بنام خدا

گزارش تمرین کامپیوتری دوم

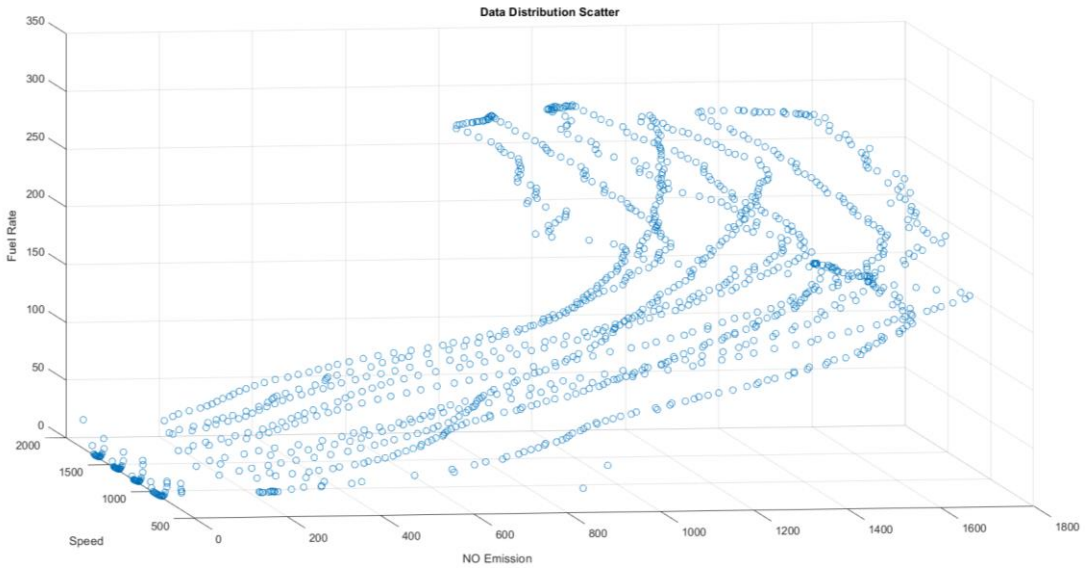
هوش محاسباتی

آرشام لؤلؤهری

۹۹۱۰۲۱۵۶

سوال ۱:

الف) توزیع سه بعدی داده ها به صورت زیر است:



ب) با تعیین اندیس های ۱ تا n=700، این کار در کد انجام شده است.

ج) در رگرسیون خطی، فرض بر رابطه خطی زیر بین ورودی ها و خروجی وجود دارد:

$$y = f(x_1, x_2) = a + bx_1 + cx_2$$

برای مینیمم شدن خطای مجموع مربعات، باید مشتق تابع خطا نسبت به هرسه متغیر a,b,c صفر شود:

$$\frac{\partial F}{\partial a} = 0, \quad \frac{\partial F}{\partial b} = 0, \quad \frac{\partial F}{\partial c} = 0$$

$$F(a,b,c) = \sum_{i=1}^{n} \left(f\left(x_1(i), x_2(i)\right) - y_i\right)^2$$

از ساده سازی این سه معادله، به روابط زیر میرسیم (که منظور از x,y همان x_1, x_2 و منظور از z همان y در روابط بالاست):

$$na + \left(\sum_{i=1}^{n} x_i\right) b + \left(\sum_{i=1}^{n} y_i\right) c = \sum_{i=1}^{n} z_i$$

$$\left(\sum_{i=1}^{n} x_i\right) a + \left(\sum_{i=1}^{n} x_i^2\right) b + \left(\sum_{i=1}^{n} x_i y_i\right) c = \sum_{i=1}^{n} z_i x_i$$

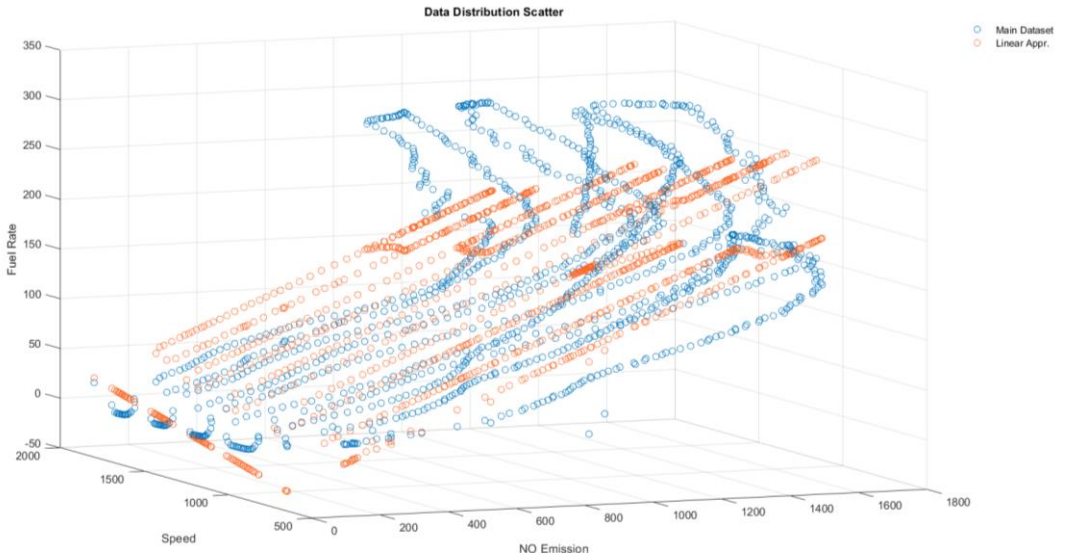
$$\left(\sum_{i=1}^{n} y_i\right) a + \left(\sum_{i=1}^{n} x_i y_i\right) b + \left(\sum_{i=1}^{n} y_i^2\right) c = \sum_{i=1}^{n} z_i y_i$$

این دستگاه خطی را با استفاده از linsolve در متلب حل کرده و ضرایب a,b,c را بدست می آوریم.

