

BPN Trace

April 22, 2020

1 For getting the trace of BPN

```
[70]: import pandas as pd
import numpy as np
data=pd.read_csv('golf.csv')
```

```
[71]: data
```

```
[71]:
```

	Temperature	Weather	Play
0	hot	sunny	no
1	hot	overcast	yes
2	mild	rainy	yes
3	cool	rainy	yes
4	cool	rainy	no
5	cool	overcast	yes
6	mild	sunny	no
7	cool	sunny	yes
8	mild	rainy	yes
9	mild	sunny	yes
10	mild	overcast	yes
11	hot	overcast	yes
12	mild	rainy	no

```
[72]: encoded_data=data.copy()
encoded_data.head()
for col in encoded_data.head():
    encoded_data[col]=encoded_data[col].astype('category')
    encoded_data[col] = encoded_data[col].cat.codes
encoded_data
```

```
[72]:
```

	Temperature	Weather	Play
0	1	2	0
1	1	0	1
2	2	1	1
3	0	1	1
4	0	1	0
5	0	0	1

6	2	2	0
7	0	2	1
8	2	1	1
9	2	2	1
10	2	0	1
11	1	0	1
12	2	1	0

```
[73]: from sklearn import preprocessing

x = encoded_data.values #returns a numpy array
min_max_scaler = preprocessing.MinMaxScaler()
x_scaled = min_max_scaler.fit_transform(x)
data = pd.DataFrame(x_scaled)
data
```

```
[73]:      0    1    2
0  0.5  1.0  0.0
1  0.5  0.0  1.0
2  1.0  0.5  1.0
3  0.0  0.5  1.0
4  0.0  0.5  0.0
5  0.0  0.0  1.0
6  1.0  1.0  0.0
7  0.0  1.0  1.0
8  1.0  0.5  1.0
9  1.0  1.0  1.0
10 1.0  0.0  1.0
11 0.5  0.0  1.0
12 1.0  0.5  0.0
```

```
[74]: feature_set=data.iloc[:,0:2].to_numpy()
feature_set
```

```
[74]: array([[0.5, 1. ],
            [0.5, 0. ],
            [1. , 0.5],
            [0. , 0.5],
            [0. , 0.5],
            [0. , 0. ],
            [1. , 1. ],
            [0. , 1. ],
            [1. , 0.5],
            [1. , 1. ],
            [1. , 0. ],
            [0.5, 0. ],
            [1. , 0.5]])
```

```
[76]: labels=data.iloc[:,2].to_numpy()
labels = labels.reshape(13, 1)
labels
```

```
[76]: array([[0.],
          [1.],
          [1.],
          [1.],
          [0.],
          [1.],
          [0.],
          [1.],
          [1.],
          [1.],
          [1.],
          [1.],
          [0.]])
```

```
[53]: def sigmoid(x):
        return 1/(1+np.exp(-x))

def sigmoid_der(x):
    return sigmoid(x) *(1-sigmoid (x))

wh = np.random.rand(len(feature_set[0]),4)
wo = np.random.rand(4, 1)
lr = 0.5
```

```
[54]: wh, wo, lr
```

```
[54]: (array([[0.89080112, 0.22237559, 0.66009567, 0.14413771],
          [0.44096171, 0.19340467, 0.873244 , 0.43116672]]),
      array([[0.36576142],
          [0.25177497],
          [0.66014917],
          [0.95274659]]),
      0.5)
```

```
[77]: zh = np.dot(feature_set, wh)
zh
```

```
[77]: array([[0.88636227, 0.30459246, 1.20329183, 0.50323558],
          [0.44540056, 0.11118779, 0.33004784, 0.07206886],
          [1.11128198, 0.31907792, 1.09671767, 0.35972107],
          [0.22048085, 0.09670233, 0.436622 , 0.21558336],
          [0.22048085, 0.09670233, 0.436622 , 0.21558336],
          [0. , 0. , 0. , 0. ]],
```

```
[1.33176283, 0.41578026, 1.53333967, 0.57530443],
[0.44096171, 0.19340467, 0.873244 , 0.43116672],
[1.11128198, 0.31907792, 1.09671767, 0.35972107],
[1.33176283, 0.41578026, 1.53333967, 0.57530443],
[0.89080112, 0.22237559, 0.66009567, 0.14413771],
[0.44540056, 0.11118779, 0.33004784, 0.07206886],
[1.11128198, 0.31907792, 1.09671767, 0.35972107]])
```

```
[78]: ah = sigmoid(zh)
      ah
```

```
[78]: array([[0.7081389 , 0.5755648 , 0.76910986, 0.6232194 ],
            [0.60954512, 0.52776835, 0.58177102, 0.51800942],
            [0.75236803, 0.57909952, 0.74964459, 0.58897291],
            [0.554898 , 0.52415676, 0.60745383, 0.55368807],
            [0.554898 , 0.52415676, 0.60745383, 0.55368807],
            [0.5 , 0.5 , 0.5 , 0.5 ],
            [0.79113208, 0.60247306, 0.82249442, 0.63998624],
            [0.60848816, 0.54820101, 0.70542026, 0.60615224],
            [0.75236803, 0.57909952, 0.74964459, 0.58897291],
            [0.79113208, 0.60247306, 0.82249442, 0.63998624],
            [0.70905547, 0.55536593, 0.65928188, 0.53597217],
            [0.60954512, 0.52776835, 0.58177102, 0.51800942],
            [0.75236803, 0.57909952, 0.74964459, 0.58897291]])
```

```
[79]: zo = np.dot(ah, wo)
      zo
```

```
[79]: array([[1.50542011],
            [1.23341432],
            [1.47700916],
            [1.2634644 ],
            [1.2634644 ],
            [1.11521608],
            [1.59376696],
            [1.40377687],
            [1.47700916],
            [1.59376696],
            [1.34504242],
            [1.23341432],
            [1.47700916]])
```

```
[80]: ao = sigmoid(zo)
      ao
```

```
[80]: array([[0.81838147],
            [0.7744156 ]],
```

```

[0.81412041],
[0.77962191],
[0.77962191],
[0.75310027],
[0.83114543],
[0.80278254],
[0.81412041],
[0.83114543],
[0.79331794],
[0.7744156 ],
[0.81412041]])

