Dense(8, kernel\_initializer=init, activation='relu'))

model.add(Dense(1, kernel\_initializer=init, activation='sigmoid'))

# Compile model

model.compile(loss='binary\_crossentropy', optimizer=optimizer, metrics=['accuracy'])

return model

# fix random seed for reproducibility

seed = 7

numpy.random.seed(seed)

# load pima indians dataset

dataset = numpy.loadtxt("pima-indians-diabetes.csv", delimiter=",")

# split into input (X) and output (Y) variables

X = dataset[:,0:8]

Y = dataset[:,8]

# create model

model = KerasClassifier(build\_fn=create\_model, verbose=0)

# grid search epochs, batch size and optimizer

optimizers = ['rmsprop', 'adam']

init = ['glorot\_uniform', 'normal', 'uniform']

epochs = [5]

batches = [5, 10, 20]

param\_grid = dict(optimizer=optimizers, epochs=epochs, batch\_size=batches, init=init)

grid = GridSearchCV(estimator=model, param\_grid=param\_grid, n\_jobs=12, verbose=1)

grid\_result = grid.fit(X, Y)

# summarize results

print("Best: %f using %s" % (grid\_result.best\_score\_, grid\_result.best\_params\_))

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("%f (%f) with: %r" % (mean, stdev, param))

```

### Fitting 3 folds for each of 18 candidates, totalling 54 fits

[Parallel(n\_jobs=12)]: Done 26 tasks | elapsed: 18.4s

[Parallel(n\_jobs=12)]: Done 54 out of 54 | elapsed: 23.7s finished

Best: 0.675781 using {'batch\_size': 5, 'epochs': 5, 'init': 'glorot\_uniform', 'optimizer': 'adam'}

0.621094 (0.036225) with: {'batch\_size': 5, 'epochs': 5, 'init': 'glorot\_uniform', 'optimizer': 'rmsprop'}

0.675781 (0.006379) with: {'batch\_size': 5, 'epochs': 5, 'init': 'glorot\_uniform', 'optimizer': 'adam'}

...

0.651042 (0.025780) with: {'batch\_size': 20, 'epochs': 5, 'init': 'uniform', 'optimizer': 'adam'}

----------------------------------------------

version info if needed

sys 3.6.4 |Anaconda custom (64-bit)| (default, Jan 16 2018, 12:04:33)

[GCC 4.2.1 Compatible Clang 4.0.1 (tags/RELEASE\_401/final)]

numpy 1.14.2

pandas 0.22.0

sklearn 0.19.1

torch 0.4.0a0+9692519

IPython 6.2.1

keras 2.1.5

compiler : GCC 4.2.1 Compatible Clang 4.0.1 (tags/RELEASE\_401/final)

system : Darwin

release : 17.5.0

machine : x86\_64

processor : i386

CPU cores : 24

interpreter: 64bit

-------------------------------------------------------------------------

2018-04-30T09:38:44Z siideffect

Thank you @thomberg1 , but adding

```

import multiprocessing

if \_\_name\_\_ == '\_\_main\_\_':

```

did not help. The problem is still the same

-------------------------------------------------------------------------

2018-05-20T20:18:26Z byrony

Same problem on my machine when using customized scoring function in `GridsearchCV`.

python 3.6.4,

scikit-learn 0.19.1,

windows 10.,

CPU cores: 24

-------------------------------------------------------------------------

2018-05-21T16:02:23Z amueller

@byrony can you provide code to reproduce? did you use ``if \_\_name\_\_ == "\_\_main\_\_"``?

-------------------------------------------------------------------------

2018-05-25T20:52:34Z Pazitos10

I've experienced a similar problem multiple times on my machine when using `n\_jobs=-1` or `n\_jobs=8` as an argument for `GridsearchCV` but using the default scorer argument.

\* Python 3.6.5,

\* scikit-learn 0.19.1,

\* Arch Linux,

\* CPU cores: 8.

Here is the code I used:

```python

from sklearn.model\_selection import GridSearchCV

from sklearn.preprocessing import MinMaxScaler

from sklearn.decomposition import PCA

from sklearn.utils import shuffle

from sklearn.neural\_network import MLPClassifier

import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

def main():

df = pd.read\_csv('../csvs/my\_data.csv', nrows=4000000)

X = np.array(list(map(lambda a: np.fromstring(a[1:-1] , sep=','), df['X'])))

y = np.array(list(map(lambda a: np.fromstring(a[1:-1] , sep=','), df['y'])))

scalerX = MinMaxScaler()

scalerY = MinMaxScaler()

X = scalerX.fit\_transform(X)

y = scalerY.fit\_transform(y)

grid\_params = {

'beta\_1': [ .1, .2, .3, .4, .5, .6, .7, .8, .9 ],

'activation': ['identity', 'logistic', 'tanh', 'relu'],

'learning\_rate\_init': [0.01, 0.001, 0.0001]

