بسمه تعالی

دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده برق و کامپیوتر

گزارش تمرین شماره 1

درس سیستم های هوشمند

پاییز 1401

امیر حسین بیرژندی

...

810198367

...

سوال 1 – رگرسیون لاجستیک باینری

الف) محاسبه‌ی گرادیان(تحلیلی)

$$J(w,b) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + exp\left(-y_i\left(b + x_i^T w\right)\right)\right)$$

$$\nabla_w J = \frac{1}{n}\sum_{i=1}^{n} \frac{-y_i x_i^T exp\left(-y_i\left(b+x_i^T w\right)\right)}{1+exp\left(-y_i\left(b+x_i^T w\right)\right)} = \frac{1}{n}\sum_{i=1}^{n} -y_i x_i^T (1-\mu_i)$$

$$\nabla_b J = \frac{1}{n}\sum_{i=1}^{n} \frac{-y_i x_i^T exp\left(-y_i\left(b+x_i^T w\right)\right)}{1+exp\left(-y_i\left(b+x_i^T w\right)\right)} = \frac{1}{n}\sum_{i=1}^{n} -y_i (1-\mu_i)$$

ب) محاسبه‌ی گرادیان نزولی

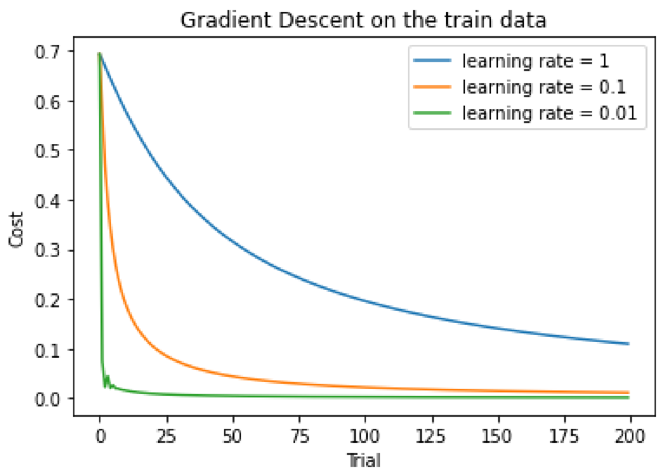

Figure 1 – Gradient decsent on the train data

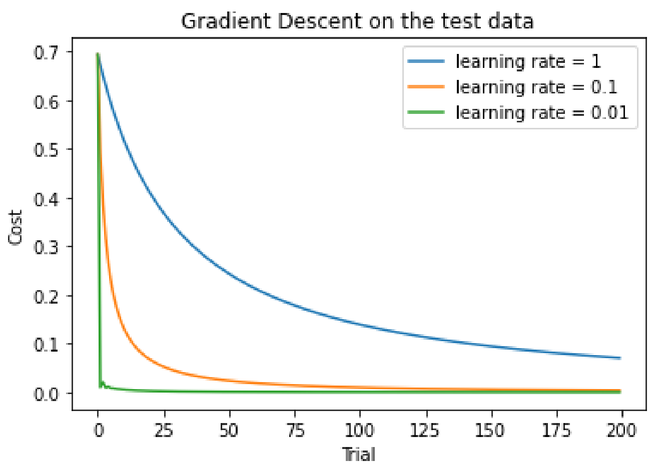

Figure 2 – Gradient decsent on the test data

ج) محاسبه دقت

w و b بدست آمده در نقطه همگرا شده را در $sign(b + x_i^T w)$ جایگذاری کرده و با مقایسه آن با لیبل های حقیقی دقت را محاسبه می‌کنیم.

```
correct prediction percentage of the train data for learning rate = 1 : 0.8866868733970622
correct prediction percentage of the train data for learning rate = 0.1: 0.8782933084635113
correct prediction percentage of the train data for learning rate = 0.01: 0.8689671252040102
```

```
correct prediction percentage of the test data for learning rate = 1 : 0.9883313885647608
correct prediction percentage of the test data for learning rate = 0.1 : 0.9836639439906651
correct prediction percentage of the test data for learning rate = 0.01 : 0.9731621936989499
```

