Computer Assignment 1

CPE 261456 (Introduction to Computational Intelligence)

โดย

นายพีรณัฐ ธารทะเลทอง

รหัสนักศึกษา 550610530

เสนอ

ผศ.ดร. ศันสนีย์ เอื้อพันธ์วิริยะกุล

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่

วิธีการทำงานของโปรแกรม

โปรแกรมเขียนด้วยภาษา python โดยรับ parameter ผ่าน command line รายละเอียดดังนี้

Parameter -N arg : arg คือ ชื่อโครงสร้าง Neural network เช่น “2-4-1”

parameter -n arg : arg คือ learning rate

parameter -m arg : arg คือ momentum

parameter -e arg : arg คือ จำนวน epoch

parameter -c : ทำ 10% cross validation

parameter -t arg : arg คือ training set file

ตัวอย่างการเรียกใช้โปรแกรม

python ComputerAssignment1.py -N 4-4-1 -n 0.2 -m 0.3 -e 100 -t iris.pat

เริ่มต้นการทำงาน โปรแกรมจะอ่าน input และ desire-output จากไฟล์มาเก็บใน array จากนั้นนำมา shuffle ก่อนนำเข้า neural network เมื่อเริ่มเข้า neural network จะทำการ random init weight ซึ่งค่าอยู่ระหว่าง $\frac{-1}{\sqrt{fanin}}$ และ $\frac{1}{\sqrt{fanin}}$ จากนั้นเข้าสู่กระบวนการ train โดยเริ่มจากการทำ feedforward networks โดย output แต่ละ layer ได้จากการ dot product ระหว่าง matrix output ของ layer ก่อนหน้า กับ matrix ของ weight ทุก weight ที่เข้า layer นั้น

จากนั้น ทำ back propagation โดยเริ่มจาก หา error จากสูตร

$$e_j(t) = d_j(t) - y_j(t)$$

เพื่อนำค่า error ไปหาค่า gradients ที่ output layer จากสูตร

$$\delta_j(t) = e_j(t)\varphi_j'(v_j(t))$$

จากนั้น หา gradients ใน hidden layer จากสูตร

$$\delta_{\mathrm{j}}^{(\mathrm{l})}(\mathrm{t}) = \varphi_{\mathrm{j}}^{(\mathrm{l})'}\left(v_{\mathrm{j}}^{(\mathrm{l})}(\mathrm{t})\right)\sum_k \delta_k^{(l+1)}(t) w_{\mathrm{kj}}^{(\mathrm{l}+1)}(\mathrm{t})$$

เพื่อนำ gradients ที่ได้ของทุก node นำไปปรับ weight จากสูตร

$$\Delta w_{\mathrm{ji}}^{(\mathrm{l})}(\mathrm{t}) = \alpha \Delta w_{\mathrm{ji}}^{(\mathrm{l})}(\mathrm{t}-1) + \eta \delta_j^{(l)}(t) y_{\mathrm{i}}^{(\mathrm{l}-1)}(\mathrm{t})$$

จากนั้นกลับไปเริ่มต้นใหม่ให้ครบรอบจำนวน epoch ที่รับเข้ามา เพื่อปรับ weight ให้ดีขึ้นเรื่อยๆ

ทดลองกับปัญหา XOR เพื่อทดสอบความถูกต้องของโปรแกรม

ผลการทดลองครั้งที่ 1 (50 epochs)

```
C:\Users>python ComputerAssignment1.py -N 2-4-1 -n 0.2 -m 0.3 -e 50

------------Variable------------

Neural name     2-4-1

Activation func tanh

Learning rate   0.2

Momentum        0.3

Epoch           50

TrainingFile    -


------------Training------------

==================================100%


------------Testing-------------

Features            Output              Desired class

0 0                 [ 0.2285700218]       0

0 1                 [ 0.6795426040]       1

1 0                 [ 0.5946130258]       1

1 1                 [ 0.3343745376]       0


Error AV 0.0538852660

Accuracy 100.0000%
```

ผลการทดลองครั้งที่ 2 (200 epochs)

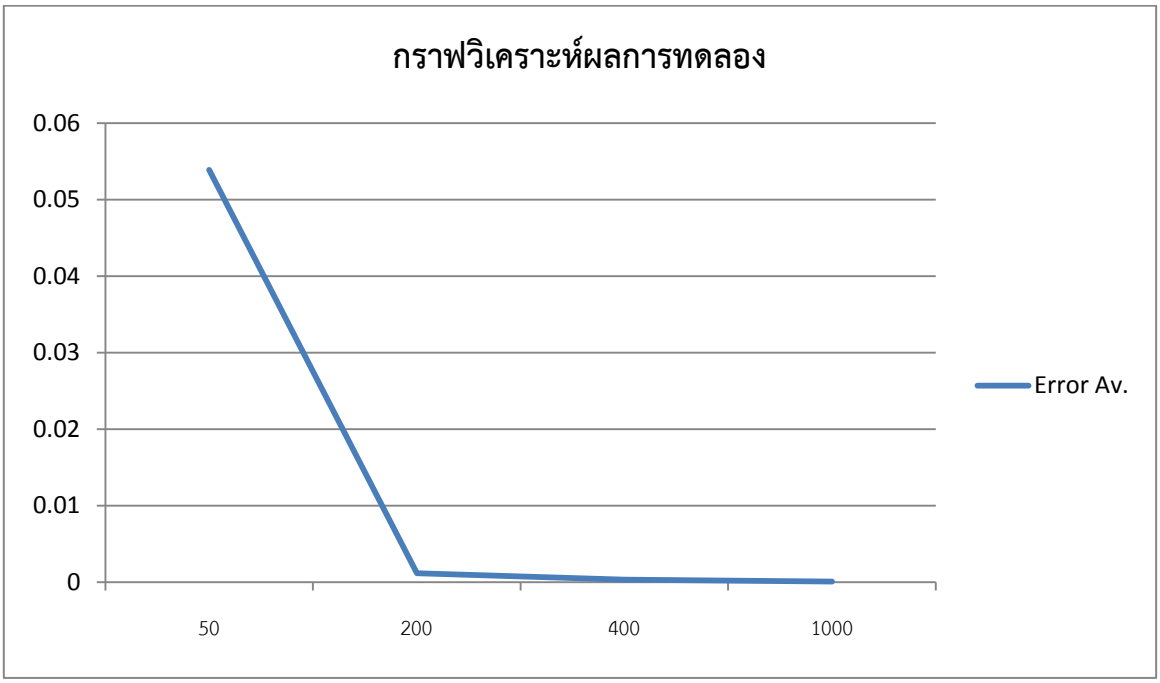
```
C:\Users>python ComputerAssignment1.py -N 2-4-1 -n 0.2 -m 0.3 -e 200

------------Variable------------

Neural name     2-4-1

Activation func tanh

Learning rate   0.2

Momentum        0.3

Epoch           200

TrainingFile    -


------------Training------------

============================100%


------------Testing-------------

Features            Output              Desired class

0 0                 [ 0.0065803017]       0

0 1                 [ 0.9336247061]       1

1 0                 [ 0.9295828521]       1

1 1                 [-0.0011971559]       0


Error AV 0.0011761235

Accuracy 100.0000%
```

ผลการทดลองครั้งที่ 3 (400 epochs)

```
C:\Users>python ComputerAssignment1.py -N 2-4-1 -n 0.2 -m 0.3 -e 400

------------Variable------------

Neural name     2-4-1

Activation func tanh

Learning rate   0.2

Momentum        0.3

Epoch           400

TrainingFile    -


------------Training------------

============================100%


------------Testing-------------

Features            Output              Desired class

0 0                 [ 0.0063966836]       0

0 1                 [ 0.9681230407]       1

1 0                 [ 0.9625096198]       1

1 1                 [ 0.0105083843]       0


Error AV 0.0003216266

Accuracy 100.0000%
```

ผลการทดลองครั้งที่ 4 (1000 epochs)

```
C:\Users>python ComputerAssignment1.py -N 2-4-1 -n 0.2 -m 0.3 -e 1000

------------Variable------------

Neural name     2-4-1

Activation func tanh

Learning rate   0.2

Momentum        0.3

Epoch           1000

TrainingFile    -


------------Training------------

============================100%


------------Testing-------------

Features            Output              Desired class

0 0                 [ 0.0005324577]       0

0 1                 [ 0.9834712507]       1

1 0                 [ 0.9806382148]       1

1 1                 [-0.0105726275]       0


Error AV 0.0000950178

Accuracy 100.0000%
```



จะเห็นได้ว่า เมื่อจำนวน epoch เพิ่มขึ้นจะทำให้ Error เฉลี่ยลดลงไปด้วย จึงสรุปได้ว่า โปรแกรมสามารถแก้ปัญหา XOR ได้ถูกต้อง

