به نام خدا

دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده برق و کامپیوتر

# سیستم‌های هوشمند

## تمرین کامپیوتری ۲

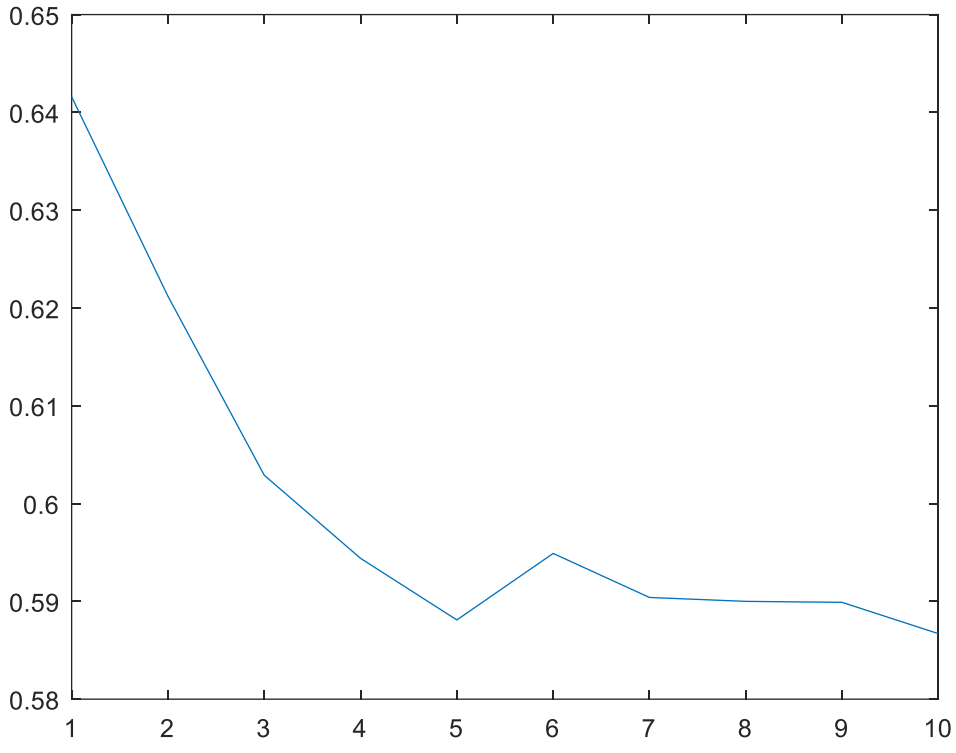
**فرزاد مهری**

**۸۱۰۱۹۴۴۱۰**

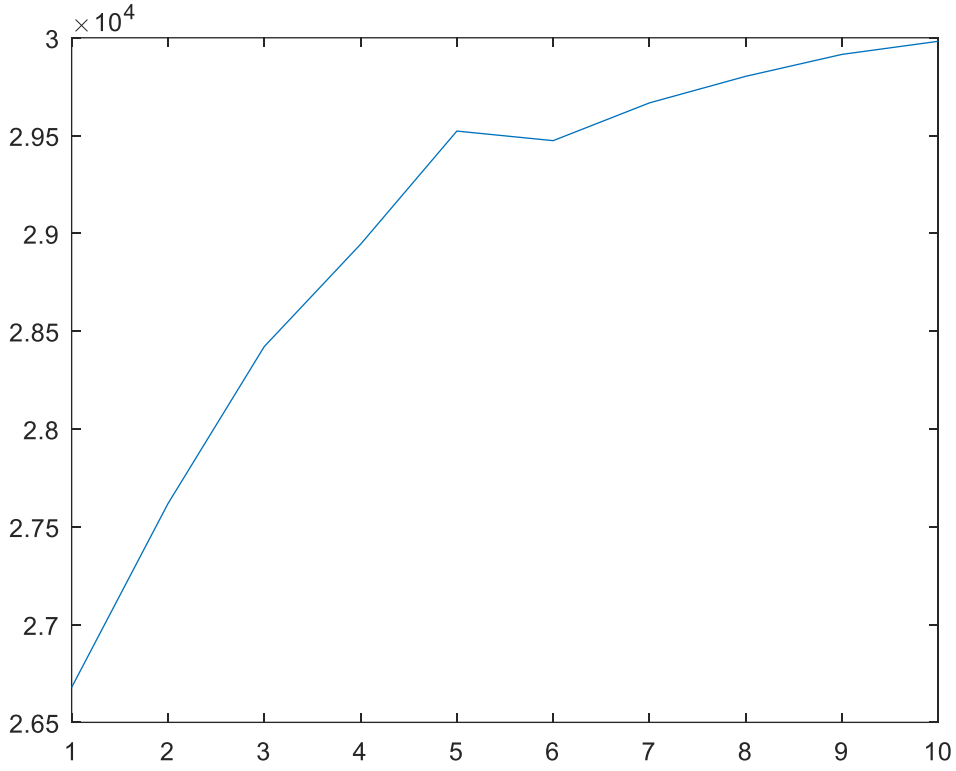
# سوال ۱

## تاثیر تعداد نورون های لایه پنهان

با افزایش تعداد نورون های لایه پنهان، امکان تطابق با ورودی های غیرخطی‌تر و پیچیده تر افزایش می‌یابد. به طوریکه در تعداد کم عملا نورون ها نمی‌توانند خروجی را به طور مناسب تشخیص دهند. از طرفی با افزایش تعداد نورون‌های لایه‌ی پنهان، تعداد داده‌های مورد نیاز برای آموزش دادن شبکه افزایش می‌یابد. با توجه به محدود بودن تعداد داده‌های ورودی، اگر تعداد نورون ها بیش از اندازه زیاد انتخاب شود، پس از تعداد زیادی epoch باعث روی دادن over-fitting می‌شود. باید تعداد نورون ها به گونه‌ای انتخاب شود که بتواند خواص غیر خطی را تشخیص دهد. نمودار همگرایی برای تعداد نورون کم در شکل ۱ آمده است. با توجه به شکل می‌توان مشاهده کرد که با این تعداد کم نورون، نمیتوان به دقت های بالا دست یافت.

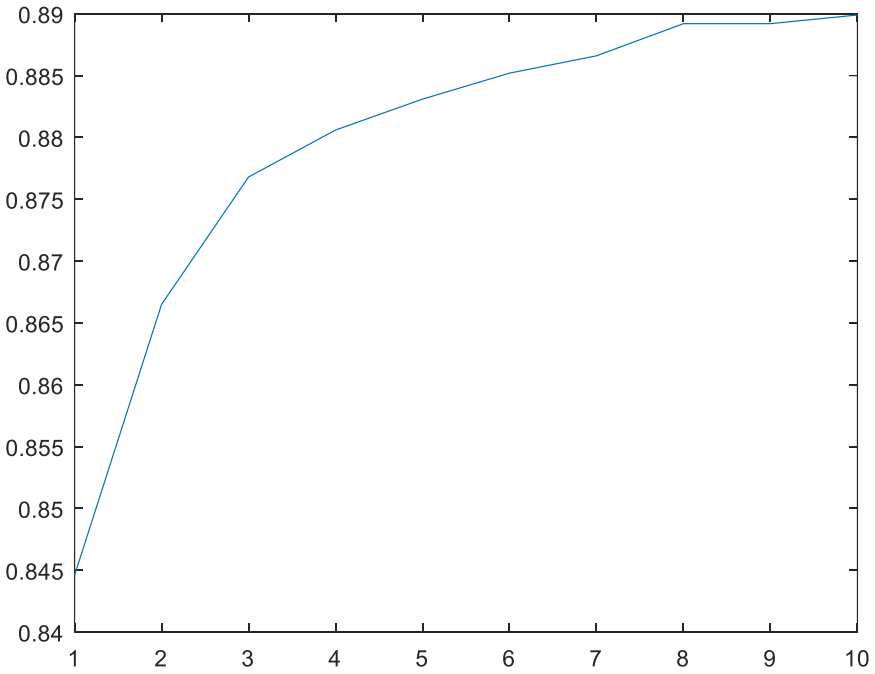



شکل ۱، نمودار دقت روی داده های تست به ازای ۷ نورون در لایه پنهان




شکل ۲، نمودار **Loss** بر خسب تعداد **epoch** به ازای ۷ نورون

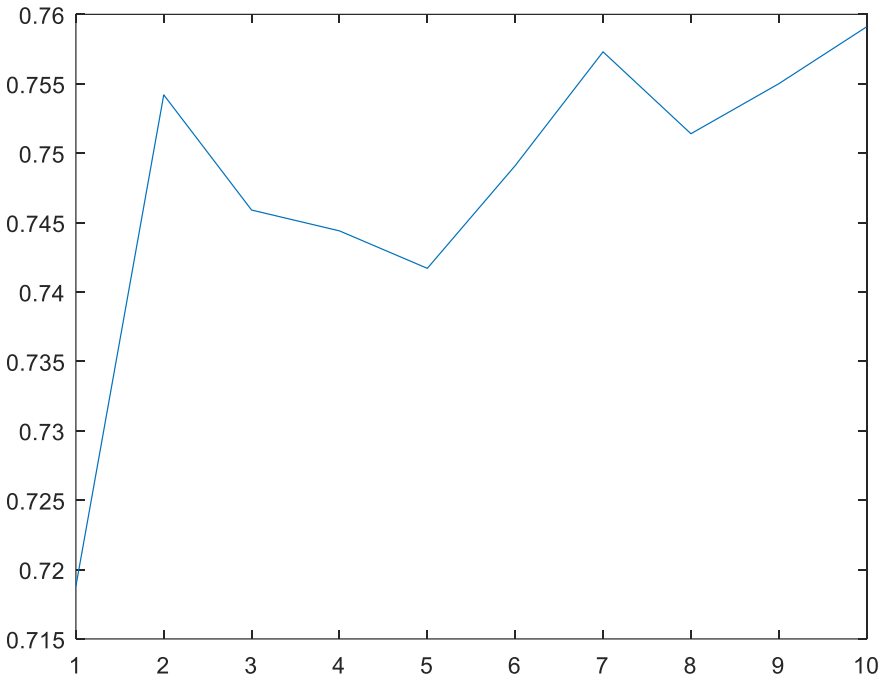
بدون تغییر دادن سایر مشخصات شبکه و فقط با افزایش تعداد نورون ها به ۷۰، نمودار به صورت شکل ۲ درمی‌آید. که در آن دقت بهتری بدست آمده.



