

Computer Assignment 1

จัดทำโดย

นายรัชพงศ์ ทอหุล 600610769

เสนอ

รศ.ดร.ศันสนีย์ เอื้อพันธุ์วิริยะกุล

รายงานนี้เป็นส่วนหนึ่งของวิชา

CPE 261456 (Introduction to Computational Intelligence)

ภาคเรียนที่ 1 ปีการศึกษา 2563

มหาวิทยาลัยเชียงใหม่

สารบัญ

สารบัญ	1
Method	2
การทำงานของระบบโดยรวม	2
Result	3
1. การทดลองเปลี่ยนจำนวนโหนดของ hidden layer	4
2. การทดลองเปลี่ยนค่า learning rate	6
3. การทดลองเปลี่ยนค่า momentum rate	8
ภาคผนวกโปรแกรม	

Method

การทำงานของระบบโดยรวม

- ระบบจะรับอินพุตดังนี้
 - อาร์เรย์แสดงจำนวนฮิดเดนเลเยอร์(hidden layer)และจำนวนโหนดของเน็ตเวิร์ค
 - Data_num เป็นเลขที่จะใช้กำหนดว่าจะเอาอินพุตมาจากไฟล์ flood.txt หรือ cross.pat
 - ค่าเลิร์นนิงเรท(Learning_rate)
 - ค่าโมเมนตัมเรท(momentum_rate)
 - ฟังก์ชันการกระตุ้น(activation)
 - จำนวนรอบการทำงาน(epoch)
- ทำการสุ่มค่า weight และค่า bias ในแต่ละโหนดใน neural network
- นำค่าอินพุตจากไฟล์ที่กำหนดมาผ่านการ normalization และ cross validation 10% เป็น train data และ test data
- นำ train data ที่ได้มาผ่านเข้า neural network โดยนำ input matrix ไปคูณกับ weight แต่ละตัวแล้วนำไปบวกกับ bias จากนั้นนำไปเข้า activation function แล้วนำไปเข้าเป็น input ของ layer ถัดไป กระบวนการนี้เรียกว่า feed forward
- จะได้ค่าผลลัพธ์สุดท้ายมาเราก็นำมาหาการคำนวณหาค่า error แล้วทำกระบวนการ back propagation เพื่อเปลี่ยนแปลงค่า weight และ bias ให้ได้ค่า error น้อยลงโดยจะมีขั้นตอนดังนี้
 - หาค่า error ของผลลัพธ์จากสมการ
$$e_j(t) = d_j(t) - y_j(t)$$
 - นำค่า error ที่ได้ไปหา gradient จากสมการ
$$\delta_j(t) = e_j(t)\phi_j'(v_j(t))$$
 - การหา gradient ใน hidden layer หาได้จากสมการ
$$\delta_j^{(l)}(t) = \phi_j^{(l)}(v_j^{(l)}(t)) \sum_k \delta_k^{(l+1)}(t) w_{kj}^{(l+1)}(t)$$
 - นำค่า gradient ที่ได้ไปปรับค่า weight จากสมการนี้
$$\Delta w_{ji}^{(l)}(t) = \alpha \Delta w_{ji}^{(l)}(t-1) + \eta \delta_j^{(l)}(t) y_i^{(l-1)}(t)$$
- ทำการวนลูปทำงานเรื่อย ๆ ตามค่า epoch โดย 1 epoch คือการทำงาน 1 รอบโดยรับ input จาก train data ครบทุกตัว 1 ครั้ง

Result

ในการทดลองนี้จะทำการทดลองเปลี่ยนแปลงค่าต่าง ๆ จาก input ทั้งสองไฟล์ทั้ง flood.txt และ cross.pat ดังนี้

1. ทดลองเปลี่ยนจำนวนโหนดของ hidden layer 10 ค่าโดยมีรายละเอียดดังนี้

- Hidden layer มี 2 ชั้นโดยจำนวนโหนดจะเปลี่ยนแปลงตามค่า i ที่อยู่ในช่วง 1-10 ตามสมการนี้

$$\text{layer} = [(2+(i*2)), (1+(i*2))]$$

- ค่า learning rate คงที่เท่ากับ 0.15
- ค่า momentum rate คงที่เท่ากับ 0.2
- Activation เป็น 'sigmoid'
- ค่า epoch คงที่เท่ากับ 1000