การทดลองโดยใช้ 10% cross validation กับ Training set iris.pat

Neural network 4-4-1

```
------------Variable------------
Neural name     4-4-1
Activation func tanh
Learning rate   0.2
Momentum        0.3
Epoch           100
TrainingFile    iris.pat


------------- Cross validation block 1 -------------

------------Testing-------------

Features                Output                  Desired class
6.1 3.0 4.6 1.4         [ 2.3757040318]         2
5.0 3.4 1.5 0.2         [ 1.0978691977]         1
5.0 2.0 3.5 1.0         [ 2.2322137614]         2
6.6 3.0 4.4 1.4         [ 2.2134703599]         2
4.4 2.9 1.4 0.2         [ 1.1035642629]         1
4.4 3.2 1.3 0.2         [ 1.0833085374]         1
4.4 3.0 1.3 0.2         [ 1.0831498495]         1
6.7 3.1 4.4 1.4         [ 2.2030891490]         2
5.7 4.4 1.5 0.4         [ 1.0879524515]         1
6.9 3.1 5.1 2.3         [ 2.9601132661]         3
6.7 3.0 5.0 1.7         [ 2.7385170150]         2
7.7 3.0 6.1 2.3         [ 2.9715773513]         3
7.2 3.2 6.0 1.8         [ 2.9682183977]         3
5.1 3.4 1.5 0.2         [ 1.0954680542]         1
4.7 3.2 1.6 0.2         [ 1.1278493760]         1

Error AV 0.0074838432
Accuracy 100.0000%
-----------------------------------------------------
------------- Cross validation block 2 -------------

------------Testing-------------

Features                Output                  Desired class
6.0 2.9 4.5 1.5         [ 2.6504565969]         2
4.6 3.2 1.4 0.2         [ 1.0601635825]         1
6.7 3.3 5.7 2.1         [ 2.9527709090]         3
6.7 2.5 5.8 1.8         [ 2.9521485913]         3
6.9 3.2 5.7 2.3         [ 2.9530486546]         3
6.8 3.2 5.9 2.3         [ 2.9533304609]         3
5.4 3.9 1.7 0.4         [ 1.0742566320]         1
4.9 2.4 3.3 1.0         [ 2.0825258374]         2
5.5 2.4 3.8 1.1         [ 2.1690375257]         2
5.5 3.5 1.3 0.2         [ 1.0456407341]         1
6.4 3.2 5.3 2.3         [ 2.9527261231]         3
6.3 2.7 4.9 1.8         [ 2.9455780968]         3
7.6 3.0 6.6 2.1         [ 2.9530336939]         3
6.3 2.9 5.6 1.8         [ 2.9523228997]         3
6.0 2.2 4.0 1.0         [ 2.1530452538]         2

Error AV 0.0042641328
Accuracy 100.0000%
-----------------------------------------------------
```

```
------------ Cross validation block 3 ------------

------------Testing-------------

Features                Output                  Desired class
5.1 2.5 3.0 1.1         [ 2.0177864699]         2
6.2 2.9 4.3 1.3         [ 2.2673390104]         2
6.4 2.7 5.3 1.9         [ 2.9813750847]         3
6.0 2.7 5.1 1.6         [ 2.9747561858]         2
5.0 3.5 1.3 0.3         [ 1.1080817017]         1
5.2 3.4 1.4 0.2         [ 1.1158191764]         1
5.5 2.3 4.0 1.3         [ 2.3984056432]         2
6.1 2.6 5.6 1.4         [ 2.9819187336]         3
6.3 3.4 5.6 2.4         [ 2.9833773647]         3
6.1 2.8 4.0 1.3         [ 2.2300949109]         2
5.6 2.5 3.9 1.1         [ 2.2473315307]         2
6.5 3.0 5.8 2.2         [ 2.9834218456]         3
5.8 2.7 4.1 1.0         [ 2.2525904285]         2
5.6 2.7 4.2 1.3         [ 2.3374689231]         2
5.1 3.8 1.9 0.4         [ 1.1808428580]         1

Error AV 0.0127624315
Accuracy 93.3333%
-------------------------------------------------
------------ Cross validation block 4 ------------

------------Testing-------------

Features                Output                  Desired class
4.6 3.6 1.0 0.2         [ 1.0475393472]         1
5.7 2.8 4.5 1.3         [ 2.6177234255]         2
4.9 3.6 1.4 0.1         [ 1.0755048974]         1
5.4 3.9 1.3 0.4         [ 1.0696694926]         1
5.7 2.9 4.2 1.3         [ 2.2546247140]         2
6.7 3.3 5.7 2.5         [ 2.9682831981]         3
6.3 2.5 4.9 1.5         [ 2.9472378075]         2
5.7 2.5 5.0 2.0         [ 2.9676160053]         3
4.9 3.1 1.5 0.1         [ 1.0944456853]         1
5.9 3.0 4.2 1.5         [ 2.2516469833]         2
6.6 2.9 4.6 1.3         [ 2.2292285697]         2
5.2 3.5 1.5 0.2         [ 1.0841543821]         1
5.6 3.0 4.1 1.3         [ 2.2126121848]         2
5.5 2.6 4.4 1.2         [ 2.7149844898]         2
7.7 2.8 6.7 2.0         [ 2.9680254937]         3

Error AV 0.0170653996
Accuracy 93.3333%
-------------------------------------------------
```

```
------------ Cross validation block 5 ------------

------------Testing-------------

Features                Output                  Desired class
5.5 2.4 3.7 1.0         [ 2.1401229612]         2
5.0 3.2 1.2 0.2         [ 1.0433350674]         1
5.7 3.0 4.2 1.2         [ 2.2017482757]         2
5.1 3.5 1.4 0.2         [ 1.0637954926]         1
5.9 3.2 4.8 1.8         [ 2.9404390028]         2
6.3 2.3 4.4 1.3         [ 2.4708308462]         2
5.1 3.8 1.5 0.3         [ 1.0717144497]         1
4.3 3.0 1.1 0.1         [ 1.0206205356]         1
4.8 3.4 1.9 0.2         [ 1.1610622309]         1
5.4 3.4 1.5 0.4         [ 1.0767653590]         1
7.7 2.6 6.9 2.3         [ 2.9614559287]         3
6.7 3.1 4.7 1.5         [ 2.2536303513]         2
6.4 3.2 4.5 1.5         [ 2.2211113965]         2
6.9 3.1 4.9 1.5         [ 2.2994461901]         2
5.7 2.8 4.1 1.3         [ 2.2448315480]         2

Error AV 0.0122842391
Accuracy 93.3333%
-------------------------------------------------------
 ------------ Cross validation block 6 ------------

 ------------Testing-------------

 Features                Output                  Desired class
 4.9 3.1 1.5 0.2         [ 1.0625933630]         1
 6.2 2.2 4.5 1.5         [ 2.9578633326]         2
 6.5 3.0 5.5 1.8         [ 2.9664154573]         3
 6.4 2.8 5.6 2.1         [ 2.9687775022]         3
 5.7 3.8 1.7 0.3         [ 1.0557321740]         1
 4.9 2.5 4.5 1.7         [ 2.9669124900]         3
 4.6 3.4 1.4 0.3         [ 1.0441002071]         1
 6.8 3.0 5.5 2.1         [ 2.9680562268]         3
 7.0 3.2 4.7 1.4         [ 2.1302321046]         2
 6.3 2.5 5.0 1.9         [ 2.9679270469]         3
 4.5 2.3 1.3 0.3         [ 1.0762577274]         1
 5.9 3.0 5.1 1.8         [ 2.9649255037]         3
 5.2 2.7 3.9 1.4         [ 2.4178778880]         2
 6.4 3.1 5.5 1.8         [ 2.9657669662]         3
 6.4 2.8 5.6 2.2         [ 2.9689968201]         3

 Error AV 0.0094373210
 Accuracy 93.3333%
 -------------------------------------------------------
```

```
------------ Cross validation block 7 ------------

------------Testing-------------

Features                Output                  Desired class
5.0 2.3 3.3 1.0         [ 2.0871297927]         2
5.8 2.6 4.0 1.2         [ 2.1809803440]         2
4.8 3.0 1.4 0.3         [ 1.0576390904]         1
5.0 3.0 1.6 0.2         [ 1.0908799450]         1
6.1 3.0 4.9 1.8         [ 2.9498554367]         3
6.5 3.0 5.2 2.0         [ 2.9561812045]         3
6.2 3.4 5.4 2.3         [ 2.9581520895]         3
5.3 3.7 1.5 0.2         [ 1.0536961072]         1
6.7 3.1 5.6 2.4         [ 2.9580668141]         3
5.8 2.7 3.9 1.2         [ 2.1432598154]         2
5.7 2.6 3.5 1.0         [ 2.0889636191]         2
5.0 3.3 1.4 0.2         [ 1.0510072011]         1
6.1 2.9 4.7 1.4         [ 2.7496149380]         2
6.2 2.8 4.8 1.8         [ 2.9454732988]         3
6.0 2.2 5.0 1.5         [ 2.9561516602]         3

Error AV 0.0055051035
Accuracy 100.0000%
-----------------------------------------------------
 ------------ Cross validation block 8 ------------

 ------------Testing-------------

 Features                Output                  Desired class
6.5 3.2 5.1 2.0         [ 2.9436674388]         3
5.4 3.7 1.5 0.2         [ 1.0769938243]         1
5.0 3.4 1.6 0.4         [ 1.0982798281]         1
5.4 3.4 1.7 0.2         [ 1.0996054804]         1
6.7 3.0 5.2 2.3         [ 2.9598425196]         3
4.9 3.0 1.4 0.2         [ 1.0799969090]         1
5.8 4.0 1.2 0.2         [ 1.0650015922]         1
5.5 2.5 4.0 1.3         [ 2.3882956273]         2
5.8 2.7 5.1 1.9         [ 2.9609241038]         3
4.6 3.1 1.5 0.2         [ 1.0957528797]         1
6.1 2.8 4.7 1.2         [ 2.4875369027]         2
5.1 3.8 1.6 0.2         [ 1.0848584403]         1
7.1 3.0 5.9 2.1         [ 2.9613308239]         3
5.1 3.3 1.7 0.5         [ 1.1182398921]         1
7.7 3.8 6.7 2.2         [ 2.9615273161]         3

Error AV 0.0038686439
Accuracy 100.0000%
-----------------------------------------------------
```

```
------------- Cross validation block 9 -------------

------------Testing-------------

Features                Output                  Desired class
5.0 3.5 1.6 0.6         [ 1.1222996871]         1
5.2 4.1 1.5 0.1         [ 1.0923598016]         1
6.0 3.4 4.5 1.6         [ 2.3183330762]         2
4.7 3.2 1.3 0.2         [ 1.0840123290]         1
6.0 3.0 4.8 1.8         [ 2.9289747958]         3
6.5 2.8 4.6 1.5         [ 2.4043124368]         2
6.4 2.9 4.3 1.3         [ 2.2247549416]         2
5.6 2.8 4.9 2.0         [ 2.9572404560]         3
6.8 2.8 4.8 1.4         [ 2.3514253750]         2
5.0 3.6 1.4 0.2         [ 1.0907676366]         1
7.4 2.8 6.1 1.9         [ 2.9579773892]         3
6.3 2.8 5.1 1.5         [ 2.9223564123]         3
6.9 3.1 5.4 2.1         [ 2.9538359134]         3
5.1 3.7 1.5 0.4         [ 1.1019451589]         1
6.3 3.3 4.7 1.6         [ 2.3777880945]         2

Error AV 0.0053959885
Accuracy 100.0000%
-----------------------------------------------------
------------- Cross validation block 10 -------------

------------Testing-------------

Features                Output                  Desired class
7.2 3.6 6.1 2.5         [ 2.9861789839]         3
6.3 3.3 6.0 2.5         [ 2.9862146920]         3
7.2 3.0 5.8 1.6         [ 2.8528489351]         3
5.8 2.7 5.1 1.9         [ 2.9858967609]         3
4.8 3.4 1.6 0.2         [ 1.0074597716]         1
4.8 3.1 1.6 0.2         [ 1.0154420269]         1
5.6 2.9 3.6 1.3         [ 1.8421279600]         2
4.8 3.0 1.4 0.1         [ 0.9910592066]         1
7.3 2.9 6.3 1.8         [ 2.9854165220]         3
5.6 3.0 4.5 1.5         [ 2.6598108082]         2
5.5 4.2 1.4 0.2         [ 0.9939292482]         1
5.8 2.8 5.1 2.4         [ 2.9859469862]         3
5.4 3.0 4.5 1.5         [ 2.8776035137]         2
5.1 3.5 1.4 0.3         [ 0.9852441998]         1
7.9 3.8 6.4 2.0         [ 2.9324514644]         3

Error AV 0.0104858067
Accuracy 93.3333%
-----------------------------------------------------
Error Average  0.00885529097936
```

