

PyNeb_manual_9

June 3, 2020

```
[2]: import numpy as np
import matplotlib.pyplot as plt
import time
import pyneb as pn
```

```
[4]: # This is the AAN interface.
try:
    import ai4neb
except:
    !pip install -U git+https://github.com/morisset/AI4neb.git
    import ai4neb
```

```
[5]: O3 = pn.Atom('O',3)
# Set the Artificial Neuron Network verbosity to True:
O3.ANN_inst_kwargs['verbose'] = True
N2 = pn.Atom('N',2)
N2.ANN_inst_kwargs['verbose'] = True
S2 = pn.Atom('S',2)
S2.ANN_inst_kwargs['verbose'] = True
```

```
[6]: # Number of data points to test the ANN option speedup
N_rand = 1000
# Define some random values to test the
ratio = 150.+ 10*np.random.randn(N_rand)
den = 150.+ 10*np.random.randn(N_rand)
ratio_S2 = 1 + 0.1*np.random.randn(N_rand)
tem_S2 = 11000 + 2000 * np.random.randn(N_rand)
```

```
[7]: start = time.time()
resa = O3.getTemDen(ratio, den=1e2, wave1=5007, wave2=4363)

resb = O3.getTemDen(ratio, den=den, wave1=5007, wave2=4363)
resc = N2.getTemDen(ratio, den=den,
                    to_eval = '(L(6584) + L(6548)) / L(5755)')
resd = S2.getTemDen(ratio_S2, tem=11000, wave1=6716, wave2=6731)

rese = S2.getTemDen(ratio_S2, tem=tem_S2, wave1=6716, wave2=6731)
```

```
end = time.time()
print(end-start)
```

18.425156116485596

```
[8]: start = time.time()
res2a = O3.getTemDen(ratio, den=1e2, wave1=5007, wave2=4363, method='ANN')

res2b = O3.getTemDen(ratio, den=den, wave1=5007, wave2=4363, method='ANN')
res2c = N2.getTemDen(ratio, den=den,
                    to_eval = '(L(6584) + L(6548)) / L(5755)',
                    method='ANN')
res2d = S2.getTemDen(ratio_S2, tem=11000, wave1=6716, wave2=6731,
                    method='ANN', end_x=1e4)
res2e = S2.getTemDen(ratio_S2, tem=tem_S2, wave1=6716, wave2=6731,
                    method='ANN', end_x=1e4)

end = time.time()
print(end-start)
```

Instantiation. V 0.17

Training set size = 30, Test set size = 0

Train data scaled. Log10 applied.

Test data scaled.

Training set size = 30, Test set size = 0

Training set size = 30, Test set size = 0

Regression Model SK_ANN

Training 2 inputs for 1 outputs with 30 data

RM trained, with 799 iterations. Score = 1.000

```
MLPRegressor(activation='tanh', alpha=0.0001, batch_size='auto', beta_1=0.9,
             beta_2=0.999, early_stopping=False, epsilon=1e-08,
             hidden_layer_sizes=(10, 10), learning_rate='constant',
             learning_rate_init=0.001, max_fun=15000, max_iter=20000,
             momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,
             power_t=0.5, random_state=None, shuffle=True, solver='lbfgs',
             tol=1e-06, validation_fraction=0.1, verbose=False,
             warm_start=False)
```

Training time 0.2 s.

Test data scaled. Log10 applied.

Training set size = 30, Test set size = 1000

Predicting from 2 inputs to 1 outputs using 1000 data in 0.00 secs.

Instantiation. V 0.17

Training set size = 900, Test set size = 0

Train data scaled. Log10 applied.

Test data scaled.

