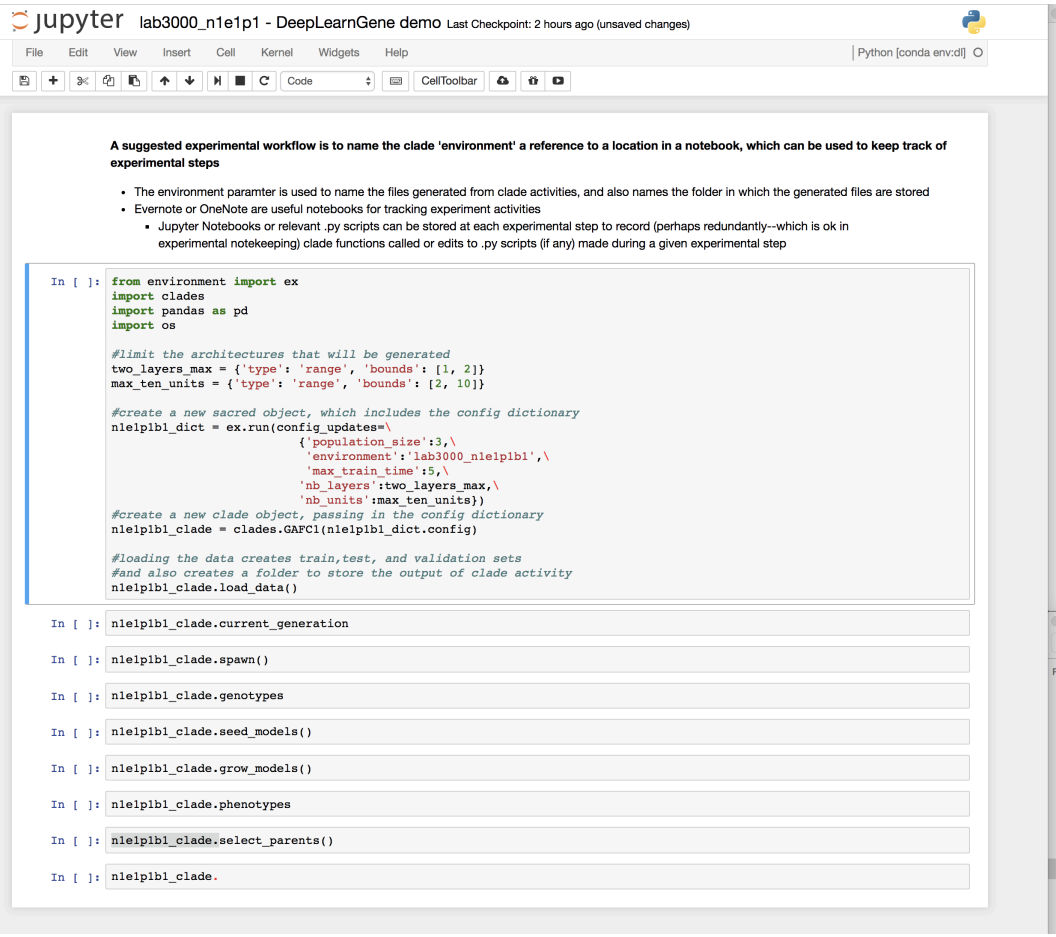
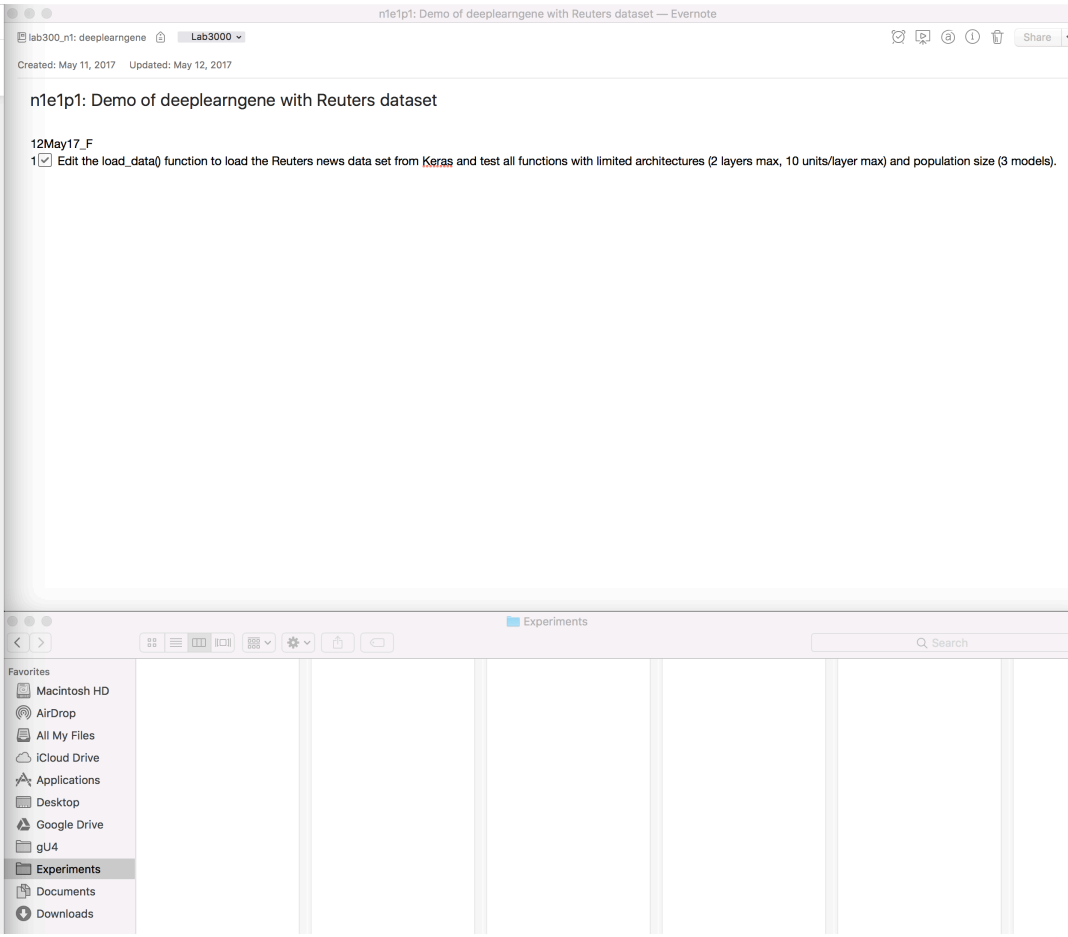