Neural network 4-3-2-1

```
------------Variable------------
Neural name     4-3-2-1
Activation func tanh
Learning rate   0.2
Momentum        0.3
Epoch           100
TrainingFile    iris.pat


------------- Cross validation block 1 -------------

------------Testing-------------

Features                Output                  Desired class
6.1 3.0 4.6 1.4         [ 1.9614685609]         2
5.5 2.3 4.0 1.3         [ 1.9606791996]         2
5.6 2.5 3.9 1.1         [ 1.9569616866]         2
6.2 2.2 4.5 1.5         [ 2.4883924841]         2
4.9 2.4 3.3 1.0         [ 1.9125094848]         2
6.7 3.0 5.2 2.3         [ 2.9354492065]         3
5.7 2.8 4.1 1.3         [ 1.9594926923]         2
5.7 2.8 4.5 1.3         [ 1.9584046759]         2
6.7 3.1 4.7 1.5         [ 1.9639408863]         2
5.0 3.2 1.2 0.2         [ 0.9864444196]         1
4.3 3.0 1.1 0.1         [ 0.9610704441]         1
7.7 3.0 6.1 2.3         [ 2.9442740038]         3
5.8 2.7 5.1 1.9         [ 2.9425204951]         3
5.5 4.2 1.4 0.2         [ 1.0051545992]         1
6.9 3.2 5.7 2.3         [ 2.9418689245]         3

Error AV 0.0022618110
Accuracy 100.0000%
-----------------------------------------------------
 ------------- Cross validation block 2 -------------

 ------------Testing-------------

 Features                Output                  Desired class
 5.0 3.5 1.3 0.3         [ 1.0016545280]         1
 7.2 3.2 6.0 1.8         [ 2.6852028137]         3
 5.1 3.5 1.4 0.2         [ 1.0036949630]         1
 6.3 2.7 4.9 1.8         [ 2.5759799523]         3
 4.4 3.0 1.3 0.2         [ 1.0076603973]         1
 6.4 2.7 5.3 1.9         [ 2.9099598010]         3
 6.7 3.1 5.6 2.4         [ 2.9231468588]         3
 5.4 3.0 4.5 1.5         [ 2.0404634887]         2
 5.0 3.4 1.6 0.4         [ 1.0329375152]         1
 7.4 2.8 6.1 1.9         [ 2.9170010088]         3
 4.6 3.4 1.4 0.3         [ 1.0087668244]         1
 6.5 3.0 5.5 1.8         [ 2.8251497298]         3
 5.8 2.7 4.1 1.0         [ 1.9349096330]         2
 5.5 2.4 3.7 1.0         [ 1.9143320981]         2
 4.4 3.2 1.3 0.2         [ 0.9981073277]         1

 Error AV 0.0028734899
 Accuracy 86.6667%
 -----------------------------------------------------
```

```
------------- Cross validation block 3 -------------

------------Testing-------------

Features                Output                  Desired class
4.9 2.5 4.5 1.7         [ 2.9276618873]         3
5.7 3.8 1.7 0.3         [ 1.0215297183]         1
6.4 2.9 4.3 1.3         [ 1.9703662407]         2
5.1 3.8 1.6 0.2         [ 1.0043394673]         1
6.7 2.5 5.8 1.8         [ 2.9317628135]         3
4.6 3.6 1.0 0.2         [ 0.9524685245]         1
5.4 3.4 1.7 0.2         [ 1.0299072985]         1
7.2 3.0 5.8 1.6         [ 2.8274341823]         3
6.1 2.8 4.0 1.3         [ 1.9453146248]         2
5.0 2.0 3.5 1.0         [ 1.9694401720]         2
5.7 2.9 4.2 1.3         [ 1.9678126131]         2
6.3 2.8 5.1 1.5         [ 2.6867824549]         3
4.7 3.2 1.6 0.2         [ 1.0268916143]         1
5.2 2.7 3.9 1.4         [ 1.9637396515]         2
4.6 3.2 1.4 0.2         [ 1.0046690604]         1

Error AV 0.0012442309
Accuracy 93.3333%
-----------------------------------------------------
 ------------- Cross validation block 4 -------------

 ------------Testing-------------

 Features                Output                  Desired class
 4.9 3.6 1.4 0.1         [ 0.9847004530]         1
 5.2 4.1 1.5 0.1         [ 0.9870422765]         1
 4.8 3.4 1.9 0.2         [ 1.0241356258]         1
 5.6 2.7 4.2 1.3         [ 1.9752253324]         2
 6.3 3.3 4.7 1.6         [ 1.9790013475]         2
 6.6 3.0 4.4 1.4         [ 1.9839806905]         2
 5.2 3.4 1.4 0.2         [ 1.0015467475]         1
 6.1 2.6 5.6 1.4         [ 2.9318530197]         3
 5.6 2.8 4.9 2.0         [ 2.9295624182]         3
 5.1 2.5 3.0 1.1         [ 1.7574671913]         2
 6.5 3.0 5.2 2.0         [ 2.9254672068]         3
 5.0 2.3 3.3 1.0         [ 1.9008203599]         2
 4.9 3.1 1.5 0.1         [ 1.0076704824]         1
 6.3 2.9 5.6 1.8         [ 2.9316197438]         3
 6.8 3.2 5.9 2.3         [ 2.9327968438]         3

 Error AV 0.0007947381
 Accuracy 93.3333%
 -----------------------------------------------------
```

```
------------- Cross validation block 5 -------------

------------Testing-------------

Features                Output                  Desired class
5.9 3.0 5.1 1.8         [ 2.9405699145]         3
5.9 3.2 4.8 1.8         [ 2.7387833469]         2
6.1 2.9 4.7 1.4         [ 2.0053139708]         2
6.6 2.9 4.6 1.3         [ 2.0220426043]         2
6.1 2.8 4.7 1.2         [ 2.0029295875]         2
6.4 2.8 5.6 2.2         [ 2.9445547985]         3
7.2 3.6 6.1 2.5         [ 2.9440440983]         3
6.0 3.0 4.8 1.8         [ 2.8974029245]         3
7.7 2.8 6.7 2.0         [ 2.9455227148]         3
6.0 3.4 4.5 1.6         [ 1.9771821062]         2
6.1 3.0 4.9 1.8         [ 2.9153510367]         3
5.8 2.7 3.9 1.2         [ 1.9696833854]         2
6.0 2.7 5.1 1.6         [ 2.9415350419]         2
5.5 2.5 4.0 1.3         [ 1.9802630002]         2
6.9 3.1 5.1 2.3         [ 2.9397051267]         3

Error AV 0.0122389424
Accuracy 93.3333%
-----------------------------------------------------
 ------------- Cross validation block 6 -------------

 ------------Testing-------------

 Features                Output                  Desired class
 7.0 3.2 4.7 1.4         [ 1.8692169776]         2
 7.3 2.9 6.3 1.8         [ 2.9464870606]         3
 4.9 3.0 1.4 0.2         [ 0.9896510802]         1
 6.2 2.8 4.8 1.8         [ 2.8051101975]         3
 6.7 3.0 5.0 1.7         [ 2.3447825036]         2
 5.8 2.8 5.1 2.4         [ 2.9455150591]         3
 5.8 2.7 5.1 1.9         [ 2.9431698762]         3
 5.6 3.0 4.1 1.3         [ 1.8341839024]         2
 6.3 2.3 4.4 1.3         [ 2.2062194041]         2
 5.7 3.0 4.2 1.2         [ 1.8283245015]         2
 6.0 2.9 4.5 1.5         [ 2.0751356403]         2
 4.8 3.0 1.4 0.3         [ 0.9924542460]         1
 5.4 3.4 1.5 0.4         [ 1.0006574411]         1
 5.1 3.3 1.7 0.5         [ 1.0111572656]         1
 6.8 2.8 4.8 1.4         [ 2.0873663090]         2

 Error AV 0.0024673684
 Accuracy 100.0000%
 -----------------------------------------------------
```

```
------------ Cross validation block 7 -------------

------------Testing-------------

Features                Output                  Desired class
6.9 3.1 5.4 2.1         [ 2.9289382266]         3
5.1 3.8 1.9 0.4         [ 1.0181003452]         1
6.7 3.1 4.4 1.4         [ 1.9072936803]         2
5.5 2.6 4.4 1.2         [ 2.0717847377]         2
5.5 2.4 3.8 1.1         [ 1.9560654356]         2
5.0 3.6 1.4 0.2         [ 0.9902167208]         1
5.7 2.6 3.5 1.0         [ 1.7599147236]         2
6.3 3.3 6.0 2.5         [ 2.9555791903]         3
5.7 2.5 5.0 2.0         [ 2.9549568487]         3
6.3 2.5 4.9 1.5         [ 2.7982025278]         2
6.4 3.2 4.5 1.5         [ 1.9574508629]         2
5.1 3.8 1.5 0.3         [ 0.9949454278]         1
6.5 3.0 5.8 2.2         [ 2.9554437871]         3
5.6 2.9 3.6 1.3         [ 1.7930357970]         2
4.6 3.1 1.5 0.2         [ 1.0093152009]         1

Error AV 0.0063888614
Accuracy 86.6667%
-----------------------------------------------------
------------ Cross validation block 8 -------------

------------Testing-------------

Features                Output                  Desired class
6.4 3.1 5.5 1.8         [ 2.9245246934]         3
7.7 2.6 6.9 2.3         [ 2.9354393146]         3
5.1 3.4 1.5 0.2         [ 1.0165110333]         1
6.7 3.3 5.7 2.1         [ 2.9290793519]         3
5.4 3.7 1.5 0.2         [ 1.0064845664]         1
5.7 4.4 1.5 0.4         [ 0.9971581746]         1
5.0 3.5 1.6 0.6         [ 1.0336823349]         1
4.5 2.3 1.3 0.3         [ 1.1274858777]         1
5.3 3.7 1.5 0.2         [ 1.0055856133]         1
4.8 3.4 1.6 0.2         [ 1.0197852677]         1
6.5 3.2 5.1 2.0         [ 2.8592667830]         3
5.4 3.9 1.3 0.4         [ 0.9988845747]         1
7.1 3.0 5.9 2.1         [ 2.9327646357]         3
4.8 3.1 1.6 0.2         [ 1.0428476199]         1
6.0 2.2 4.0 1.0         [ 2.0237717111]         2

Error AV 0.0004979631
Accuracy 100.0000%
-----------------------------------------------------
```

```
------------- Cross validation block 9 -------------

------------Testing-------------

Features                Output                  Desired class
6.4 3.2 5.3 2.3         [ 2.9284811592]         3
6.2 2.9 4.3 1.3         [ 1.9819189414]         2
5.4 3.9 1.7 0.4         [ 1.0176929370]         1
6.0 2.2 5.0 1.5         [ 2.9286848578]         3
5.5 3.5 1.3 0.2         [ 1.0036558855]         1
6.9 3.1 4.9 1.5         [ 2.0143310896]         2
5.1 3.7 1.5 0.4         [ 1.0115032603]         1
5.2 3.5 1.5 0.2         [ 1.0100963841]         1
6.8 3.0 5.5 2.1         [ 2.9287072060]         3
7.7 3.8 6.7 2.2         [ 2.9298563116]         3
4.7 3.2 1.3 0.2         [ 1.0012512004]         1
4.9 3.1 1.5 0.2         [ 1.0257858747]         1
5.9 3.0 4.2 1.5         [ 1.9691121197]         2
6.3 2.5 5.0 1.9         [ 2.9286167404]         3
4.8 3.0 1.4 0.1         [ 1.0145104237]         1

Error AV 0.0002351936
Accuracy 100.0000%
-----------------------------------------------------
------------- Cross validation block 10 -------------

------------Testing-------------

Features                Output                  Desired class
5.8 4.0 1.2 0.2         [ 0.9724382066]         1
6.5 2.8 4.6 1.5         [ 2.0455720092]         2
5.1 3.5 1.4 0.3         [ 0.9898075885]         1
5.8 2.6 4.0 1.2         [ 1.9913174503]         2
5.0 3.0 1.6 0.2         [ 1.0302216132]         1
5.0 3.3 1.4 0.2         [ 0.9920640066]         1
6.2 3.4 5.4 2.3         [ 2.9224584050]         3
7.6 3.0 6.6 2.1         [ 2.9247487419]         3
6.7 3.3 5.7 2.5         [ 2.9239018882]         3
6.4 2.8 5.6 2.1         [ 2.9240685705]         3
5.6 3.0 4.5 1.5         [ 2.3011959109]         2
7.9 3.8 6.4 2.0         [ 2.8926552203]         3
5.0 3.4 1.5 0.2         [ 0.9950594579]         1
6.3 3.4 5.6 2.4         [ 2.9235440115]         3
4.4 2.9 1.4 0.2         [ 1.0167205486]         1

Error AV 0.0011301283
Accuracy 100.0000%
-----------------------------------------------------
Error Average  0.0030132727038
```