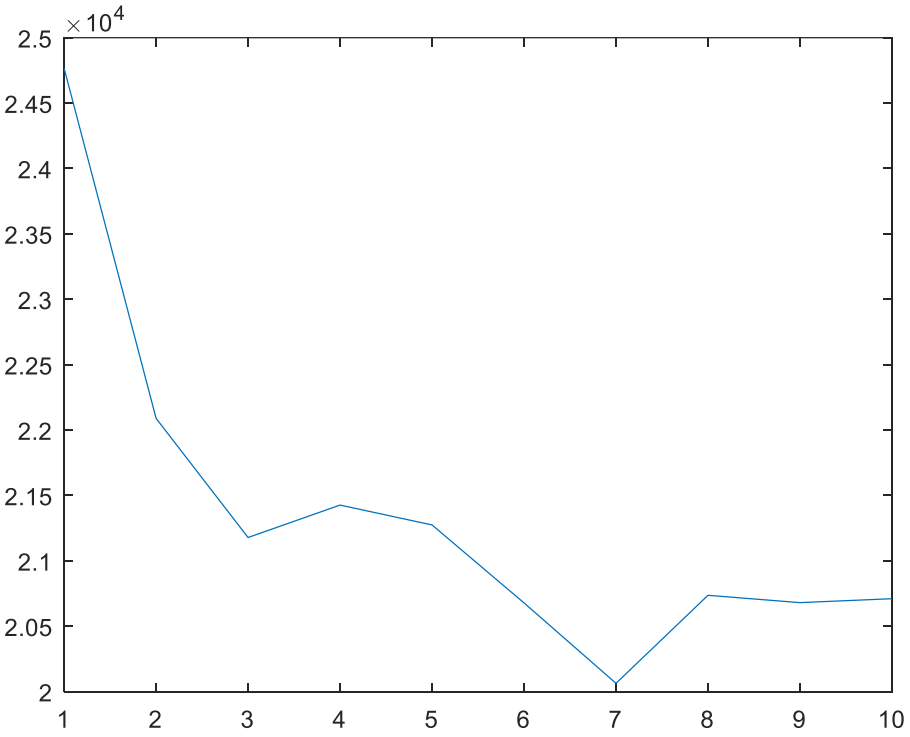

شکل ۳، نمودار خطا به ازای ۷۰ نورون در لایه‌ی پنهان

## تاثیر سایز هر batch

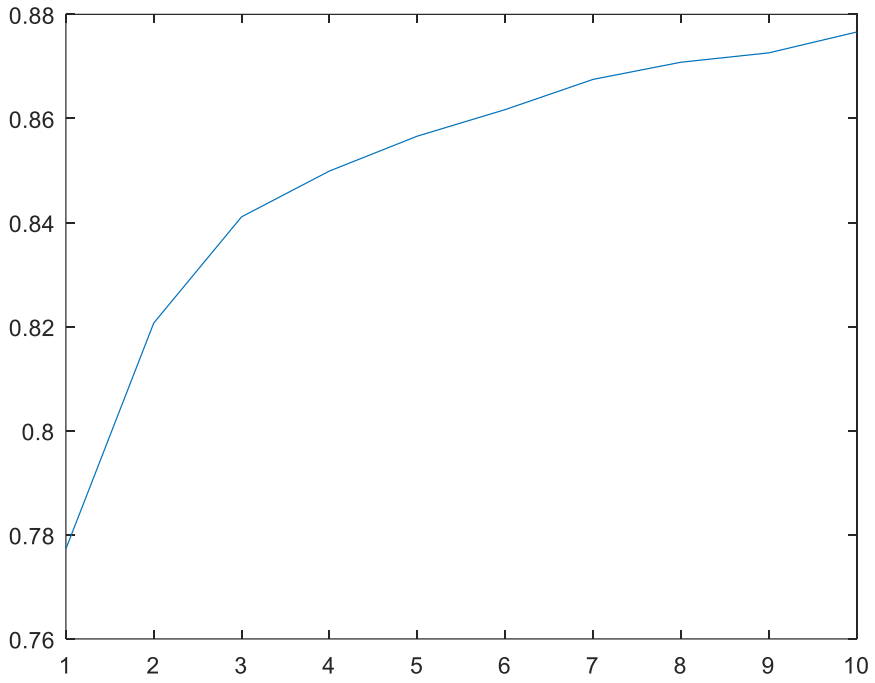
هر چه سایز batch کوچکتر انتخاب شود، روش gradient descent حساسیت بیشتری به نویز خواهد داشت و احتمال رسیدن به یک local optimum افزایش می‌یابد. در این حالت با هر step روش gradient descent ممکن است مقادیر در جهت های مختلف تغییر کند. سرعت همگرایی در این حالت زیاد است. با افزایش سایز batch، سرعت همگرایی کاهش پیدا می‌کند اما گرادیان های محاسبه شده در این حالت دقت بهتری دارند و احتمال رسیدن به local optimum کاهش می‌یابد. نمودار دقت برای سایز batch ۳ و ۳۰ در زیر آمده است. در شکل ۴ و ۵ اثر نویز و سرعت همگرایی بالا را که ناشی از کم بودن سایز batch است میتوان دید. اما در شکل ۶ و ۷، نمودار به آرامی و به صورت smooth تغییر کرده و با افزایش سایز batch، واریانس کاهش یافته.



