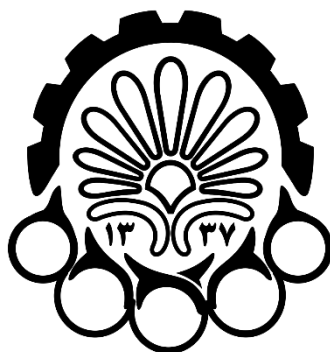


به نام خدا



دانشگاه صنعتی امیرکبیر  
( پلی تکنیک تهران )

تمرین درس شبکه‌های عصبی-سری اول

فردین آیار

شماره دانشجویی: ۹۹۱۳۱۰۴۰

استاد: دکتر صفابخش

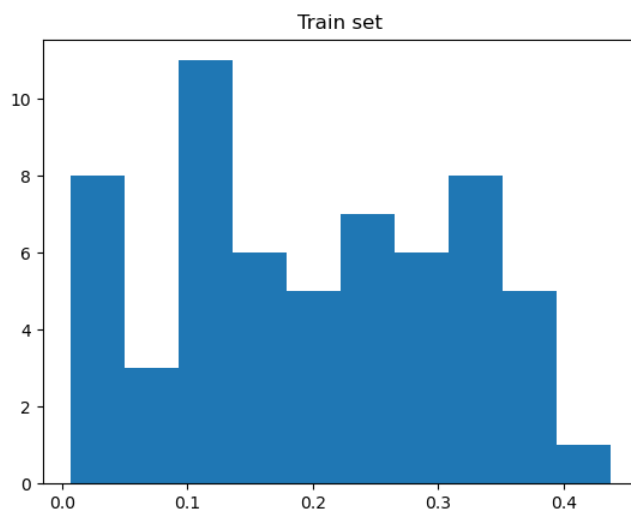
دانشکده کامپیوتر- زمستان ۹۹

۱) هر دو واحد پرسپترون و آدلاین از چند ورودی و وزن‌های مرتبط با آن‌ها و یک تابع فعال‌سازی برای خروجی تشکیل شده‌اند. تفاوت اصلی، رویکرد آن‌ها در بروزرسانی بردار وزن‌ها است. واحد پرسپترون تلاش می‌کند به دقت صد درصد در داده‌های آموزشی برسد و بنابراین در داده‌های جدایی ناپذیر خطی، همگرا نمی‌شود. در طرف مقابل، واحد آدلاین به دنبال کمینه‌سازی خطای میانگین مربعات است (با الگوریتم‌های بهینه‌سازی مانند گرادینان نزولی)؛ در نتیجه در صورت استفاده از ضریب یادگیری مناسب، حتی در داده‌های جدایی ناپذیر خطی، همگرا خواهد شد. اگرچه، در صورت وجود داده‌های پرت، ممکن است دقت آن دچار افت شدید شود.

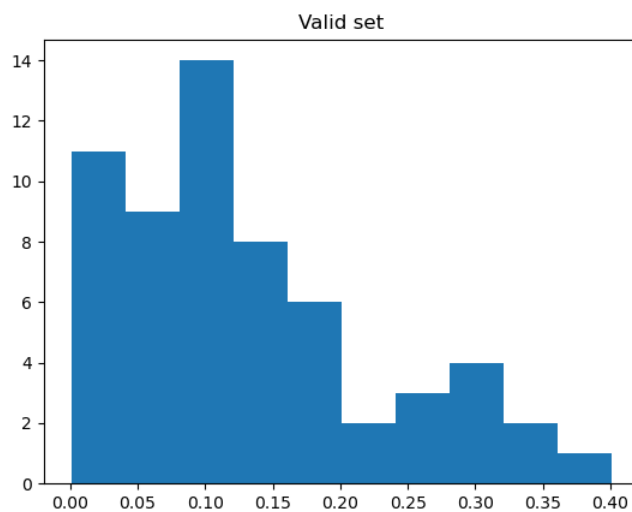
۲) کد مربوط به این قسمت در فایل `correlation.py` قرار دارد. قبل از شروع، داده‌ها را یک بار با فایل `shafiling.py` شافل کرده و در فایل `s_dataset.csv` ذخیره می‌کنیم. بین صورت سوال و تذکر ۲ در مورد نسبت تقسیم داده‌ها اختلاف وجود دارد. در اینجا طبق تذکر ۲، داده‌ها را با نسبت ۷۰، ۲۰ و ۱۰ برای آموزش، اعتبارسنجی و آزمون تقسیم می‌کنیم. در شکل‌های ۱ تا ۳، هیستوگرام قدرمطلق ضرایب همبستگی برای هر سه مجموعه آموزش، اعتبارسنجی و آزمون رسم شده است. همانطور که مشاهده می‌شود، هیستوگرام هر سه مجموعه با هم متفاوت است. دلیل این امر، کم بودن تعداد داده‌ها نسبت به تعداد ویژگی‌هاست. با توجه به این تفاوت، می‌توان پیش‌بینی کرد که مدل‌های آموزش دیده، احتمالاً تعمیم‌پذیری پایینی خواهند داشت. به بیان بهتر، دقت داده‌های آموزش و اعتبارسنجی تفاوت زیادی خواهد داشت. به هرحال برای انتخاب آستانه مناسب انتخاب ویژگی، از نمودار مربوط به مجموع آموزش استفاده می‌کنیم. طبق این نمودار اکثر ویژگی‌ها همبستگی پایینی با برچسب خروجی دارند و احتمالاً حذف ویژگی ایده خوبی نباشد. به هر حال آستانه ۰.۲ را برای انتخاب ویژگی انتخاب می‌کنیم. شماره ستون مربوط به ویژگی‌های انتخاب شده به شرح زیر است: (در قسمت‌های بعدی از همه ویژگی‌ها برای آموزش استفاده می‌کنیم)