จะเห็นได้ว่า เมื่อเปลี่ยนจำนวน hidden layer จาก 4-4-1 เป็น 4-3-2-1 ทำให้ค่า error เฉลี่ยลดลง

จึงสรุปได้ว่าโครงค่ายที่มีจำนวน hidden layer และ node มากกว่าจะให้ความถูกต้องมากกว่า

การทดลองโดยใช้ 10% cross validation กับ Training set cross.pat

Neural network 2-3-2

```
------------Variable------------
Neural name     2-3-2
Activation func tanh
Learning rate   0.2
Momentum        0.3
Epoch           1000
TrainingFile    cross.pat

------------- Cross validation block 1 -------------

------------Testing-------------

Features        Output                          Desired class
0.0842 0.4683   [ 0.5146610983  0.4360603709]   [0 1]
0.5474 0.2817   [ 0.5106900045  0.5321637955]   [0 1]
0.7459 0.2783   [ 0.4921878088  0.5656750004]   [1 0]
0.8175 0.2770   [ 0.4835596211  0.5754944296]   [1 0]
0.2438 0.1689   [ 0.5020511548  0.4290173069]   [1 0]
0.8239 0.6233   [ 0.4795308392  0.6077925265]   [0 1]
0.4163 0.7219   [ 0.5282030247  0.5645867050]   [0 1]
0.2582 0.0687   [ 0.4939302740  0.4122268578]   [1 0]
0.7948 0.5374   [ 0.4844277249  0.5973063167]   [0 1]
0.4005 0.2314   [ 0.5136312785  0.4901919831]   [0 1]
0.4777 0.0000   [ 0.5013295346  0.4708129585]   [0 1]
0.9319 0.8400   [ 0.4633460100  0.6344723465]   [1 0]
0.4509 0.0853   [ 0.5059383947  0.4785102972]   [0 1]
0.7246 0.2069   [ 0.4943576348  0.5544069764]   [1 0]
0.6031 0.4165   [ 0.5080540293  0.5593059605]   [0 1]
0.9788 0.2190   [ 0.4633414678  0.5899864516]   [1 0]
0.8897 0.1033   [ 0.4753080981  0.5674937349]   [1 0]
0.3561 0.4977   [ 0.5262878566  0.5222093838]   [0 1]
0.2252 0.2299   [ 0.5059345498  0.4354726434]   [1 0]
0.7686 0.3057   [ 0.4894623678  0.5718099340]   [1 0]

Error AV 0.2550786971
Accuracy 40.0000%
```

```
------------ Cross validation block 2 ------------

------------Testing-------------

Features        Output                              Desired class
0.8818 0.8665   [ 0.5164988432  0.7510260699]   [1 0]
0.8869 0.2814   [ 0.4740639292  0.6247106100]   [1 0]
0.3086 0.5060   [ 0.5334249946  0.5269338544]   [0 1]
0.1889 0.8486   [ 0.5557050978  0.6016292750]   [1 0]
0.8189 0.2253   [ 0.4782048263  0.5909972525]   [1 0]
0.4470 0.5504   [ 0.5252959269  0.5860286344]   [0 1]
0.8386 0.7856   [ 0.5108909096  0.7309679588]   [1 0]
0.7044 0.2208   [ 0.4893024905  0.5567010374]   [1 0]
0.9716 0.7389   [ 0.5011107976  0.7410257543]   [1 0]
0.3480 0.7506   [ 0.5408543796  0.6185034611]   [1 0]
0.7032 0.5949   [ 0.5062147041  0.6650208360]   [0 1]
0.0523 0.8042   [ 0.5615531843  0.5429266889]   [1 0]
0.3052 0.6927   [ 0.5416923682  0.5882437578]   [1 0]
0.3672 0.9987   [ 0.5503430220  0.6894506718]   [1 0]
0.2655 0.2685   [ 0.5209784091  0.4227084349]   [0 1]
0.0538 0.1519   [ 0.5073960886  0.2928937791]   [1 0]
0.9737 0.0683   [ 0.4552475031  0.5846745418]   [1 0]
0.8855 0.1773   [ 0.4693559122  0.5943064566]   [1 0]
0.1412 0.6891   [ 0.5517167455  0.5337284975]   [1 0]
0.8915 0.8043   [ 0.5103285042  0.7417692315]   [1 0]

Error AV 0.2715636704
Accuracy 25.0000%
-----------------------------------------------------
------------ Cross validation block 3 ------------

------------Testing-------------

Features        Output                              Desired class
0.1452 0.3705   [ 0.5457422617  0.4336308790]   [1 0]
0.4909 0.5871   [ 0.5199046823  0.5866188345]   [0 1]
0.6870 0.8106   [ 0.4965647358  0.6657220501]   [1 0]
0.3831 0.8177   [ 0.5260453308  0.6127811877]   [1 0]
0.7249 0.8469   [ 0.4952655844  0.6776382960]   [1 0]
0.3620 0.8441   [ 0.5277970966  0.6139310767]   [1 0]
0.0928 0.8304   [ 0.5604799195  0.5524213212]   [1 0]
0.8016 0.5192   [ 0.4871687826  0.6344576797]   [0 1]
0.8077 0.7634   [ 0.4899906769  0.6774745302]   [1 0]
0.7590 0.3777   [ 0.4932568378  0.5980882717]   [1 0]
0.4824 0.9477   [ 0.5120041419  0.6548184648]   [0 1]
0.4154 0.6172   [ 0.5283889599  0.5768709676]   [0 1]
0.5088 0.5415   [ 0.5190369692  0.5804201440]   [0 1]
0.7466 0.8522   [ 0.4945988045  0.6818374031]   [1 0]
0.5675 0.2185   [ 0.5189355776  0.5151142267]   [0 1]
0.7772 0.4676   [ 0.4897964056  0.6199944662]   [0 1]
0.4509 0.5433   [ 0.5261230460  0.5679062301]   [0 1]
0.1830 0.8217   [ 0.5507648822  0.5715509896]   [1 0]
0.8590 0.0838   [ 0.4880274024  0.5539508549]   [1 0]
0.4784 0.3855   [ 0.5263804202  0.5359779441]   [0 1]

Error AV 0.2628272637
Accuracy 50.0000%
-----------------------------------------------------
```

```
------------ Cross validation block 4 ------------

------------Testing-------------

Features        Output                          Desired class
1.0000 0.4854   [-0.1742838280  0.9409923972]   [0 1]
0.2358 0.5684   [-0.1402883860  0.9494866777]   [0 1]
0.3458 0.6498   [-0.0077388910  0.9277471601]   [0 1]
0.2956 0.7931   [ 0.7253300877  0.2181175074]   [1 0]
0.5313 0.7802   [ 0.7052642181  0.2464703468]   [0 1]
0.5359 0.5659   [-0.1656004119  0.9492115843]   [0 1]
0.9228 0.6112   [-0.1574937186  0.9446657057]   [0 1]
0.0566 0.4951   [-0.1412459361  0.9485528155]   [0 1]
0.5640 0.5521   [-0.1734527568  0.9493151561]   [0 1]
0.8930 0.2511   [ 0.8539167853  0.0281958832]   [1 0]
0.8344 0.5514   [-0.1953215434  0.9488714507]   [0 1]
0.3903 0.4089   [ 0.1234295248  0.8949296576]   [0 1]
0.1897 0.7067   [ 0.4161058854  0.7452505395]   [1 0]
0.6484 0.7466   [ 0.6316137757  0.4089353101]   [0 1]
0.4469 0.5430   [-0.1668842144  0.9494418430]   [0 1]
0.5952 0.7574   [ 0.6644338491  0.3413423469]   [0 1]
0.4107 0.5648   [-0.1560243417  0.9493842008]   [0 1]
0.2488 0.4502   [-0.1003428737  0.9391570846]   [0 1]
0.6990 0.4454   [-0.0613434207  0.9267152661]   [0 1]
0.7559 0.7985   [ 0.7131761008  0.2036828222]   [1 0]

Error AV 0.4033088729
Accuracy 80.0000%
-----------------------------------------------------
 ------------ Cross validation block 5 ------------

 ------------Testing-------------

 Features        Output                          Desired class
 0.1364 0.7663   [ 0.4963339006  0.5067045942]   [1 0]
 0.7909 0.2158   [ 0.4949778897  0.5052817044]   [1 0]
 0.4892 0.6156   [ 0.4975266173  0.5079076143]   [0 1]
 0.8143 0.5887   [ 0.5000902465  0.5105156941]   [0 1]
 0.5738 0.1506   [ 0.4921472845  0.5023936382]   [0 1]
 0.1337 0.7618   [ 0.4962503143  0.5066191358]   [1 0]
 0.6155 0.5313   [ 0.4975543465  0.5079294361]   [0 1]
 0.8428 0.7575   [ 0.5025468573  0.5130291883]   [1 0]
 0.5119 0.4503   [ 0.4955544987  0.5058868145]   [0 1]
 0.6064 0.5329   [ 0.4974935486  0.5078676514]   [0 1]
 0.9551 0.3439   [ 0.4981447156  0.5085163118]   [1 0]
 0.4343 0.5832   [ 0.4966064288  0.5069681590]   [0 1]
 0.6330 0.2632   [ 0.4941767791  0.5044694202]   [1 0]
 0.5459 0.7060   [ 0.4992216053  0.5096402546]   [0 1]
 0.6445 0.8764   [ 0.5023276624  0.5128150688]   [1 0]
 0.7396 0.1480   [ 0.4936167372  0.5038902081]   [1 0]
 0.6262 0.5184   [ 0.4974810116  0.5078537826]   [0 1]
 0.6352 1.0000   [ 0.5038512614  0.5143751287]   [0 1]
 0.6544 0.4623   [ 0.4969969512  0.5073566162]   [0 1]