گرادیان نزولی تصادفی

بسته 1 تایی:

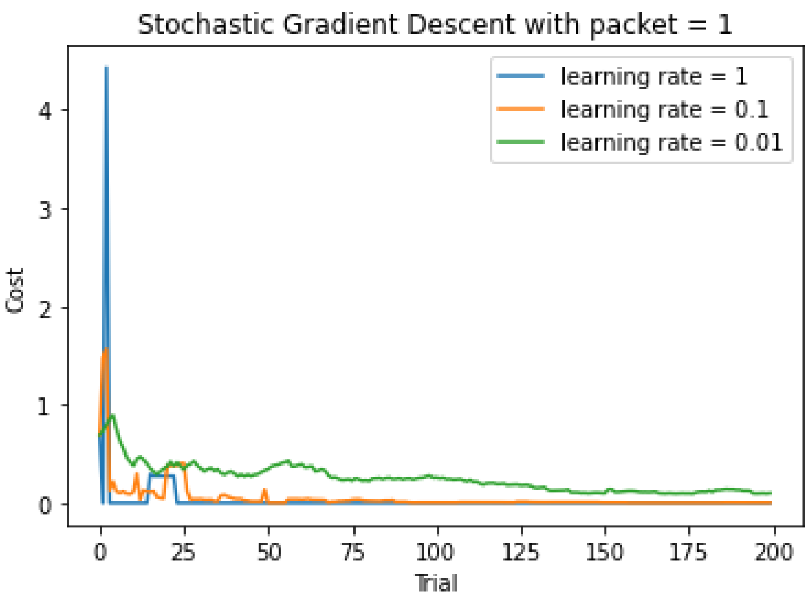

Figure 3 –Stochastic Gradient decsent with packet = 1

```
correct prediction percentage of the test data for learning rate = 1 : 0.9544924154025671
correct prediction percentage of the test data for learning rate = 0.1: 0.969661610268378
correct prediction percentage of the test data for learning rate = 0.01:  0.969661610268378
```

بسته 100 تایی:

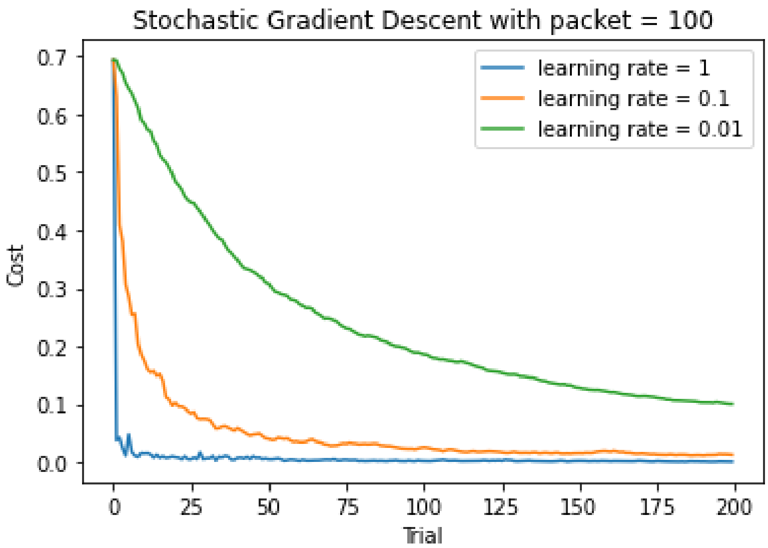

Figure 4 –Stochastic Gradient decsent with packet = 100

```
correct prediction percentage of the test data for learning rate = 1 : 0.9883313885647608
correct prediction percentage of the test data for learning rate = 0.1 : 0.9848308051341891
correct prediction percentage of the test data for learning rate = 0.01 : 0.9754959159859977
```