selected features:

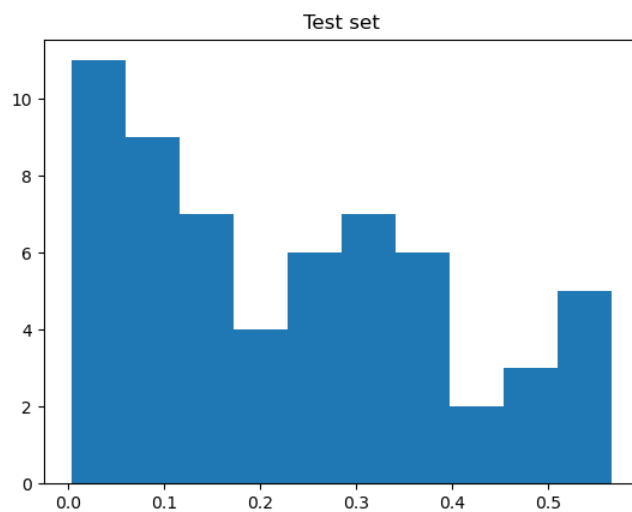
```
[ 0  1  2  3  4  5  8  9 10 11 12 13 18 19 20 21 22 34 35 36 42 43 44 45  
46 47 48 50 51]
```