 0.3431 0.5531   [ 0.4953874223  0.5057245079]   [0 1]

 Error AV 0.2500696630
 Accuracy 60.0000%
 -----------------------------------------------------
```

```
------------- Cross validation block 6 -------------

------------Testing-------------

Features        Output                              Desired class
0.2097 0.8957   [ 0.5081801965  0.5410410334]   [1 0]
0.0609 0.1103   [ 0.4839765366  0.5158523159]   [1 0]
0.4653 0.6214   [ 0.5218991079  0.5551099929]   [0 1]
0.7736 0.8976   [ 0.5474290695  0.5813399777]   [1 0]
0.4339 0.6054   [ 0.5194262450  0.5525634244]   [0 1]
0.2574 0.3660   [ 0.5028342054  0.5354137811]   [1 0]
0.1539 0.7021   [ 0.5009281104  0.5335167743]   [1 0]
0.1408 0.5054   [ 0.4966634885  0.5290688000]   [0 1]
0.5200 0.7388   [ 0.5276069677  0.5609989943]   [0 1]
0.5934 0.4977   [ 0.5288851765  0.5622626008]   [0 1]
0.4419 0.9173   [ 0.5249974740  0.5583549473]   [0 1]
0.1823 0.7609   [ 0.5039714784  0.5366710987]   [1 0]
0.3983 0.4703   [ 0.5147023093  0.5476732010]   [0 1]
0.3496 0.1067   [ 0.5051916658  0.5377924827]   [1 0]
0.1240 0.2975   [ 0.4918946412  0.5240923836]   [1 0]
0.8217 0.6893   [ 0.5474878417  0.5813593040]   [1 0]
0.6600 0.8513   [ 0.5390259433  0.5727324563]   [1 0]
0.8873 0.7189   [ 0.5523278537  0.5863101921]   [1 0]
0.4983 0.3113   [ 0.5192155545  0.5522861715]   [0 1]
0.2988 0.5136   [ 0.5082852581  0.5410693032]   [0 1]

Error AV 0.2528991369
Accuracy 45.0000%
-----------------------------------------------------
 ------------- Cross validation block 7 -------------

 ------------Testing-------------

 Features        Output                              Desired class
 0.5577 0.9503   [ 0.5104899015  0.5432222895]   [0 1]
 0.6008 0.7255   [ 0.5110571277  0.5438057125]   [0 1]
 0.0030 0.9024   [ 0.4817264123  0.5135093511]   [1 0]
 0.6737 0.5399   [ 0.5133857773  0.5462027494]   [0 1]
 0.5031 0.5944   [ 0.5052191233  0.5377890827]   [0 1]
 0.2431 0.1419   [ 0.4886405661  0.5206642850]   [1 0]
 0.0901 0.2039   [ 0.4811147635  0.5128731126]   [1 0]
 0.6119 0.6125   [ 0.5108140067  0.5435549165]   [0 1]
 0.8510 0.3547   [ 0.5208615378  0.5538916211]   [1 0]
 0.3461 0.5444   [ 0.4968792909  0.5291821726]   [0 1]
 0.1556 0.2654   [ 0.4849970995  0.5168940950]   [1 0]
 0.5411 0.3724   [ 0.5055538961  0.5381334740]   [0 1]
 0.1472 0.1856   [ 0.4839695011  0.5158298554]   [1 0]
 0.9834 0.5201   [ 0.5284631262  0.5616985978]   [0 1]
 0.7200 0.0997   [ 0.5125945226  0.5453866105]   [1 0]
 0.6744 0.7515   [ 0.5149065862  0.5477687666]   [1 0]
 0.2142 0.2516   [ 0.4879455195  0.5199455844]   [1 0]
 0.5059 0.4930   [ 0.5046369313  0.5371883669]   [0 1]
 0.1209 0.7077   [ 0.4864471613  0.5183967144]   [1 0]
 0.0000 0.1668   [ 0.4760912528  0.5076659578]   [1 0]

 Error AV 0.2513047332
 Accuracy 45.0000%
 -----------------------------------------------------
```

```
------------ Cross validation block 8 ------------

------------Testing-------------

Features        Output                              Desired class
0.3749 0.7395   [ 0.5678923211  0.6054220456]   [0 1]
0.7357 0.1887   [ 0.5322565663  0.5503246591]   [1 0]
0.8273 0.9289   [ 0.5653886291  0.7182441959]   [1 0]
0.8311 0.5247   [ 0.5471868438  0.6480996274]   [0 1]
0.8392 0.5134   [ 0.5464516219  0.6472700303]   [0 1]
0.2962 0.0697   [ 0.5143254081  0.3804520405]   [1 0]
0.4394 0.7473   [ 0.5669389463  0.6206704464]   [0 1]
0.9222 0.8117   [ 0.5575585475  0.7126291834]   [1 0]
0.5526 0.4095   [ 0.5480038906  0.5645854979]   [0 1]
0.7974 0.7302   [ 0.5570247426  0.6811512551]   [1 0]
0.7038 0.3263   [ 0.5409170285  0.5784838714]   [1 0]
0.7387 0.4991   [ 0.5486820238  0.6258101753]   [0 1]
0.7385 0.7779   [ 0.5606715001  0.6804374788]   [1 0]
0.0866 0.8116   [ 0.5751536599  0.5562548853]   [1 0]
0.1989 0.1818   [ 0.5236731154  0.3889306810]   [1 0]
0.5533 0.5493   [ 0.5554242480  0.5993381304]   [0 1]
0.5325 0.7068   [ 0.5630716405  0.6304065736]   [0 1]
0.7779 0.2660   [ 0.5361574415  0.5801226089]   [1 0]
0.5629 0.5807   [ 0.5567251144  0.6086555704]   [0 1]
0.0119 0.7999   [ 0.5746989103  0.5337936739]   [1 0]

Error AV 0.2666119199
Accuracy 65.0000%
-----------------------------------------------------
 ------------ Cross validation block 9 ------------

 ------------Testing-------------

 Features        Output                              Desired class
0.3897 0.4966   [ 0.5067644885  0.4932949386]   [0 1]
0.3718 0.5199   [ 0.5069322828  0.4934637409]   [0 1]
0.1958 0.7386   [ 0.5085450768  0.4950849746]   [1 0]
0.5236 0.3460   [ 0.5055825150  0.4921090576]   [0 1]
0.6822 0.3444   [ 0.5047447317  0.4912934029]   [1 0]
0.1269 0.1416   [ 0.5070073930  0.4934475490]   [1 0]
0.4347 0.5128   [ 0.5065802416  0.4931194496]   [0 1]
0.8847 0.8514   [ 0.5052929941  0.4919468079]   [1 0]
0.4280 0.5098   [ 0.5066058095  0.4931436243]   [0 1]
0.6740 0.5383   [ 0.5054064148  0.4919831616]   [0 1]
0.2525 0.2539   [ 0.5067092726  0.4931838968]   [1 0]
0.1332 0.5733   [ 0.5083498368  0.4948560305]   [0 1]
0.2653 0.2286   [ 0.5065612821  0.4930338811]   [1 0]
0.4771 0.5672   [ 0.5065308885  0.4930842614]   [0 1]
0.3956 0.5292   [ 0.5068371796  0.4933733777]   [0 1]
0.4755 0.5915   [ 0.5066163927  0.4931732089]   [0 1]
0.3304 0.5528   [ 0.5072534840  0.4937841112]   [0 1]
0.5050 0.2814   [ 0.5054735060  0.4919876926]   [0 1]
0.2448 0.2711   [ 0.5068045904  0.4932807324]   [1 0]
0.2756 0.9146   [ 0.5086819571  0.4952596020]   [1 0]

Error AV 0.2500473283
Accuracy 40.0000%
-----------------------------------------------------
```

```
------------- Cross validation block 10 -------------

------------Testing-------------

Features        Output                              Desired class
0.2047 0.2394   [ 0.4904395603  0.4356029177]   [1 0]
0.4692 0.4133   [ 0.4815495783  0.5303686118]   [0 1]
0.8496 0.8346   [ 0.4352760150  0.6381584645]   [1 0]
0.2580 0.8219   [ 0.4946451094  0.5764608010]   [1 0]
0.3932 0.7691   [ 0.4845092281  0.5861105536]   [0 1]
0.5935 0.5255   [ 0.4706961213  0.5705689911]   [0 1]
0.1931 0.2549   [ 0.4913129015  0.4370470234]   [1 0]
0.1619 0.2285   [ 0.4909187202  0.4226858322]   [1 0]
0.0902 0.2690   [ 0.4939802643  0.4167282862]   [1 0]
0.4503 0.6102   [ 0.4817564502  0.5658183917]   [0 1]
0.8188 0.1137   [ 0.4564689810  0.5264262589]   [1 0]
0.5068 0.4199   [ 0.4788865549  0.5379678675]   [0 1]
0.4285 0.6941   [ 0.4825424138  0.5778313547]   [0 1]
0.8837 0.6746   [ 0.4356905724  0.6214480964]   [0 1]
0.8717 0.7477   [ 0.4349855737  0.6294532507]   [1 0]
0.3349 0.8492   [ 0.4883239171  0.5920759006]   [1 0]
0.3661 0.5083   [ 0.4883966594  0.5324230513]   [0 1]
0.6844 0.9535   [ 0.4518243155  0.6416989187]   [1 0]
0.8724 0.4701   [ 0.4433522455  0.5933361579]   [0 1]
0.5370 0.4896   [ 0.4760315709  0.5561608462]   [0 1]

Error AV 0.2572391979
Accuracy 70.0000%
-----------------------------------------------------
Error Average  0.272095048338
```