jupyter lab3000_n1e1p1 - DeepLearnGene demo Last Checkpoint: 2 hours ago (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
Python [conda env:dl]
Code
CellToolbar
A suggested experimental workflow is to name the clade 'environment' a reference to a location in a notebook, which can be used to keep track of experimental steps
The environment paramter is used to name the files generated from clade activities, and also names the folder in which the generated files are stored
Evernote or OneNote are useful notebooks for tracking experiment activities
Jupyter Notebooks or relevant .py scripts can be stored at each experimental step to record (perhaps redundantly--which is ok in experimental notekeeping) clade functions called or edits to .py scripts (if any) made during a given experimental step
In [ ]:
from environment import ex
import clades
import pandas as pd
import os
#limit the architectures that will be generated
two_layers_max = {'type': 'range', 'bounds': [1, 2]}
max_ten_units = {'type': 'range', 'bounds': [2, 10]}
#create a new sacred object, which includes the config dictionary
n1e1p1b1_dict = ex.run(config_updates=\
{'population_size':3,\
'environment':'lab3000_n1e1p1b1',\
'max_train_time':5,\
'nb_layers':two_layers_max,\
'nb_units':max_ten_units})
#create a new clade object, passing in the config dictionary
n1e1p1b1_clade = clades.GAFC1(n1e1p1b1_dict.config)
#loading the data creates train,test, and validation sets
#and also creates a folder to store the output of clade activity
n1e1p1b1_clade.load_data()
In [ ]: n1e1p1b1_clade.current_generation
In [ ]: n1e1p1b1_clade.spawn()
In [ ]: n1e1p1b1_clade.genotypes
In [ ]: n1e1p1b1_clade.seed_models()
In [ ]: n1e1p1b1_clade.grow_models()
In [ ]: n1e1p1b1_clade.phenotypes
In [ ]: n1e1p1b1_clade.select_parents()
In [ ]: n1e1p1b1_clade.


n1e1p1: Demo of deeplearngene with Reters dataset — Evernote
lab300_n1: deeplearngene
Lab3000
Share
Created: May 11, 2017 Updated: May 12, 2017
n1e1p1: Demo of deeplearngene with Reuters dataset
12May17_F
1 Edit the load_data() function to load the Reuters news data set from Keras and test all functions with limited architectures (2 layers max, 10 units/layer max) and population size (3 models).
Experiments
Search
Favorites
Macintosh HD
AirDrop
All My Files
iCloud Drive
Applications
Desktop
Google Drive
gU4
Experiments
Documents
Downloads

jupyter lab3000_n1e1p1 - DeepLearnGene demo Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help | Python [conda env:dl]

Code CellToolbar

A suggested experimental workflow is to name the clade 'environment' a reference to a location in a notebook, which can be used to keep track of experimental steps

- The environment paramter is used to name the files generated from clade activities, and also names the folder in which the generated files are stored
- Evernote or OneNote are useful notebooks for tracking experiment activities
  - Jupyter Notebooks or relevant .py scripts can be stored at each experimental step to record (perhaps redundantly--which is ok in experimental notekeeping) clade functions called or edits to .py scripts (if any) made during a given experimental step

In [1]:
```
from environment import ex
import clades
import pandas as pd
import os

#limit the architectures that will be generated
two_layers_max = {'type': 'range', 'bounds': [1, 2]}
max_ten_units = {'type': 'range', 'bounds': [2, 10]}

#create a new sacred object, which includes the config dictionary
n1e1p1b1_dict = ex.run(config_updates=\
                       {'population_size':3,\
                        'environment':'lab3000_n1e1p1b1',\
                        'max_train_time':5,\
                       'nb_layers':two_layers_max,\
                       'nb_units':max_ten_units})
#create a new clade object, passing in the config dictionary
n1e1p1b1_clade = clades.GAFC1(n1e1p1b1_dict.config)

#loading the data creates train,test, and validation sets
#and also creates a folder to store the output of clade activity
n1e1p1b1_clade.load_data()
```

```
Using TensorFlow backend.
WARNING - DLGn1e1p1 - No observers have been added to this run
INFO - DLGn1e1p1 - Running command 'main'
INFO - DLGn1e1p1 - Started
INFO - DLGn1e1p1 - Completed after 0:00:00

Vectorizing sequence data...
x_ shape: (8982, 10000)
46 classes
Converting class vector to binary class matrix (for use with categorical_crossentropy)
```

