In [1]:
```
import h2o
from h2o.estimators.deeplearning import H2ODeepLearningEstimator
```

In [2]:
```
h2o.init()
```

```
Checking whether there is an H2O instance running at http://localhost:54321. (h
ttp://localhost:54321.) connected.
Warning: Your H2O cluster version is too old (7 months and 8 days)! Please down
load and install the latest version from http://h2o.ai/download/ (http://h2o.a
i/download/)
```

H2O cluster uptime:	5 days 0 hours 44 mins
H2O cluster version:	3.13.0.369
H2O cluster version age:	7 months and 8 days !!!
H2O cluster name:	H2O_from_python_Craig_6veplf
H2O cluster total nodes:	1
H2O cluster free memory:	10.01 Gb
H2O cluster total cores:	16
H2O cluster allowed cores:	16
H2O cluster status:	locked, healthy
H2O connection url:	http://localhost:54321
H2O connection proxy:	None
H2O internal security:	False
H2O API Extensions:	Algos, AutoML, Core V3, Core V4
Python version:	3.5.4 final

In [3]:
```
# Convert from XLSX to csv
df = h2o.import_file('Copy of Training Dataset.csv',col_types = {'Education Level

response = 'Employment Status'
predictors = list(set(df.col_names)-set(response))
```

```
Parse progress: |██████████████████████████████████████████████████████████| 10
0%
```

In [4]:
```
df_earning = df[df['Weekly Earnings']==0]
```

In [5]:
```
train,test = df_earning.split_frame(ratios=[.8],seed =3)
```

In [6]:
```
train['cv']=train.kfold_column(n_folds=5,seed=3)
```

In [7]:
```
train1 = train[train['cv']==0].rbind(train[train['cv']==1]).rbind(train[train['cv
train2 = train[train['cv']==4].rbind(train[train['cv']==1]).rbind(train[train['cv
train3 = train[train['cv']==0].rbind(train[train['cv']==4]).rbind(train[train['cv
train4 = train[train['cv']==0].rbind(train[train['cv']==1]).rbind(train[train['cv
train5 = train[train['cv']==0].rbind(train[train['cv']==1]).rbind(train[train['cv

test1 = train[train['cv']==4]
test2 = train[train['cv']==0]
test3 = train[train['cv']==1]
test4 = train[train['cv']==2]
test5 = train[train['cv']==3]
```

In [8]:
```
train_list = [train1, train2, train3, train4, train5]
test_list = [test1,test2,test3,test4, test5]
```

In [9]:
```
models = []
for i in range(5):
    dl = H2ODeepLearningEstimator(hidden=[300,300], epochs=1000, seed = 3,
                                  standardize=True,ignored_columns=['id'])
    dl.train(predictors,response,training_frame=train_list[i])
    predict = dl.predict(test_data=test_list[i])
    predict['true']=test_list[i]['Employment Status']
    h2o.export_file(predict,'cv'+str(i+1)+'.csv')
    models.append(dl)
```

```
deeplearning Model Build progress: |██████████████████████████████████████| 10
0%
deeplearning prediction progress: |███████████████████████████████████████| 10
0%
Export File progress: |███████████████████████████████████████████████████| 10
0%
deeplearning Model Build progress: |██████████████████████████████████████| 10
0%
deeplearning prediction progress: |███████████████████████████████████████| 10
0%
Export File progress: |███████████████████████████████████████████████████| 10
0%
deeplearning Model Build progress: |██████████████████████████████████████| 10
0%
deeplearning prediction progress: |███████████████████████████████████████| 10
0%
Export File progress: |███████████████████████████████████████████████████| 10
0%
deeplearning Model Build progress: |██████████████████████████████████████| 10
0%
deeplearning prediction progress: |███████████████████████████████████████| 10
0%
Export File progress: |███████████████████████████████████████████████████| 10
0%
deeplearning Model Build progress: |██████████████████████████████████████| 10
0%
deeplearning prediction progress: |███████████████████████████████████████| 10
0%
Export File progress: |███████████████████████████████████████████████████| 10
0%
```

Predict

In [78]:

```
df_pr = h2o.import_file('Copy of Test Dataset.csv',col_types = {'Education Level'
response = 'Employment Status'
predictors = list(set(df.col_names)-set(response))
```

```
Parse progress: |████████████████████████████████████████████████████████| 10
0%
```

In [79]:

```
df_pr = df_pr.drop([0],0)
```

In [80]:

```
names = ['id']
for i in df_pr.col_names[1:]:
    names.append(i)
df_pr.set_names(names=names)
```

id	Education Level	Age	Age Range	Employment Status	Gender	Children	Weekly Earnings	Year	Weekly Hours Worked	Sleeping	G
1	11th grade	17	0-19	nan	Male	2	240	2005	22	570	
2	Master	49	40-49	nan	Female	0	0	2005	0	555	
3	Bachelor	40	40-49	nan	Male	3	1470	2005	35	613	
4	Bachelor	46	40-49	nan	Female	2	673	2005	60	550	
5	High School	39	30-39	nan	Female	2	1385	2005	40	570	
6	Some College	80	80+	nan	Female	0	125	2005	12	585	
7	Master	38	30-39	nan	Female	3	0	2005	0	540	
8	11th grade	76	70-79	nan	Female	0	0	2005	0	645	
9	Bachelor	43	40-49	nan	Male	2	1769	2005	45	585	
10	9th grade	16	0-19	nan	Female	3	50	2005	20	485	

Out[80]:

In [81]:

```
dl = models[0]
```

In [82]:
```
test_predict = dl.predict(df_pr)
```

```
deeplearning prediction progress: |███████████████████████████████████████| 10
0%

c:\users\craig\appdata\local\programs\python\python35\lib\site-packages\h2o\jo
b.py:69: UserWarning: Test/Validation dataset column 'Education Level' has leve
ls not trained on: [Education Level]
  warnings.warn(w)
c:\users\craig\appdata\local\programs\python\python35\lib\site-packages\h2o\jo
b.py:69: UserWarning: Test/Validation dataset column 'Age Range' has levels not
trained on: [Age Range]
  warnings.warn(w)
c:\users\craig\appdata\local\programs\python\python35\lib\site-packages\h2o\jo
b.py:69: UserWarning: Test/Validation dataset column 'Gender' has levels not tr
ained on: [Gender]
  warnings.warn(w)
c:\users\craig\appdata\local\programs\python\python35\lib\site-packages\h2o\jo
b.py:69: UserWarning: Test/Validation dataset is missing column 'Id': substitut
ing in a column of NaN
  warnings.warn(w)
```

In [83]:
```
df_pr['predict'] = test_predict['predict']
```

In [85]:
```
h2o.export_file(df_pr,'test_predict_ANN.csv')
```

```
Export File progress: |███████████████████████████████████████████████████| 10
0%
```