Neural network 2-3-3-2

```
------------Variable------------
Neural name     2-3-3-2
Activation func tanh
Learning rate   0.2
Momentum        0.3
Epoch           1000
TrainingFile    cross.pat

------------- Cross validation block 1 -------------

------------Testing-------------

Features        Output                          Desired class
0.4909 0.5871   [ 0.5584567458  0.9361326641]   [0 1]
0.6990 0.4454   [ 0.8695548627  0.0441903078]   [0 1]
0.0842 0.4683   [ 0.8952840072 -0.1082478455]   [0 1]
0.1412 0.6891   [ 0.8952779768 -0.1082425376]   [1 0]
0.2988 0.5136   [ 0.8869818274 -0.0931501928]   [0 1]
0.8273 0.9289   [ 0.8826606639 -0.0609456557]   [1 0]
0.9228 0.6112   [ 0.8832532397 -0.0629172679]   [0 1]
0.8428 0.7575   [ 0.8831331912 -0.0625647764]   [1 0]
0.5577 0.9503   [ 0.5583331480  0.9361819567]   [0 1]
1.0000 0.4854   [ 0.8832525370 -0.0629129159]   [0 1]
0.4469 0.5430   [ 0.5591267596  0.9358908571]   [0 1]
0.6031 0.4165   [ 0.5663161318  0.9332122339]   [0 1]
0.4107 0.5648   [ 0.5627715013  0.9345283776]   [0 1]
0.7200 0.0997   [ 0.8827773356 -0.0613834647]   [1 0]
0.5459 0.7060   [ 0.5584580073  0.9361351737]   [0 1]
0.7249 0.8469   [ 0.7577950528  0.6940696613]   [1 0]
0.8869 0.2814   [ 0.8832482665 -0.0628986838]   [1 0]
0.3620 0.8441   [ 0.7641188082  0.6717013428]   [1 0]
0.5533 0.5493   [ 0.5587729628  0.9360240511]   [0 1]
0.6844 0.9535   [ 0.5697891319  0.9318116584]   [1 0]

Error AV 0.4888648939
Accuracy 70.0000%
```

```
------------ Cross validation block 2 ------------

------------Testing-------------

Features        Output                          Desired class
0.8855 0.1773   [ 0.8701246758  0.0232319506]   [1 0]
0.7974 0.7302   [ 0.9372701057  0.0768045834]   [1 0]
0.4824 0.9477   [ 0.8597113129  0.5838023718]   [0 1]
0.8077 0.7634   [ 0.9386941118  0.0223490075]   [1 0]
0.2358 0.5684   [ 0.9508239130 -0.0535324714]   [0 1]
0.3496 0.1067   [ 0.8749576511  0.1005420079]   [1 0]
0.4653 0.6214   [-0.0321216485  0.9782191933]   [0 1]
0.5640 0.5521   [-0.0321434251  0.9782202843]   [0 1]
0.7779 0.2660   [ 0.8697860218  0.0254711529]   [1 0]
0.3661 0.5083   [-0.0299984912  0.9781125754]   [0 1]
0.3956 0.5292   [-0.0314695006  0.9781864980]   [0 1]
0.3831 0.8177   [ 0.7528171072  0.7765923014]   [1 0]
0.2252 0.2299   [ 0.8983411219  0.0483584268]   [1 0]
0.7396 0.1480   [ 0.8666799878  0.0395856340]   [1 0]
0.2756 0.9146   [ 0.9507679020 -0.0528759422]   [1 0]
0.8915 0.8043   [ 0.9379982214 -0.0045682334]   [1 0]
0.8386 0.7856   [ 0.9382571631  0.0021765425]   [1 0]
0.7459 0.2783   [ 0.8688898696  0.0314069007]   [1 0]
0.9788 0.2190   [ 0.8703612566  0.0223560453]   [1 0]
0.2956 0.7931   [ 0.9507334267 -0.0524597036]   [1 0]

Error AV 0.4650424510
Accuracy 85.0000%
-----------------------------------------------------
 ------------ Cross validation block 3 -------------

 ------------Testing-------------

 Features        Output                          Desired class
0.0523 0.8042   [ 0.8977605686  0.0109859681]   [1 0]
0.1897 0.7067   [ 0.8975969915  0.0110418566]   [1 0]
0.8510 0.3547   [ 0.9381493832  0.0464733380]   [1 0]
0.5200 0.7388   [-0.0240123544  0.9774078259]   [0 1]
0.1989 0.1818   [ 0.9413854743 -0.0118333169]   [1 0]
0.4394 0.7473   [-0.0252064204  0.9775190322]   [0 1]
0.9737 0.0683   [ 0.9384465982  0.0412511297]   [1 0]
0.8873 0.7189   [ 0.8688047797  0.0765450965]   [1 0]
0.7357 0.1887   [ 0.9392236080  0.0274722280]   [1 0]
0.0902 0.2690   [ 0.9413506208 -0.0111910524]   [1 0]
0.4005 0.2314   [ 0.1880892442  0.9467982551]   [0 1]
0.3304 0.5528   [ 0.1726207979  0.9571932425]   [0 1]
0.4285 0.6941   [-0.0252389928  0.9775220306]   [0 1]
0.3086 0.5060   [ 0.6278720986  0.5967898337]   [0 1]
0.4509 0.0853   [ 0.0784127332  0.9682266560]   [0 1]
0.8239 0.6233   [ 0.8684457374  0.0764061416]   [0 1]
0.3897 0.4966   [-0.0252500135  0.9775229973]   [0 1]
0.4280 0.5098   [-0.0227747014  0.9772909798]   [0 1]
0.3903 0.4089   [-0.0195333729  0.9769825198]   [0 1]
0.5411 0.3724   [-0.0105457775  0.9761172460]   [0 1]

Error AV 0.4615466652
Accuracy 90.0000%
-----------------------------------------------------
```

```
------------- Cross validation block 4 -------------

------------Testing-------------

Features        Output                              Desired class
0.1539 0.7021   [ 0.8522385532  0.5609975186]   [1 0]
0.3672 0.9987   [ 0.7994166035  0.7658571501]   [1 0]
0.6822 0.3444   [ 0.9774836890  0.0249106878]   [1 0]
0.7736 0.8976   [ 0.9501104469 -0.3061531075]   [1 0]
0.7948 0.5374   [ 0.6408150380  0.9548071389]   [0 1]
0.5119 0.4503   [ 0.6263899088  0.9711682207]   [0 1]
0.1958 0.7386   [ 0.9151098346 -0.0095122771]   [1 0]
0.3480 0.7506   [ 0.8191570840  0.7140043406]   [1 0]
0.4509 0.5433   [ 0.6237281828  0.9711884719]   [0 1]
0.9716 0.7389   [ 0.9268867384  0.0872020982]   [1 0]
0.7387 0.4991   [ 0.6710180769  0.9485289128]   [0 1]
0.4503 0.6102   [ 0.6092391956  0.9712637883]   [0 1]
0.2580 0.8219   [ 0.9251025092 -0.1745399947]   [1 0]
0.4771 0.5672   [ 0.6212306910  0.9712029876]   [0 1]
0.5474 0.2817   [ 0.6273493714  0.9708758100]   [0 1]
0.3052 0.6927   [ 0.7309246315  0.8754202885]   [1 0]
0.5059 0.4930   [ 0.6259276118  0.9711739501]   [0 1]
0.0566 0.4951   [ 0.6460891807  0.9566535472]   [0 1]
0.4343 0.5832   [ 0.6177023429  0.9712197637]   [0 1]
0.5088 0.5415   [ 0.6241930668  0.9711857162]   [0 1]

Error AV 0.4707116880
Accuracy 95.0000%
-----------------------------------------------------
------------- Cross validation block 5 -------------

------------Testing-------------

Features        Output                              Desired class
0.9834 0.5201   [ 0.5093939052  0.5033126507]   [0 1]
0.1452 0.3705   [ 0.5064725441  0.5003919942]   [1 0]
0.4692 0.4133   [ 0.5083452771  0.5022591715]   [0 1]
0.5629 0.5807   [ 0.5085004152  0.5024204717]   [0 1]
0.6352 1.0000   [ 0.5078487706  0.5017884287]   [0 1]
0.8344 0.5514   [ 0.5094489408  0.5033661232]   [0 1]
0.2097 0.8957   [ 0.5050427802  0.4989904742]   [1 0]
0.4892 0.6156   [ 0.5080083017  0.5019315626]   [0 1]
0.6445 0.8764   [ 0.5082724006  0.5022052644]   [1 0]
0.1332 0.5733   [ 0.5056884800  0.4996197478]   [0 1]
0.8847 0.8514   [ 0.5095021737  0.5034300186]   [1 0]
0.1931 0.2549   [ 0.5071363156  0.5010482920]   [1 0]
0.5236 0.3460   [ 0.5086885525  0.5025988004]   [0 1]
0.1269 0.1416   [ 0.5070259443  0.5009337475]   [1 0]
0.2574 0.3660   [ 0.5072374634  0.5011535722]   [1 0]
0.7590 0.3777   [ 0.5092579344  0.5031692530]   [1 0]
0.4419 0.9173   [ 0.5067339879  0.5006750286]   [0 1]
0.8724 0.4701   [ 0.5093986345  0.5033139460]   [0 1]
0.1240 0.2975   [ 0.5065577119  0.5004737936]   [1 0]
0.7686 0.3057   [ 0.5091825118  0.5030920030]   [1 0]

Error AV 0.2500049732
Accuracy 50.0000%
-----------------------------------------------------
```

```
------------ Cross validation block 6 ------------

------------Testing-------------

Features        Output                              Desired class
0.2448 0.2711   [ 0.5054244011  0.5001791976]   [1 0]
0.5526 0.4095   [ 0.5066513838  0.5014021724]   [0 1]
0.0119 0.7999   [ 0.5018824769  0.4966586250]   [1 0]
0.4339 0.6054   [ 0.5057703928  0.5005263321]   [0 1]
0.1209 0.7077   [ 0.5031306551  0.4978998957]   [1 0]
0.6008 0.7255   [ 0.5063814707  0.5011357794]   [0 1]
0.5325 0.7068   [ 0.5060939626  0.5008493095]   [0 1]
0.7909 0.2158   [ 0.5065457242  0.5012976182]   [1 0]
0.7044 0.2208   [ 0.5067997103  0.5015496650]   [1 0]
0.6330 0.2632   [ 0.5068691260  0.5016184566]   [1 0]
0.7246 0.2069   [ 0.5067454257  0.5014957370]   [1 0]
0.1364 0.7663   [ 0.5030567746  0.4978268564]   [1 0]
0.1337 0.7618   [ 0.5030497314  0.4978198109]   [1 0]
0.4154 0.6172   [ 0.5056303832  0.5003870420]   [0 1]
0.3718 0.5199   [ 0.5056003456  0.5003563546]   [0 1]
0.2047 0.2394   [ 0.5052443889  0.4999997501]   [1 0]
0.3932 0.7691   [ 0.5050798594  0.4998402640]   [0 1]
0.5952 0.7574   [ 0.5063016388  0.5010565250]   [0 1]
0.3431 0.5531   [ 0.5053304077  0.5000879094]   [0 1]
0.6262 0.5184   [ 0.5067360502  0.5014874478]   [0 1]

Error AV 0.2500020959
Accuracy 50.0000%
-----------------------------------------------------
 ------------ Cross validation block 7 -------------

 ------------Testing-------------

 Features        Output                              Desired class
 0.4777 0.0000   [-0.0203697909  0.9415170693]   [0 1]
 0.1556 0.2654   [ 0.9535008863 -0.0438104444]   [1 0]
 0.2962 0.0697   [ 0.9535009035 -0.0438102677]   [1 0]
 0.3983 0.4703   [-0.0247234944  0.9430662151]   [0 1]
 0.5370 0.4896   [-0.0247226138  0.9430663061]   [0 1]
 0.8175 0.2770   [ 0.9535009132 -0.0438101683]   [1 0]
 0.3461 0.5444   [ 0.1678253500  0.8756314225]   [0 1]
 0.5935 0.5255   [-0.0247229349  0.9430662729]   [0 1]
 0.7385 0.7779   [ 0.8611335792  0.0571545544]   [1 0]
 0.5934 0.4977   [-0.0247228321  0.9430662835]   [0 1]
 0.5313 0.7802   [-0.0247226935  0.9430662979]   [0 1]
 0.8717 0.7477   [ 0.8642009565  0.0536523712]   [1 0]
 0.6155 0.5313   [-0.0247236284  0.9430658557]   [0 1]
 0.8496 0.8346   [ 0.8661352828  0.0507236440]   [1 0]
 0.6744 0.7515   [ 0.7945530154  0.1434726028]   [1 0]
 0.8392 0.5134   [ 0.8495286676  0.0762957375]   [0 1]
 0.4470 0.5504   [-0.0247226272  0.9430663047]   [0 1]
 0.6870 0.8106   [ 0.8409128390  0.0822480914]   [1 0]
 0.8143 0.5887   [ 0.8560902142  0.0658734520]   [0 1]
 0.2438 0.1689   [ 0.9535008826 -0.0438104820]   [1 0]

 Error AV 0.4579957950
 Accuracy 90.0000%
 -----------------------------------------------------
```

```
------------ Cross validation block 8 ------------

------------Testing-------------

Features        Output                              Desired class
0.1619 0.2285   [ 0.5362104870  0.4815654516]   [1 0]
0.0928 0.8304   [ 0.5374790212  0.4827647510]   [1 0]
0.5031 0.5944   [ 0.5377715685  0.4830413311]   [0 1]
0.8311 0.5247   [ 0.5382833261  0.4835252834]   [0 1]
0.8837 0.6746   [ 0.5387414622  0.4839586346]   [0 1]
0.8217 0.6893   [ 0.5386484393  0.4838706474]   [1 0]
0.3458 0.6498   [ 0.5375773520  0.4828576932]   [0 1]
0.6064 0.5329   [ 0.5378403158  0.4831063288]   [0 1]
0.0609 0.1103   [ 0.5357240553  0.4811057187]   [1 0]
0.6484 0.7466   [ 0.5384261825  0.4836604369]   [0 1]
0.5068 0.4199   [ 0.5373707860  0.4826623373]   [0 1]
0.1472 0.1856   [ 0.5360793787  0.4814415304]   [1 0]
0.4163 0.7219   [ 0.5378911551  0.4831544398]   [0 1]
0.0901 0.2039   [ 0.5360043565  0.4813706310]   [1 0]
0.4347 0.5128   [ 0.5374397643  0.4827275738]   [0 1]
0.1408 0.5054   [ 0.5368165817  0.4821384037]   [0 1]
0.2488 0.4502   [ 0.5369099479  0.4822266561]   [0 1]
0.4755 0.5915   [ 0.5377079708  0.4829811923]   [0 1]
0.7032 0.5949   [ 0.5381844015  0.4834317380]   [0 1]
0.7772 0.4676   [ 0.5380390329  0.4832942366]   [0 1]

Error AV 0.2502984471
Accuracy 30.0000%
-----------------------------------------------------
 ------------ Cross validation block 9 ------------

 ------------Testing-------------

 Features        Output                              Desired class
 0.8016 0.5192   [ 0.5264565937  0.5115329797]   [0 1]
 0.8188 0.1137   [ 0.5341363892  0.4893771669]   [1 0]
 0.3749 0.7395   [ 0.5146153603  0.4768270642]   [0 1]
 0.0866 0.8116   [ 0.5074842808  0.4479624684]   [1 0]
 0.2655 0.2685   [ 0.5200798698  0.4366984172]   [0 1]
 0.4784 0.3855   [ 0.5228743680  0.4672824590]   [0 1]
 0.6544 0.4623   [ 0.5249084208  0.4917829442]   [0 1]
 0.7466 0.8522   [ 0.5191062266  0.5247616716]   [1 0]
 0.2653 0.2286   [ 0.5206475063  0.4344776819]   [1 0]
 0.9551 0.3439   [ 0.5323624576  0.5180092479]   [1 0]
 0.7559 0.7985   [ 0.5203111509  0.5227248090]   [1 0]
 0.8897 0.1033   [ 0.5356193981  0.4966399267]   [1 0]
 0.3561 0.4977   [ 0.5184783021  0.4600767618]   [0 1]
 0.6740 0.5383   [ 0.5238657000  0.4985395675]   [0 1]
 0.7038 0.3263   [ 0.5282659076  0.4891774993]   [1 0]
 0.6737 0.5399   [ 0.5238306150  0.4986017927]   [0 1]
 0.5050 0.2814   [ 0.5251526827  0.4641344125]   [0 1]
 0.1889 0.8486   [ 0.5089750207  0.4620530460]   [1 0]
 0.5359 0.5659   [ 0.5208295969  0.4846451072]   [0 1]
 0.9319 0.8400   [ 0.5224128075  0.5431047092]   [1 0]

 Error AV 0.2510464663
 Accuracy 35.0000%
 -----------------------------------------------------
```

```
------------- Cross validation block 10 -------------

------------Testing-------------

Features        Output                              Desired class
0.3349 0.8492   [-0.0602010063  0.9707556472]   [1 0]
0.2142 0.2516   [ 0.9044495887  0.4819365651]   [1 0]
0.0538 0.1519   [ 0.9167533156  0.4192368315]   [1 0]
0.6600 0.8513   [ 0.8891590035  0.0604410962]   [1 0]
0.2582 0.0687   [ 0.8978201324  0.5064489232]   [1 0]
0.8590 0.0838   [ 0.9133395754 -0.0508613679]   [1 0]
0.2431 0.1419   [ 0.9005305017  0.4960843163]   [1 0]
0.1830 0.8217   [ 0.9204801381  0.3696900516]   [1 0]
0.1823 0.7609   [ 0.9202964084  0.3852476533]   [1 0]
0.8930 0.2511   [ 0.9140268804 -0.0493930657]   [1 0]
0.6119 0.6125   [-0.0801317527  0.9782892786]   [0 1]
0.5675 0.2185   [ 0.9073180658  0.0066442236]   [0 1]
0.0000 0.1668   [ 0.9168903644  0.4185696788]   [1 0]
0.8189 0.2253   [ 0.9137646427 -0.0500424945]   [1 0]
0.9222 0.8117   [ 0.8946282597  0.0279200455]   [1 0]
0.4983 0.3113   [-0.1262860964  0.9756518609]   [0 1]
0.0030 0.9024   [ 0.9224934620  0.3543991515]   [1 0]
0.5738 0.1506   [ 0.9089296159 -0.0084783853]   [0 1]
0.8818 0.8665   [ 0.8948781331  0.0215144468]   [1 0]
0.2525 0.2539   [ 0.8855755699  0.5544533269]   [1 0]

Error AV 0.3899632209
Accuracy 85.0000%
-----------------------------------------------------
Error Average  0.37354766967
```