- Initially the output folder is empty
- Generations are 0-indexed

In [2]:
```
n1e1p1b1_clade.current_generation
```

Out[2]: 0

In [ ]:
```
n1e1p1b1_clade.spawn()
```

In [ ]:
```
n1e1p1b1_clade.genotypes
```

In [ ]:
```
n1e1p1b1_clade.seed_models()
```

In [ ]:
```
n1e1p1b1_clade.grow_models()
```

In [ ]:
```
n1e1p1b1_clade.phenotypes
```

In [ ]:
```
n1e1p1b1_clade.select_parents()
```

In [ ]:
```
n1e1p1b1_clade.
```

n1e1p1: Demo of deeplearngene with Reters dataset — Evernote

lab300_n1: deeplearngene Lab3000 Share

Created: May 11, 2017 Updated: May 12, 2017

n1e1p1: Demo of deeplearngene with Reuters dataset

12May17_F

1 ☑ Edit the load_data() function to load the Reuters news data set from Keras and test all functions with limited architectures (2 layers max, 10 units/layer max) and population size (3 models).

lab3000_n1e1p1b1

Favorites
- Macintosh HD
- AirDrop
- All My Files
- iCloud Drive
- Applications
- Desktop
- Google Drive
- gU4
- Experiments
- Documents
- Downloads

Devices
- Remote Disc

Tags
- Red
- Orange
- Yellow
- Green
- Blue
- Purple

lab3000_n1e1p1b1

jupyter lab3000_n1e1p1 - DeepLearnGene demo Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help | Python [conda env:dl]

Code CellToolbar

```
two_layers_max = {'type': 'range', 'bounds': [1, 2]}
max_ten_units = {'type': 'range', 'bounds': [2, 10]}

#create a new sacred object, which includes the config dictionary
n1e1p1b1_dict = ex.run(config_updates=\
                       {'population_size':3,\
                        'environment':'lab3000_n1e1p1b1',\
                        'max_train_time':5,\
                       'nb_layers':two_layers_max,\
                       'nb_units':max_ten_units})
#create a new clade object, passing in the config dictionary
n1e1p1b1_clade = clades.GAFC1(n1e1p1b1_dict.config)

#loading the data creates train,test, and validation sets
#and also creates a folder to store the output of clade activity
n1e1p1b1_clade.load_data()
```

```
Using TensorFlow backend.
WARNING - DLGn1e1p1 - No observers have been added to this run
INFO - DLGn1e1p1 - Running command 'main'
INFO - DLGn1e1p1 - Started
INFO - DLGn1e1p1 - Completed after 0:00:00

Vectorizing sequence data...
x_ shape: (8982, 10000)
46 classes
Converting class vector to binary class matrix (for use with categorical_crossentropy)
```

- Initially the output folder is empty
- Generations are 0-indexed

In [2]:
```
n1e1p1b1_clade.current_generation
```
Out[2]: 0

- spawn() creates a pandas dataframe of genes which 'encode' the model architectures of a given population
- the dataframe is saved as a property and also pickled into the experiment folder
  - Note that the pickled dataframe file, and gene and model name includes reference to the generation (Gen0)

In [3]:
```
n1e1p1b1_clade.spawn()
```

In [4]:
```
n1e1p1b1_clade.genotypes
```
Out[4]:

	LR	activations	batch_size	epochs	gene_name	layer_units	loss	model_name
0	0.374050	[relu]	512	4	lab3000_n1e1p1b1+Gen0+gene0	[8]	categorical_crossentropy	lab3000_n1e1p1b1+Gen0+gene0-
0	0.296663	[relu, softplus]	512	16	lab3000_n1e1p1b1+Gen0+gene1	[8, 9]	categorical_crossentropy	lab3000_n1e1p1b1+Gen0+gene1-
0	0.006975	[softmax, sigmoid]	512	9	lab3000_n1e1p1b1+Gen0+gene2	[7, 6]	categorical_crossentropy	lab3000_n1e1p1b1+Gen0+gene2-