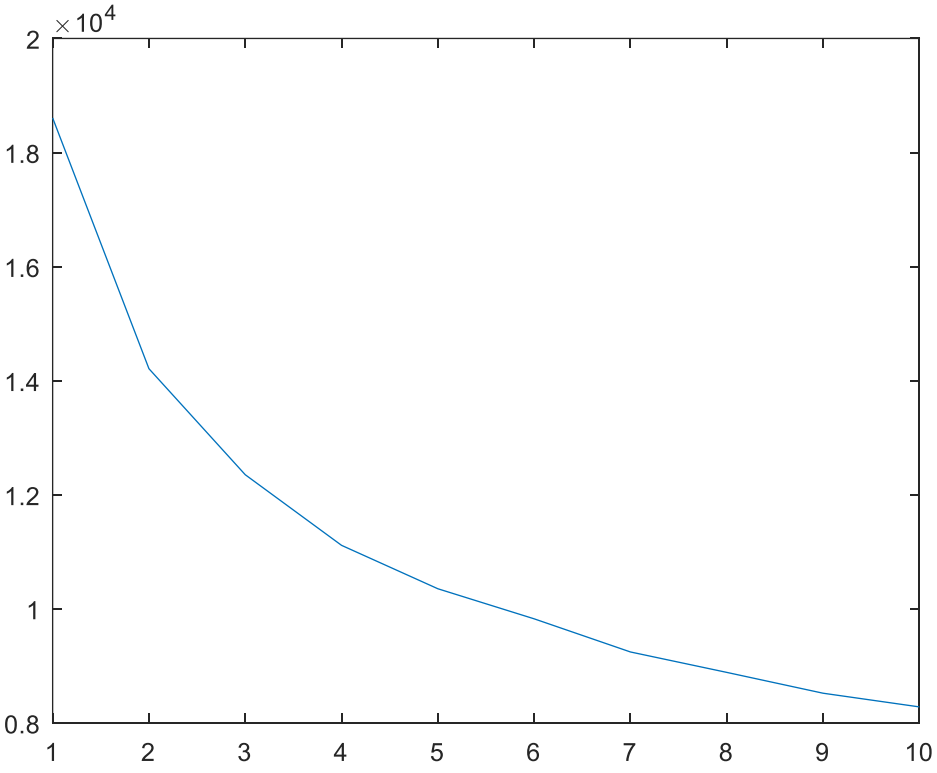

شکل ۴، نمودار دقت به ازای سایز **batch** برابر ۳




شکل ۵، نمودار **loss** بر اساس **epoch** برای سایز **batch** برابر ۳



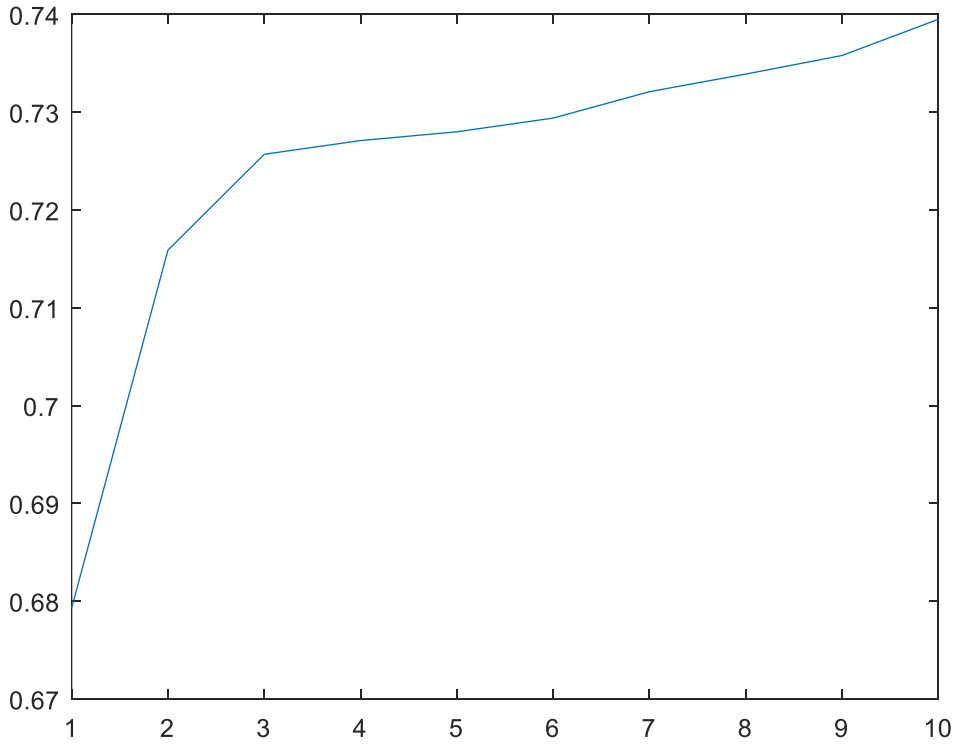

شکل ۶، نمودار دقت برای سایز **batch** برابر ۳۰



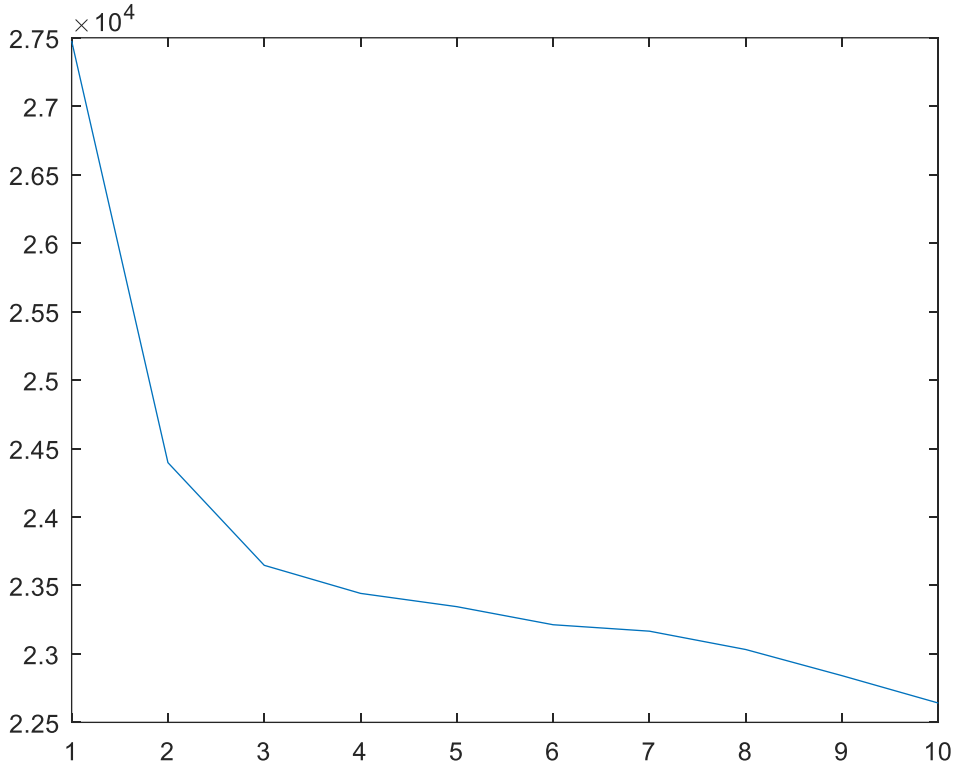

شکل ۷، نمودار **Loss** بر اساس تعداد **epoch** برای سایز **batch** برابر ۳۰

## تاثیر تابع فعالساز

به طور کلی انتخاب تابع فعالساز مناسب -که برای تشخیص الگوهای غیرخطی ضروری است- به داده-های ورودی بستگی دارد. از معایب تابع فعالساز tanh و sigmoid این است که گرادیان این توابع در مقدار های زیاد اشباع می‌شود و به صفر می‌رسد. تابع sigmoid حول صفر متقارن نیست که این باعث سخت شدن بهینه‌سازی پارامترها می‌شود. همچنین سرعت همگرایی در sigmoid کم است. تابع tanh حول صفر متقارن است و از این رو بهینه‌سازی آن آسانتر و همگرایی آن سریع‌تر است. tanh به این دلیل بر sigmoid برتری دارد. نمودار دقت برای این توابع فعالساز sigmoid در شکل‌های ۸ و ۹ آمده است. مقایسه این شکل ها با شکل های ۶ و ۷ که در شرایط مشابه با تابع فعالساز tanh رسم شده اند، تایید کننده برتری tanh از نظر همگرایی و بهینه‌سازی بر sigmoid است. همچنین تابع فعالساز f(x)=x برای شبکه‌ی عصبی مناسب نیست به علت اینکه یک تابع خطی است.



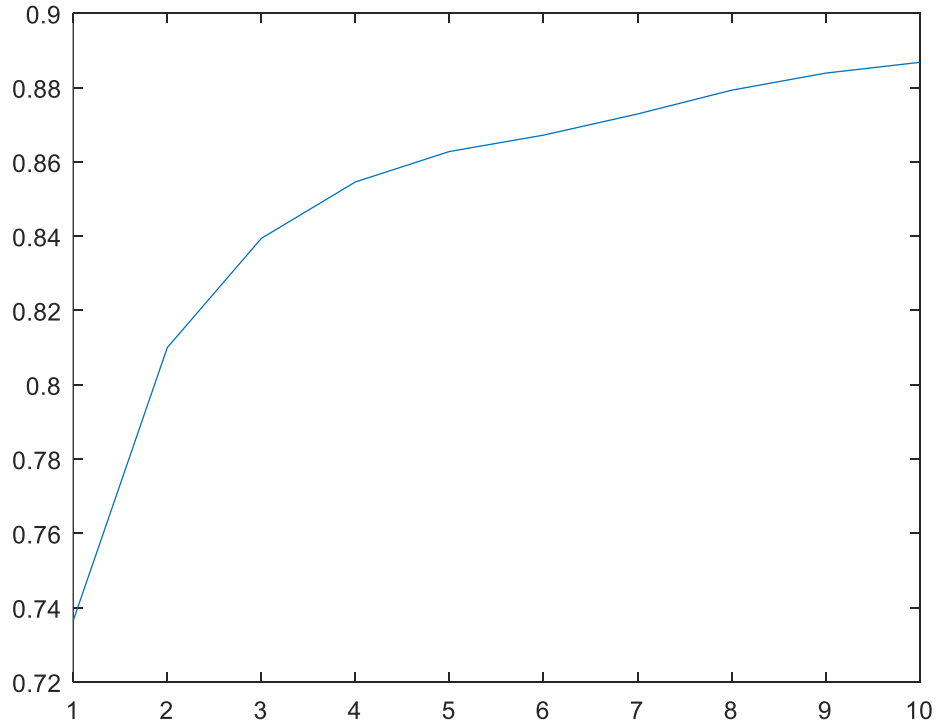

شکل ۸، نمودار دقت با تابع فعالساز **sigmoid**



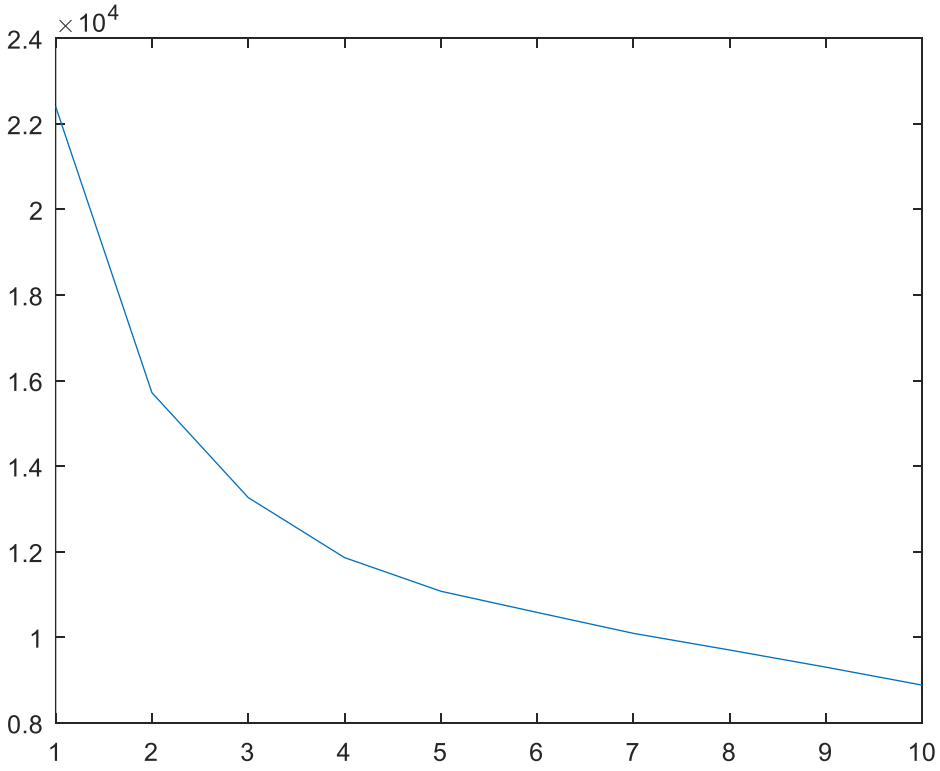

شکل ۹، نمودار **Loss** با تابع فعالساز **sigmoid**

## اثر نرمالیزه‌کردن

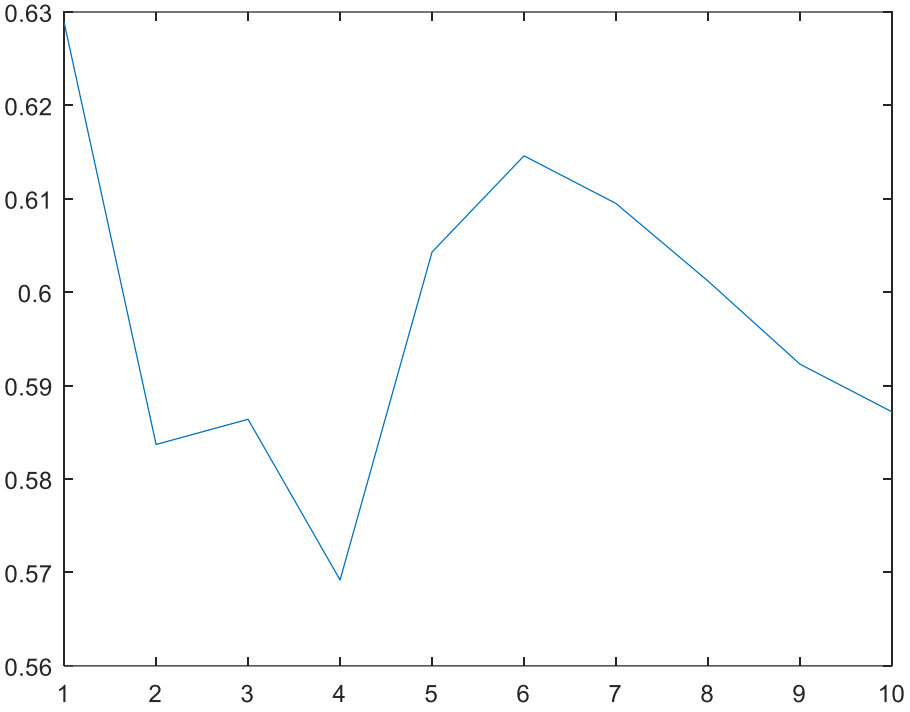
نرمالیزه‌کردن داده‌ها با از بین بردن بایاس به سمت feature های ورودی باعث بهبود کارکرد شبکه می‌شود و از اشباع شبکه هم جلوگیری می‌کند که منجر به افزایش سرعت همگرایی می‌شود. روش Z-score و ورش Min-Max هر کدام کاربردهای متفاوت دارند. روش Min-Max داده‌ها را به بازه‌ی بین صفر تا یک می‌نگارد اما واریانس را کاهش می‌دهد. در اینجا تفاوت زیادی بین روش‌ها وجود ندارد و فقط Z-score کمی سریع‌تر همگرا شده است. نمودار مربوط به حالت نرمالیزاسیون Z-score در شکل ۶ و ۷ آمده است. شکل ۱۰ و ۱۱ مربوط به نرمالیزاسیون Min-Max است و شکل۱۲ و ۱۳ مربوط به حالت بدون نرمالیزاسیون. با توجه به نمودار ها مشخص است که نرمالیزاسیون بسیار مهم است.