شکل ۱



شکل ۲

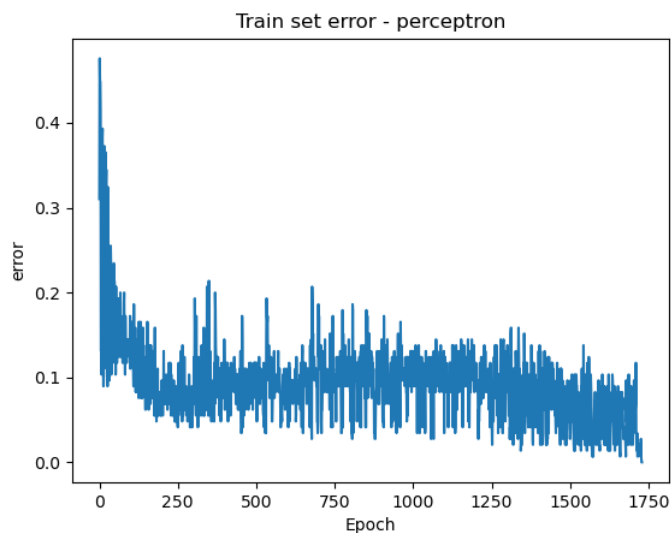


شکل ۳

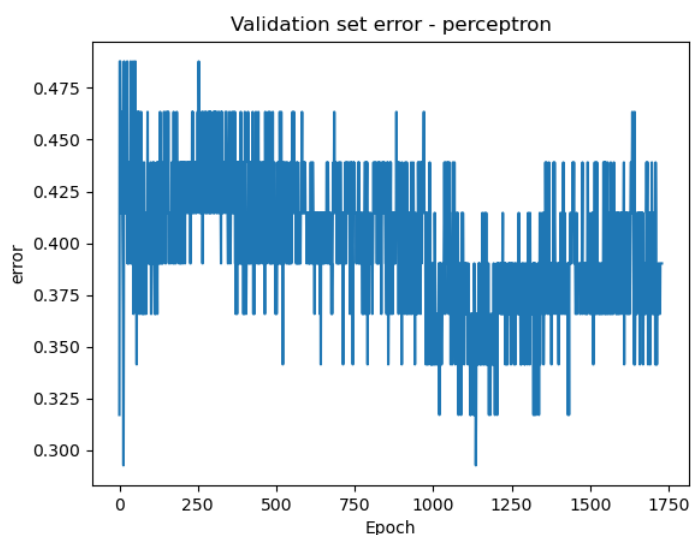
۳) کد مربوط به این سوال در فایل `perceptron.py` قرار دارد. تابع فعالیت پله برای این مسئله مناسب است، زیرا مسئله دارای دو کلاس صفر و یک است و در نتیجه نیازی به احتمالات تعلق به هر کلاس نداریم.

قبل از اجرای الگوریتم، یک ویژگی جدید با مقدار یک به همه داده‌ها اضافه می‌کنیم. با اینکار می‌توان بایاس را جزئی از بردار وزن‌ها فرض کرد. همچنین شرط توقف الگوریتم را عدم تغییر وزن‌ها در دو تکرار (Epoch) متوالی قرار می‌دهیم و با توجه به اینکه داده‌ها ممکن است جداپذیر خطی نباشند، یک حد ۱۰۰۰۰ برای تعداد تکرار در نظر می‌گیریم.

در شکل ۴ نمودار خطای دسته‌بندی بر حسب تکرار برای مجموعه آموزش و در شکل ۵ همین نمودار برای مجموعه اعتبارسنجی رسم شده است. همانطور که مشخص است داده‌ها (یا دسته‌کم داده‌های مجموعه آموزش)، خطی جداپذیر هستند و پرسپترون به دقت ۱۰۰ درصد در آموزش رسیده است. اگرچه دقت نهایی برای داده‌های اعتبارسنجی در محدوده مناسبی قرار ندارد و متأسفانه مدل تعمیم‌پذیری پایینی دارد.