سوال 2 – بهینه سازی در توابع غیر محدب

الف) روش نیوتن تحلیلی

$$x^{(k+1)} = x^{(k)} - \frac{f'(x^{(k)})}{f''(x^{(k)})}$$

$$f(\underline{x}) = 2x_1^2 + 2x_2^2 - 17x_2\cos(0.2\pi x_1) - x_1x_2$$

$$\nabla \underline{x} f = (4x_1 + 3.4\pi x_2\sin(0.2\pi x_1) - x_2\,, 4x_2 - 17\cos(0.2\pi x_1) - x_1)$$

$$\mathrm{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix} = \begin{bmatrix} 4 + 0.68x_2\pi^2\cos(0.2\pi x_1) & 3.4\pi\sin(0.2\pi x_1) - 1 \\ 3.4\pi\sin(0.2\pi x_1) - 1 & 4 \end{bmatrix}$$

$$H^{-1} = \frac{1}{15}\begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix}$$

$$x^{(k+1)} = x^{(k)} - H^{-1}\nabla \underline{x} f$$

$$x^{k+1} = (0,0) - \frac{1}{15}\begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix}\begin{bmatrix} 0 \\ -17 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \frac{17}{15} \\ \frac{68}{15} \end{bmatrix} = \begin{bmatrix} 1.13 \\ 4.53 \end{bmatrix}$$

$$x^{(k+1)} = \begin{bmatrix} 1.13 \\ 4.53 \end{bmatrix}$$

ب) روش نیوتن (شبیه سازی)

ابتدا از نقطه (1,3) شروع کرده و با پیاده‌سازی روش نیوتن با 1000 ترایال نقطه کمینه را بدست می‌آوریم.

```
The point which minimizes the function is:  [[0.13087466]
 [4.26835765]]
The Minimum is:  -36.40349774185023
```

حال به ازای 0<x1<10 و -5<x2<5 کمینه تابع را بدست می‌آوریم و با مقایسه فاصله آن با -36.4 آن را به سه دسته نزدیک، دور و دورتر تقسیم می‌کنیم.

نحوه مرزبندی با psuedo code زیر نشان داده است.

```
if(final_value[count] + 36.4 < 10 ):
    group = “ close “
if((10 < final_value[count] + 36.4) & (final_value[count] + 36.4 < 50)):
    group = “ far “
if(50 < final_value[count] + 36.4):
    group = “farther “
```

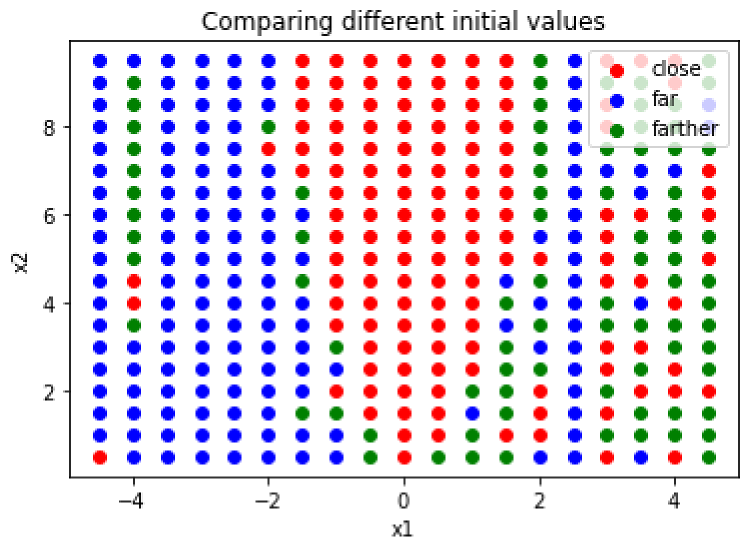

Figure 5

با توجه به نتیجه بدست آمده در نمودار بالا مشاهده می‌کنیم در نقاطی که نزدیک $\begin{bmatrix}0.13\\4.29\end{bmatrix}$ هستند همگی به نزدیکای نقطه کمینه همگرا شده اند و هر چه از این ناحیه دورتر می‌شویم احتمال گیر افتادن در کمینه-های محلی بیشتر شده است.