```

```

[81]: error_out = ((1 / 2) * (np.power((ao - labels), 2)))
      error_out

```

```

[81]: array([[0.33487412],
            [0.02544416],
            [0.01727561],
            [0.02428325],
            [0.30390516],
            [0.03047974],
            [0.34540136],
            [0.01944736],
            [0.01727561],
            [0.01425593],
            [0.02135874],
            [0.02544416],
            [0.33139602]])

```

```

[82]: error_out.sum()

```

```

[82]: 1.5108412258201425

```

```

[83]: dcost_dao = ao - labels
      dcost_dao

```

```

[83]: array([[ 0.81838147],
            [-0.2255844 ],
            [-0.18587959],
            [-0.22037809],
            [ 0.77962191],
            [-0.24689973],
            [ 0.83114543],
            [-0.19721746],
            [-0.18587959],
            [-0.16885457],
            [-0.20668206],

```

```
[-0.2255844 ],
[ 0.81412041]])
```

```
[84]: dao_dzo = sigmoid_der(zo)
dao_dzo
```

```
[84]: array([[0.14863324],
[0.17469608],
[0.15132837],
[0.17181159],
[0.17181159],
[0.18594025],
[0.14034271],
[0.15832274],
[0.15132837],
[0.14034271],
[0.16396458],
[0.17469608],
[0.15132837]])
```

```
[85]: dzo_dwo = ah
dzo_dwo
```

```
[85]: array([[0.7081389 , 0.5755648 , 0.76910986, 0.6232194 ],
[0.60954512, 0.52776835, 0.58177102, 0.51800942],
[0.75236803, 0.57909952, 0.74964459, 0.58897291],
[0.554898 , 0.52415676, 0.60745383, 0.55368807],
[0.554898 , 0.52415676, 0.60745383, 0.55368807],
[0.5 , 0.5 , 0.5 , 0.5 ],
[0.79113208, 0.60247306, 0.82249442, 0.63998624],
[0.60848816, 0.54820101, 0.70542026, 0.60615224],
[0.75236803, 0.57909952, 0.74964459, 0.58897291],
[0.79113208, 0.60247306, 0.82249442, 0.63998624],
[0.70905547, 0.55536593, 0.65928188, 0.53597217],
[0.60954512, 0.52776835, 0.58177102, 0.51800942],
[0.75236803, 0.57909952, 0.74964459, 0.58897291]])
```

```
[86]: dcost_wo = np.dot(dzo_dwo.T, dcost_dao * dao_dzo)
dcost_wo
```

```
[86]: array([[0.14932763],
[0.11464956],
[0.16537583],
[0.12704834]])
```

```
[87]: dcost_dzo = dcost_dao * dao_dzo
dzo_dah = wo
```

```
dcost_dah = np.dot(dcost_dzo , dzo_dah.T)
dcost_dah
```

```
[87]: array([[ 0.04449074,  0.03062558,  0.08029968,  0.11589085],
             [-0.01441419, -0.00992213, -0.02601563, -0.03754651],
             [-0.01028845, -0.00708214, -0.01856924, -0.02679967],
             [-0.01384901, -0.00953308, -0.02499556, -0.03607433],
             [ 0.04899304,  0.03372477,  0.08842571,  0.12761858],
             [-0.01679159, -0.01155864, -0.03030652, -0.04373926],
             [ 0.04266431,  0.02936834,  0.07700323,  0.11113332],
             [-0.01142054, -0.00786142, -0.0206125 , -0.02974857],
             [-0.01028845, -0.00708214, -0.01856924, -0.02679967],
             [-0.00866763, -0.00596644, -0.01564389, -0.02257772],
             [-0.01239512, -0.00853229, -0.02237149, -0.03228719],
             [-0.01441419, -0.00992213, -0.02601563, -0.03754651],
             [ 0.04506163,  0.03101855,  0.08133006,  0.11737792]])
```

```
[88]: dah_dzh = sigmoid_der(zh)
      dzh_dwh = feature_set
      dcost_wh = np.dot(dzh_dwh.T, dah_dzh * dcost_dah)
      dcost_wh
```

```
[88]: array([[0.00878944, 0.00887382, 0.01302663, 0.03204541],
             [0.01871316, 0.01621013, 0.03064404, 0.05954574]])
```

```
[89]: wh -= lr * dcost_wh
      wh
```

```
[89]: array([[0.8864064 , 0.21793868, 0.65358236, 0.128115  ],
             [0.43160513, 0.1852996 , 0.85792198, 0.40139385]])
```

```
[90]: wo -= lr * dcost_wo
      wo
```

```
[90]: array([[0.29109761],
             [0.19445019],
             [0.57746126],
             [0.88922242]])
```

```
[ ]:
```