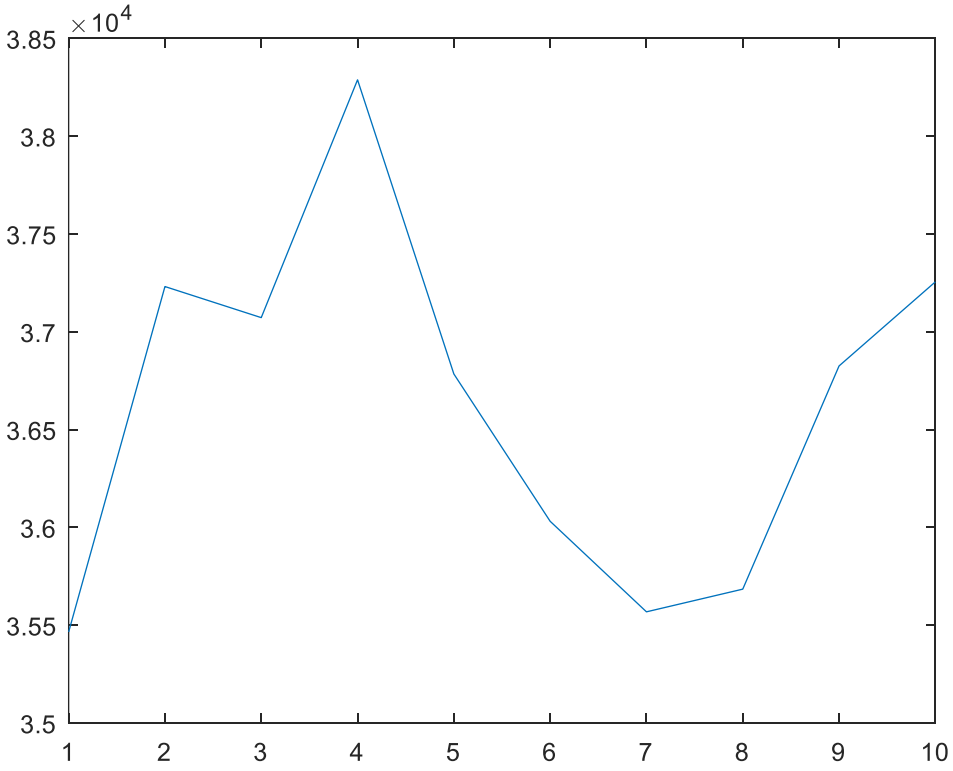

شکل ۱۰، دقت با نرمالیزاسیون **Min-Max**




شکل ۱۱، **Loss** با نرمالیزاسیون **Min-Max**



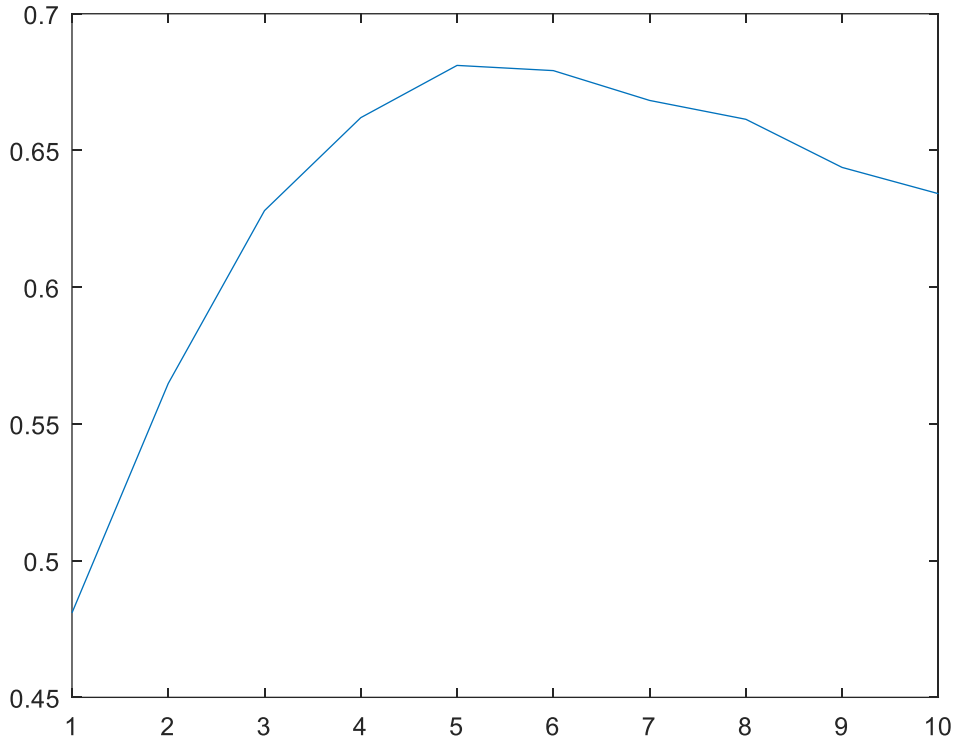

شکل ۱۲، نمودار دقت بدون نرمالیزاسیون



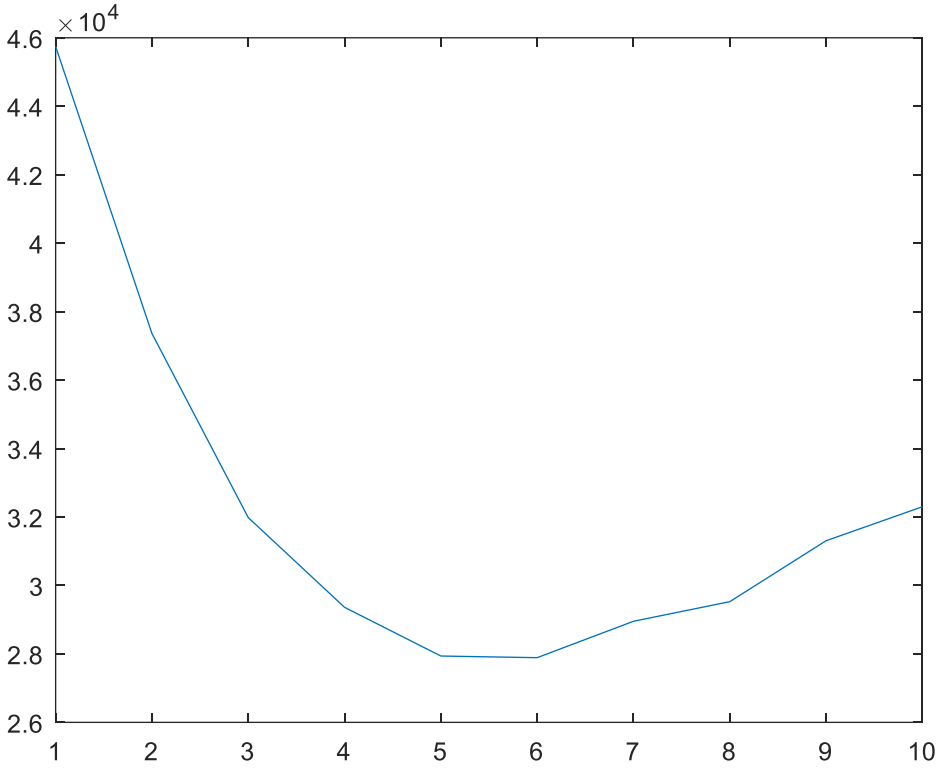

شکل ۱۳، نمودار **Loss** بدون نرمالیزاسیون

## تاثیر مقداردهی اولیه وزن‌ها

مقدار اولیه‌ی پارامترها از این نظر اهمیت دارد که مسئله بهینه‌سازی شبکه عصبی، یک مساله بهینه-سازی non-convex است و جواب های بهینه‌ی محلی متفاوتی ممکن است وجود داشته باشد. با توجه به مقدار اولیه وزن ها، ممکن است حل back-propagation به یکی از این جواب‌های بهینه‌ی محلی همگرا شود. در شکل ۶ و ۷، وزن ها به صورت تصادفی در نزدیکی صفر انتخاب شده‌اند. در شکل ۱۴ و ۱۵، وزن‌ها برابر با مقادیر تصادفی بزرگتری قرار گرفته‌اند در نتیجه به یک حل غیر ایده‌آل همگرا شده است.