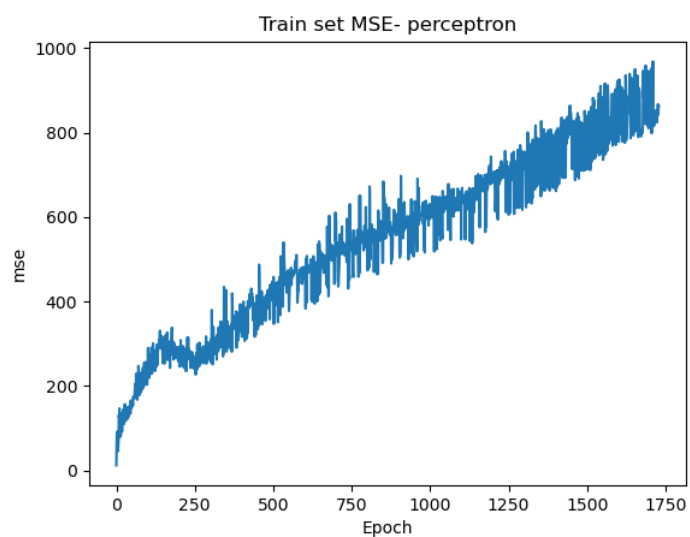


شکل ۴

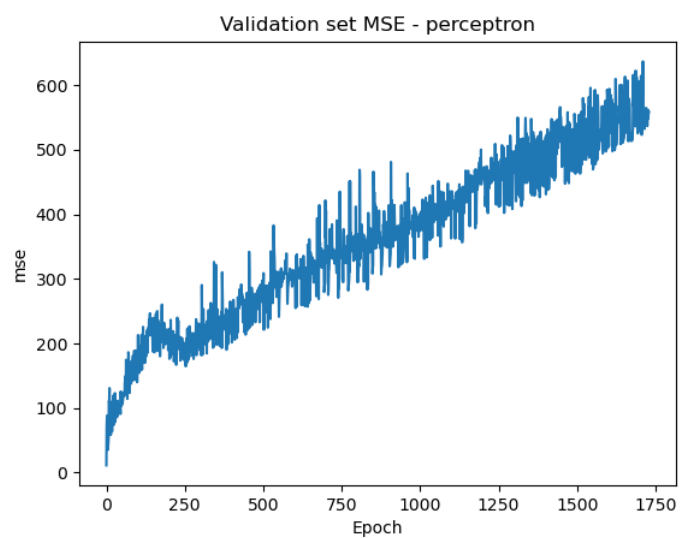


شکل ۵

در ادامه به بررسی نمودار خطای  $MSE$  بر حسب تکرار که برای مجموعه آموزش و اعتبارسنجی، به ترتیب در شکل های ۶ و ۷ رسم شده می‌پردازیم. با توجه به این شکل‌ها، خطای  $MSE$  در هر دو مجموعه صعودی بوده است. از این مشاهده می‌توان نتیجه گرفت که دیتاست احتمالاً دارای داده‌های پرت می‌باشد و حتی قبل از اجرای الگوریتم آدلاین، می‌توان پیش‌بینی کرد که دقت آن (دسته‌کم در مجموعه آموزش) بسیار کمتر از پرسپترون است.



شکل ۶



شکل ۷

در انتهای این بخش، خروجی‌های پرسپترون را به صورت کمی ارائه می‌کنیم. در ماتریس درهم‌ریختگی، سطرها برچسب واقعی و ستون‌ها برچسب پیش‌بینی شده است.

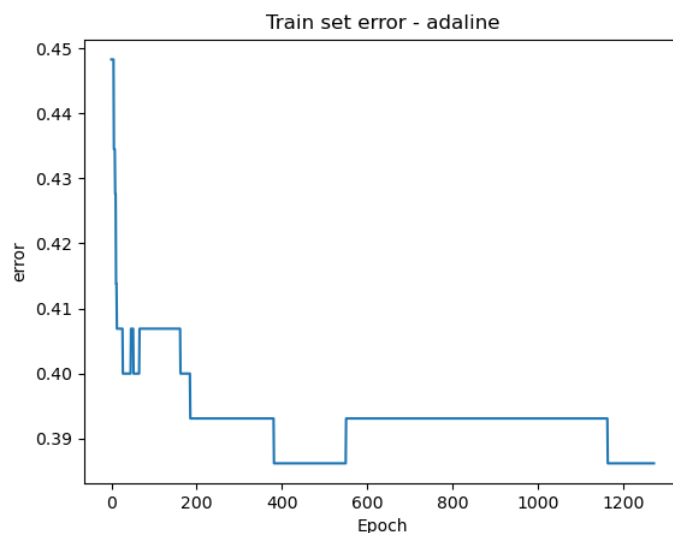
```
stopped after 1729 iteration
accuracy for train set: 1.0
confusion matrix for train set:
  0  1
0 65.0 0.0
1 0.0 80.0
```

```
accuracy for valid set: 0.6097560975609756
```

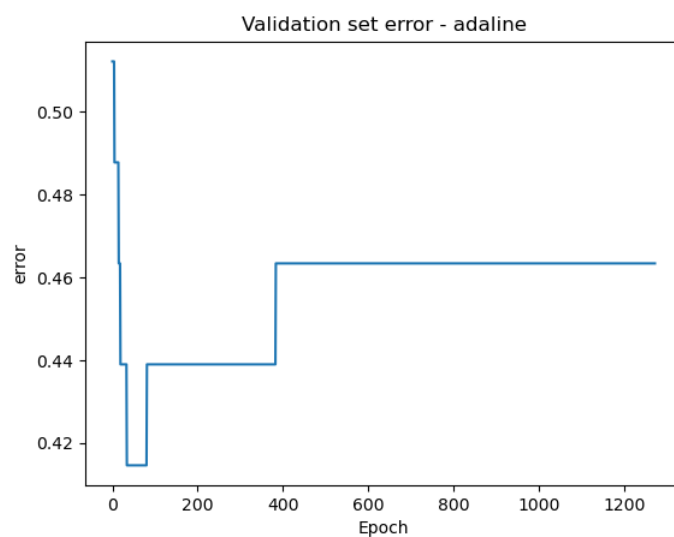
```
confusion matrix for valid set:
```

```
   0   1
0 12.0  9.0
1  7.0 13.0
```

