تمرین اول درس برنامه نویسی پیشرفته دکتر جهانشاهی – نیلا مسروری سعادت ۹۵۲۳۱۱۱

در این تمرین قصد این است که با داشتن یک سری داده ها از دانشجو ها بتوان شبکه عصبی را طراحی کرد که یک تابع خطی از این داده ها باشد و بتوان با ان نمرات دانشجو ها با دیتاهای جدید را تخمین زد. هدف در واقع یادگیری وزن های این تابع خطی می باشد .

```
std::vector <std::vector <double >> getData(const char*);
```

در ابتدا باید دیتاهای موجود در فایل اکسل را استخراج کنیم و داخل یک وکتور دو بعدی با ابعاد ۲۳۴در ۸ بریزیم.البته دیتاهای هر دانش اموز ۷ تا می باشد ولی ما برای یادگیری بهتر شبکه یک bias با اندازه ۱ در ابتدا اضافه می کنیم. با فراخوانی تابع getdata و دادن ادرس فایل اکسل، با استفاده از دستور getline هر خطی از فایل را خوانده و در یک متغیر string ذخیره می کنیم سپس باید ان را برای ذخیره کردن در یک وکتور دو بعدی به double تبدیل کنیم با دستور stod . برای یادگیری بهتر شبکه بهتر است که همه ی ۷ مشخصه بین ۰و۱ باشند بنابراین مشخصات ۳و۴ را تقسیم بر ۱۰۰ می کنیم.در نهایت باید وکتور اطلاعات را به صورت خروجی تابع return کنیم.

```
void displayDataset(std::vector <std::vector <double >>row)
```

در این تابع با گرفتن اطلاعات ذخیره شده در وکتو باید سعی کنیم انها را نمایش دهیم با استفاده از دو حلقه تو در تو همه ی اطلاعات را نمایش می دهیم.

نتیجه:

تمرین اول درس برنامه نویسی پیشرفته دکتر جهانشاهی – نیلا مسروری سعادت ۹۵۲۳۱۱۱

| bias | class | ta | coding | studying | background | mind | grade |
|---|---|---|---|---|---|---|---|
| 1 | 0.42 | 0.83 | 0.13 | 0.28 | 0.76 | 0.48 | 14.23 |
| 1 | 0.47 | 0.54 | 0.68 | 0.4 | 0.39 | 0.4 | 15.76 |
| 1 | 0.7 | 0.06 | 0.48 | 0.01 | 0.28 | 0.29 | 9.99 |
| 1 | 1 | 0.07 | 0.35 | 0.04 | 0.73 | 0.49 | 13.39 |
| 1 | 0.03 | 0.76 | 0.69 | 0.22 | 0.87 | 0.12 | 11.26 |
| 1 | 0.72 | 0.87 | 0.26 | 0.07 | 0.25 | 0.06 | 12.74 |
| 1 | 0.13 | 0.77 | 0.26 | 0.2 | 0.32 | 0.46 | 10.11 |
| 1 | 0.72 | 0.92 | 0.21 | 0.07 | 0.48 | 0.94 | 14.7 |
| 1 | 0.34 | 0.43 | 0.46 | 0.23 | 0.41 | 0.95 | 13.34 |
| 1 | 0.38 | 0.4 | 0.58 | 0.25 | 0.28 | 0.59 | 12.92 |
| 1 | 0.93 | 0.47 | 0.35 | 0.33 | 0.72 | 0.01 | 17.72 |
| 1 | 0.91 | 0 | 0.6 | 0.1 | 0.25 | 0.62 | 12.81 |
| 1 | 0.67 | 0.67 | 0.52 | 0.32 | 0.27 | 0.09 | 16.03 |
| 1 | 0.64 | 0.9 | 0.13 | 0.18 | 0.85 | 0.64 | 14.98 |
| 1 | 0.03 | 0.12 | 0.06 | 0.39 | 0.95 | 0.65 | 9.51 |
| 1 | 0.71 | 0.8 | 0.12 | 0.35 | 0.38 | 0.85 | 17.95 |
| 1 | 0.04 | 0.63 | 0.34 | 0.03 | 0.61 | 0.23 | 7.83 |
| 1 | 0.96 | 0.79 | 0.59 | 0.37 | 0.11 | 0.55 | 20 |
| 1 | 0.09 | 0.25 | 0.23 | 0.32 | 0.72 | 0.49 | 8.91 |
| 1 | 0.09 | 0.48 | 0.33 | 0.37 | 0.14 | 0.44 | 10.27 |
| 1 | 0.98 | 0.59 | 0.31 | 0.36 | 0.34 | 0.26 | 18.62 |
| 1 | 0.68 | 0.22 | 0.56 | 0.13 | 0.47 | 0.45 | 13.31 |
| 1 | 0.2 | 0.09 | 0.54 | 0.2 | 0.8 | 0.04 | 8.1 |
| 1 | 0.52 | 0.18 | 0.37 | 0.3 | 0.89 | 0.06 | 12.34 |
| 1 | 0.88 | 0.82 | 0.55 | 0.24 | 0.81 | 0.22 | 19.48 |
| 1 | 0.2 | 0.05 | 0.5 | 0.22 | 0.9 | 0.59 | 9.27 |

```
double grade(std::vector <double > w, std::vector <double > x)
```