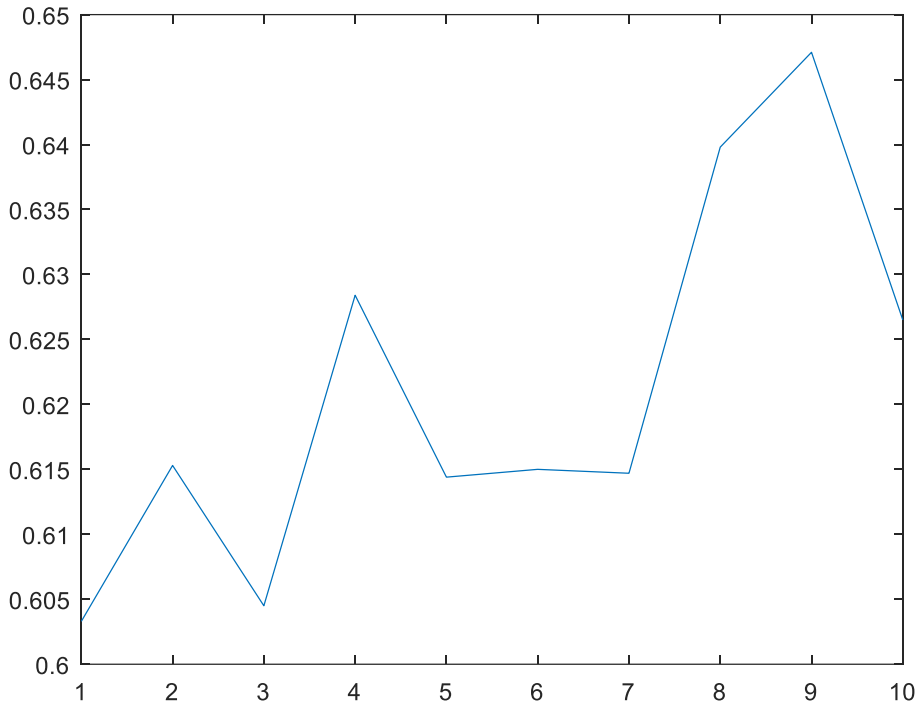

شکل ۱۴، دقت با وزن های با میانگین ۰,۱



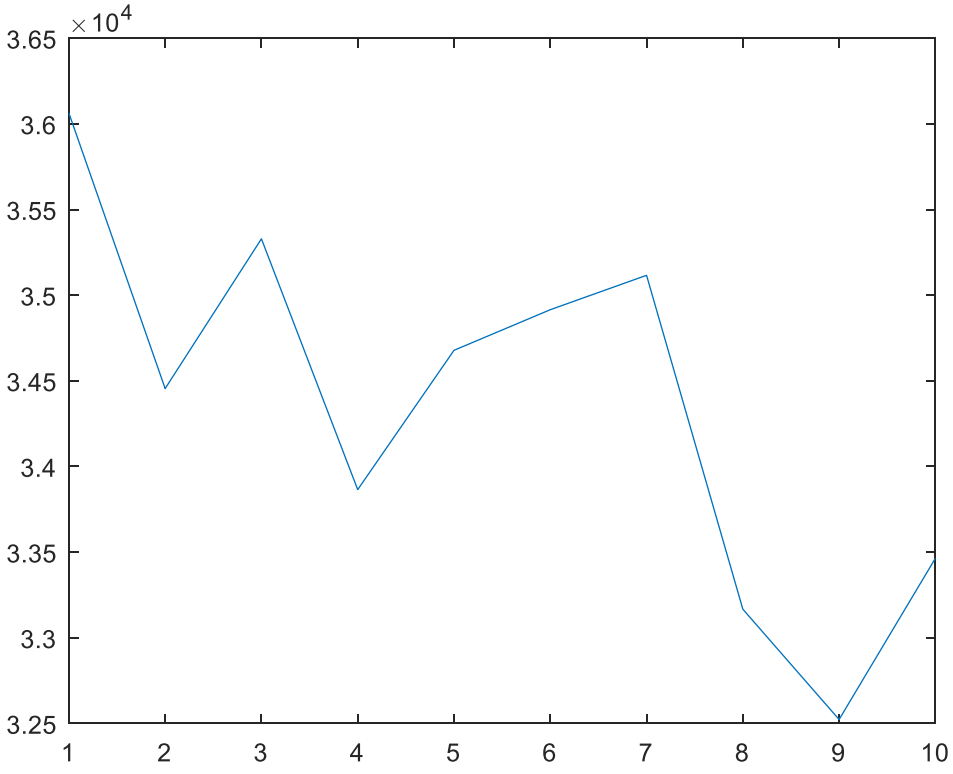

شکل ۱۵، **Loss** با وزن‌های با میانگین ۰,۱

## تاثیر مقدار نرخ یادگیری

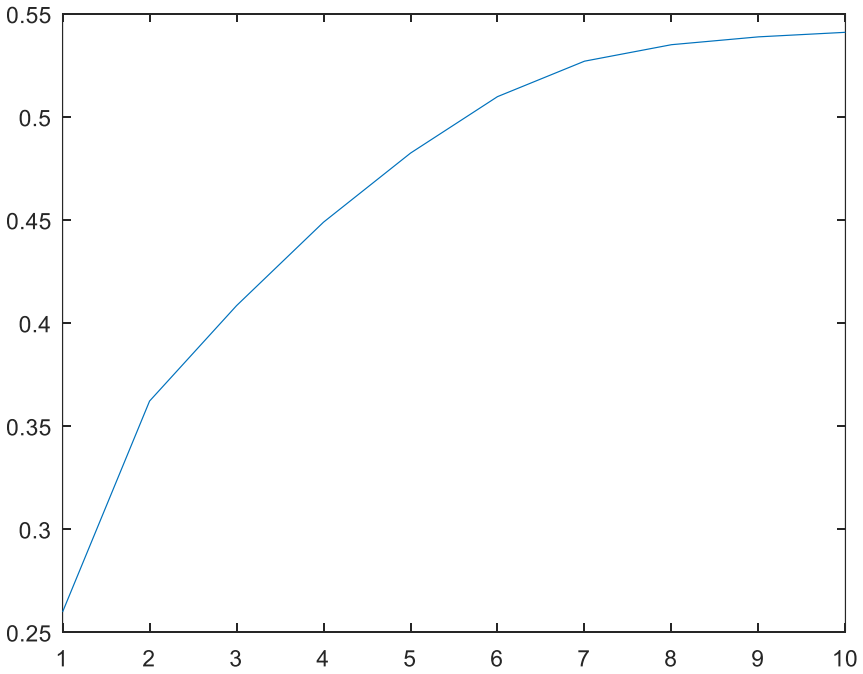
نرخ یادگیری اگر خیلی زیاد انتخاب شود، ممکن است از روی نقطه بهینه جهش کند و منجر به oscillation حول نقطه بهینه شود. همچنین اگر کم انتخاب شود، پیشرفت gradient descent کند می‌شود و نمیتوان با داده‌های موجود به دقت مورد نظر رسید. در شکل ۱۶ و ۱۷، نرخ یادگیری مقدار زیادی در نظر گرفته‌شده و در شکل ۱۸ و ۱۹ مقدار کم. نوسان دقت در شکل ۱۶ و ۱۷ به خوبی مشاهده می‌شود. همچنین کند بودن همگرایی در شکل ۱۸ و ۱۹ قابل تشخیص است.



