Computational Intelligence Laboratory Exercise 2

Student Name: ChangXu
CCNU Student Number: 2019180034
UOW Student Number: 6643048

Central China Normal University Wollongong Joint Institute

1 Task One & Two

```
#include <iostream>
#include <fstream>
#include <vector>
#include <cstdio>
#include <cstdlib>
#include <cmath>

using namespace std;

const int DataRow=4;
const int DataColumn=4;
const double learning_rate=0.1;
extern double DataTable[DataRow][DataColumn];
double Theta=0.5;
extern double Weight[DataColumn-1];
const int iterator_n =20000;
const int batch_size = 5;

double DataTable[DataRow][DataColumn];
double Weight[DataColumn-1];
void Init()
{
    ifstream fin("data.txt");
    for(int i=0;i<DataRow;i++)
    {
        for(int j=0;j<DataColumn;j++)
        {
            fin>>DataTable[i][j];
        }
    }
    if(!fin)
    {
        cout<<"fin error";
        exit(1);
    }
    fin.close();
    for(int i=0;i<DataColumn-1;i++)
    {
         Weight[i]=0.0;
```

```
    }
}
void stochastic_gradient()
{
    for(int i=0;i<iterator_n;i++)
    {
        for(int n; n<batch_size; n++)
        {
            for(int j=0;j<DataRow;j++)
            {
                double hat_y,p=0;
                for(int k=0;k<DataColumn-1;k++)
                {
                    p+=DataTable[j][k]*Weight[k];
                }
                p=p-Theta;
                hat_y=1/(1+exp(-p));
                double delta=0;
                delta=learning_rate*(hat_y-DataTable[j][DataColumn-1])*hat_y*(1-hat_y);
                for(int k=0;k<DataColumn-1;k++)
                {
                    Weight[k]-=delta*DataTable[j][k];
                    Theta+=delta;
                }
            }

        }
    }
}
void printWeight()
{
    for(int i=0;i<DataColumn-1;i++)
        cout<<Weight[i]<<" ";
    cout<<endl;
    printf("theta= %f\n", Theta);
}

int main()
{
    Init();
    stochastic_gradient();
    printWeight();
    for(int i=0; i<DataRow; i++)
    {
        double y=0;
        for(int j=0; j<DataColumn-1; j++)
        {
            y += Weight[j]*DataTable[i][j];
        }
        double p=y-Theta;
        double hat_y=1/(1+exp(-p));
        printf("%lf ",hat_y);
```

```
    }
    return 0;
}
```

```
[Running] cd "c:\Users\72715\Desktop\UOW\第二学期
8.13471 -0.0619557 -0.876493
theta= 3.129479
0.017881 0.016825 0.984152 0.983156
[Done] exited with code=0 in 0.457 seconds
```

```
#include <iostream>
#include <fstream>
#include <vector>
#include <cstdio>
#include <cstdlib>
#include <cmath>

using namespace std;

const int DataRow=4;
const int DataColumn=4;
const double learning_rate=0.1;
extern double DataTable[DataRow][DataColumn];
double Theta=0.5;
extern double Weight[DataColumn-1];
const int iterator_n =20000;
const int batch_size = 2;
double delta=0;

double DataTable[DataRow][DataColumn];
double Weight[DataColumn-1];
void Init()
{
    ifstream fin("data.txt");
    for(int i=0;i<DataRow;i++)
    {
        for(int j=0;j<DataColumn;j++)
        {
            fin>>DataTable[i][j];
        }
    }
    if(!fin)
    {
        cout<<"fin error";
        exit(1);
    }
    fin.close();
    for(int i=0;i<DataColumn-1;i++)
```

```
    {
        Weight[i]=0.0;
    }
}
void stochastic_gradient()
{
    for(int i=0;i<iterator_n;i++)
    {

        for(int j=0;j<DataRow;j++)
        {
            for(int n=0; n<batch_size; n++)
            {
                double hat_y,p=0;
                for(int k=0;k<DataColumn-1;k++)
                {
                    p+=DataTable[j][k]*Weight[k];
                }
                p=p-Theta;
                hat_y=1/(1+exp(-p));
                delta +=learning_rate*(hat_y-DataTable[j][DataColumn-1])*hat_y*(1-h
            }
            delta = delta/batch_size;
            for(int k=0;k<DataColumn-1;k++)
            {
                Weight[k]-=delta*DataTable[j][k];
                Theta+=delta;
            }
        }
    }
}

void printWeight()
{
    for(int i=0;i<DataColumn-1;i++)
        cout<<Weight[i]<<" ";
    cout<<endl;
    printf("theta= %f\n", Theta);
}

int main()
{
    Init();
    stochastic_gradient();
    printWeight();
    for(int i=0; i<DataRow; i++)
    {
        double y=0;
        for(int j=0; j<DataColumn-1; j++)
        {
            y += Weight[j]*DataTable[i][j];
        }
```

```
        double p=y-Theta;
        double hat_y=1/(1+exp(-p));
        printf("%lf ",hat_y);
    }
    return 0;
}
```

```
[Running] cd "c:\Users\72715\Desktop\UOW\第二学期
7.89966 -0.075136 -0.849994
theta= 3.049983
0.019841 0.018431 0.982008 0.980632
[Done] exited with code=0 in 0.468 seconds
```