In [ ]:
```
n1e1p1b1_clade.seed_models()
```

In [ ]:
```
n1e1p1b1_clade.grow_models()
```

In [ ]:
```
n1e1p1b1_clade.phenotypes
```

In [ ]:
```
n1e1p1b1_clade.select_parents()
```

In [ ]:
```
n1e1p1b1_clade.
```

n1e1p1: Demo of deeplearngene with Reters dataset — Evernote

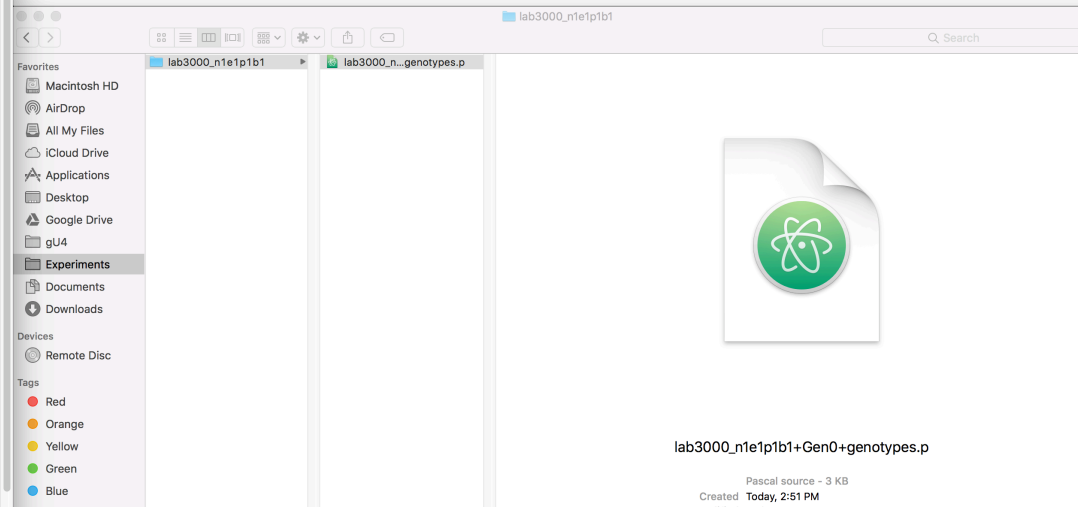
lab300_n1: deeplearngene Lab3000 Share

Created: May 11, 2017 Updated: May 12, 2017

n1e1p1: Demo of deeplearngene with Reuters dataset

12May17_F

1 ☑ Edit the load_data() function to load the Reuters news data set from Keras and test all functions with limited architectures (2 layers max, 10 units/layer max) and population size (3 models).



jupyter lab3000_n1e1p1 - DeepLearnGene demo Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help | Python [conda env:dl]

Code CellToolbar

```
                                'nb_units':max_ten_units})
#create a new clade object, passing in the config dictionary
n1e1p1b1_clade = clades.GAFC1(n1e1p1b1_dict.config)

#loading the data creates train,test, and validation sets
#and also creates a folder to store the output of clade activity
n1e1p1b1_clade.load_data()
```

```
Using TensorFlow backend.
WARNING - DLGn1e1p1 - No observers have been added to this run
INFO - DLGn1e1p1 - Running command 'main'
INFO - DLGn1e1p1 - Started
INFO - DLGn1e1p1 - Completed after 0:00:00

Vectorizing sequence data...
x_ shape: (8982, 10000)
46 classes
Converting class vector to binary class matrix (for use with categorical_crossentropy)
```

- Initially the output folder is empty
- Generations are 0-indexed

In [2]:
```
n1e1p1b1_clade.current_generation
```
Out[2]: 0

- spawn() creates a pandas dataframe of genes which 'encode' the model architectures of a given population
- the dataframe is saved as a property and also pickled into the experiment folder
  - Note that the pickled dataframe file, and gene and model name includes reference to the generation (Gen0)

In [3]:
```
n1e1p1b1_clade.spawn()
```

In [4]:
```
n1e1p1b1_clade.genotypes
```
Out[4]:

	LR	activations	batch_size	epochs	gene_name	layer_units	loss	model_name
0	0.106895	[relu]	512	4	lab3000_n1e1p1b1+Gen0+gene0	[4]	categorical_crossentropy	lab3000_n1e1p1b1+Gen0+gene0-
0	0.065681	[relu, softplus]	512	16	lab3000_n1e1p1b1+Gen0+gene1	[4, 10]	categorical_crossentropy	lab3000_n1e1p1b1+Gen0+gene1-
0	0.004449	[softmax]	64	16	lab3000_n1e1p1b1+Gen0+gene2	[9]	categorical_crossentropy	lab3000_n1e1p1b1+Gen0+gene2-

Type *Markdown* and LaTeX: α^2

- seed_models() acts as an intermediary between genotypes and model evaluations, which are executed in grow_models()
- compiled models are saved as .h5 files in the experiment folder

In [5]:
```
n1e1p1b1_clade.seed_models()
```

In [ ]:
```
n1e1p1b1_clade.grow_models()
```

In [ ]:
```
n1e1p1b1_clade.phenotypes
```

In [ ]:
```
n1e1p1b1_clade.select_parents()
```

In [ ]:
```
n1e1p1b1_clade.
```

n1e1p1: Demo of deeplearngene with Reuters dataset — Evernote

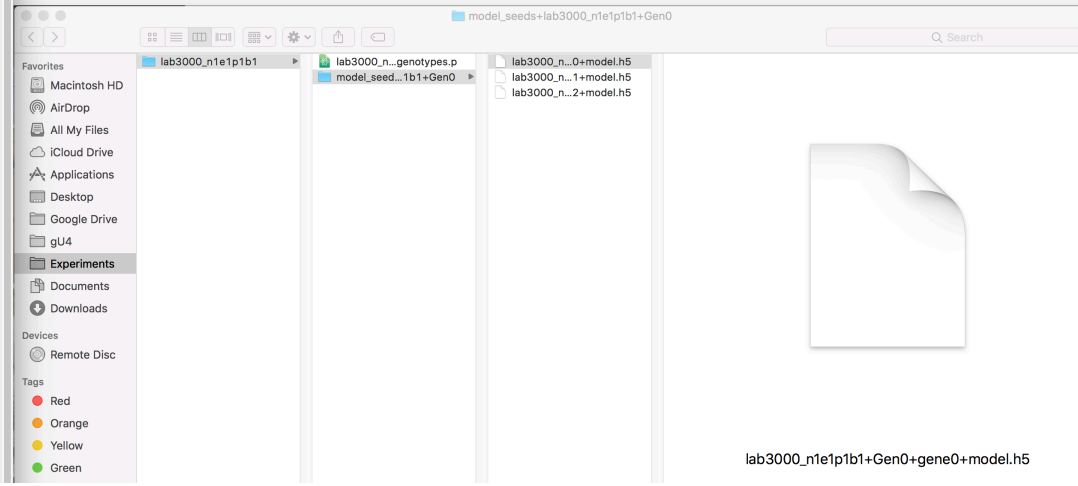
lab300_n1: deeplearngene Lab3000 Share

Created: May 11, 2017 Updated: May 12, 2017

n1e1p1: Demo of deeplearngene with Reuters dataset

12May17_F

1 ☑ Edit the load_data() function to load the Reuters news data set from Keras and test all functions with limited architectures (2 layers max, 10 units/layer max) and population size (3 models).