در این تابع با گرفتن وکتور دو بعدی اطلاعات و وکتور وزن ها یک تابع خطی به‌گونه ای پیاده سازی می شود که نمرات دانشجو ها تخمین زده شود در این جا نباید المان اخر اطلاعات هر دانشجوکه نمره اش هست را در نظر بگیریم. تابع به شکل زیر میباشدکه هر المان در وزن مربوطه ضرب می شود.

$$grade(\mathbf{x}) = x_0 w_0 + x_1 w_1 + \ldots + x_6 w_6$$

نتیجه تست:

```
std::vector <double > w(7,1);
std::vector <double > student{1,2,3,4,5,6,7};
```

با وزن های ۱ برابر28 شد.

```
double J(std::vector<double>w, std::vector<std::vector<double>>data)
```

تمرین اول درس برنامه نویسی پیشرفته دکتر جهانشاهی – نیلا مسروری سعادت ۹۵۲۳۱۱۱

برای محاسبه تابع هزینه باید مطابق شکل زیر، در ابتدا اختلاف نمره ی تخمین زده شده توسط تابع grade در مرحله قبل و نمره ی واقعی هر دانشجو را بدست اورده و برای این کار باید وزن ها و وکتور کلیه ی اطلاعات را به صورت ورودی دریافت کنیم. نمره ی واقعی هر دانشجو در واقع ستون هفتم از هر ردیف وکتور دیتا می باشد.

$$J(\mathbf{w}) = \frac{1}{2m} \sum_{i=1}^{m} (grade(\mathbf{x}^{(i)}) - y^{(i)})^2$$

نتیجه:

تست این تابع هزینه با وزن های ۱ عدد 47.8298 را نتیجه می دهد.

```
std::vector<double> train(std::vector<std::vector<double>> data, std::vector
<double> w, double alpha, size_t max_iters, double min_cost, size_t verbose)
```

تابع اصلی اموزش همین train می باشدکه در این مرحله هرکدام از سطرها اعمال می شوند و نتیجه ها با هم جمع زده می شوند. در واقع همه ی سطر های بردار X به تابع grade میره اما در هر لحظه یه سطر، یعنی ۷ تا عدد به تابع grade میره، از مقدار نمره (یعنی ستون آخر) کم میشه و در ستون i ام اون سطر ضرب میشه. این عمل برای تمام سطر ها انجام و با هم جمع میشود. برای این کار به دو تا حلقه تو در تو for نیاز می باشد و بعد از انجام محاسبات بالا در حلقه داخلی نتایج داخل وکتور یک بعدی sum ذخیره می شوند.در بیرون این دو حلقه به اپدیت وزن ها پرداخته می شود. و بعد از هر اپدیت باید وکتور sum را صفرکنیم تا در هر iteration با مقادیر قبلی جمع نشود.برای اینکه مقادیر بهتری از وزن ها محاسبه شوند بهتر است انها را در چندین مرحله محاسبه کنیم به تعداد iteration که به عنوان ورودی تابع گرفتیم برای این کار بعد از هر بار محاسبه وزن ها، به یک شمارنده یک مقدار اضافه می کنیم وکل این کارها را در whileای انجام می دهیم که این شمارنده را با iteration چک می کند . یک شرط دیگر هم نیاز است برای اینکه اگر زودتر از تعداد iteration ها تابع هزینه مان برابر مقدار دلخواه شد، دیگر به اپدیت وزن ها نپردازد.همچنین همان طورکه در صورت سوال توضیح داده شده است در صورت true بودنverbose ،لازم است مقادیر قبل و بعد تابع هزینه چاپ شود.محاسبات اصلی این تابع به صورت شکل زیر می باشد.

تمرین اول درس برنامه نویسی پیشرفته دکتر جهانشاهی – نیلا مسروری سعادت ۹۵۲۳۱۱۱

$$\frac{\partial J}{\partial w_i} = \frac{\partial J}{\partial grade}\frac{\partial grade}{\partial w_i} = \frac{1}{m}\sum_{j=1}^{m}(grade(\mathbf{x}^{(j)}) - y^{(j)})x_i^{(j)}$$

$$w_i := w_i - \alpha\frac{\partial J}{\partial w_i}$$

نتیجه تست زیر:

```
//testing the train function
w = train(data, w, 0.01, 500, 0.01, false);
    std::cout<<"Weights...\n";
    for(size_t i{}; i<w.size(); std::cout << w[i++] << " ,");
    std::cout<<"\n";
```

به صورت مقابل:

```
Weights...
4.92714 ,4.61845 ,3.73029 ,2.94849 ,2.45714 ,2.46021 ,2.37687 ,
```

```
void displayOutput(std::vector<std::vector<double>>data, std::vector<
double>w)
```

در این تابع وزن های اپدیت شده و دیتاهای دانشجوها به عنوان ورودی گرفته شده و لازم است که نمره های اصلی دانشجوها و نمره های تخمین زده شده توسط تابع grade و وزن های جدید در خروجی چاپ شوند.برای تمیزی و زیبایی خروجی از دستور setw استفاده می شود.