Training set size = 900, Test set size = 0

Training set size = 900, Test set size = 0

Regression Model SK_ANN

Training 2 inputs for 1 outputs with 900 data
 RM trained, with 388 iterations. Score = 1.000
 MLPRegressor(activation='tanh', alpha=0.0001, batch_size='auto', beta_1=0.9,
 beta_2=0.999, early_stopping=False, epsilon=1e-08,
 hidden_layer_sizes=(10, 10), learning_rate='constant',
 learning_rate_init=0.001, max_fun=15000, max_iter=20000,
 momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,
 power_t=0.5, random_state=None, shuffle=True, solver='lbfgs',
 tol=1e-06, validation_fraction=0.1, verbose=False,
 warm_start=False)
 Training time 0.3 s.
 Test data scaled. Log10 applied.
 Training set size = 900, Test set size = 1000
 Predicting from 2 inputs to 1 outputs using 1000 data in 0.00 secs.
 Instantiation. V 0.17
 Training set size = 900, Test set size = 0
 Train data scaled. Log10 applied.
 Test data scaled.
 Training set size = 900, Test set size = 0
 Training set size = 900, Test set size = 0
 Regression Model SK_ANN
 Training 2 inputs for 1 outputs with 900 data
 RM trained, with 334 iterations. Score = 1.000
 MLPRegressor(activation='tanh', alpha=0.0001, batch_size='auto', beta_1=0.9,
 beta_2=0.999, early_stopping=False, epsilon=1e-08,
 hidden_layer_sizes=(10, 10), learning_rate='constant',
 learning_rate_init=0.001, max_fun=15000, max_iter=20000,
 momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,
 power_t=0.5, random_state=None, shuffle=True, solver='lbfgs',
 tol=1e-06, validation_fraction=0.1, verbose=False,
 warm_start=False)
 Training time 0.2 s.
 Test data scaled. Log10 applied.
 Training set size = 900, Test set size = 1000
 Predicting from 2 inputs to 1 outputs using 1000 data in 0.00 secs.
 Instantiation. V 0.17
 Training set size = 30, Test set size = 0
 Train data scaled. Log10 applied.
 Test data scaled.
 Training set size = 30, Test set size = 0
 Training set size = 30, Test set size = 0
 Regression Model SK_ANN
 Training 2 inputs for 1 outputs with 30 data
 RM trained, with 267 iterations. Score = 1.000
 MLPRegressor(activation='tanh', alpha=0.0001, batch_size='auto', beta_1=0.9,
 beta_2=0.999, early_stopping=False, epsilon=1e-08,
 hidden_layer_sizes=(10, 10), learning_rate='constant',
 learning_rate_init=0.001, max_fun=15000, max_iter=20000,

```

        momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,
        power_t=0.5, random_state=None, shuffle=True, solver='lbfgs',
        tol=1e-06, validation_fraction=0.1, verbose=False,
        warm_start=False)
Training time 0.1 s.
Test data scaled. Log10 applied.
Training set size = 30, Test set size = 1000
Predicting from 2 inputs to 1 outputs using 1000 data in 0.00 secs.
Instantiation. V 0.17
Training set size = 900, Test set size = 0
Train data scaled. Log10 applied.
Test data scaled.
Training set size = 900, Test set size = 0
Training set size = 900, Test set size = 0
Regression Model SK_ANN
Training 2 inputs for 1 outputs with 900 data
RM trained, with 1140 iterations. Score = 1.000
MLPRegressor(activation='tanh', alpha=0.0001, batch_size='auto', beta_1=0.9,
             beta_2=0.999, early_stopping=False, epsilon=1e-08,
             hidden_layer_sizes=(10, 10), learning_rate='constant',
             learning_rate_init=0.001, max_fun=15000, max_iter=20000,
             momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,
             power_t=0.5, random_state=None, shuffle=True, solver='lbfgs',
             tol=1e-06, validation_fraction=0.1, verbose=False,
             warm_start=False)
Training time 0.8 s.
Test data scaled. Log10 applied.
Training set size = 900, Test set size = 1000
Predicting from 2 inputs to 1 outputs using 1000 data in 0.00 secs.
1.6784470081329346

```

```

[9]: # Compare the results
print(np.mean(resa/res2a), np.std(resa/res2a))
print(np.mean(resb/res2b), np.std(resb/res2b))
print(np.mean(resc/res2c), np.std(resc/res2c))
print(np.mean(resd/res2d), np.std(resd/res2d))
print(np.mean(rese/res2e), np.std(rese/res2e))

```

```

0.989628851618506 0.0015739650916350154
1.0029206289581216 0.0010239215279917362
1.0031256214295916 0.0007397789335735901
0.9971578858108526 0.005512895929255806
0.9960431130363838 0.008118436244891825

```