jupyter lab3000_n1e1p1 - DeepLearnGene demo Last Checkpoint: 2 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help | Python [conda env:dl]

Code CellToolbar

```
2080/2246 [==========================>...] - ETA: 0sthis is the index:  1
and this is the gene:  LR                                        0.0656806
activations                        [relu, softplus]
batch_size                                      512
epochs                                           16
gene_name               lab3000_n1e1p1b1+Gen0+gene1
layer_units                                 [4, 10]
loss                        categorical_crossentropy
model_name     lab3000_n1e1p1b1+Gen0+gene1+model.h5
nb_layers                                        2
optimizer                                  RMSProp
Name: 0, dtype: object
Train on 8083 samples, validate on 899 samples
Epoch 1/16
8083/8083 [==============================] - 2s - loss: 3.9295 - acc: 0.0040 - val_loss: 3.7433 - val_acc: 0.0044
Epoch 2/16
8083/8083 [==============================] - 1s - loss: 3.5759 - acc: 0.0073 - val_loss: 3.4173 - val_acc: 0.0189
Epoch 3/16
8083/8083 [==============================] - 1s - loss: 3.2314 - acc: 0.1466 - val_loss: 3.0809 - val_acc: 0.3471
Epoch 4/16
7680/8083 [===========================>..] - ETA: 0s - loss: 2.8953 - acc: 0.4474_______Stopping after 5 seconds.
8083/8083 [==============================] - 1s - loss: 2.8852 - acc: 0.4506 - val_loss: 2.7559 - val_acc: 0.4461
2246/2246 [==============================] - 0s
in the else
this is the index:  2
and this is the gene:  LR                                        0.00444869
activations                              [softmax]
batch_size                                      64
epochs                                           16
gene_name               lab3000_n1e1p1b1+Gen0+gene2
layer_units                                   [9]
loss                        categorical_crossentropy
model_name     lab3000_n1e1p1b1+Gen0+gene2+model.h5
nb_layers                                        1
optimizer                                     Adam
Name: 0, dtype: object
Train on 8083 samples, validate on 899 samples
Epoch 1/16
8083/8083 [==============================] - 3s - loss: 3.5798 - acc: 0.1379 - val_loss: 3.3895 - val_acc: 0.2191
Epoch 2/16
7936/8083 [============================>.] - ETA: 0s - loss: 3.2423 - acc: 0.2387_______Stopping after 5 seconds.
8083/8083 [==============================] - 2s - loss: 3.2388 - acc: 0.2380 - val_loss: 3.1041 - val_acc: 0.2570
2080/2246 [==========================>...] - ETA: 0sin the else
```

```
^^^verbose output of n1e1p1b1_clade.grow_models()
```

- grow_models() trains the models and generates pickled 'growth analyses' dataframes, one for each model trained, which include train and validation loss and accuracy for each batch and epoch, as well as the time take to run each batch and epoch
- grow_models() also pickles, and saves as a property, a phenotypes dataframe, which summarizes the performance of each model
  - the misclassed dictionaries store the true and labeled classes for each mislabeled datapoint
- grow_models() also saves each trained model as a .h5 file

In [7]:
```
n1e1p1b1_clade.phenotypes
```

Out[7]:

	gene_name	misclassed	test_accuracy	test_loss	time	train_accuracy	train_loss
0	lab3000_n1e1p1b1+Gen0+gene0	{'true_class': [3, 10, 1, 4, 3, 3, 3, 5, 1, 1,...	0.468833	3.168827	6.302856	0.467401	3.147856
0	lab3000_n1e1p1b1+Gen0+gene1	{'true_class': [10, 1, 4, 4, 5, 4, 1, 1, 11, 2...	0.455476	2.760931	6.278670	0.464803	2.702639
0	lab3000_n1e1p1b1+Gen0+gene2	{'true_class': [3, 10, 1, 3, 3, 3, 3, 3, 5, 1,...	0.253339	3.106247	5.817873	0.265124	3.091641