۴) کد مربوط به این سوال در فایل `adaline.py` قرار دارد. برای شرط توقف از اختلاف وزن‌ها در دو تکرار متوالی استفاده می‌کنیم؛ به این صورت که اگر مجموع قدر مطلق اختلاف وزن‌ها از ۰.۰۱ کمتر باشد، الگوریتم متوقف می‌شود. در اجراهای مختلف آدلاین مشاهده شد که سختگیرانه‌تر کردن این شرط تاثیری در کاهش بیشتر خطای  $MSE$  در مجموعه آموزش ندارد. به بیان بهتر، همانطور که خواهیم دید، الگوریتم قبل از توقف، به یک مقدار حدی نزدیک می‌شود. همچنین ضریب یادگیری با آزمون و خطا ۰.۰۱ تنظیم شده است. در شکل‌های ۸ و ۹ نمودار خطای دسته بندی برحسب تکرار برای دو مجموعه آموزش و اعتبارسنجی رسم شده است. همانطور که در سوال مربوط به پرسپترون پیش‌بینی شد، الگوریتم آدلاین به دقت مناسبی در مجموعه آزمون نرسیده است. همچنین مجموعه اعتبارسنجی نیز مانند قبل، دقت مناسبی ندارد.

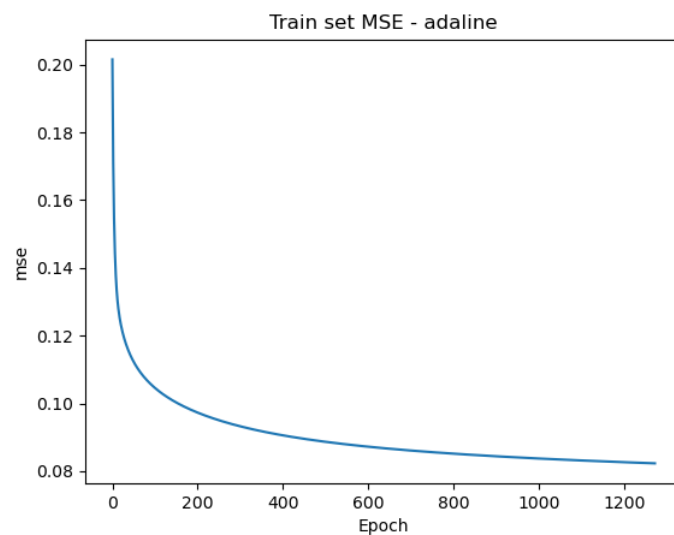


شکل ۸

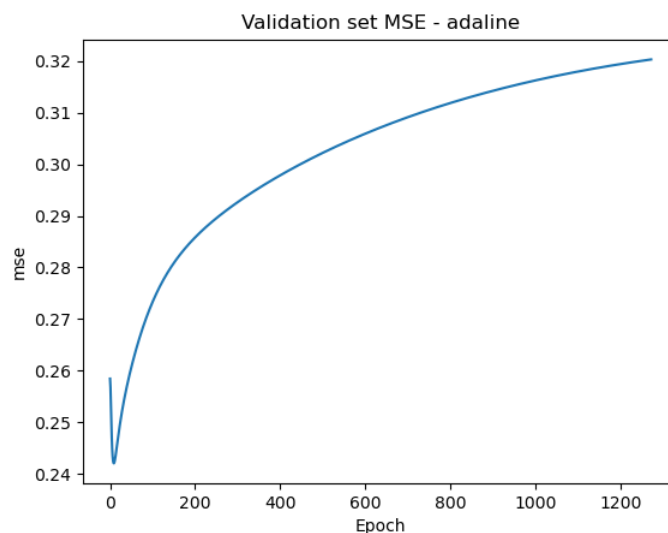


شکل ۹

در شکل‌های ۱۰ و ۱۱، خطای MSE برحسب تکرار، به ترتیب برای مجموعه آموزش و اعتبارسنجی رسم شده‌است. مطابق شکل ۷، قبل از توقف الگوریتم، خطای MSE به یک مقدار حدی نزدیک شده‌است و بنابراین، نیازی به افزایش تعداد تکرار وجود ندارد. همچنین شکل ۸ نشان می‌دهد که خطای MSE برای مجموعه اعتبارسنجی افزایشی بوده و مجدداً می‌توان نتیجه گرفت که مدل دارای تعمیم‌پذیری پایینی است.