2. ทดลองเปลี่ยนค่า learning rate 10 ค่าโดยมีรายละเอียดดังนี้

- Hidden layer มี 2 ชั้นโดยจำนวนโหนดเท่ากับ [6,5]
- ค่า learning rate เปลี่ยนตามค่า i ที่อยู่ในช่วง 1-10 ตามสมการนี้

$$\text{learning_rate} = (0.01*i)+0.1$$

- ค่า momentum rate คงที่เท่ากับ 0.2
- Activation เป็น 'sigmoid'
- ค่า epoch คงที่เท่ากับ 1000

3. ทดลองเปลี่ยนค่า momentum rate 10 ค่าโดยมีรายละเอียดดังนี้

- Hidden layer มี 2 ชั้นโดยจำนวนโหนดเท่ากับ [6,5]
- ค่า learning rate คงที่เท่ากับ 0.15
- ค่า momentum rate เปลี่ยนตามค่า i ที่อยู่ในช่วง 1-10 ตามสมการนี้

$$\text{momentum_rate} = (0.01*i)+0.2$$

- Activation เป็น 'sigmoid'
- ค่า epoch คงที่เท่ากับ 1000

โดยได้ผลการทดลองดังนี้

1. การทดลองเปลี่ยนจำนวนโหนดของ hidden layer

มีการตั้งค่าตามรายละเอียดดังนี้

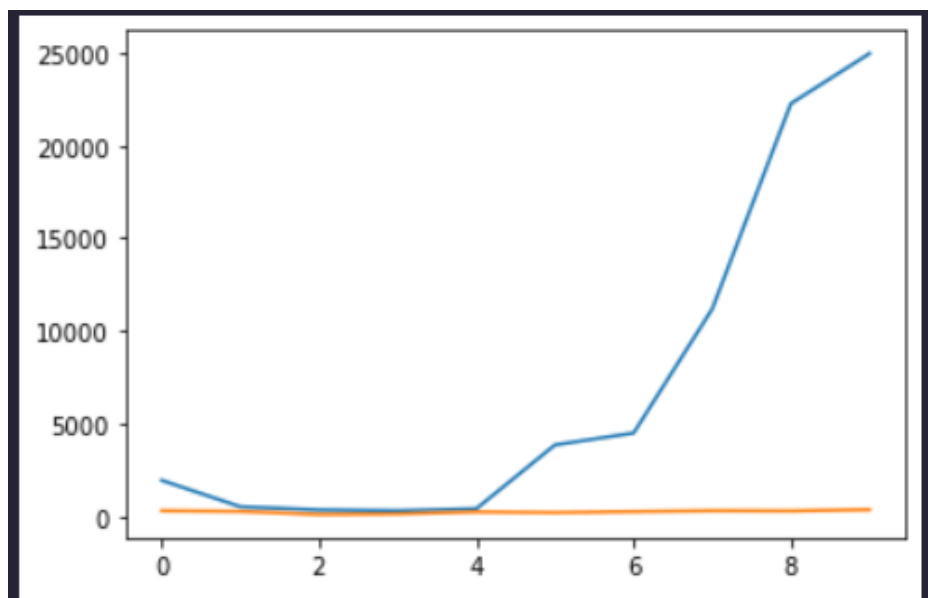
```
# Find best layer for Flood_dataset.txt
error_arr = np.zeros(10)
minn_arr = np.zeros(10)
for i in range(1,11):
    layer = [(2+(i*2)),(1+(i*2))]
    data_num = 1 # 0 = cross.pat , 1 = flood data set
    learning_rate = 0.15
    momentum_rate = 0.2
    activation = 'sigmoid'
    epoch = 1000
    error,minn = MLP(layer,learning_rate,momentum_rate,activation,epoch,data_num)
    error_arr[i-1] = error
    minn_arr[i-1] = minn

plt.plot(error_arr, Label="Error Avg")
plt.plot(minn_arr, Label="Min Error")
plt.show()
```