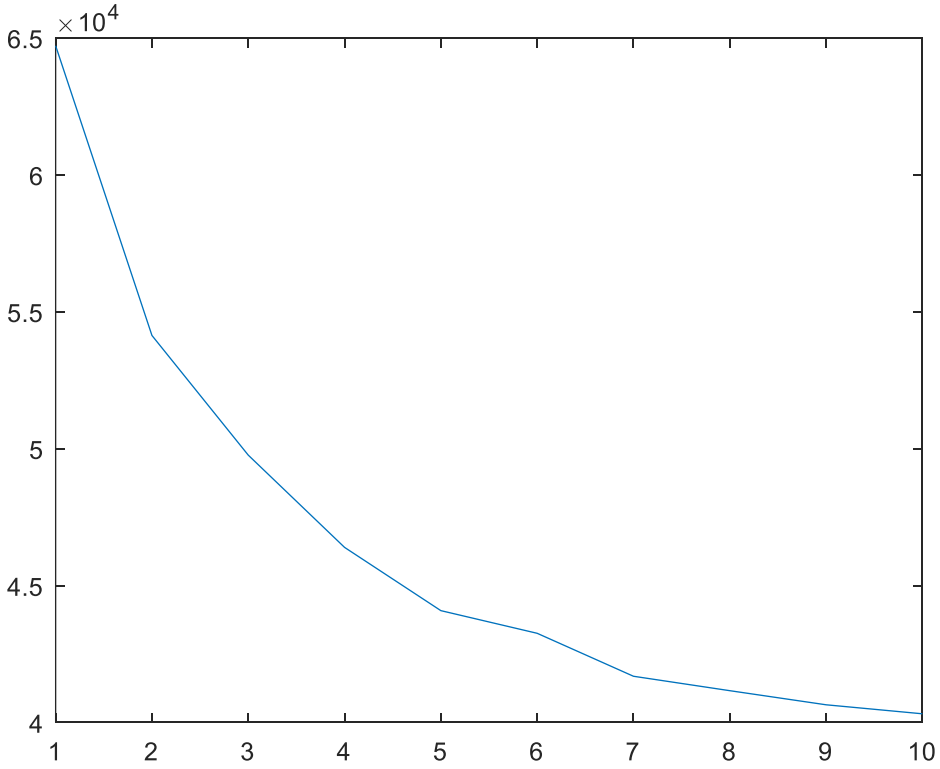

شکل ۱۶، نمودار دقت به ازای نرخ یادگیری بزرگ




شکل ۱۷، نمودار **Loss** به ازای نرخ یادگیری بزرگ



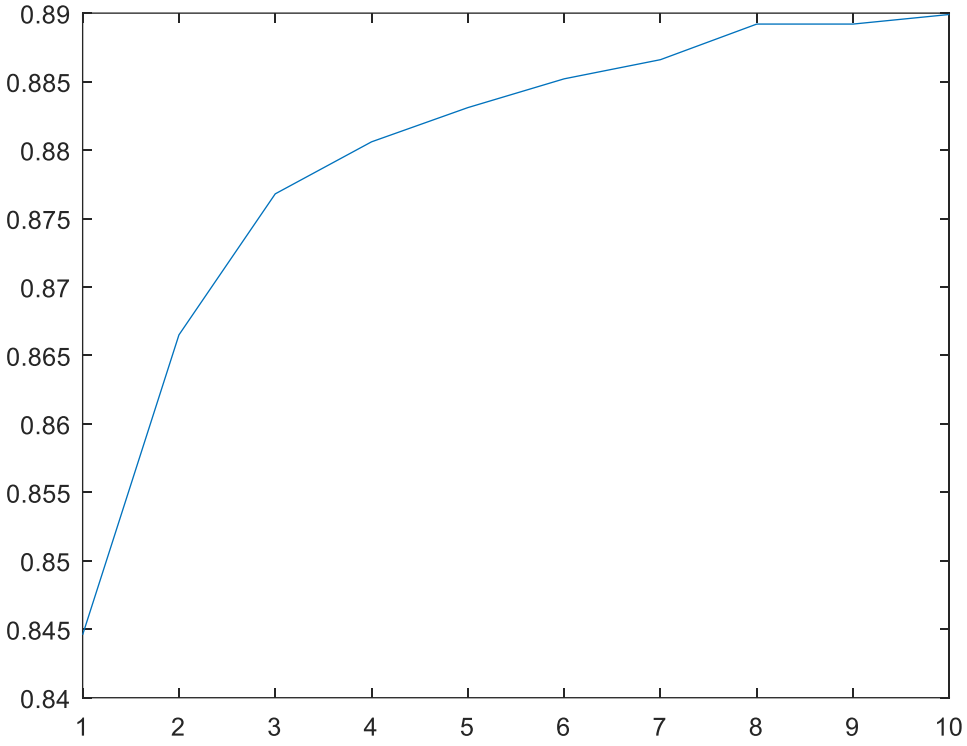

شکل ۱۸، نمودار دقت به ازای نرخ یادگیری کوچک



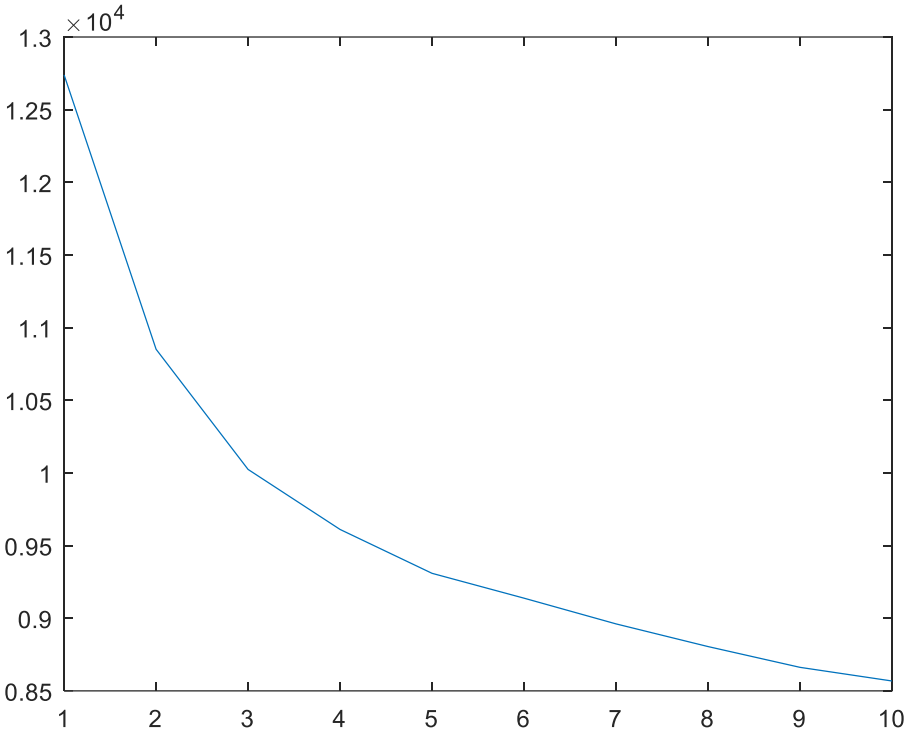

شکل ۱۹، نمودار **Loss** به ازای نرخ یادگیری کوچک

## تاثیر coding

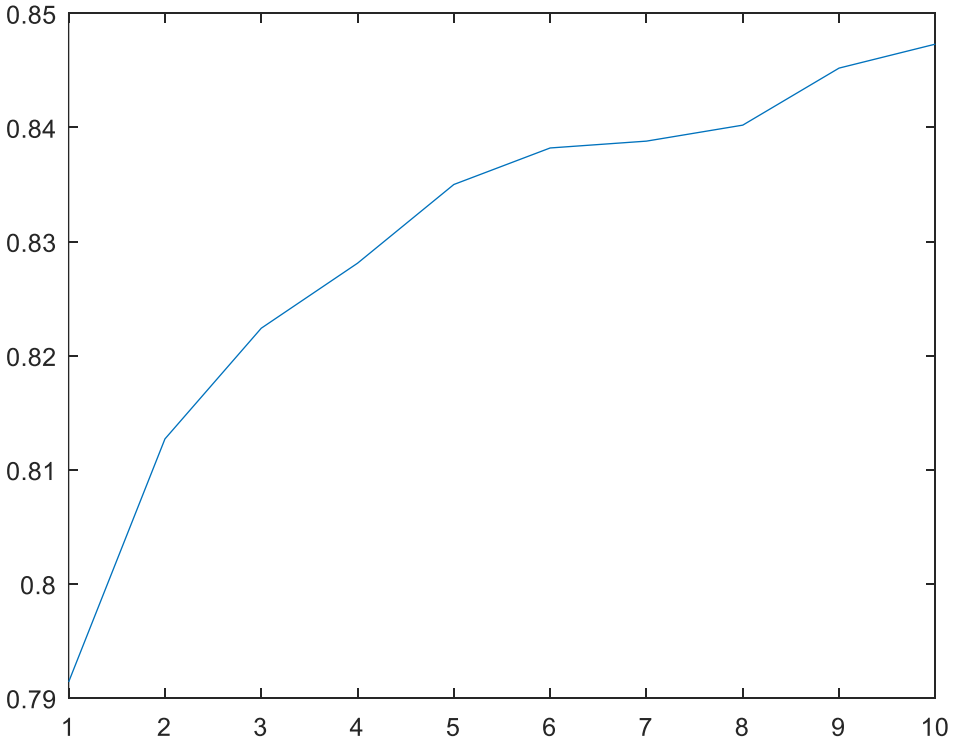
کدینگ باینری، با فشردسازی بردار خروجی، تعداد نورون‌های لایه‌ی خروجی را (به ویژه اگر تعداد کلاس های خروجی زیاد باشد) به شدت کاهش می‌دهد در نتیجه تعداد وزن‌های مورد نیاز کاهش می‌یابد. از طرفی با توجه به اینکه در کدینگ باینری، هر بردار خروجی می‌تواند بیش از یک مقدار غیر صفر داشته باشد، نوعی ارتباط و ترتیب بین داده‌های خروجی فرض می‌شود. در صورتی که خروجی ها مستقل باشند و ترتیب و ارتباطی بینشان نباشد این مساله باعث کاهش دقت یادگیری شبکه می‌شود. در این تمرین نیز به همین دلیل، استفاده از کدینگ one-hot نتایج بهتری میدهد. نمودار دقت برای کدینگ onehot در شکل ۲۰ و ۲۱ و نمودار دقت کدینگ باینری در شکل ۲۲ و ۲۳ آمده است.



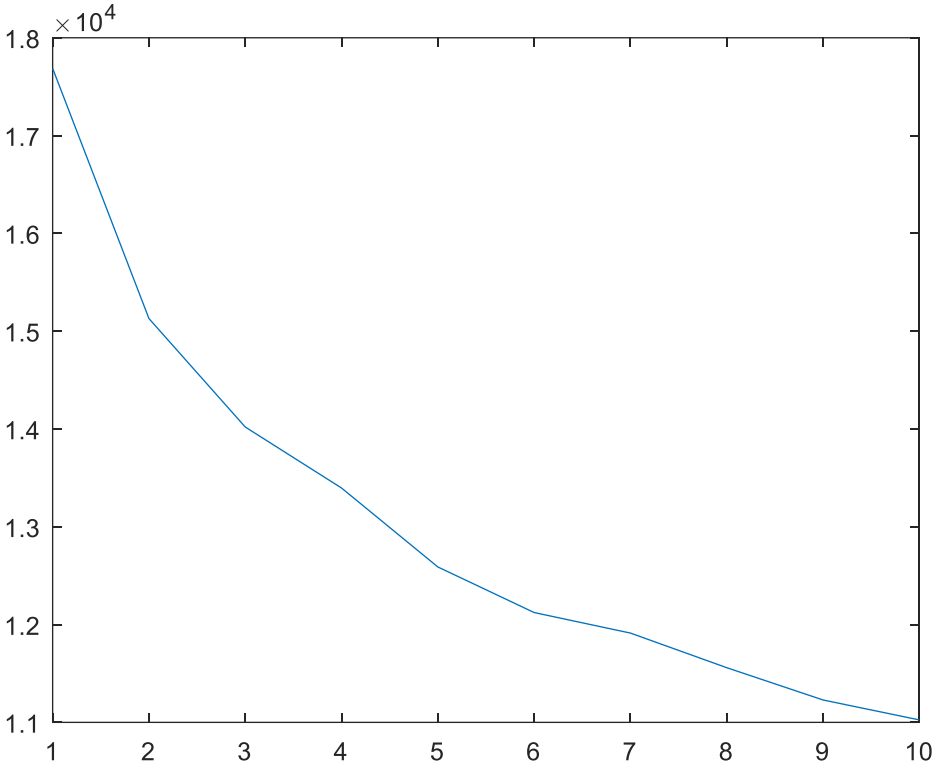

شکل ۲۰، نمودار دقت با کدینگ **onehot**




شکل ۲۱، نمودار **Loss** با کدینگ **onehot**




شکل ۲۲، نمودار دقت با کدینگ **binary**




شکل ۲۳، نمودار **Loss** با کدینگ باینری

### تاثیر momentum

## سوال ۲

بهترین دقت در این مساله به ازای پارامترهای زیر بدست آمد:

سایز batch: سایز هر batch برابر ۳۰ قرار گرفت تا تعادل بین سرعت همگرایی و دقت جهت گرادیان برقرار شود.

تعداد نورون های لایه پنهان: ۷۰ نورون در لایه‌ی پنهان قرار گرفت تا بتواند خواص غیر خطی مساله را تشخیص دهد. افزایش بیشتر تعداد نورون‌ها تاثیر زیادی در دقت نداشت و منجر به افزایش حجم محاسبات میشد.

نرخ یادگیری برابر ۰,۰۶ انتخاب شد تا نوسان حول نقطه‌ی حل رخ ندهد و جواب با سرعت کافی همگرا شود.

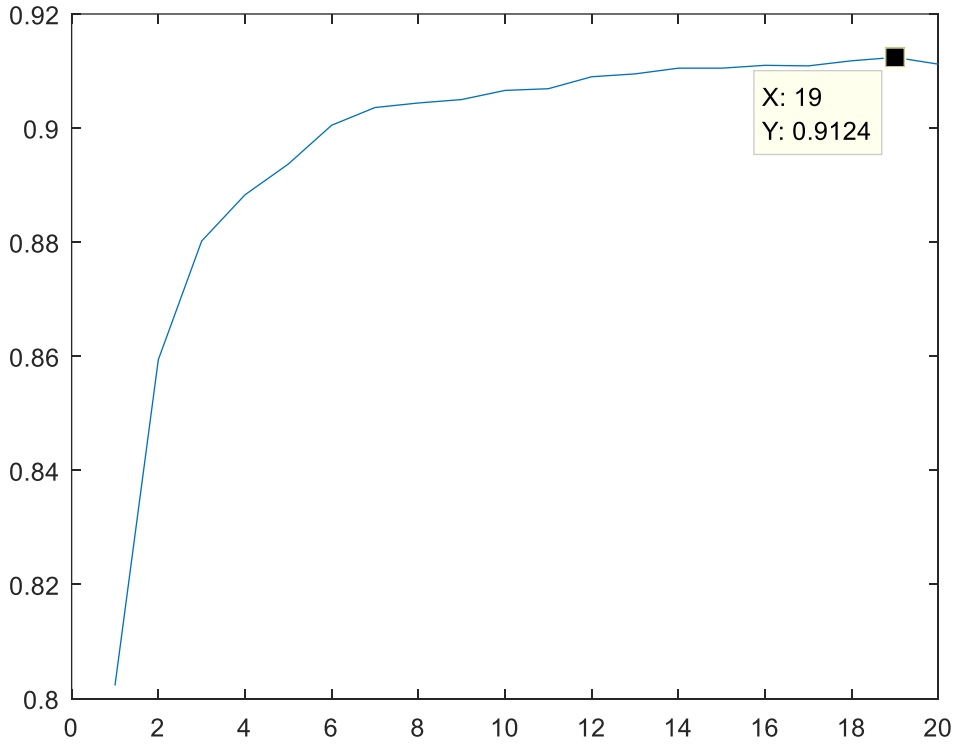
مومنتوم برابر ۰ در نظر گرفته شد. با توجه به نرخ یادگیری و سایز batch، در این مساله نیاز زیادی به مومنتم وجود نداشت.

تابع فعالسازی برابر tanh قرار گرفت که بر sigmoid برتری دارد.

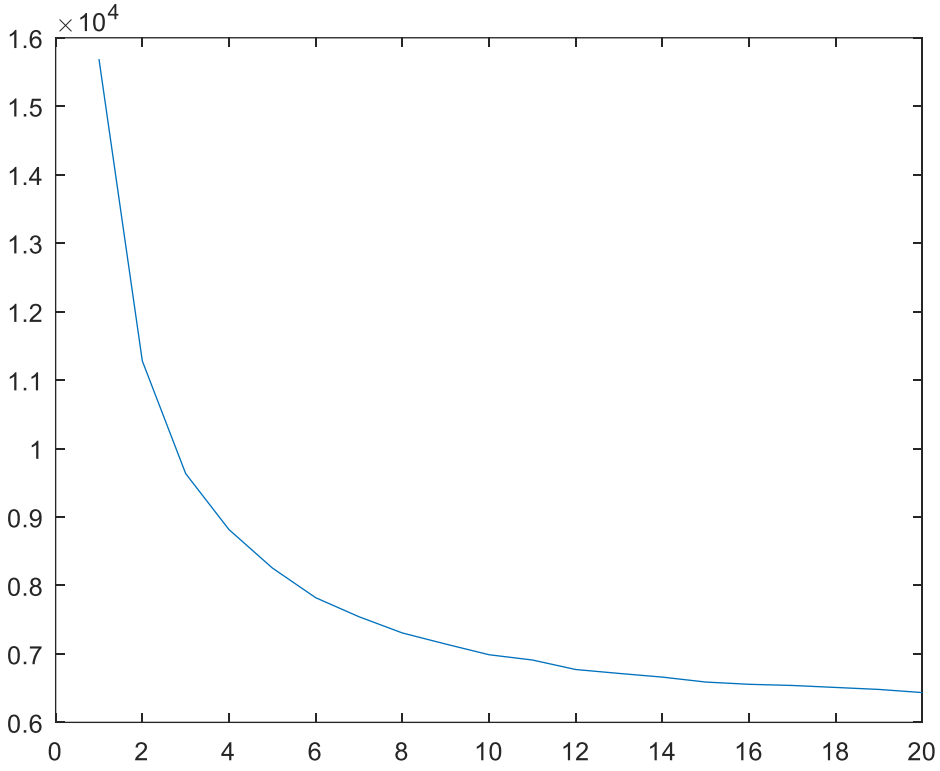
داده‌های خروجی به صورت one-hot کد شدند تا ترتیبی بین خروجی های مختلف در نظر گرفته نشود و ارتباطی بین آن ها نباشد.