}

estimator = MLPClassifier(random\_state=1,

max\_iter=1000,

verbose=10,

early\_stopping=True)

gs = GridSearchCV(estimator,

grid\_params,

cv=5,

verbose=10,

return\_train\_score=True,

n\_jobs=8)

X, y = shuffle(X, y, random\_state=0)

y = y.astype(np.int16)

gs.fit(X, y.ravel())

print("GridSearchCV Report \n\n")

print("best\_estimator\_ {}".format(gs.best\_estimator\_))

print("best\_score\_ {}".format(gs.best\_score\_))

print("best\_params\_ {}".format(gs.best\_params\_))

print("best\_index\_ {}".format(gs.best\_index\_))

print("scorer\_ {}".format(gs.scorer\_))

print("n\_splits\_ {}".format(gs.n\_splits\_))

print("Exporting")

results = pd.DataFrame(data=gs.cv\_results\_)

results.to\_csv('../csvs/gs\_results.csv')

if \_\_name\_\_ == '\_\_main\_\_':

main()

```

I know is a big dataset so I expected it would take some time to get results but then after 2 days running, it just stopped working (the script keeps executing but is not using any resource apart from RAM and swap).

![captura de pantalla de 2018-05-25 17-53-11](https://user-images.githubusercontent.com/5199304/40565799-50f84b0a-6044-11e8-84c1-cd81845d0352.png)

![captura de pantalla de 2018-05-25 17-54-59](https://user-images.githubusercontent.com/5199304/40565857-914405f0-6044-11e8-98ea-e981a803a360.png)

Thanks in advance!

-------------------------------------------------------------------------

2018-05-26T00:04:26Z byrony

@amueller I didn't use the `if \_\_name\_\_ == "\_\_main\_\_"`. Below is my code, it only works when `n\_jobs=1`

```

def neg\_mape(true, pred):

true, pred = np.array(true)+0.01, np.array(pred)

return -1\*np.mean(np.absolute((true - pred)/true))

xgb\_test1 = XGBRegressor(

#learning\_rate =0.1,

n\_estimators=150,

max\_depth=3,

min\_child\_weight=1,

gamma=0,

subsample=0.8,

colsample\_bytree=0.8,

objective= 'reg:linear',

nthread=4,

scale\_pos\_weight=1,

seed=123,

)

param\_test1 = {

'learning\_rate':[0.01, 0.05, 0.1, 0.2, 0.3],

}

gsearch1 = GridSearchCV(estimator = xgb\_test1, param\_grid = param\_test1, scoring=neg\_mape, n\_jobs=4, cv = 5)

```

-------------------------------------------------------------------------

2018-05-26T15:22:31Z amueller

You're using XGBoost. I don't know what they do internally, it's very possible that's the issue. Can you try to see if adding the ``if \_\_name\_\_`` helps?

Otherwise I don't think there's a fix for that yet.

-------------------------------------------------------------------------

2018-05-26T15:23:26Z amueller

@Pazitos10 can you reproduce with synthetic data and/or smaller data? I can't reproduce without your data and it would be good to reproduce in shorter time.

-------------------------------------------------------------------------

2018-05-26T15:31:57Z Pazitos10

@amueller Ok, I will run it again with 500k rows and will post the results. Thanks!

-------------------------------------------------------------------------

2018-05-26T16:12:25Z Pazitos10

@amueller, running the script with 50k rows works as expected. The script ends correctly, showing the results as follows (sorry, I meant 50k not 500k):

![captura de pantalla de 2018-05-26 13-09-00](https://user-images.githubusercontent.com/5199304/40578161-ca1c8d92-60e5-11e8-953a-1b6b5a329a7d.png)

![captura de pantalla de 2018-05-26 13-09-51](https://user-images.githubusercontent.com/5199304/40578174-e4750b38-60e5-11e8-894b-494cdf0540f3.png)

The problem is that I don't know if these results are going to be the best for my whole dataset. Any advice?

-------------------------------------------------------------------------

2018-05-26T16:13:33Z amueller

Seems like you're running out of ram. Maybe try using Keras instead, it's likely a better solution for large scale neural nets.

-------------------------------------------------------------------------

2018-05-26T16:17:47Z Pazitos10

@amueller Oh, ok. I will try using Keras instead. Thank you again!

-------------------------------------------------------------------------

://user-images.githubusercontent.com/5199304/40578174-e4750b38-60e5-11e8-894b-494cdf0540f3.png)

The problem is that I don't know if these results are going to be the best for my whole dataset. Any advice?

-------------------------------------------------------------------------

2018-05-26T16:13:33Z amueller

Seems like you're running out of ram. Maybe try using Keras instead, it's likely a better solution for large scale neural nets.

-------------------------------------------------------------------------

2018-05-26T16:17:47Z Pazitos10

@amueller Oh, ok. I will try using Keras instead. Thank you again!

-------------------------------------------------------------------------