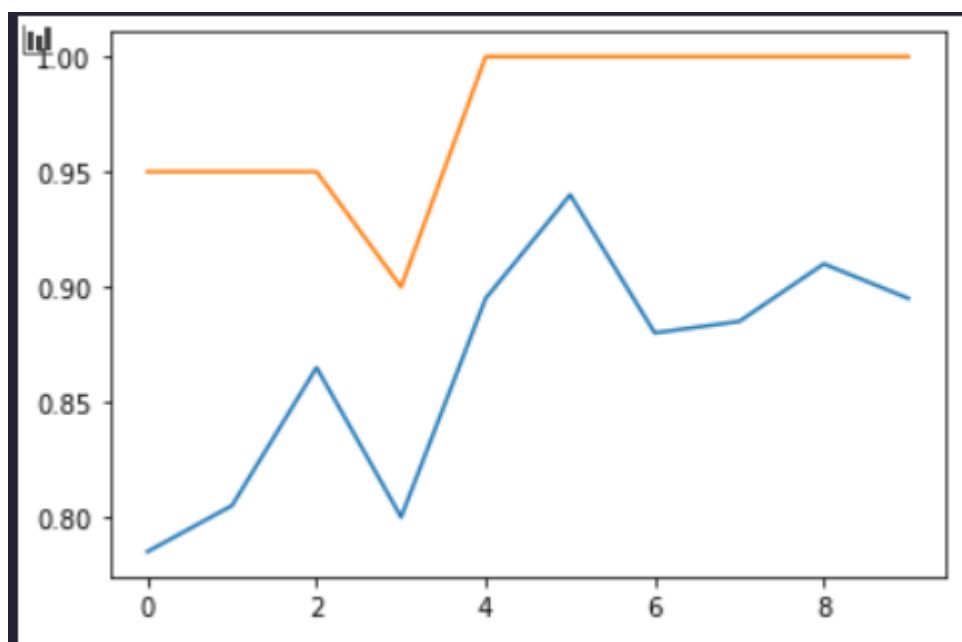
```
# Find best layer for cross.pat
error_arr = np.zeros(10)
minn_arr = np.zeros(10)
for i in range(1,11):
    layer = [(2+(i*2)),(1+(i*2))]
    data_num = 0 # 0 = cross.pat , 1 = flood data set
    learning_rate = 0.15
    momentum_rate = 0.2
    activation = 'sigmoid'
    epoch = 1000
    error,minn = MLP(layer,learning_rate,momentum_rate,activation,epoch,data_num)
    error_arr[i-1] = error
    minn_arr[i-1] = minn

plt.plot(error_arr, Label="Error Avg")
plt.plot(minn_arr, Label="Min Error")
plt.show()
```

โดยจะนำค่า sum square error และ minimum ของ sum square error มาแสดงเป็นกราฟดังนี้



กราฟแสดงค่า min error และ sum square error ของไฟล์ flood.txt



กราฟแสดงค่า min error และ sum square error ของไฟล์ cross.pat

โดยการทดลองพบว่าจำนวนโหนดของ hidden layer ที่เหมาะสมในช่วงที่ทำการทดลองในไฟล์ flood.txt อยู่ในช่วง $i = 1-4$ เพราะมีค่า error ทั้งสองแบบที่น้อยที่สุดในกราฟโดยค่า i แปลงเป็น

จำนวนโหนดของ hidden layer ได้เป็น [4,3] – [10,9] และในการทดลองในไฟล์ cross.pat ได้ค่า i ที่ทำให้ค่า error ทั้งสองแบบน้อยที่สุดเท่ากับ 3 แปลงเป็นจำนวนโหนดของ hidden layer ได้เป็น [8,7]