نتیجه:

| NO | Real Grade | Estimated Grade |
|---|---|---|
| 1 | 14.23 | 14.045 |
| 2 | 15.76 | 14.0102 |
| 3 | 9.99 | 11.2019 |
| 4 | 13.39 | 13.8976 |
| 5 | 11.26 | 12.9013 |
| 6 | 12.74 | 13.194 |
| 7 | 10.11 | 11.5385 |
| 8 | 14.7 | 15.8906 |
| 9 | 13.34 | 13.2896 |
| 10 | 12.92 | 12.5899 |
| 11 | 17.72 | 14.6135 |
| 12 | 12.81 | 13.2334 |
| 13 | 16.03 | 13.7185 |
| 14 | 14.98 | 15.6782 |
| 15 | 9.51 | 10.5307 |
| 16 | 17.95 | 15.3595 |
| 17 | 7.83 | 10.5856 |
| 18 | 20 | 16.5344 |
| 19 | 8.91 | 10.6758 |
| 20 | 10.27 | 10.4057 |
| 21 | 18.62 | 14.9072 |
| 22 | 13.31 | 13.0848 |
| 23 | 8.1 | 10.3334 |
| 24 | 12.34 | 12.1605 |
| 25 | 19.48 | 16.7773 |
| 26 | 9.27 | 11.6687 |

```
void save(std::vector<double> w, const char* name)
```

در این تابع وزن های اپدیت شده را در یک فایل اکسل ذخیره می کنیم با کمک دستور ofstream .

برای مثال نتیجه ذخیره در فایل weights.csv به صورت زیر:

```
//saving the weights
save(w,"weights.csv");
```

با وزن هایی با مقادیر اولیه 0 نتیجه زیر را می دهد در فایل اکسل:

| | | | | | | |
|---|---|---|---|---|---|---|
| 5.48528 | 4.46104 | 3.60295 | 2.61499 | 1.82324 | 2.32825 | 2.19782 |
| | | | | | | |
| | | | | | | |

```
std::vector<double> load(const char*name)
```

تمرین اول درس برنامه نویسی پیشرفته دکتر جهانشاهی – نیلا مسروری سعادت ۹۵۲۳۱۱۱

در این مرحله با کمک دستوراتی که برای خوندن دیتاها در تابع getdata استفاده کردیم با کمک دستورات ifstream و getline وزن های ذخیره شده در مرحله ی قبل را خوانده و به صورت یک وکتور تک بعدی باز می گردانیم .

به صورت زیر تست کردم :

```
//loading the weights
 std::vector <double > w2(7,0);
w2=load("weights.csv");
//test the loading
std::cout<<"Weights...\n";
    for(size_t i{}; i<w2.size(); std::cout << w2[i++] << " ,");
    std::cout<<"\n";
```

نتیجه:

```
Weights...
5.48528 ,4.46104 ,3.60295 ,2.61499 ,1.82324 ,2.32825 ,2.19782 ,
```

```
double predict(std::vector<double>newstu, std::vector<double>w)
```

در مرحله ی اخر فقط لازمه که اطلاعات یک دانشجو جدید را ازکاربرگرفته و با استفاده از وزن های اپدیت شده جدید نمره دانشجو را تخمین بزنیم با استفاده از gradeاطلاعات دانشجوی جدید را در قسمت main ازکاربرگرفته و در یک وکتور یک بعدی ذخیره می کنیم وبه عنوان ورودی به تابع می دهیم و نمره تخمین زده شده را به کاربر می دهیم .

در صورتی که بخواهیم نمره ها برای همه دانشجوها چه درسخون و چه تنبل در همان محدوده ی ۰و۲۰ بماند باید خروجی تابع را بین ۰و۲۰ scale کنیم یعنی ببینم بیشترین وکمترین حد نمره ی ممکن چه مقدار است و بعد به محدوده ۰و۲۰ map کنیم.

نتیجه:

تمرین اول درس برنامه نویسی پیشرفته دکتر جهانشاهی – نیلا مسروری سعادت ۹۵۲۳۱۱۱

```
enter the students data:
class:.42
ta:.83
coding:13
studying:28
background:.76
mind:.48

the estimated grade is: 14.0242
```