ج) روش فراابتکاری

در این بخش روش ژنتیک پیاده‌سازی شد. روش ژنتیک سه مرحله دارد که متشکل از selection، crossover و mutation می‌باشد که این سه بخش در فایل ژوپیتر قابل رویت می‌باشد.

ابتدا به بررسی پارامتر های این مسئله می‌پردازیم. این مسئله 5 پارامتر دارد که در مورد هر کدام جداگانه بررسی خواهیم کرد.

1) **جمعیت** : در ابتدا باید مشخص کنیم که قرار است که چه تعداد از افراد حاضر در جامعه را انتخاب کنیم. مشخصا اگر این تعداد کم باشد نیاز به تعداد ایتریشن های بسیاری برای همگرایی داریم و ممکن است که اگر تعداد ایتریشن کم باشد با این جمعیت همگرا نشویم. از آن طرف اگر جمعیت بسیار بالایی انتخاب شوند هزینه یادگیری بسیار بالا خواهد بود.

2) **ایتریشن** : تعداد دفعه‌ای که سه مرحله ذکر شده تکرار شود با پارامتر ایتریشن مشخص می‌شود هر چه این مقدار بیشتر باشد احتمال همگرایی بسیار بالاتر ولی در عوض هزینه یادگیری بیشتر خواهد بود.

3) **تعداد منتخب برای mutation** : این تعداد مشخص می‌کند که بر روی چه تعداد از جمعیت قرار است نویز وارد شود (mutation). مرحله mutation در مقایسه با دو مرحله دیگر تاثیر کمتری دارد و اگر این پارامتر حتی برابر صفر باشد می‌توانیم همگرا بشویم. اما باید دقت کرد اگر این تعداد زیاد باشد هزینه یادگیری زیاد می‌شود.

4) **تعداد بیت** : این تعداد مشخص می‌کند که چه تعداد از 8 بیت را دچار تغییرات کنیم. همچنان هم اگر این تعداد زیاد باشد و هم کم احتمال همگرایی بسیاری وجود دارد.

5) **احتمال mutation** : این تعداد مشخص می‌کند که با چه احتمالی مرحله mutation انجام شود اگر این احتمال خیلی بالا باشد و در کنار آن دو پارامتر بالایی نیز در بیشترین مقدار خود باشند ممکن است که خیلی دیر همگرا شویم.

```
argmin x1:  0.11811023622047244
argmin x2:  4.251968503937007
The minimum of this function is:  -36.40034296457406
```

سوال 3 – ماشین بردار پشتیبان

الف) تحلیلی

-1

اگر مقدار ξ برابر صفر باشد مینیمم $||w||^2$ برابر ماکسیمم $(0,1-y_i(w^Tx_i+b))$ خواهد بود که چون مینیمم است برابر مقدار صفر خواهد بود. حال در این صورت عبارت $1-y_i(w^Tx_i+b)$ منفی بوده و سپس می‌توان گفت $y_i(w^Tx_i+b)$ بزرگتر از 1 است و به عبارتی با توجه به y_i که یا یک است و یا منفی یک (w^Tx_i+b) یا بزرگتر از یک است و یا کوچکتر از منفی یک خواهد بود که می‌توانیم نتیجه بگیریم داده i در طرفین دو بردار پشتیبان قرار دارد.