شکل ۱۰



شکل ۱۱

در انتها خروجی‌های الگوریتم آدالین را به صورت کمی ارائه می‌کنیم.

```
stopped after 1271 iteration
accuracy for train set: 0.6137931034482759
confusion matrix for train set:
  0   1
0 9.0 56.0
1 0.0 80.0
accuracy for valid set: 0.5365853658536586
confusion matrix for valid set:
  0   1
0 2.0 19.0
1 0.0 20.0
```

در مقایسه با الگوریتم پرسپترون، الگوریتم آدالین در تعداد تکرار کمتری متوقف شده است؛ اما از آنجا که شرط توقف آدالین آسان‌تر است، نمی‌توان برتری خاصی در این زمینه به آدالین داد. در زمینه خطای آموزش، پرسپترون بسیار بهتر از آدالین عمل کرده است؛ اگرچه خطای اعتبارسنجی هر دو بالا است. به طور کلی تفاوت بسیار زیاد در معیارهای مربوط به مجموعه آموزش و اعتبارسنجی، به این دلیل است که تعداد داده‌های موجود در دیتاست نسبت به تعداد متغیرها کم است و مدل‌ها نمی‌توانند به خوبی الگوی موجود در ویژگی‌ها را تشخیص دهند.

اگر کلاس ۱ را کلاس **positive** فرض کنیم، نکته دیگری که در ماتریس درهم‌ریختگی مجموعه آموزش آدالین مشاهده می‌شود این است که تعداد داده‌های **false positive** در آن زیاد است. از این مشاهده می‌توان نتیجه گرفت که کلاس ۰ دارای داده‌های پرت زیادی است و همین نکته باعث شده ابرصفحه جداکننده، برای کمینه شدن **MSE** به سمت داده‌های پرت متمایل شود و به همین دلیل اکثر داده‌ها، کلاس ۱ تشخیص داده شده‌اند.



و در آخر با توجه به اینکه به نظر می‌رسد خطای false positive هزینه بیشتری دارد( حفاری کردن منطقه‌ای که اشتباهاً حاوی سنگ‌های معدن تشخیص داده شده است)، مدل پرسپترون به آدالین ترجیح دارد؛ زیرا در ماتریس درهم‌ریختگی داده‌های اعتبارسنجی آن، تعداد false positive کمتری وجود دارد. (این استدلال با توجه به کم بودن تعداد داده‌ها ممکن است چندان درست نباشد)

۵) کد مربوط به این سوال در فایل‌های adaline\_degree2.py و perceptron\_degree2.py قرار دارد. برای ساخت مدل‌های درجه دو، کافی است توان دو همه ویژگی‌ها را به عنوان ویژگی‌های جدید به ماتریس داده‌ها اضافه کنیم.<sup>۱</sup> به این ترتیب، الگوریتم‌های قبلی بدون تغییر قابل استفاده هستند. در اینجا طبق خواسته سوال، فقط خروجی‌های کمی را ارائه می‌کنیم و به بررسی نمودارها نمی‌پردازیم. خروجی پرسپترون درجه ۲ به این صورت است:

```
stopped after 372 iteration
accuracy for train set: 1.0
confusion matrix for train set:
  0  1
0 65.0  0.0
1  0.0 80.0
accuracy for valid set: 0.7317073170731707
confusion matrix for valid set:
  0  1
0 15.0  6.0
1  5.0 15.0
```

پرسپترون درجه ۲ نیز مطابق انتظار در مرحله آموزش به دقت ۱۰۰ درصد رسیده است؛ اما در مقایسه با نوع خطی آن، در تکرار کمتر همگرا شده و دقت اعتبارسنجی آن نیز بیشتر است. لذا استفاده از آن پیشنهاد می‌شود.<sup>۲</sup> در ادامه خروجی آدالین درجه ۲ را ارائه می‌کنیم:

```
stopped after 4190 iteration
accuracy for train set: 0.6551724137931034
confusion matrix for train set:
  0  1
0 15.0 50.0
1  0.0 80.0
accuracy for valid set: 0.6585365853658537
confusion matrix for valid set:
  0  1
0 7.0 14.0
1 0.0 20.0
```

در اینجا نیز، استفاده از مدل درجه ۲، سبب افزایش دقت در داده‌های اعتبارسنجی شده است. علاوه بر آن، دقت آموزش نیز اندکی بهبود یافته- است. اگرچه بر خلاف پرسپترون درجه ۲، تعداد تکرار لازم برای همگرایی نیز افزایش یافته است.

<sup>۱</sup> علاوه بر این، می‌توان ضرب دو به دوی ویژگی‌ها را نیز به عنوان ویژگی به دیتاست اضافه کنیم. در اینجا به علت کم بودن داده‌ها، از این کار صرف‌نظر شده است.

<sup>۲</sup> در حقیقت، در مواردی که تعداد داده‌ها نسبت به تعداد ویژگی‌ها کم است (مانند این مسئله)، شافلینگ‌های متفاوت ممکن است نتایج بسیار متفاوتی را سبب شود. بنابراین در این مورد نیز قطعی وجود ندارد.