سپس ورودی ها به تابع خطی f داده شده و خروجی های شبکه رگرسیون بدست می آید. سپس مقدار خطای MSE به صورت زیر بدست می آید (عدد اول روی داده های آموزشی و عدد دوم روی داده های validation):

```
train_MSE_lin =

   3.4647e+03


val_MSE_lin =

   3.6694e+03
```

داده های تقریب زده شده با این روش را روی داده های اصلی انداخته و مقایسه میکنیم. بیشترین خطا را در بخش انحنای بالای توزیع میتوان مشاهده کرد:



د) در روش رگرسیون logistic تابع f به صورت زیر تعریف میشود:

$$y = f(x_1, x_2) = \frac{Y}{1 + e^{a+bx_1+cx_2}}$$

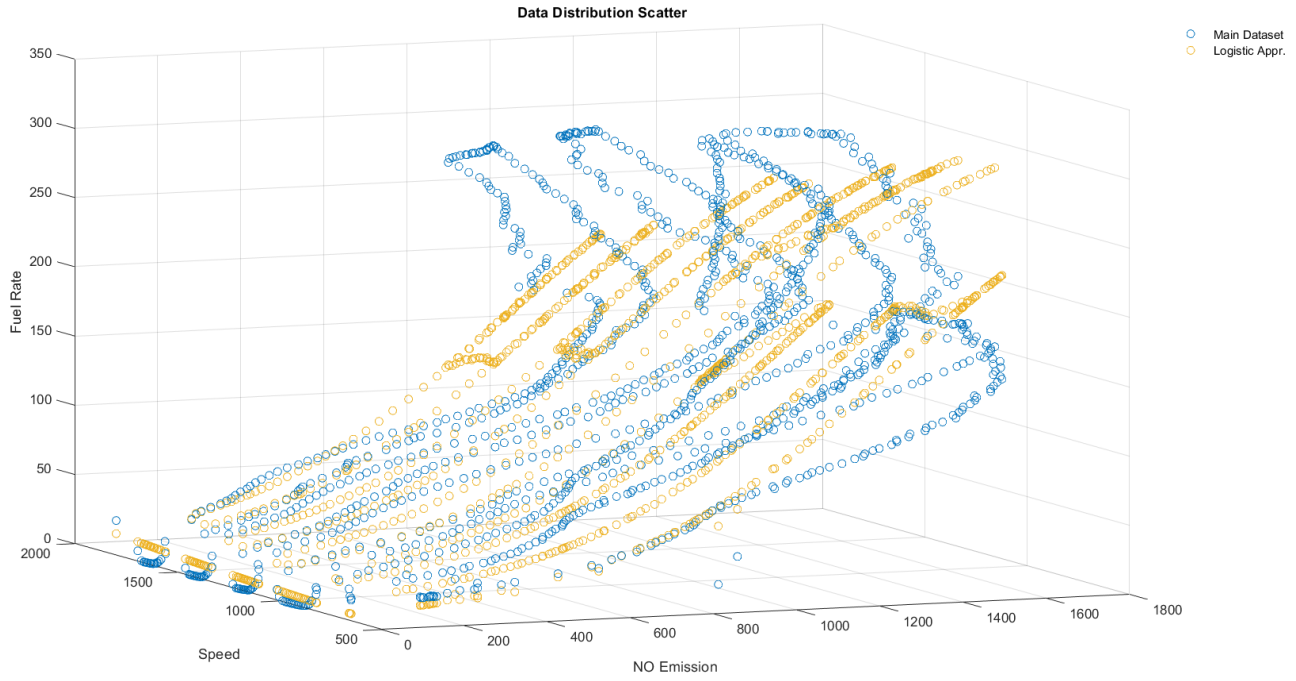
پس داریم:

$$z = \ln\left(\frac{Y - y}{y}\right) = a + bx_1 + cx_2$$

پس کافیست از روی خروجی های مورد نظر y، z ها را بسازیم و سپس به روش قبل، با فرض خروجی های z، پارامترهای a,b,c را محاسبه کنیم. مقدار Y برابر ماکزیمم خروجی ها (y ها) در داده های آموزشیست، زیرا کران پایین و بالای تابع f به ترتیب صفر و Y است (Y را مقداری بیشتر از ماکزیمم دیتاها میگیریم تا برای آن نقطه ی ماکزیمم، به لگاریتم صفر نرسیم). پس از ساخت شبکه و استخراج پارامترها، a,b,c را در رابطه ی تابع f قرار داده و MSE را حساب میکنیم (عدد اول برای داده های آموزشی و عدد دوم برای داده های اعتبارسنجی):

```
train_MSE_log =

   3.9562e+03


val_MSE_log =

   4.2846e+03
```

داده های تقریب زده شده با این روش را روی داده های اصلی انداخته و مشاهده میکنیم که در خروجی های بزرگتر، تفاوت فاحشی بینشان وجود دارد:

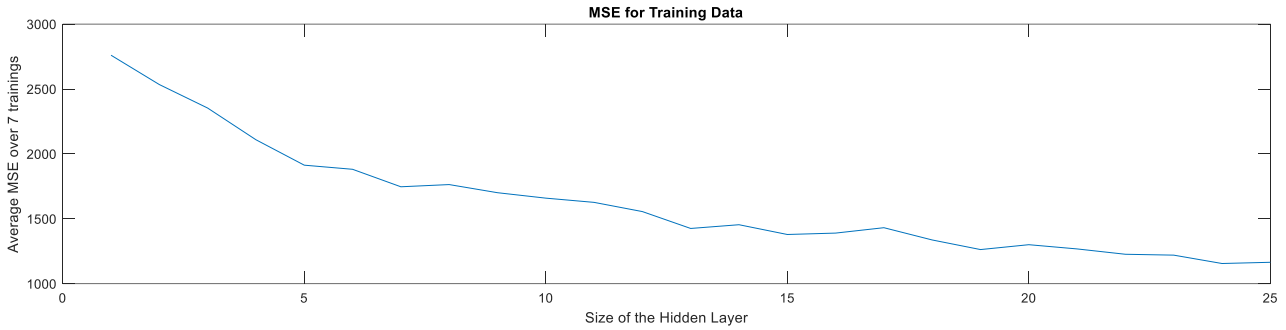


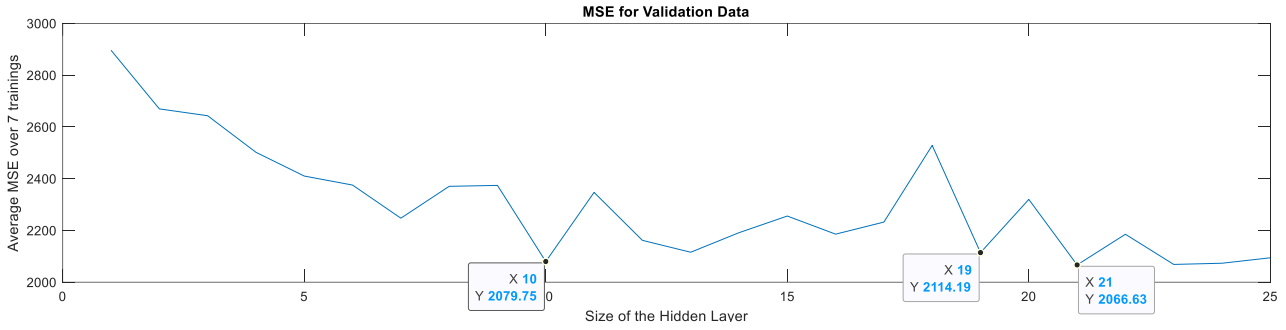
تا اینجا مشاهده شد که به خطاهای زیادی در هردو روش رسیدیم. در هردو روش، مطابق انتظار، خطای داده آموزشی کمتر از اعتبارسنجی بود اما هر ۴ مقدار خطا بسیار زیاد بود. طبق توزیع داده ها در بخش الف، نمیتوان داده ها را به صورت خطی (با یک

صفحه) به خوبی تقریب زد. رگرسیون در مقادیر کوچک عملکرد بهتری دارد اما در خروجی های بزرگ (قسمت انحنای توزیع داده ها)، خطا به قدری زیاد میشود که MSE را حتی از روش خطی نیز بیشتر میکند.

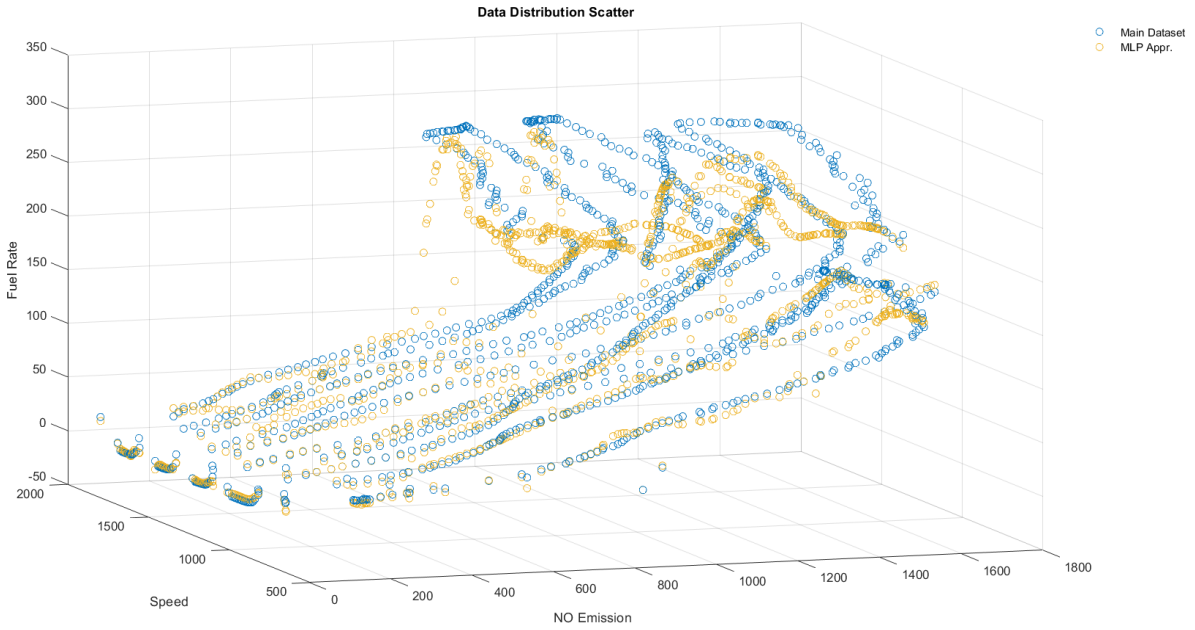
در کل هیچ یک از دو روش به اندزه کافی دقت ندارند.

۵) در این روش برای پیدا کردن تعداد نورون بهینه برای لایه پنهان، تعداد آنها را در یک for از ۱ تا ۲۵ تغییر میدهیم (با توجه به فرم کلی توزیع داده ها، بنظر می آید تا حدود ۲۵ نورون پنهان بتواند تا حدو خوبی پیچیدگی های توزیع را استخراج کرده و مدل خوبی فیت کند). به ازای هر تعداد نورون، شبکه ای ساخته و آن را ۷ بار آموزش میدهیم تا میانگین MSE بین این ۷ بار را گزارش کنیم. هربار داده های آموزشی به شبکه داده شده و شبکه learn شده و سپس مقادیر خروجی محاسبه میشوند (هم برای داده های آموزشی و هم برای داده های اعتبارسنجی). سپس با استفاده از Perform، خطای MSE بطور جداگانه برای داده های آموزشی و validation محاسبه میشود. این خطاها در نهایت روی کل ۷ بار آموزش شبکه میانگین گرفته میشود و در نهایت یک عدد بعنوان خطا برای تعداد نورون مشخص خواهیم داشت. نمودار خطا برحسب سایز لایه پنهان را برای داده های آموزشی (نمودار اول) و داده های اعتبارسنجی (نمودار دوم) رسم میکنیم:





مشاهده میشود که کمترین خطا روی داده های اعتبارسنجی، به ازای ۲۱ نورون پنهان رخ داده است (البته اگر کم بودن تعداد نورون اهمیت زیادی داشته باشد، میتوان از ۱۰ نورون نیز استفاده کرد، چون میانگین MSE آن تفاوت چندانی با ۲۱ نورون ندارد). به ازای این سایز، میانگین مقادیر تخمین زده شده برای خروجی را، روی توزیع اولیه‌ی داده ها انداخته و رسم میکنیم (البته که این مقدار میانگین ممکن است در هیچ یک از آموزش ها تولید نشده باشد، ولی برای مشاهده‌ی برایندی از عملکرد شبکه، آن را رسم میکنیم):



مشاهده میشود که پیچیدگی توزیع به خوبی مدل شده و بخصوص در انحنای بالای توزیع، شبکه عملکرد بهتری نیست به رگرسیون ها داشته است. ضمنا مقدار MSE با استفاده از دستور perform محاسبه شده است. برای داده های آموزش و اعتبارسنجی، به ازای ۲۱ نورون پنهان به MSE های زیر رسیده ایم:

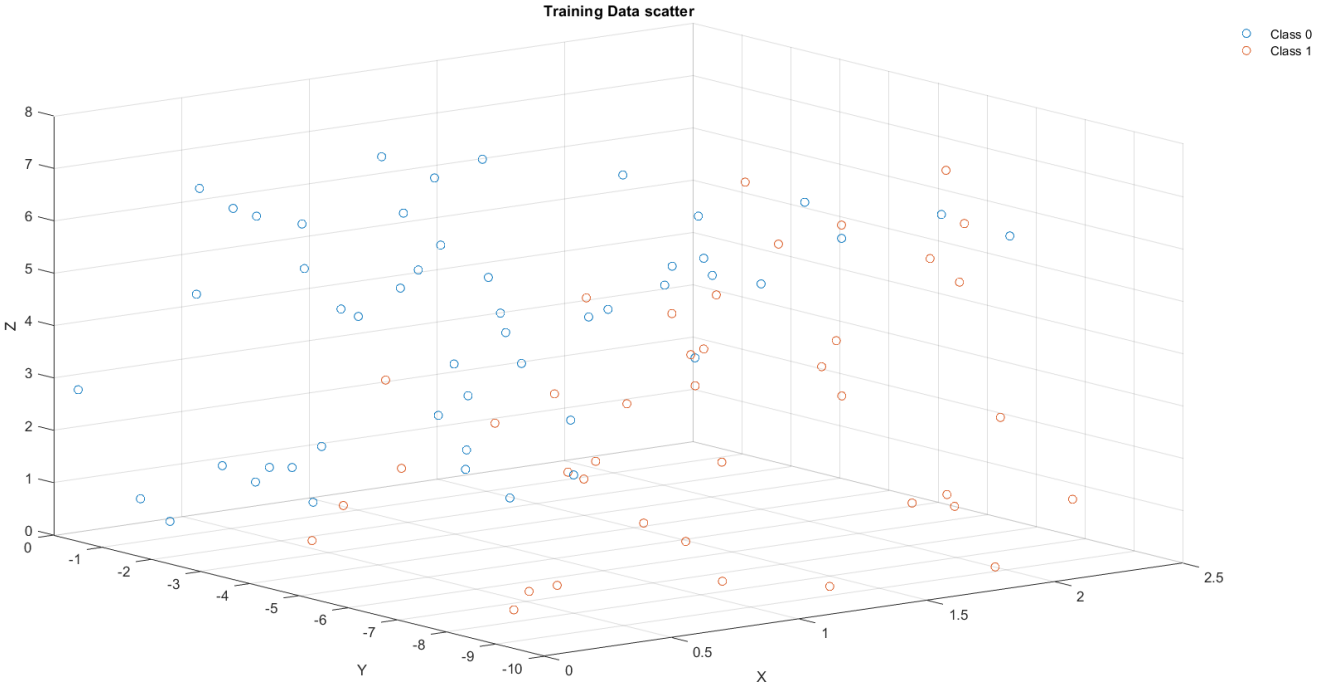
```
final_train_perf =

   1.2681e+03


final_val_perf =

   2.0666e+03
```

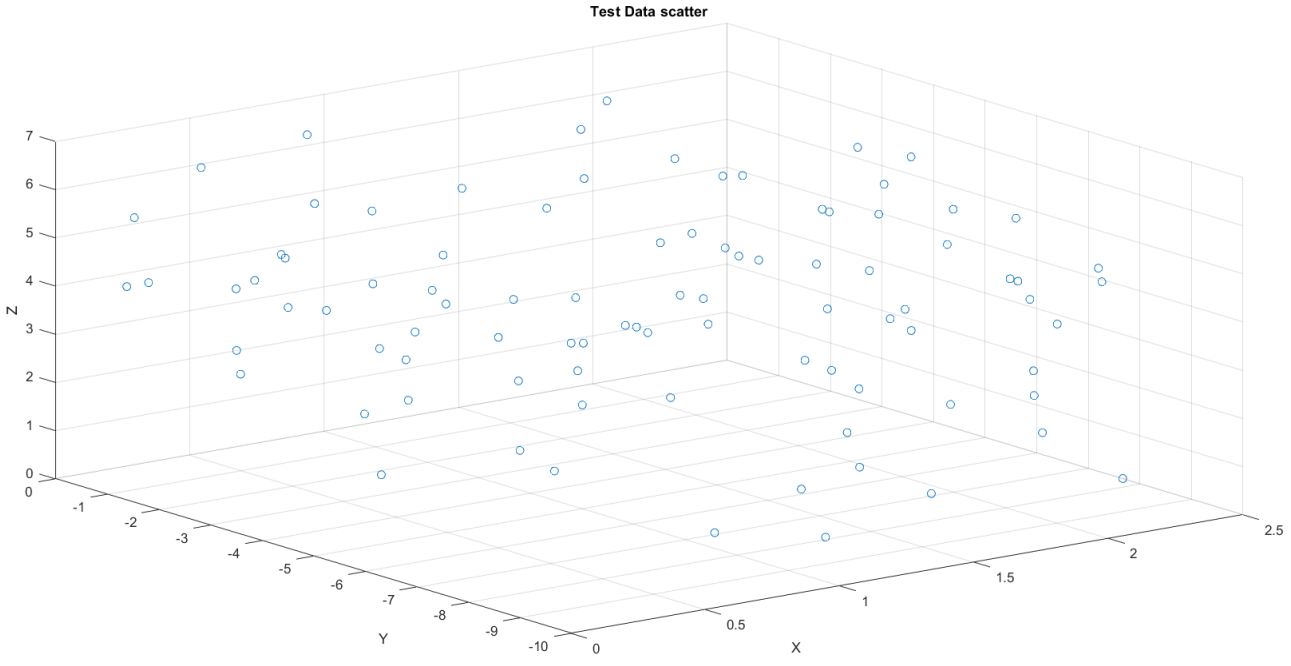
که هردو مقدار، بطور قابل توجهی کمتر از تقریب های رگرسیون هستند. پس این روش، بهتر از دو روش دیگر توانسته به توزیع فیت شود.

سوال ۲:

الف) لیبل های ۰ و ۱ در داده ها را به ترتیب با کلاس های ۰ و ۱ متناظر میکنیم.