In [ ]:
```
n1e1p1b1_clade.select_parents()
```

In [ ]:
```
n1e1p1b1_clade.breed()
```

n1e1p1: Demo of deeplearngene with Reuters dataset — Evernote

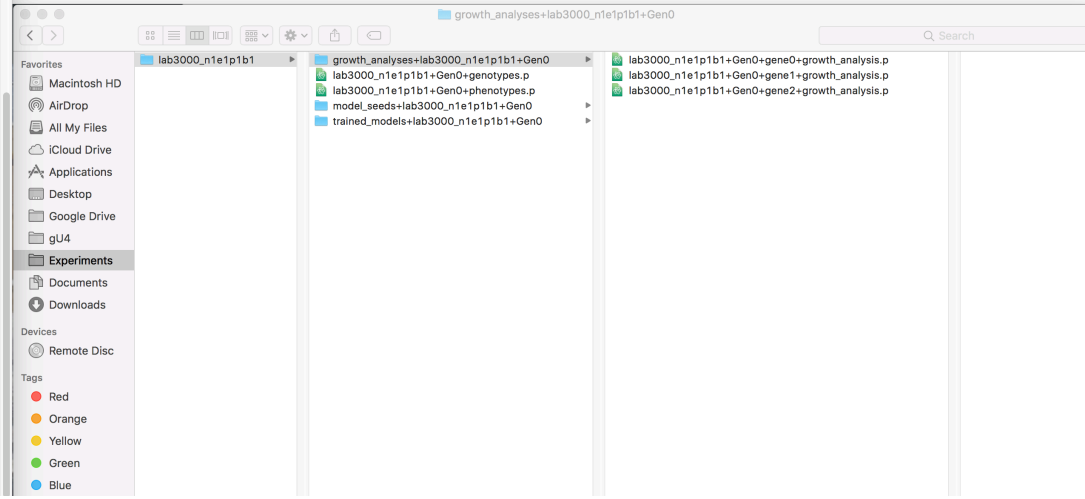
lab300_n1: deeplearngene Lab3000 Share

Created: May 11, 2017 Updated: May 12, 2017

n1e1p1: Demo of deeplearngene with Reuters dataset

12May17_F

1 ☑ Edit the load_data() function to load the Reuters news data set from Keras and test all functions with limited architectures (2 layers max, 10 units/layer max) and population size (3 models).



jupyter lab3000_n1e1p1 - DeepLearnGene demo Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help | Python [conda env:dl]

Code CellToolbar

```
Epoch 3/16
8083/8083 [==============================] - 1s - loss: 3.2314 - acc: 0.1466 - val_loss: 3.0809 - val_acc: 0.3471
Epoch 4/16
7680/8083 [===========================>..] - ETA: 0s - loss: 2.8953 - acc: 0.4474_______Stopping after 5 seconds.
8083/8083 [==============================] - 1s - loss: 2.8852 - acc: 0.4506 - val_loss: 2.7559 - val_acc: 0.4461
2246/2246 [==============================] - 0s
in the else
this is the index:  2
and this is the gene:  LR                                          0.00444869
activations                              [softmax]
batch_size                                      64
epochs                                          16
gene_name                lab3000_n1e1p1b1+Gen0+gene2
layer_units                                    [9]
loss                        categorical_crossentropy
model_name     lab3000_n1e1p1b1+Gen0+gene2+model.h5
nb_layers                                        1
optimizer                                     Adam
Name: 0, dtype: object
Train on 8083 samples, validate on 899 samples
Epoch 1/16
8083/8083 [==============================] - 3s - loss: 3.5798 - acc: 0.1379 - val_loss: 3.3895 - val_acc: 0.2191
Epoch 2/16
7936/8083 [============================>.] - ETA: 0s - loss: 3.2423 - acc: 0.2387_______Stopping after 5 seconds.
8083/8083 [==============================] - 2s - loss: 3.2388 - acc: 0.2380 - val_loss: 3.1041 - val_acc: 0.2570
2080/2246 [==========================>...] - ETA: 0sin the else
```

```
^^^verbose output of n1e1p1b1_clade.grow_models()
```

- grow_models() trains the models and generates pickled 'growth analyses' dataframes, one for each model trained, which include train and validation loss and accuracy for each batch and epoch, as well as the time take to run each batch and epoch
- grow_models() also pickles, and saves as a property, a phenotypes dataframe, which summarizes the performance of each model
  - the misclassed dictionaries store the true and labeled classes for each mislabeled datapoint
- grow_models() also saves each trained model as a .h5 file

In [7]:
```
n1e1p1b1_clade.phenotypes
```

Out[7]:

	gene_name	misclassed	test_accuracy	test_loss	time	train_accuracy	train_loss
0	lab3000_n1e1p1b1+Gen0+gene0	{'true_class': [3, 10, 1, 4, 3, 3, 3, 5, 1, 1,...	0.468833	3.168827	6.302856	0.467401	3.147856
0	lab3000_n1e1p1b1+Gen0+gene1	{'true_class': [10, 1, 4, 4, 5, 4, 1, 1, 11, 2...	0.455476	2.760931	6.278670	0.464803	2.702639
0	lab3000_n1e1p1b1+Gen0+gene2	{'true_class': [3, 10, 1, 3, 3, 3, 3, 3, 5, 1,...	0.253339	3.106247	5.817873	0.265124	3.091641

- select_parents() selects, by default, the top 20% of models by test accuracy, plut 10% random models; or if the population size is small, such as in this demo case, at least two parent models are selected

In [8]:
```
n1e1p1b1_clade.select_parents()
```

In [9]:
```
n1e1p1b1_clade.parent_genes
```

Out[9]:

	LR	activations	batch_size	epochs	gene_name	layer_units	loss	model_name
0	0.106895	[relu]	512	4	lab3000_n1e1p1b1+Gen0+gene0	[4]	categorical_crossentropy	lab3000_n1e1p1b1+Gen0+gene0-
0	0.004449	[softmax]	64	16	lab3000_n1e1p1b1+Gen0+gene2	[9]	categorical_crossentropy	lab3000_n1e1p1b1+Gen0+gene2-