برای پیش‌پردازش داده‌ها از نرمالیزاسون Z-Score استفاده شد که در این مساله سریع‌تر همگرا می‌شود.

نمودار بهترین دقت بدست آمده در شکل ۲۴ و ۲۵ رسم شده است. بهترین دقت بدست آمده برابر 91% است.



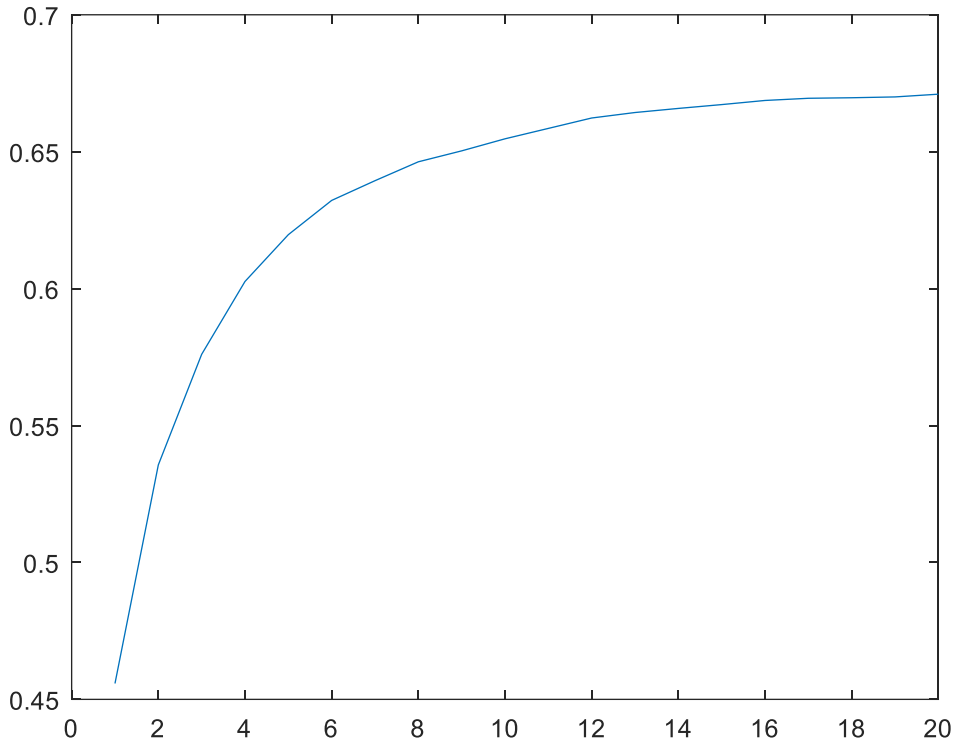

شکل ۲۴، نمودار بهترین دقت بدست آمده




شکل ۲۵، نمودار **Loss** در بهترین دقت

## سوال ۳

با استفاده از یک کلاسیفایر خطی، دقت طبقه بندی 66% بدست آمد که به مراتب از دقت 91% شبکه عصبی کمتر است. نمودار خطای کلاسیفایر خطی بر حسب epoch در شکل ۲۶ رسم شده است.




شکل ۲۶، نمودار دقت بر حسب تعداد **epoch** در کلاسیفایر خطی

## سوال ۴

ماتریس در هم‌ریختگی برای طبقه‌بندی با استفاده از شبکه عصبی:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 953 | 0 | 0 | 2 | 0 | 2 | 4 | 1 | 1 | 0 |
| 0 | 1115 | 2 | 0 | 0 | 0 | 4 | 0 | 1 | 0 |
| 8 | 1 | 922 | 4 | 4 | 1 | 4 | 6 | 9 | 1 |
| 1 | 2 | 8 | 901 | 0 | 5 | 2 | 9 | 6 | 5 |
| 0 | 1 | 2 | 0 | 909 | 0 | 10 | 1 | 2 | 13 |
| 5 | 1 | 1 | 16 | 1 | 755 | 9 | 3 | 7 | 9 |
| 9 | 4 | 1 | 0 | 3 | 14 | 898 | 0 | 2 | 0 |
| 0 | 13 | 12 | 3 | 3 | 0 | 0 | 902 | 0 | 12 |
| 4 | 7 | 4 | 5 | 7 | 8 | 5 | 3 | 848 | 2 |
| 8 | 4 | 0 | 12 | 15 | 2 | 0 | 7 | 3 | 897 |

شکل ۲۷، ماتریس درهم‌ریختگی، سطر **i** و ستون **j** نشان دهنده تعداد داده هایی است که در واقع برابر (**i-1**) بوده ولی (**j-1**) تشخیص داده شده است.

ماتریس درهم‌ریختگی برای طبقه‌بندی با استفاده از کلاسیفایر خطی:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 842 | 0 | 0 | 0 | 0 | 4 | 4 | 1 | 1 | 0 |
| 0 | 1034 | 1 | 0 | 0 | 0 | 2 | 0 | 0 | 0 |
| 5 | 13 | 613 | 9 | 3 | 0 | 25 | 1 | 10 | 3 |
| 1 | 0 | 4 | 634 | 0 | 1 | 1 | 9 | 5 | 2 |
| 0 | 6 | 2 | 0 | 678 | 0 | 1 | 1 | 1 | 13 |
| 3 | 5 | 2 | 14 | 2 | 398 | 9 | 5 | 10 | 6 |
| 6 | 5 | 3 | 0 | 4 | 7 | 766 | 0 | 0 | 0 |
| 2 | 16 | 1 | 2 | 6 | 0 | 0 | 717 | 0 | 18 |
| 3 | 11 | 4 | 2 | 4 | 11 | 6 | 3 | 473 | 2 |
| 4 | 1 | 0 | 3 | 13 | 0 | 0 | 29 | 0 | 542 |

شکل ۲۸، ماتریس درهم‌ریختگی، سطر **i** و ستون **j** نشان دهنده تعداد داده هایی است که در واقع برابر (**i-1**) بوده ولی (**j-1**) تشخیص داده شده است.

## سوال ۵

فرمول‌های محاسبه‌ی گرادیان:

۱) کلاسیفایر خطی (رگراسیون لاجیستیک)، تابع خطا آنتروپی:

$$\frac{\partial J(\theta)}{\partial \theta \mathrm{j}} = \frac{\partial}{\partial \theta j}\frac{1}{2}(h_\theta(x)-y)^2 = 2*\frac{1}{2}*(h_\theta-y).\frac{\partial}{\partial \theta j}(h_\theta(x)-y)$$

$$= (h_\theta-y).\frac{\partial}{\partial \theta j}(\sum \theta_i x_i - y) = (h_\theta-y)x_j$$

۲) شبکه عصبی:

خروجی شبکه: $y_p$

خروجی لایه‌ی پنهان قبل از فعالساز: $Z^{(1)}$، پس از فعالساز: $Y_h$

خروجی لایه‌ی خروجی قبل از فعال‌ساز: $Z^{(2)}$

تابع خطا: $\frac{1}{N}\sum(y_i - target_i)^2$

بایاس لایه پنهان: $b^{(1)}$، بایاس لایه‌ی خروجی: $b^{(2)}$

تابع فعالساز: f

وزن لایه‌ی اول: $w^{(1)}$، وزن لایه‌ی خروجی: $w^{(2)}$

روابط گرادیان به صورت زیر است:

$$\frac{\partial L}{\partial y_p} = (y_p - target)$$

$$\frac{\partial L}{\partial Z^{(2)}} = f'(y_p) * \frac{\partial L}{\partial y_p}$$

$$\frac{\partial L}{\partial w^{(2)}} = y_h^T * \frac{\partial L}{\partial Z^{(2)}}$$

$$\frac{\partial L}{\partial b^{(2)}} = \frac{\partial L}{\partial Z^{(2)}}$$

$$\frac{\partial L}{\partial y_h} = \frac{\partial L}{\partial Z^{(2)}} * w^{(2)T}$$

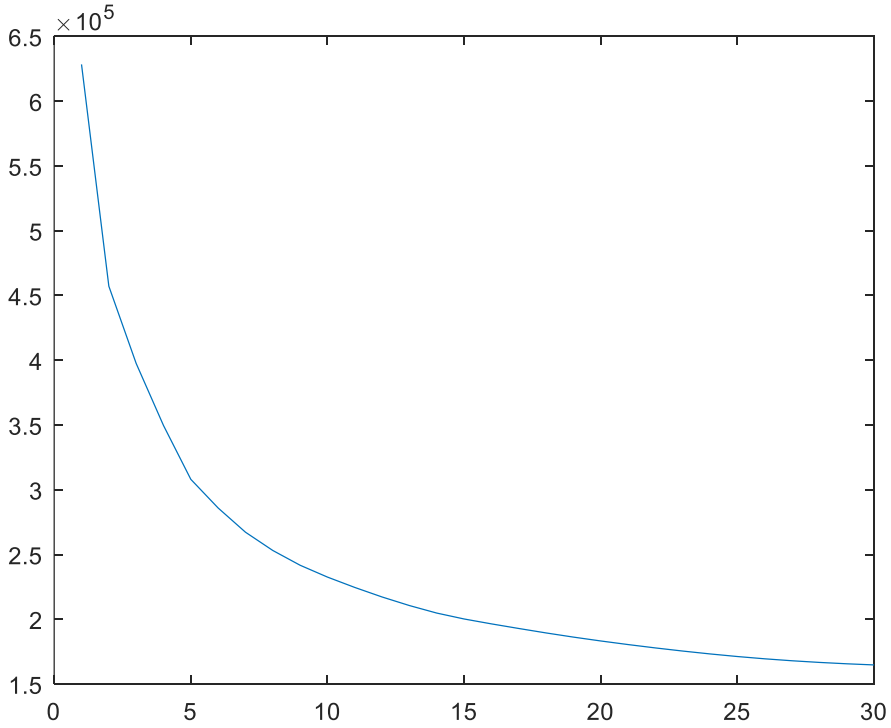
$$\frac{\partial L}{\partial Z^{(1)}} = f'(y_h) * \frac{\partial L}{\partial y_h}$$

$$\frac{\partial}{\partial w^{(1)}} = X^T * \frac{\partial L}{\partial Z^{(1)}}$$