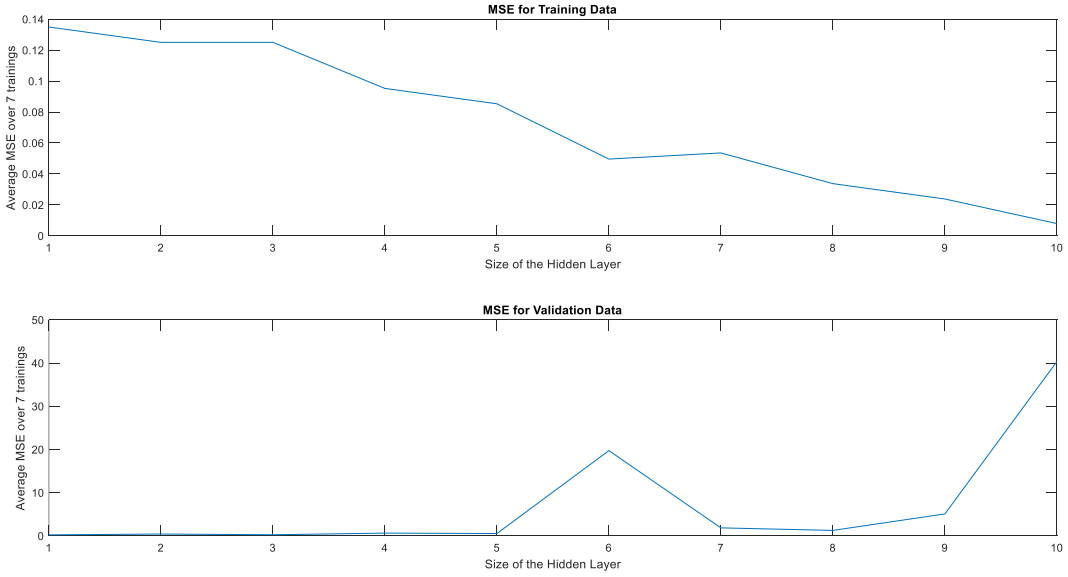
ابتدا داده های TrainData و TestData را رسم میکنیم:





میبینیم که تا حد خوبی داده ها از یکدیگر جدا هستند. پس انتظار میرود پیچیدگی شبکه موردنیاز برای طراحی، نسبتا کم باشد. به همین دلیل، این بار برخلاف سوال قبل، تعداد نورون های لایه پنهان را تنها تا ۱۰ زیاد میکنیم.

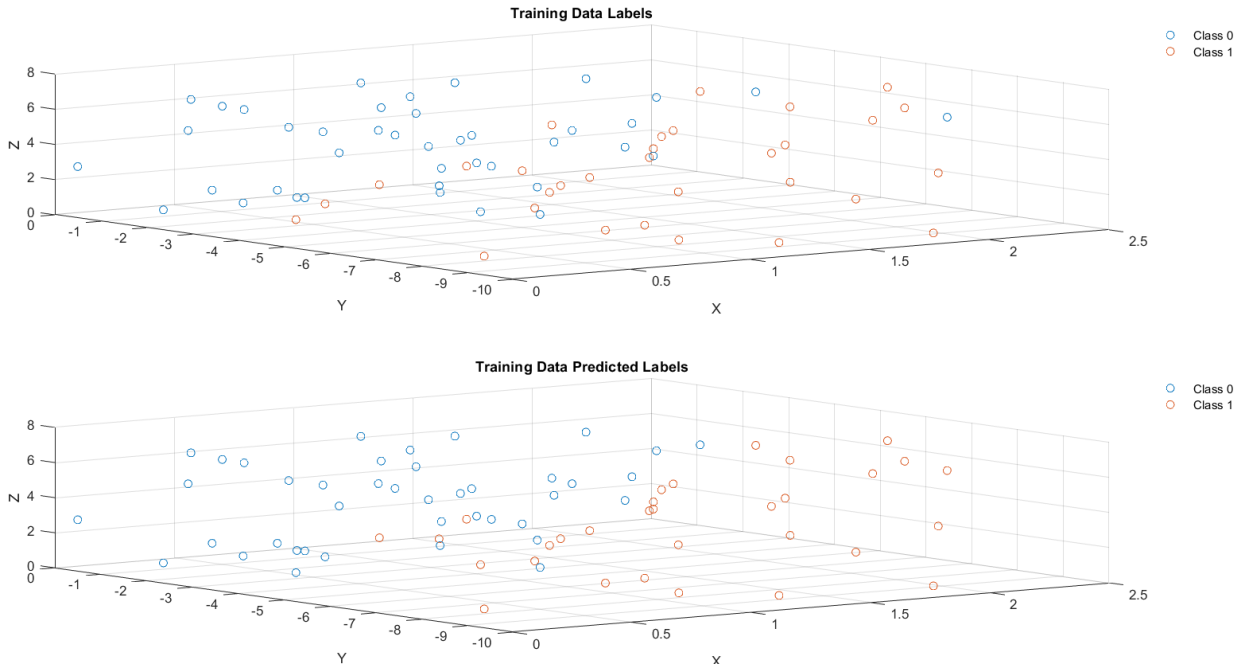
این بار برای طراحی شبکه از feedforwardnet استفاده میکنیم. این بار خروجی های شبکه، آرایه های صفر و یک به ازای داده های آموزشی یا اعتبارسنجی هستند. توابع train, perform و بقیه روند، مشابه سوال قبل طی میشود. در ابتدا، ۲۰ درصد داده ها بعنوان داده های validation به صورت رندوم انتخاب شده و از فرایند آموزش شبکه خارج میشوند. در نهایت مشابه سوال قبل، نمودار میانگین MSE برحسب تعداد نورون مورد استفاده را، برای داده های training و نیز validation رسم میکنیم:



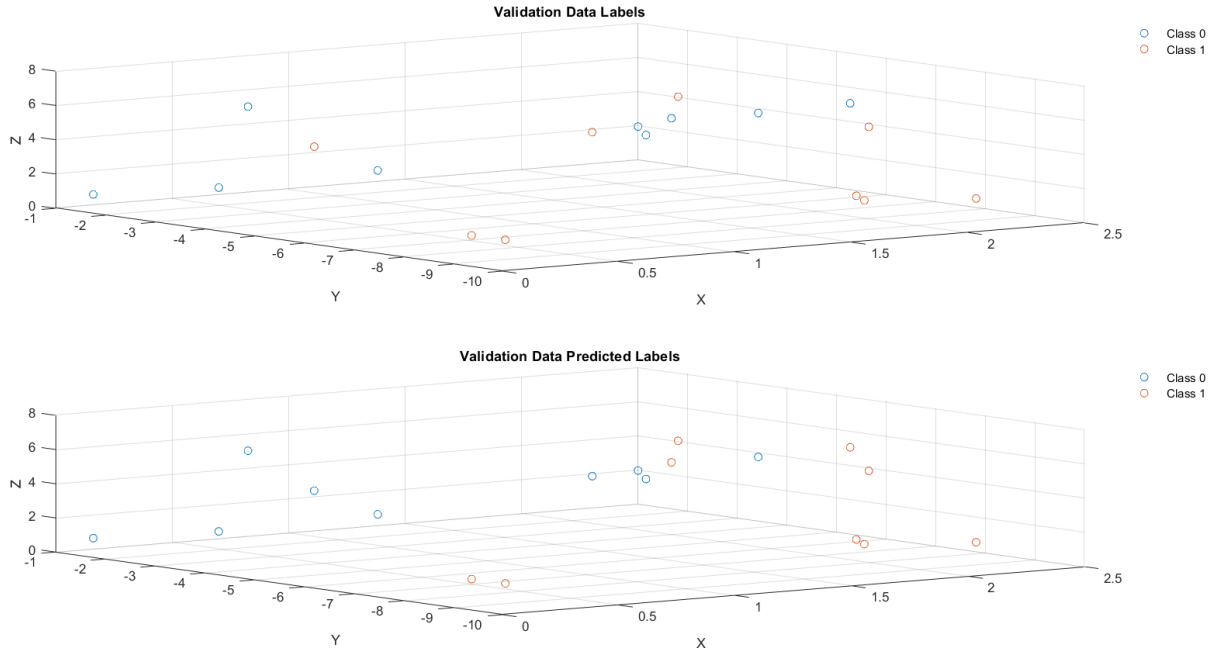
پس با یک نورون پنهان میتوان به کمترین خطای MSE برای داده های validation رسید. حال یکبار دیگر شبکه ای با یک نورون پنهان را با داده های آموزشی train کرده و خروجی های پیش بینی شده برای داده های training, validation, test را محاسبه میکنیم. برای داده های train, validation، مقدار MSE به صورت زیر بدست می آید:

```
train_perf =

    0.1389


val_perf =

    0.2222
```

حال داده های آموزشی را، یکبار با لیبل گذاری اصلی (نمودار اول) و یکبار با لیبل های شبکه (نمودار دوم) رسم میکنیم:



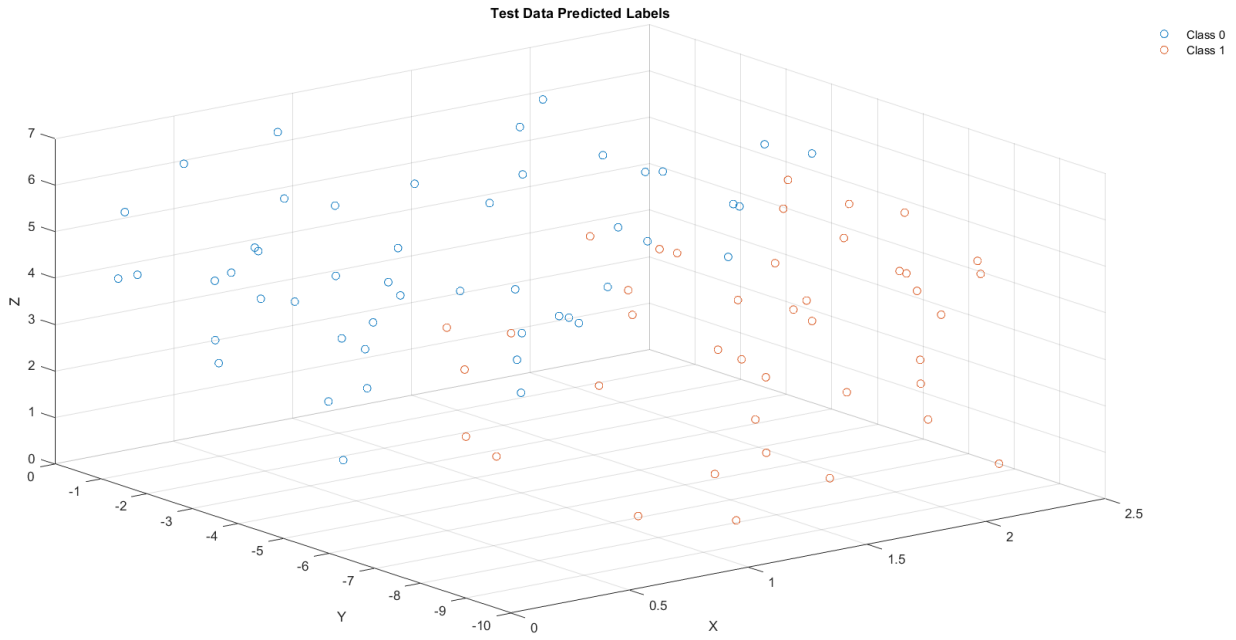
همین کار را برای داده های validation نیز انجام میدهیم:



داده های آموزشی و اعتبارسنجی، هردو تا حد خوبی به درستی تفکیک شده اند و باعث شده که مطابق اعداد بالا، MSE کمی حاصل شود. در داده های validation تنها چهار داده به اشتباه طبقه بندی شده اند و این نشان میدهد شبکه با همان یک نورون به خوبی عمل کرده است. به عبارت دیگر، دقت (accuracy) برابر است با:

$$acc = \frac{14}{18} = 77.8\%$$

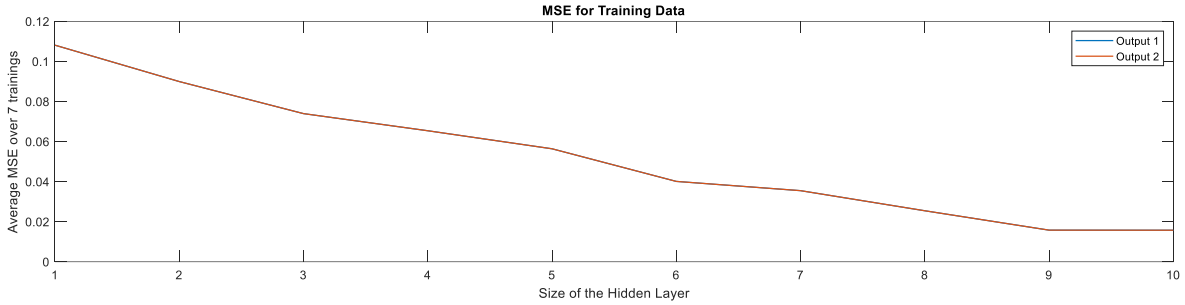
حال داده های تست را به این شبکه داده، و طبقه بندی خروجی را به صورت زیر رسم میکنیم:

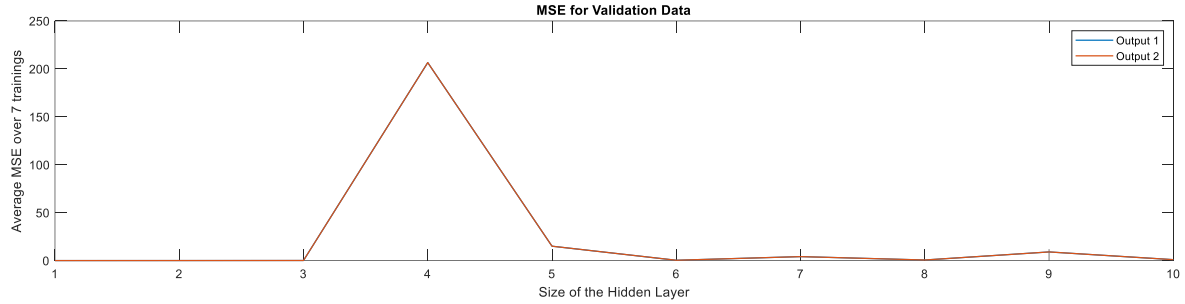


مشاهده میشود علی رغم برخی تفاوت ها، محل کلی داده های کلاس ۰ و ۱، تقریبا مشابه داده های آموزشی قرار دارد.

در شبکه حاصله، سه لایه داریم (ورودی، خروجی و یک لایه پنهان). برای تابع فعالسازی از پیش فرض های توابع متلب استفاده شده (sigmoid در لایه پنهان و خطی همانی در لایه خروجی). یک نورون ورودی و یک خروجی، و یک نورون پنهان داریم.

ب) این بار لیبل های داده های یادگیری و اعتبارسنجی را به گونه ای میدهیم که برای هر نقطه، یک بردار دوبعدی داریم که یکی از دو المان آن (المان کلاس مربوطه) ۱، و دیگری صفر است. مشابه قبل با feedforwardnet شبکه را میسازیم و نمودارهای قبل را تکرار میکنیم. ابتدا نمودار MSE برحسب تعداد نورون پنهان:

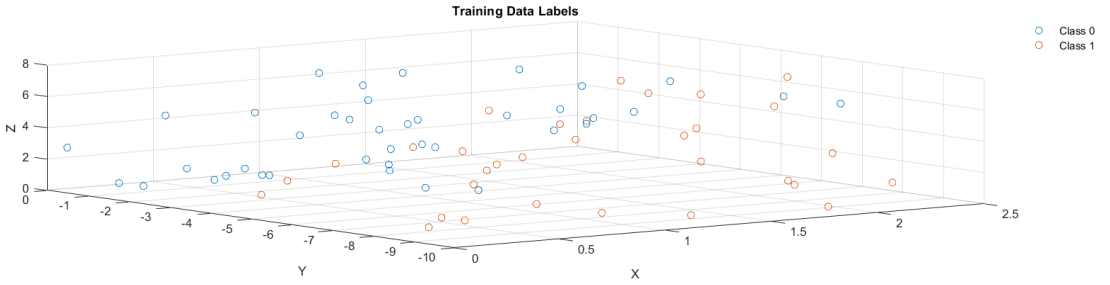


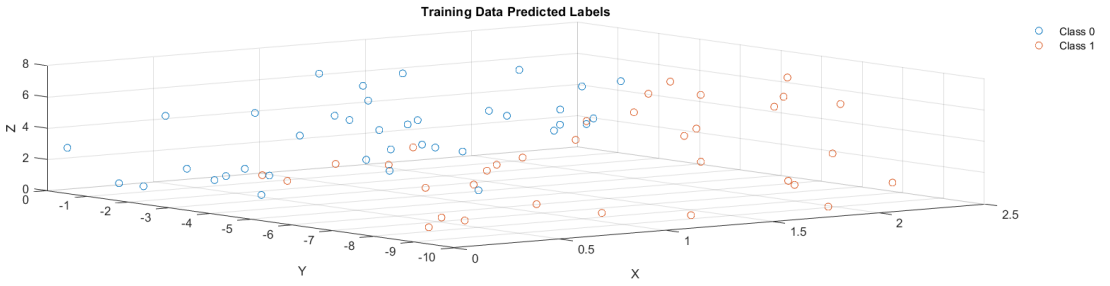


مجددا با یک نورون کمترین خطای MSE بدست می آید. البته این بار خطای MSE به صورت میانگین MSE در دو خروجی تولیدی محاسبه شده است.

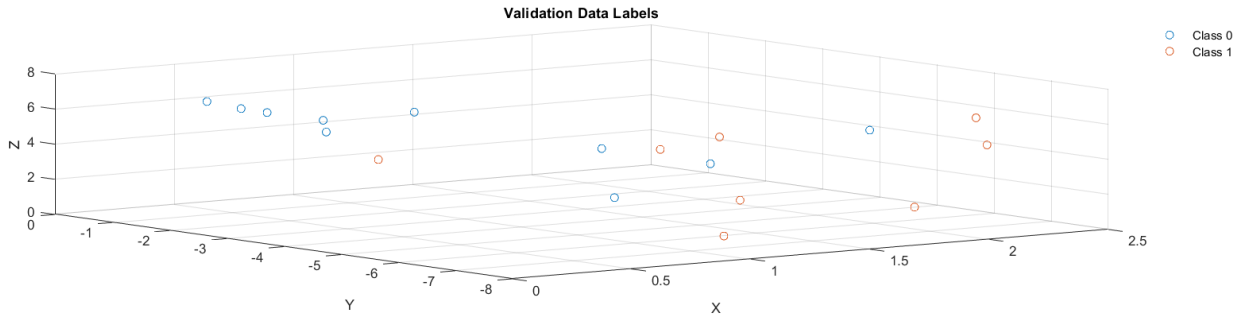
حال با یک نورون، شبکه را آموزش داده و داده ها را مشابه بخش الف لیبل گذاری میکنیم. البته این بار معیار تعیین لیبل بدین شکل است که اگر خروجی اول، بزرگتر از خروجی دوم باشد، نقطه به کلاس صفر، و در غیر این صورت به کلاس یک نظیر میشود.