-2

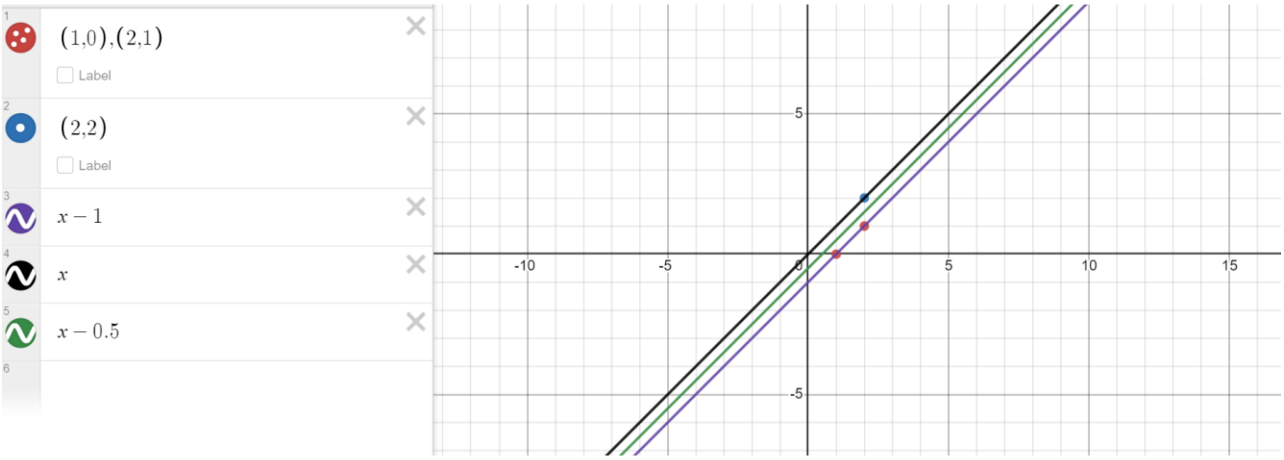

Figure 6 –Support vector machines

برای انتخاب مرز تصمیم بهینه باید به گونه ای عمل کنیم که margin ما بیشترین مقدار ممکن را داشته باشد تا احتمال رخداد خطا در تصمیم‌گیری های آتی کمینه شود. حال با توجه به بردار های پشتیبان ما که دو کلاس مثبت ما بر روی خط y = x-1 قرار دارند برای بیشینه‌سازی margin نقطه کلاس منفی را نیز روی خط y = x در نظر می‌گیریم. حال بهترین تصمیم برای مرز بهینه خطی است که از دو بردار پشتیبان به یک فاصله باشد در واقع بین دو خط y = x-1 و y = x است که به سادگی می‌توان نشان داد خط مورد نظر y = x – 0.5 است.

ب) شبیه سازی

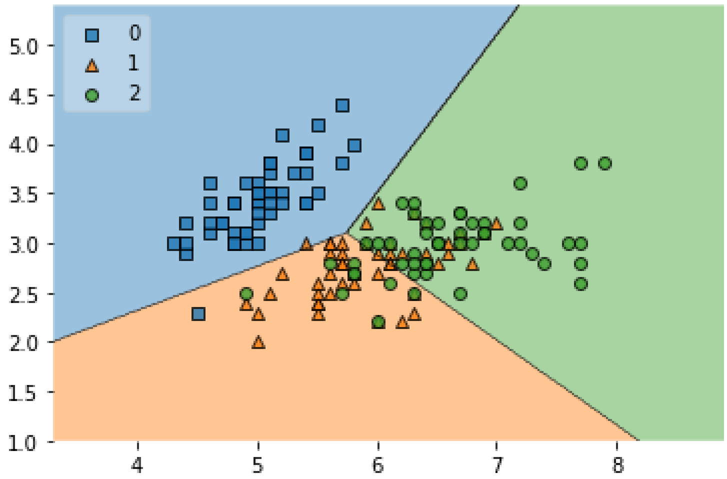

Figure 7 –visualizing the data after one vs rest

```
print("Accuracy:",metrics.accuracy_score(label, yhat))
```
✓ 0.3s

```
Accuracy: 0.7866666666666666
```

```
confusion_matrix(label, yhat)
```
✓ 0.3s

```
array([[49,  1,  0],
       [ 1, 28, 21],
       [ 0,  9, 41]], dtype=int64)
```

```
confidence_matrix
```
✓ 0.3s

```
array([[0.98      , 0.02631579, 0.        ],
       [0.02      , 0.73684211, 0.33870968],
       [0.        , 0.23684211, 0.66129032]])
```