จะเห็นได้ว่า เมื่อเปลี่ยนจำนวน hidden layer จาก 2-3-2 เป็น 2-3-3-2 ทำให้ค่า error เฉลี่ยเพิ่มขึ้น

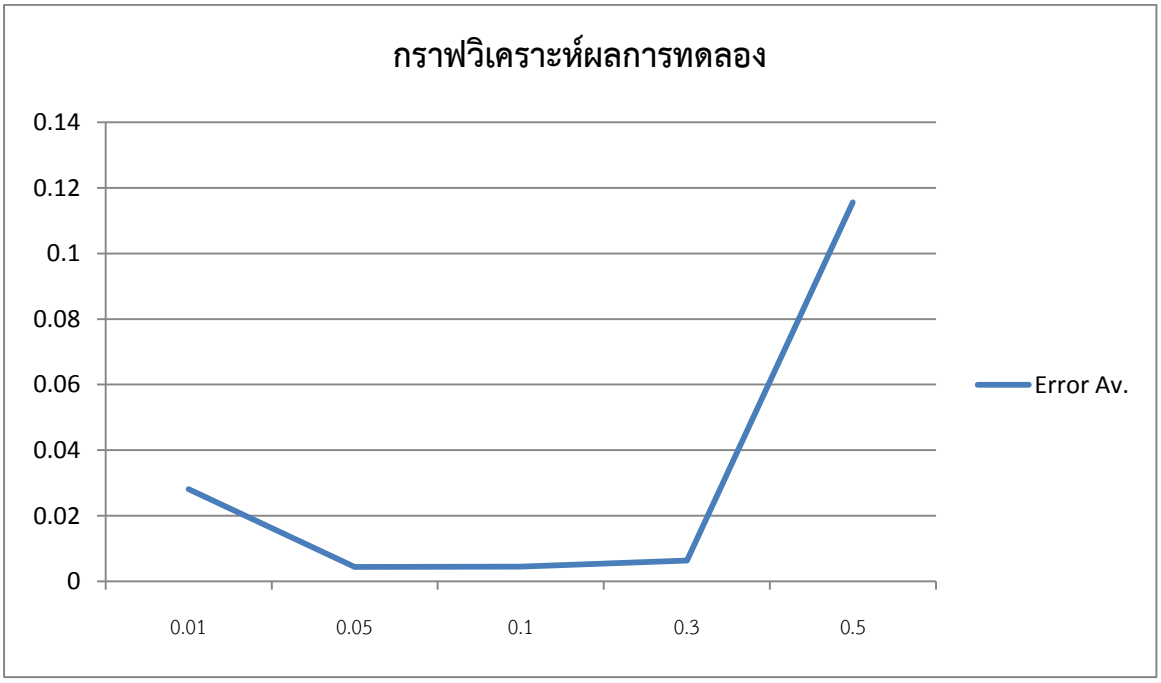
ซึ่งขัดกับผลการทดลองก่อนหน้า จึงยังสรุปไม่ได้ว่าโครงค่ายที่มีจำนวน hidden layer และ node มากกว่าจะให้ความถูกต้องมากกว่า ซึ่งอาจต้องปรับ learning rate และ momentum ให้ดีด้วย

การทดลองปรับค่า learning rate กับ Training set iris.pat

กำหนด โครงข่าย Neural 4-4-1, Momentum = 0.3, Epoch = 100

ซึ่งทดลองปรับ learning rate เป็น 0.01, 0.05, 0.1, 0.3, 0.5 ตามลำดับ

ได้ผลการทดลองดังนี้



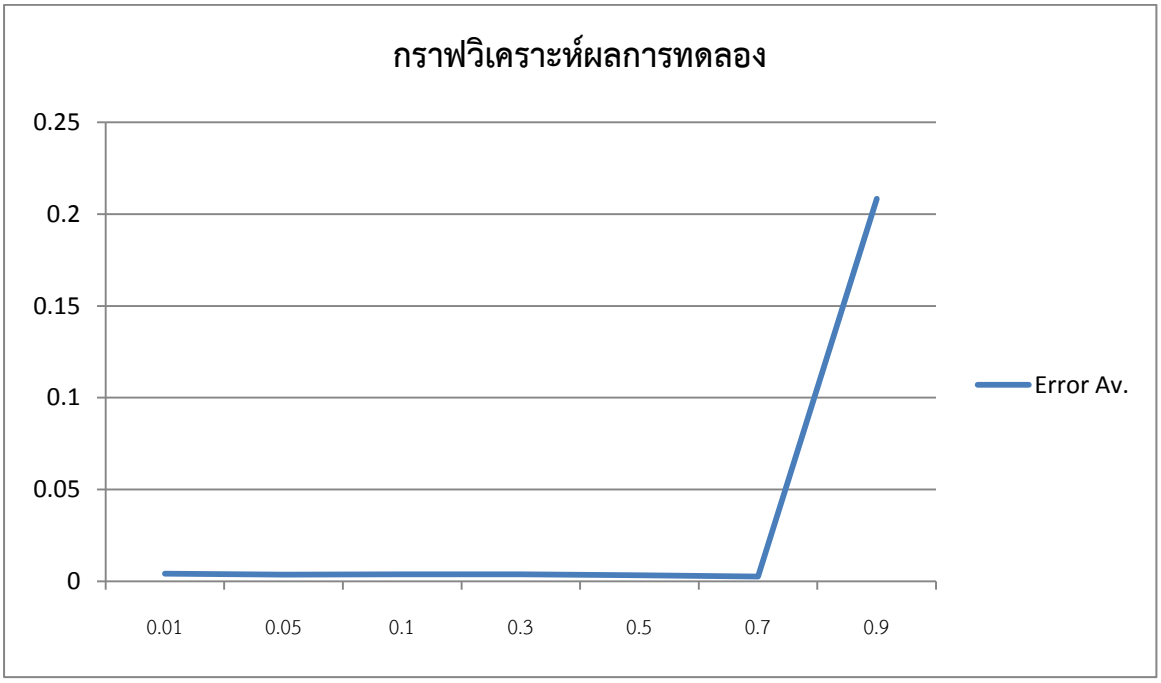
จึงสรุปได้ว่าถ้าปรับค่า learning rate น้อยไปหรือมากไป จะทำให้ Error Av เพิ่มขึ้น

