

توضیحات پیاده سازی سوال پنجم:

(الف)

برنامه نوشته شده شامل موارد زیر است:

۱. کلاس activation function:

این کلاس برای ما توابع مختلف activation را فراهم میکند به همراه مشتقات آنها که برای آموزش شبکه عصبی به کار میرود مخصوصا برای محاسبات گرادیان.

توابع سیگموئید به همراه مشتق آنها که چون خیلی خوب بین مقادیر ۰ و ۱ عمل میکنند برای لایه آخر در نظر گرفته شده اند.

توابع RELU به همراه مشتق آنها از اونجایی که مقادیر منفی را صفر میکند و در لایه های میانی مورد استفاده قرار میگیرد.

ضمنا برنامه قابلیت این را دارد که بتوانیم نرخ یادگیری و توابع activation و تعداد لایه ها را تغییر بدهیم.

تابع خطی به همراه مشتق آنها نیز به برنامه اضافه شده است.

۲. کلاس لایه:

این کلاس برای تولید یک لایه در نظر گرفته شده است.

در تابع init میتوانیم به وزن ها یک مقدار تصادفی اتلاق کنیم و توابع activation را برای هر لایه انتخاب کنیم.

weight and biases که دو متغیر آزاد ما هستند.

و در activation که مقدار تابع activation را برای ما ذخیره میکند.

۳. multilayerPerceptron class:

در این کلاس ما مدل خودمان میسازیم و هسته اصلی برنامه ما هست جایی که لایه های مختلف از شبکه ساخته میشوند.

تابع `init` شبکه را به همراه تعداد لایه و تابع `activation` را میسازد.

تابع `forward propagation` که وظیفه الگوریتم پس خور را به عهده دارد که ورودی را از طریق شبکه با جمع وزن دار را محاسبه میکند و تابع فعالیت را نسبت به هر لایه اعمال میکند.

تابع `backward propagation` که وظیفه الگوریتم پس انتشار را به عهده دارد. که تابع `loss` را محاسبه میکند و محاسبات گرادیان و پیاده سازی گرادیان کاهشی در اینجا انجام میشود.

تابع `train` مدل ما را بر اساس سائز دسته ها و `epoch` و ورودی آموزش میدهد و از `mini batch` استفاده میکند.

`Predict`: خروجی را بر اساس الگوریتم `forward propagation` محاسبه میکند.

از ۲ تابع کمکی هم استفاده کردیم:

۱. `preprocess` داده که داده ها را آماده ورود به مدل میکند.

۲. `Train` که مدل شبکه را بر اساس داده ورودی داده شده آموزش میدهد.

جریان برنامه به این صورت هست که داده پیش پردازش میشود، مدل تنظیم میشود، یادگیری انجام میشود، یادگیری ارزیابی میشود و خروجی که همان `accuracy` تابع ما هست تولید میشود.

ب) تابع سیگموید پیاده سازی شده است.

پ) امکان تغییری در تعداد نوروں ها و لایه و نرخ یادگیری وجود دارد.

خروجی بر اساس موارد خواسته شده:

(الف)

وقتی تمام تابع activation خطی باشد مقدار خروجی برنامه:

```
Epoch 10/100, Loss: 0.3971
Epoch 20/100, Loss: 0.3915
Epoch 30/100, Loss: 0.3950
Epoch 40/100, Loss: 0.3974
Epoch 50/100, Loss: 0.3986
Epoch 60/100, Loss: 0.3991
Epoch 70/100, Loss: 0.3993
Epoch 80/100, Loss: 0.3993
Epoch 90/100, Loss: 0.3993
Epoch 100/100, Loss: 0.3993

Training Accuracy: 0.8815
Test Accuracy: 0.8587
```

(ب)

نتایج وقتی از تابع سیگموید نیز استفاده میکنیم:

```
Epoch 10/100, Loss: 0.4396
Epoch 20/100, Loss: 0.3794
Epoch 30/100, Loss: 0.3649
Epoch 40/100, Loss: 0.3540
Epoch 50/100, Loss: 0.3420
Epoch 60/100, Loss: 0.3282
Epoch 70/100, Loss: 0.3180
Epoch 80/100, Loss: 0.3047
Epoch 90/100, Loss: 0.2896
Epoch 100/100, Loss: 0.2772
```

```
Training Accuracy: 0.9183
Test Accuracy: 0.8913
```

پ (ا)

```
Epoch 10/100, Loss: 0.3880
Epoch 20/100, Loss: 0.3580
Epoch 30/100, Loss: 0.3464
Epoch 40/100, Loss: 0.3335
Epoch 50/100, Loss: 0.3225
Epoch 60/100, Loss: 0.3144
Epoch 70/100, Loss: 0.3080
Epoch 80/100, Loss: 0.3029
Epoch 90/100, Loss: 0.3049
Epoch 100/100, Loss: 0.3150
```

```
Training Accuracy: 0.9407
Test Accuracy: 0.8913
```

پ ۲)

تغیر لایہ ها :

```
Epoch 10/100, Loss: 0.3821
Epoch 20/100, Loss: 0.3633
Epoch 30/100, Loss: 0.3567
Epoch 40/100, Loss: 0.3511
Epoch 50/100, Loss: 0.3423
Epoch 60/100, Loss: 0.3340
Epoch 70/100, Loss: 0.3229
Epoch 80/100, Loss: 0.3114
Epoch 90/100, Loss: 0.3032
Epoch 100/100, Loss: 0.2968
```

```
Training Accuracy: 0.9005
```

```
Test Accuracy: 0.8533
```

پ ۳)

تغییر نرخ یادگیری:

```
Epoch 10/100, Loss: 0.3440
Epoch 20/100, Loss: 0.3352
Epoch 30/100, Loss: 0.3482
Epoch 40/100, Loss: 0.4300
Epoch 50/100, Loss: 0.4937
Epoch 60/100, Loss: 0.5503
Epoch 70/100, Loss: 0.6710
Epoch 80/100, Loss: 0.7142
Epoch 90/100, Loss: 0.8207
Epoch 100/100, Loss: 0.9450

Training Accuracy: 0.9591
```

همانطور که مشاهده میشود overshooting در برنامه زیاد است.

ت)

نرخ یادگیری adaptive مشاهده شد و همانطور که در عکس مشخص است بعد از یک اجرا در یکی از epoch نرخ یادگیری تغییر کرد:

```
Epoch 10/100, Loss: 0.3936, Learning Rate: 0.200000
Epoch 20/100, Loss: 0.3150, Learning Rate: 0.200000
Epoch 30/100, Loss: 0.3016, Learning Rate: 0.200000
Epoch 40/100, Loss: 0.2940, Learning Rate: 0.200000
Epoch 50/100, Loss: 0.2883, Learning Rate: 0.200000
Epoch 60/100, Loss: 0.2822, Learning Rate: 0.200000
Epoch 70/100, Loss: 0.2772, Learning Rate: 0.200000
Epoch 80/100, Loss: 0.2756, Learning Rate: 0.200000
Epoch 90/100, Loss: 0.2760, Learning Rate: 0.200000
Learning rate decreased to 0.100000
Epoch 100/100, Loss: 0.2736, Learning Rate: 0.100000

Training Accuracy: 0.9298
Test Accuracy: 0.8641
```