2. การทดลองเปลี่ยนค่า learning rate

มีการตั้งค่าตามรายละเอียดดังนี้

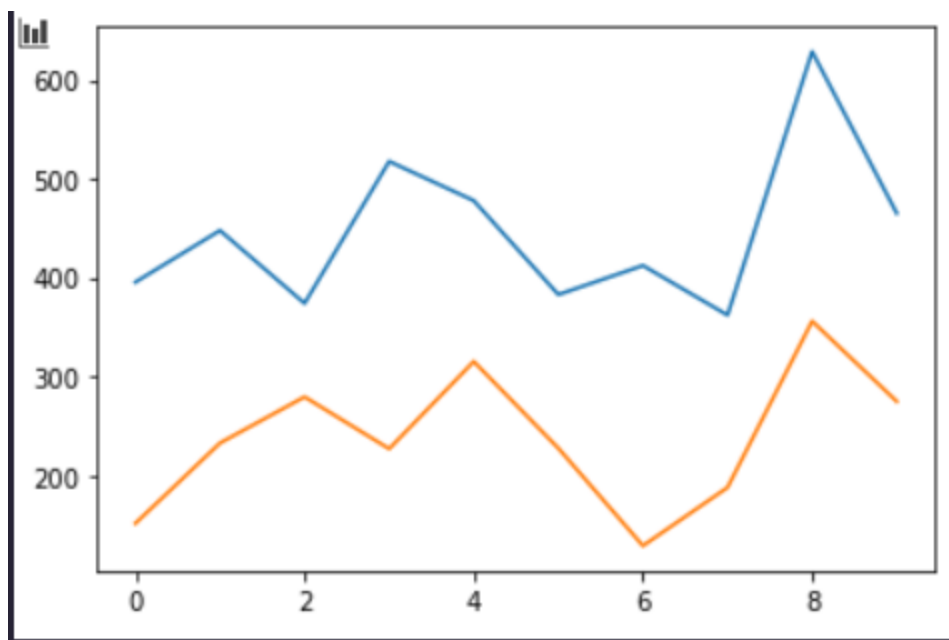
```
# Find best learning rate for Flood_dataset.txt
error_arr = np.zeros(10)
minn_arr = np.zeros(10)
for i in range(1,11):
    layer = [6,5]
    data_num = 1 # 0 = cross.pat , 1 = flood data set
    learning_rate = (0.01*i)+0.1
    momentum_rate = 0.2
    activation = 'sigmoid'
    epoch = 1000
    error,minn = MLP(layer,learning_rate,momentum_rate,activation,epoch,data_num)
    error_arr[i-1] = error
    minn_arr[i-1] = minn

plt.plot(error_arr, Label="Error Avg")
plt.plot(minn_arr, Label="Min Error")
plt.show()
```

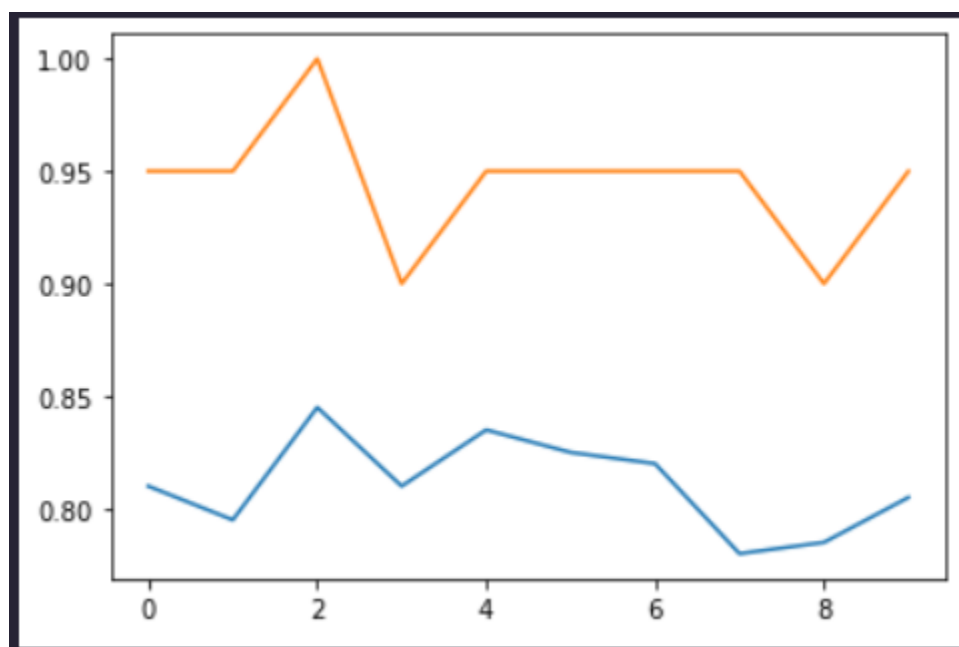
```
# Find best learning rate for cross.pat
error_arr = np.zeros(10)
minn_arr = np.zeros(10)
for i in range(1,11):
    layer = [6,5]
    data_num = 0 # 0 = cross.pat , 1 = flood data set
    learning_rate = (0.01*i)+0.1
    momentum_rate = 0.2
    activation = 'sigmoid'
    epoch = 1000
    error,minn = MLP(layer,learning_rate,momentum_rate,activation,epoch,data_num)
    error_arr[i-1] = error
    minn_arr[i-1] = minn

plt.plot(error_arr, Label="Error Avg")
plt.plot(minn_arr, Label="Min Error")
plt.show()
```