การทดลองปรับค่า Momentum กับ Training set iris.pat

กำหนด โครงข่าย Neural 4-4-1, learning rate = 0.1, Epoch = 100

ซึ่งทดลองปรับ momentum เป็น 0.01, 0.05, 0.1, 0.3, 0.5, 0.7, 0.9 ตามลำดับ

ได้ผลการทดลองดังนี้



จึงสรุปได้ว่าถ้าปรับค่า momentum มากไป จะทำให้ Error Av เพิ่มขึ้น

Code (https://github.com/porpeeranut/Computational_Intelligence_Assignment1)

```
# ComputerAssignment1.py

import numpy as np
import math, sys, getopt, os, fileinput, copy, random


#    Neural Networks Backpropagation


def usage():
    fileName = os.path.basename(sys.argv[0])
    print "\nusage: ", fileName,
    print "[option]"
    print " -N  arg : arg is neural name"
    print " -a  arg : arg is activation function \"tanh\" or \"sigmoid\""
    print " -n  arg : arg is learning rate"
    print " -m  arg : arg is momentum"
    print " -e  arg : arg is number of epoch to exit"
    print " -c      : to test 10%s cross validation" % ('%')
    #print " -E  arg : arg is min average error to exit"
    print " -t  arg : arg is training set file"
    print "\nex."
    print fileName, "-N 2-4-1 -a tanh -n 0.2 -m 0.1 -e 10000 -t train.pat"


class NeuralNetwork:


    def __init__(self, layers, activFunct, learning_rate, momentum, epoch, error):
        self.learning_rate = float(learning_rate)
        self.momentum = float(momentum)
        self.epoch = epoch
        self.error = float(error)
```

```
    if activFunct == 'sigmoid':
        self.activation = sigmoid
        self.activation_derive = sigmoid_derivertive
    else:
        self.activation = tanh
        self.activation_derive = tanh_derivertive

    # Init weight
    self.init_weights = []
    for i in range(1, len(layers) - 1):
        fanin = layers[i-1]
        weightInit = 1/math.sqrt(fanin)
        cellPrev = layers[i-1]
        cellCurr = layers[i]
        r = np.random.uniform(-1*weightInit, weightInit, [cellPrev + 1, cellCurr + 1])
        self.init_weights.append(r)
    fanin = layers[i-1]
    weightInit = 1/math.sqrt(fanin)
    cellOutput = layers[i+1]
    cellPreOutput = layers[i]
    r = np.random.uniform(-1*weightInit, weightInit, [cellPreOutput + 1, cellOutput])
    self.init_weights.append(r)
    #print "\nWeights:", self.init_weights

def train(self, x, y):
    #print "\n----------Training----------"
    # add bias 1 to input layer
    bias = np.atleast_2d(np.ones(x.shape[0]))
    x = np.concatenate((bias.T, x), axis=1)
```

```
        self.weights = copy.deepcopy(self.init_weights)
        self.old_weights = copy.deepcopy(self.weights)

        for e in range(int(self.epoch) + 1):
            # if e % int(int(self.epoch)/13) == 0:
            #     #print " Epoch", e, "--", e*10 / int(int(self.epoch)/10), '%'
            #     sys.stdout.write('==')
            # if e == int(self.epoch):
            #     print "%d%s" % (e*10 / int(int(self.epoch)/10), '%')

            for i in range(int(x.shape[0])):
                y_all = [x[i]]
                v_all = [[]]
                # feedforward networks
                for l in range(len(self.weights)):
                    v_layer = np.dot(y_all[l], self.weights[l])
                    y_layer = self.activation(v_layer)
                    v_all.append(v_layer)
                    y_all.append(y_layer)

                # gradients at output layer
                error = y[i] - y_all[-1]
                gradients = [error * self.activation_derive(v_all[-1])]

                # gradients at hidden layer
                for l in range(len(y_all)-2, 0, -1):
                    gradients.append(self.activation_derive(v_all[l])*gradients[-
1].dot(self.weights[l].T))
```

```
                gradients.reverse()

                # set new weight for back propagation
                self.tmp_old_weights = copy.deepcopy(self.weights)
                for i in range(len(self.weights)):
                    layer = np.atleast_2d(y_all[i])
                    gradient = np.atleast_2d(gradients[i])
                    delta_weight = self.weights[i] - self.old_weights[i]
                    # print delta_weight
                    # print
                    #self.weights[i] += self.learning_rate * layer.T.dot(gradient)
                    self.weights[i] += self.momentum * delta_weight + self.learning_rate *
layer.T.dot(gradient)

                self.old_weights = copy.deepcopy(self.tmp_old_weights)

    def test(self, listX, listY, trainingFile):
        print "\n----------Testing----------"
        print "\nFeatures",
        if trainingFile == "cross.pat":
            print "\tOutput\t\t\t\tDesired class"
        else:
            print "\t\tOutput\t\t\tDesired class"
        EsumSqr = 0
        correct = 0
        i = 0
        np.set_printoptions(formatter={'float': '{: 0.10f}'.format})
        for x in listX:
            if trainingFile == "iris.pat":
```

```
            print " ".join('%0.1f' % f for f in x), "\t",
        elif trainingFile == "cross.pat":
            print " ".join('%0.4f' % f for f in x), "\t",
        else:
            print " ".join('%d' % f for f in x), "\t\t\t",

        # add bias 1 to input layer
        x = np.concatenate((np.ones(1), np.array(x)))
        for l in range(0, len(self.weights)):
            v = np.dot(x, self.weights[l])
            x = self.activation(v)
        error = listY[i] - x[-1]
        Esum = 0;
        if isinstance(error, np.float64):
            Esum += error**2
        else:
            for e in error:
                Esum += e**2
        EsumSqr += Esum/2
        #print "error",error
        desireY = listY[i]
        if trainingFile == "iris.pat":
            out = x*2+1
            desireY = listY[i]*2+1
            if (out < 1.8 and desireY == 1.0) or (out >= 1.8 and out < 2.8 and desireY == 2.0)
or (out >= 2.8 and desireY == 3.0):
                correct = correct+1
        elif trainingFile == "cross.pat":
            if x[0] > x[1]:
```

```
                out = np.array([1, 0])
            else:
                out = np.array([0, 1])
            if (out == listY[i]).all():
                correct = correct+1
            out = x
        else:
            if (x < 0.5 and listY[i] == 0) or (x >= 0.5 and listY[i] == 1):
                correct = correct+1
            out = x

        print out, "\t",
        if trainingFile == "iris.pat":
            print "%d" % (desireY)
        else:
            print desireY
        i = i+1
    Eav = EsumSqr/len(listX)
    print "\nError AV %.10f" % (Eav)
    print "Accuracy %.4f%s" % (correct/(len(listY)*1.0)*100.0, '%')
    return Eav

def sigmoid(x):
    return 1.0/(1.0 + np.exp(-x))

def sigmoid_derivertive(x):
    return sigmoid(x)*(1.0-sigmoid(x))

def tanh(x):
```

```
    return np.tanh(x)


def tanh_derivertive(x):
    return (1.0/np.cosh(x))**2


def main(argv):
    NNnameList = []
    learning_rate = 0.2
    momentum = 0.1
    epoch = 1000
    error = 0.001
    isCrossValid = 0
    activFunct = 'tanh'
    trainingFile = '-'
    try:
        opts, args = getopt.getopt(argv,"chN:a:n:m:e:E:t:")
        if len(sys.argv) == 1:
            usage()
            sys.exit(2)
    except getopt.GetoptError:
        usage()
        sys.exit(2)
    for opt, arg in opts:
        if opt == '-h':
            usage()
            sys.exit()
        elif opt in ("-N"):
            NNname = arg
            NNnameList = arg.split('-')
```

```
        NNnameList = map(int, NNnameList)
    elif opt in ("-a"):
        activFunct = arg
    elif opt in ("-n"):
        learning_rate = arg
    elif opt in ("-m"):
        momentum = arg
    elif opt in ("-e"):
        epoch = arg
    elif opt in ("-E"):
        error = arg
    elif opt in ("-c"):
        isCrossValid = 1
    elif opt in ("-t"):
        trainingFile = arg

print "\n----------Variable----------"
print 'Neural name\t', NNname
print 'Activation func\t', activFunct
print 'Learning rate\t', learning_rate
print 'Momentum\t', momentum
print 'Epoch\t\t', epoch
#print 'Min error\t', error
print 'TrainingFile\t', trainingFile

nn = NeuralNetwork(NNnameList, activFunct, learning_rate, momentum, epoch, error)

listX = []
listY = []
```

```
shuffleX = []
shuffleY = []
if trainingFile == "cross.pat":
    i = 1
    with open(trainingFile) as f:
    #with open("testcrs.pat") as f:
        for line in f:
            if i % 3 == 2:  # features
                tmp = line.split()
                tmp = map(float, tmp)
                listX.append(tmp)
            if i % 3 == 0:  # classes
                tmp = line.split()
                tmp = map(int, tmp)
                listY.append(tmp)
            i = i+1

        rdIndex = random.sample(range(len(listX)), len(listX))
        for i in rdIndex:
            shuffleX.append(listX[i])
            shuffleY.append(listY[i])
        inputX = np.array(shuffleX)
        outputY = np.array(shuffleY)
elif trainingFile == "iris.pat":
    i = 1
    with open(trainingFile) as f:
    #with open("testiris.pat") as f:
        for line in f:
            if i != 1:
```

```
            tmp = line.split()
            # if int(tmp[4]) == 1:
            #     listY.append([1, 0, 0])
            # elif int(tmp[4]) == 2:
            #     listY.append([0, 1, 0])
            # elif int(tmp[4]) == 3:
            #     listY.append([0, 0, 1])

            #   set range y to (0,1)
            listY.append((int(tmp[4])-1)/2.0)
            tmp.pop()
            tmp = map(float, tmp)
            listX.append(tmp)
        i = i+1

    rdIndex = random.sample(range(len(listX)), len(listX))
    for i in rdIndex:
        shuffleX.append(listX[i])
        shuffleY.append(listY[i])
    inputX = np.array(shuffleX)
    outputY = np.array(shuffleY)
else:
    inputX = np.array([[0, 0],
            [0, 1],
            [1, 0],
            [1, 1]])

    outputY = np.array([0, 1, 1, 0])
```

```
# print "\nInput X"
# print inputX
# print "\nDesire output"
# print outputY

if isCrossValid == 1:
    # 10% cross validation
    errorAV = 0.0
    for p in range(0, 10):
        print "\n\n---------- Cross validation block", p+1, "----------"
        block = int(round(len(listX)/10.0, 0))
        end = (p*block+block)-1
        if p == 9:
            end = len(listX)-1
        tmpTestListX = []
        tmpTestListY = []
        trainListX = copy.deepcopy(shuffleX)
        trainListY = copy.deepcopy(shuffleY)
        for i in range(end, p*block-1, -1):
            tmpTestListX.append(trainListX[i])
            tmpTestListY.append(trainListY[i])
            trainListX.pop(i)
            trainListY.pop(i)

        testDataX = np.array(tmpTestListX)
        testDataY = np.array(tmpTestListY)
        trainDataX = np.array(trainListX)
        trainDataY = np.array(trainListY)
        nn.train(trainDataX, trainDataY)
```

```
            errorAV = errorAV + nn.test(testDataX, testDataY, trainingFile)
            print "-------------------------------------------------"
        print "Error Average ", errorAV/10
    else:
        nn.train(inputX, outputY)
        nn.test(inputX, outputY, trainingFile)

if __name__ == "__main__":
    main(sys.argv[1:])
```