In [ ]:
```
n1e1p1b1_clade.breed()
```

n1e1p1: Demo of deeplearngene with Reuters dataset — Evernote

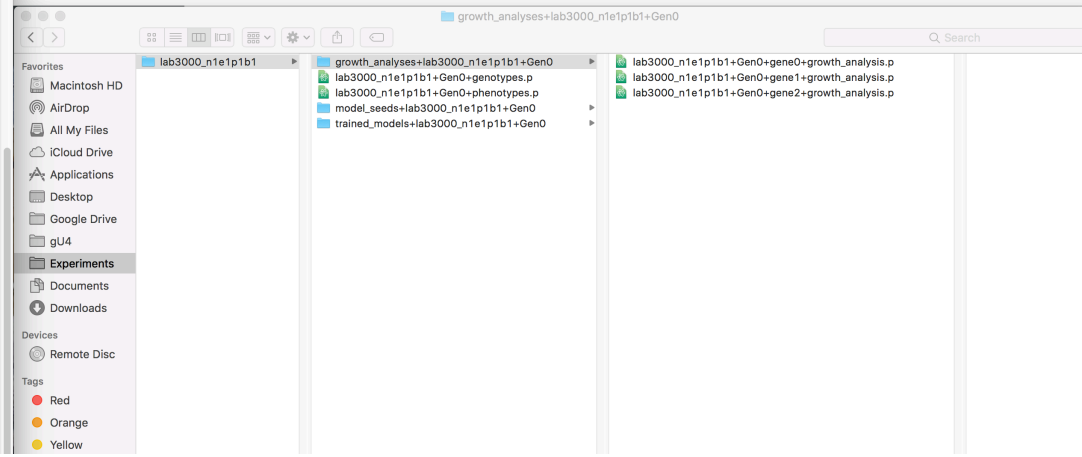
lab300_n1: deeplearngene Lab3000 Share

Created: May 11, 2017 Updated: May 12, 2017

n1e1p1: Demo of deeplearngene with Reuters dataset

12May17_F

1 ☑ Edit the load_data() function to load the Reuters news data set from Keras and test all functions with limited architectures (2 layers max, 10 units/layer max) and population size (3 models).



jupyter lab3000_n1e1p1 - DeepLearnGene demo Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help | Python [conda env:dl]

Code CellToolbar

```
Epoch 2/16
7936/8083 [============================>.] - ETA: 0s - loss: 3.2423 - acc: 0.2387_______Stopping after 5 seconds.
8083/8083 [==============================] - 2s - loss: 3.2388 - acc: 0.2380 - val_loss: 3.1041 - val_acc: 0.2570
2080/2246 [==========================>...] - ETA: 0sin the else
```

^^^verbose output of n1e1p1b1_clade.grow_models()

- grow_models() trains the models and generates pickled 'growth analyses' dataframes, one for each model trained, which include train and validation loss and accuracy for each batch and epoch, as well as the time take to run each batch and epoch
- grow_models() also pickles, and saves as a property, a phenotypes dataframe, which summarizes the performance of each model
  - the misclassed dictionaries store the true and labeled classes for each mislabeled datapoint
- grow_models() also saves each trained model as a .h5 file

In [7]:
```
n1e1p1b1_clade.phenotypes
```

Out[7]:

	gene_name	misclassed	test_accuracy	test_loss	time	train_accuracy	train_loss
0	lab3000_n1e1p1b1+Gen0+gene0	{'true_class': [3, 10, 1, 4, 3, 3, 3, 5, 1, 1,...	0.468833	3.168827	6.302856	0.467401	3.147856
0	lab3000_n1e1p1b1+Gen0+gene1	{'true_class': [10, 1, 4, 4, 5, 4, 1, 1, 11, 2...	0.455476	2.760931	6.278670	0.464803	2.702639
0	lab3000_n1e1p1b1+Gen0+gene2	{'true_class': [3, 10, 1, 3, 3, 3, 3, 3, 5, 1,...	0.253339	3.106247	5.817873	0.265124	3.091641

- select_parents() selects, by default, the top 20% of models by test accuracy, plut 10% random models; or if the population size is small, such as in this demo case, at least two parent models are selected

In [8]:
```
n1e1p1b1_clade.select_parents()
```

In [9]:
```
n1e1p1b1_clade.parent_genes
```

Out[9]:

	LR	activations	batch_size	epochs	gene_name	layer_units	loss	model_name
0	0.106895	[relu]	512	4	lab3000_n1e1p1b1+Gen0+gene0	[4]	categorical_crossentropy	lab3000_n1e1p1b1+Gen0+gene0-
0	0.004449	[softmax]	64	16	lab3000_n1e1p1b1+Gen0+gene2	[9]	categorical_crossentropy	lab3000_n1e1p1b1+Gen0+gene2-

- breed() generates a new population of genes, encoding a new generation of models; note that current_generation is incremented when clade.breed() is run

In [10]:
```
n1e1p1b1_clade.breed()
```

In [11]:
```
n1e1p1b1_clade.current_generation
```

Out[11]: 1

In [12]:
```
n1e1p1b1_clade.genotypes
```

Out[12]:

	LR	activations	batch_size	epochs	gene_name	layer_units	model_name	nb_layers	opt
0	0.004449	[relu]	512	16	lab3000_n1e1p1b1+Gen1+gene0	[9]	lab3000_n1e1p1b1+Gen1+gene0+model.h5	1	Ada
1	0.004449	[relu]	64	4	lab3000_n1e1p1b1+Gen1+gene1	[9]	lab3000_n1e1p1b1+Gen1+gene1+model.h5	1	Ada
2	0.004449	[softmax]	64	4	lab3000_n1e1p1b1+Gen1+gene2	[4]	lab3000_n1e1p1b1+Gen1+gene2+model.h5	1	Ada