โดยจะนำค่า sum square error และ minimum ของ sum square error มาแสดงเป็นกราฟดังนี้



กราฟแสดงค่า min error และ sum square error ของไฟล์ flood.txt



กราฟแสดงค่า min error และ sum square error ของไฟล์ cross.pat

โดยจากการทดลองพบว่าค่า learning rate ที่เหมาะสมในช่วงที่ทำการทดลองในไฟล์ flood.txt อยู่ในช่วง $i = 6-7$ แต่ค่า 6 จะมี sum square error เยอะกว่าค่า 7 ค่า $i = 7$ จึงดีที่สุดจากความคิดของ

ผม โดยค่า i แปลงเป็นค่า learning rate ได้เป็น 0.17 และในการทดลองในไฟล์ cross.pat ได้ค่า i ที่เหมาะสม อยู่ในช่วง $i = 7-8$ แต่ค่า 7 จะมี sum square error เยอะกว่าค่า 8 ค่า $i = 8$ จึงดีที่สุดจากความคิดของผม โดยค่า i แปลงเป็นค่า learning rate ได้เป็น 0.18

3. การทดลองเปลี่ยนค่า momentum rate

มีการตั้งค่าตามรายละเอียดดังนี้

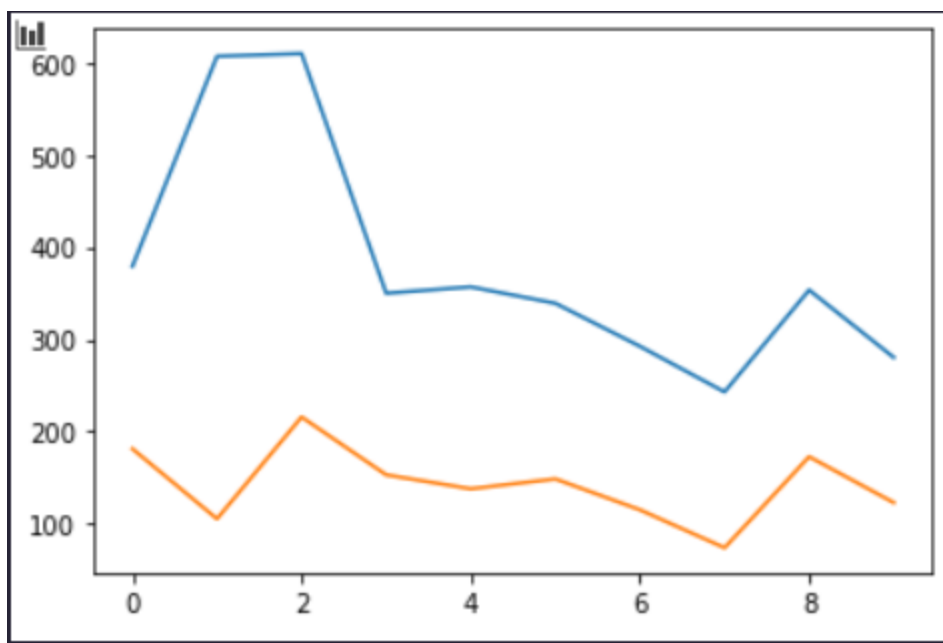
```
# Find best momentum rate for Flood_dataset.txt
error_arr = np.zeros(10)
minn_arr = np.zeros(10)
for i in range(1,11):
    layer = [6,5]
    data_num = 1 # 0 = cross.pat , 1 = flood data set
    learning_rate = 0.15
    momentum_rate = (0.01*i)+0.2
    activation = 'sigmoid'
    epoch = 1000
    error,minn = MLP(layer,learning_rate,momentum_rate,activation,epoch,data_num)
    error_arr[i-1] = error
    minn_arr[i-1] = minn

plt.plot(error_arr, Label="Error Avg")
plt.plot(minn_arr, Label="Min Error")
plt.show()
```