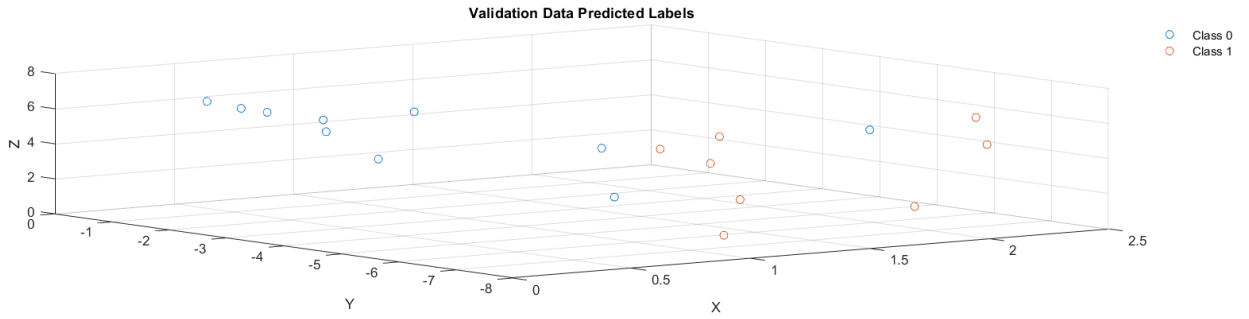
برای داده های آموزشی:





برای validation:





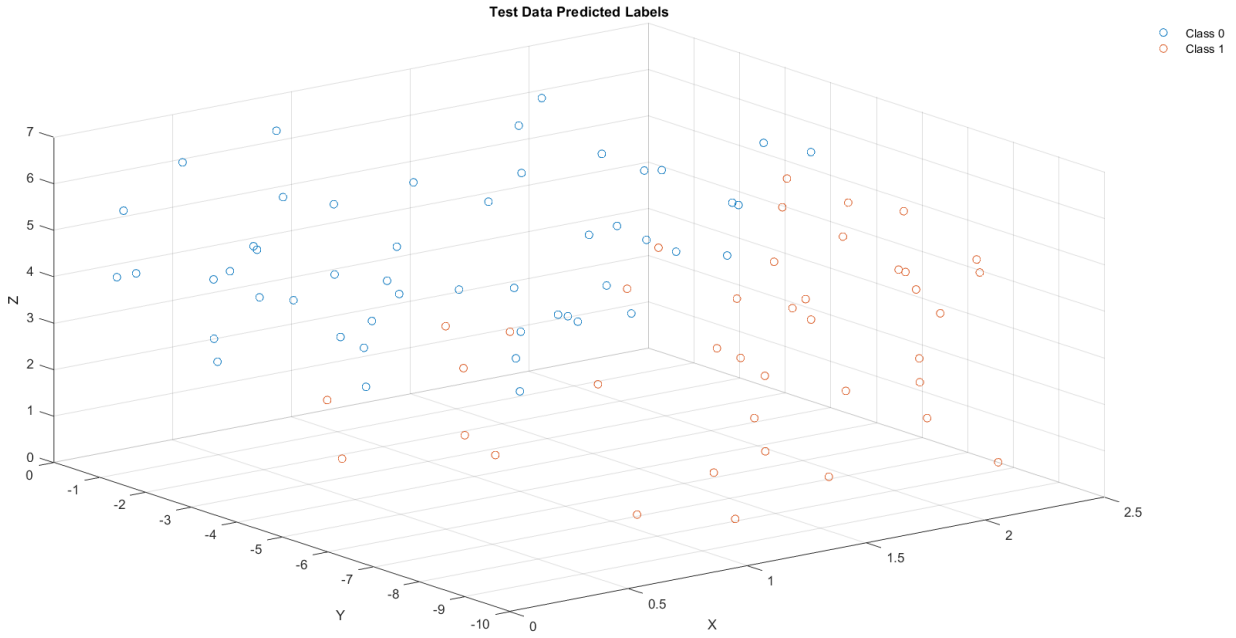
مشاهده میشود با دقت خوبی داده ها تفکیک شده اند و در validation تنها دو نقطه طبقه بندی اشتباه دارند. پس

$$acc = \frac{16}{18} = 88.9\%$$

که بیشتر از روش قبل است. خطای MSE برای داده های آموزشی و validation به صورت زیر است که نسبت به روش قبل کمتر است:

```
train_perf =

    0.3919


val_perf =

    0.4229
```

در نهایت داده های تست به صورت زیر لیبل گذاری میشوند:



که شباهت بسیار زیادی به لیبل گذاری های روش الف دارد (بجز تعداد اندکی داده).

در شبکه حاصله، سه لایه داریم (ورودی، خروجی و یک لایه پنهان). برای تابع فعالسازی از پیش فرض های توابع متلب استفاده شده (sigmoid در لایه پنهان و خطی همانی در لایه خروجی). همانطور که ذکر شد، یک نورون پنهان و دو نورون خروجی داریم.

توجه داریم در هردو روش، طبق نمودارهای MSE بر حسب تعداد نورون، میتوانستیم با تعداد نورون بیشتری نیز تقریبا به خطای MSE مشابه و اندکی برسیم، به نحوی که خطای داده های آموزشی نیز کاهش یابد. اما طبق خواسته سوال، تعداد نورون ها را برابر با حالتی گرفتیم که کمترین خطا در داده های validation بدست آید.

بعنوان یک مقایسه، هردو روش به خوبی توانستند داده های اعتبارسنجی را با دقت بالایی طبقه بندی کنند. در نتایج بالا دقت روش دوم کمی بهتر بود اما در حالت کلی برتری چندانی وجود ندارد. روش دوم از این جهت برتری دارد که بیانی احتمالاتی میدهد و مشخص میکند که هر داده، با چه احتمالی در کدام کلاس قرار دارد. بدین ترتیب ضریب اطمینانی از لیبل گذاری خود نیز خواهیم داشت. برای مثال اگر بردار خروجی مربوط به یک داده، اعداد 0.51, 0.49 باشد، لیبل تعیین شده برای آن داده اطمینان پایینی داشته و احتمال بالایی وجود دارد که داده واقعا مربوط به کلاس دیگر باشد. اما در روش اول یک تصمیم گیری نهایی انجام میشود و احتمالی در اختیار نداریم.

با این وجود، عیب روش دوم استفاده بیش از حد لازم از نورون ها در لایه خروجی است که میتواند محاسبات آموزش شبکه را نیز طولانی تر کند.