In [ ]:

n1e1p1: Demo of deeplearngene with Reuters dataset — Evernote

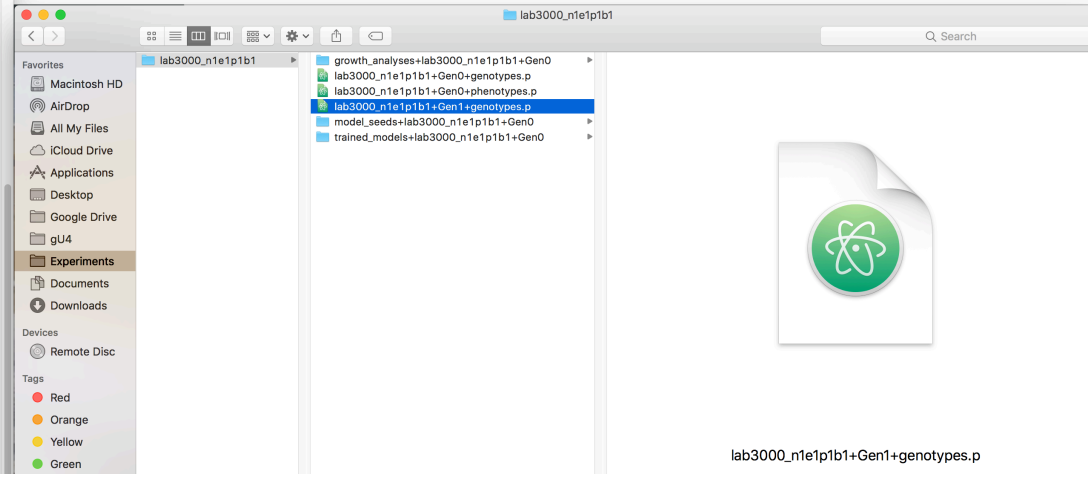
lab300_n1: deeplearngene Lab3000 Share

Created: May 11, 2017 Updated: May 12, 2017

n1e1p1: Demo of deeplearngene with Reuters dataset

12May17_F

1 ☑ Edit the load_data() function to load the Reuters news data set from Keras and test all functions with limited architectures (2 layers max, 10 units/layer max) and population size (3 models).



jupyter lab3000_n1e1p1 - DeepLearnGene demo Last Checkpoint: 2 hours ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help | Python [conda env:dl]

Code CellToolbar

- the misclassed dictionaries store the true and labeled classes for each mislabeled datapoint
- grow_models() also saves each trained model as a .h5 file

In [7]:
```
n1e1p1b1_clade.phenotypes
```

Out[7]:

	gene_name	misclassed	test_accuracy	test_loss	time	train_accuracy	train_loss
0	lab3000_n1e1p1b1+Gen0+gene0	{'true_class': [3, 10, 1, 4, 3, 3, 3, 5, 1, 1,...	0.468833	3.168827	6.302856	0.467401	3.147856
0	lab3000_n1e1p1b1+Gen0+gene1	{'true_class': [10, 1, 4, 4, 5, 4, 1, 1, 11, 2...	0.455476	2.760931	6.278670	0.464803	2.702639
0	lab3000_n1e1p1b1+Gen0+gene2	{'true_class': [3, 10, 1, 3, 3, 3, 3, 3, 5, 1,...	0.253339	3.106247	5.817873	0.265124	3.091641

- select_parents() selects, by default, the top 20% of models by test accuracy, plut 10% random models; or if the population size is small, such as in this demo case, at least two parent models are selected

In [8]:
```
n1e1p1b1_clade.select_parents()
```

In [9]:
```
n1e1p1b1_clade.parent_genes
```

Out[9]:

	LR	activations	batch_size	epochs	gene_name	layer_units	loss	model_name
0	0.106895	[relu]	512	4	lab3000_n1e1p1b1+Gen0+gene0	[4]	categorical_crossentropy	lab3000_n1e1p1b1+Gen0+gene0-
0	0.004449	[softmax]	64	16	lab3000_n1e1p1b1+Gen0+gene2	[9]	categorical_crossentropy	lab3000_n1e1p1b1+Gen0+gene2-

- breed() generates a new population of genes, encoding a new generation of models; note that current_generation is incremented when clade.breed() is run

In [10]:
```
n1e1p1b1_clade.breed()
```

In [11]:
```
n1e1p1b1_clade.current_generation
```

Out[11]: 1

In [12]:
```
n1e1p1b1_clade.genotypes
```

Out[12]:

	LR	activations	batch_size	epochs	gene_name	layer_units	model_name	nb_layers	opt
0	0.004449	[relu]	512	16	lab3000_n1e1p1b1+Gen1+gene0	[9]	lab3000_n1e1p1b1+Gen1+gene0+model.h5	1	Ada
1	0.004449	[relu]	64	4	lab3000_n1e1p1b1+Gen1+gene1	[9]	lab3000_n1e1p1b1+Gen1+gene1+model.h5	1	Ada
2	0.004449	[softmax]	64	4	lab3000_n1e1p1b1+Gen1+gene2	[4]	lab3000_n1e1p1b1+Gen1+gene2+model.h5	1	Ada

after model evolution is run interactively, the commands can be saved to the experiment notebook (here, Evernote)

n1e1p1: Demo of deeplearngene with Reuters dataset — Evernote

lab300_n1: deeplearngene Lab3000 Share

Created: May 11, 2017 Updated: May 12, 2017

n1e1p1: Demo of deeplearngene with Reuters dataset

12May17_F

1 ☑ Edit the load_data() function to load the Reuters news data set from Keras and test all functions with limited architectures (2 layers max, 10 units/layer max) and population size (3 models).

lab3000_n1...demo.ipynb
23.6 KB