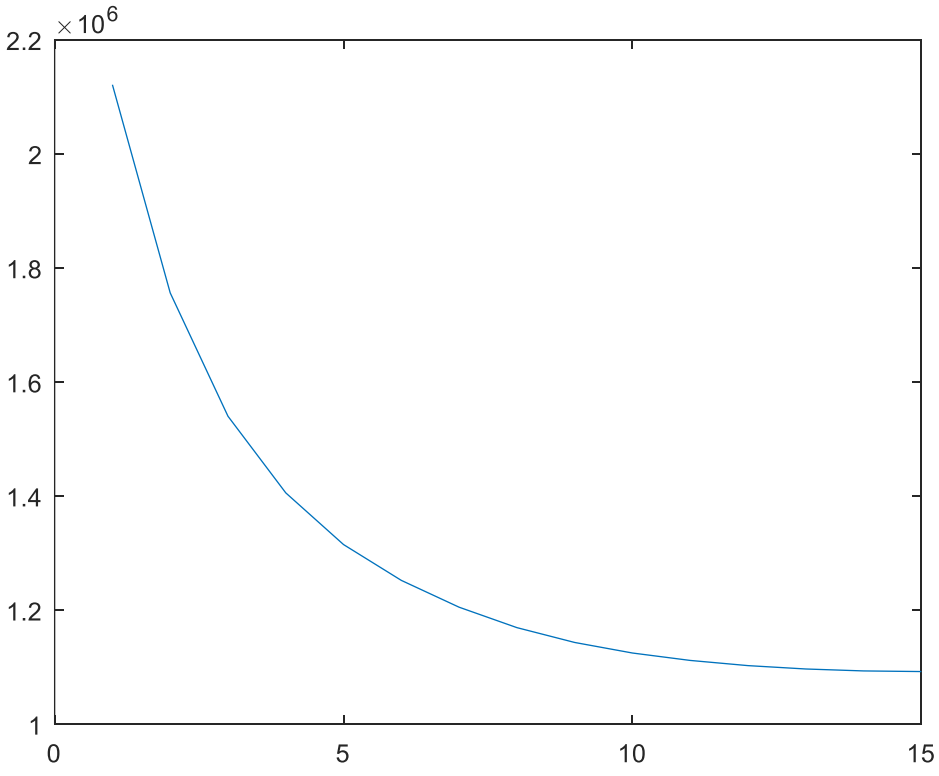
$$\frac{\partial L}{\partial b^{(1)}} = \frac{\partial L}{\partial Z^{(1)}}$$

## سوال ۶

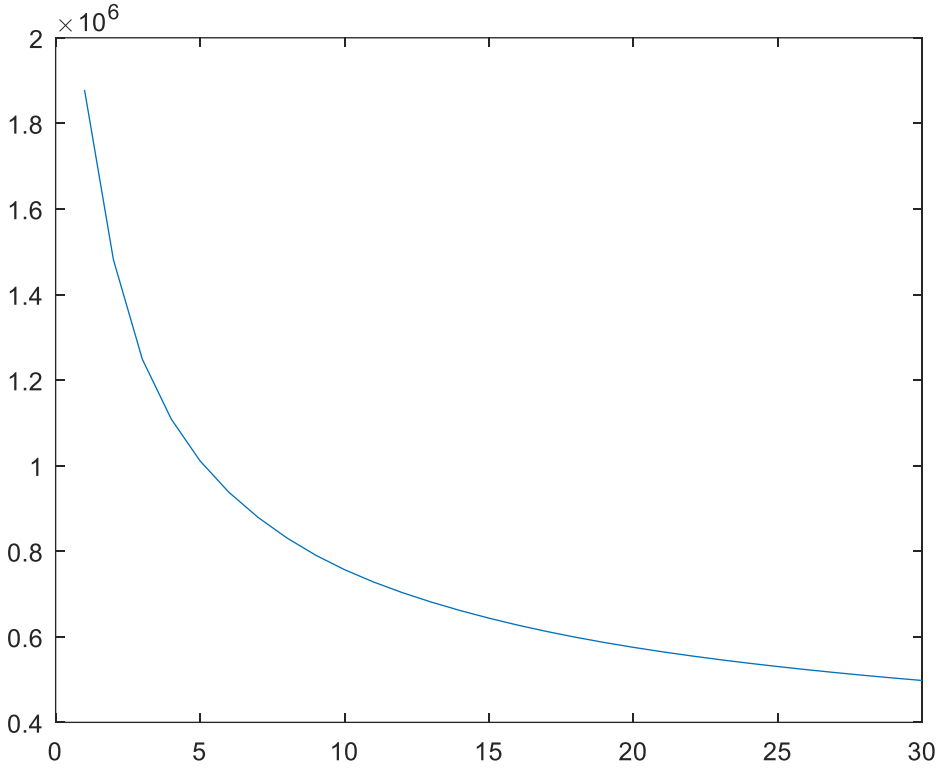
با استفاده از یک لایه‌ی پنهان که تعداد نورون‌های آن با توجه به میزان فشرده‌سازی مورد نظر انتخاب می‌شود، میتوان ابعاد داده را کاهش داد. در اینجا به ازای سه حالت مختلف ۶۴، ۱۲۸ و ۲۵۶ نورون در لایه‌ی پنهان ابعاد داده کاهش داده شده است. بدیهی است به طور کلی هر چه از تعداد نورون‌های بیشتری استفاده شود، می‌توان با دقت بهتری داده‌ها را بازیابی کرد. تعداد نورون‌ها باید به گونه‌ای انتخاب شود که با کمترین تعداد ممکن به دقت مورد نظر دست یافت. در اموزش شبکه از نرمالیزاسیون max-min استفاده شده است. نمودار Loss بر حسب تعداد epoch در شکل‌های ۲۹، ۳۰ و ۳۱ رسم شده است.




شکل ۲۹، نمودار **Loss** بر حسب تعداد **epoch** برای حالت ۲۵۶ نورون در لایه‌ی پنهان



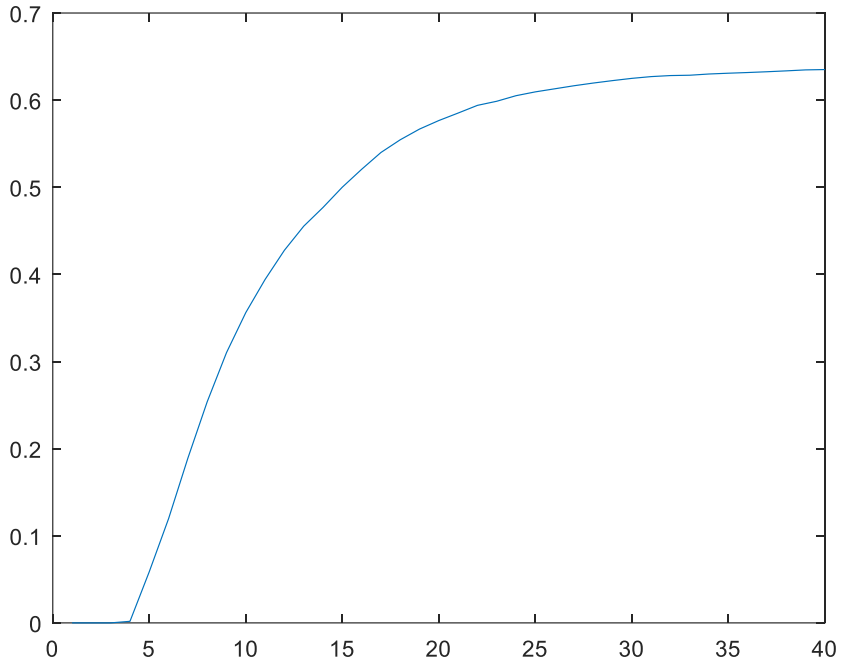

شکل ۳۰، نمودار **Loss** بر حسب تعداد **epoch** برای حالت ۶۴ نورون



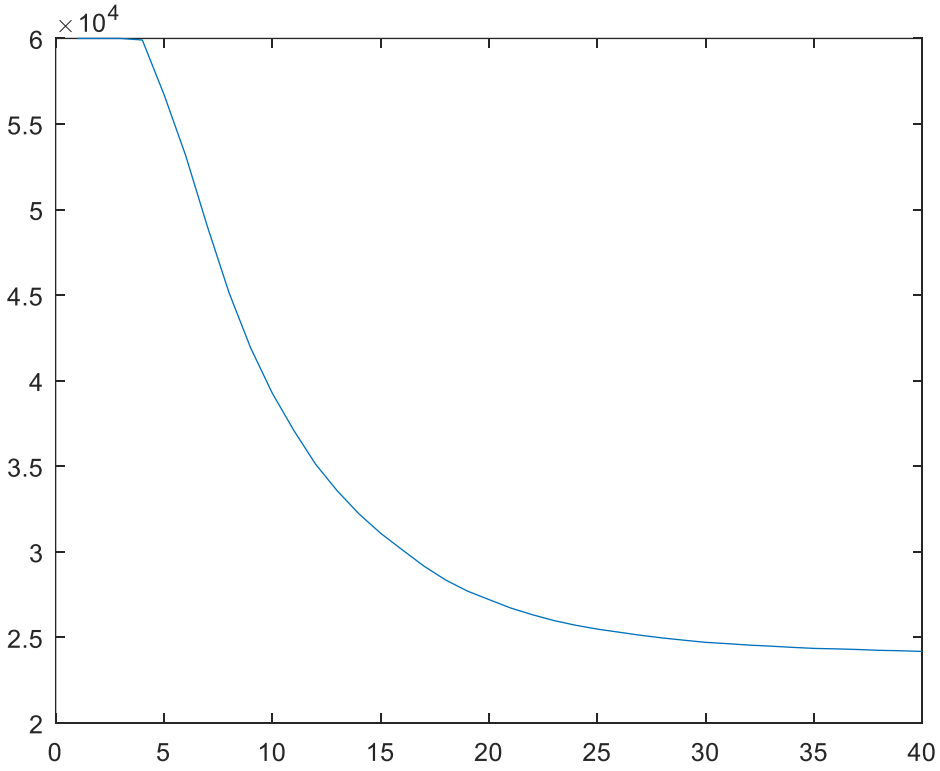

شکل ۳۱، نمودار **Loss** برای ۱۲۸ نورون

با توجه به این که در ۲۵۶ نورون، دقت بالاتر است از آن استفاه شده است. همچنین برای اینکه فشرده‌سازی قابل ملاحظه‌ای وجود داشته باشد، از نورون های بیش‌تر استفاده نشده است.

از خروجی ۲۵۶ بعدی لایه‌ی پنهان، برای اموزش شبکه استفاده شده است. نمودار دقت و Loss برای این حالت در شکل ۳۲ و ۳۳ رسم شده است. با توجه به نمودارها مشخص است که دقت شبکه به مقدار قابل توجهی کاهش یافته است از آنجایی که از feature های تقریبی استفاده شده است، بخشی از اطلاعات موجوددر ورودی از بین رفته است و مقداری نویز در نتیجه‌ی این تغییر به داده‌های ورودی اضافه شده است که منجر به کاهش دقت شده است.




شکل ۳۲، نمودار دقت با **feature** های فشرده شده




شکل ۳۳، نمودار **Loss** با **feature** های فشرده شده