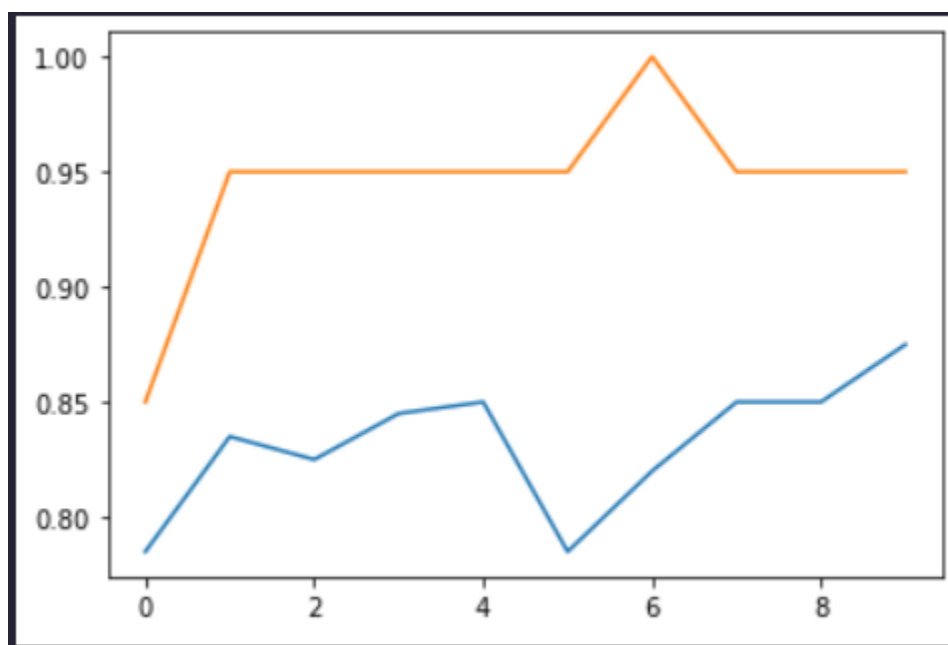
```
# Find best momentum rate for cross.pat
error_arr = np.zeros(10)
minn_arr = np.zeros(10)
for i in range(1,11):
    layer = [6,5]
    data_num = 0 # 0 = cross.pat , 1 = flood data set
    learning_rate = 0.15
    momentum_rate = (0.01*i)+0.2
    activation = 'sigmoid'
    epoch = 1000
    error,minn = MLP(layer,learning_rate,momentum_rate,activation,epoch,data_num)
    error_arr[i-1] = error
    minn_arr[i-1] = minn

plt.plot(error_arr, Label="Error Avg")
plt.plot(minn_arr, Label="Min Error")
plt.show()
```

โดยจะนำค่า sum square error และ minimum ของ sum square error มาแสดงเป็นกราฟดังนี้



กราฟแสดงค่า min error และ sum square error ของไฟล์ flood.txt



กราฟแสดงค่า min error และ sum square error ของไฟล์ cross.pat

โดยจากการทดลองพบว่าค่า momentum rate ที่เหมาะสมในช่วงที่ทำการทดลองในไฟล์ flood.txt คือค่า $i = 7$ เพราะได้ค่า error ทั้งสองแบบนี้ต่ำสุดในช่วงการทดลอง โดยค่า i แปลงเป็นค่า

momentum rate ได้เป็น 0.27 และในการทดลองในไฟล์ cross.pat ได้ค่า i ที่เหมาะสม เป็น $i = 1$ และ $i = 5$ แต่ค่า 5 จะมี sum square error เยอะกว่าค่า 1 ค่า $i = 1$ จึงดีที่สุดจากความคิดของผม โดยค่า i แปลงเป็นค่า learning rate ได้เป็น 0.21