به نام خدا

دانشگاه صنعتی شریف

دانشکده مهندسی برق

هوش مصنوعی و محاسبات زیستی ـ دکتر حاجی پور

نیم‌سال اول ۱۳۹۹

گزارش بخش کامپیوتری تمرین سری دوم

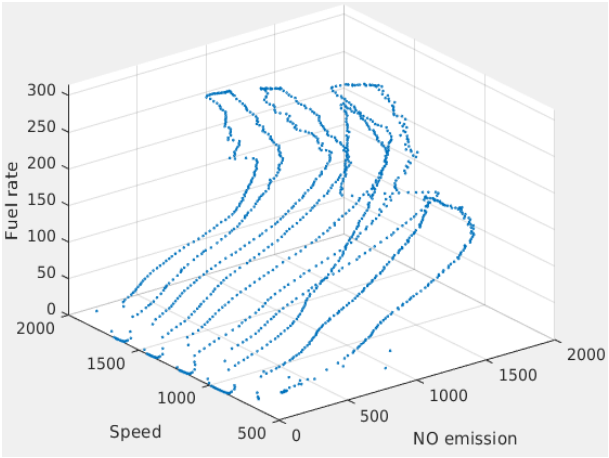
امید شرفی (۹۶۱۰۱۸۳۸)

---

۱ سوال یک

۱.۱ الف

در نمودار زیر نرخ مصرف سوخت را در محور سوم بر اساس میزان اکسید نیتروژن خروجی و سرعت ترسیم کرده ایم.



شکل ۱: ترسیم داده ها در فضای سه بعدی

۲.۱ ب

با توجه به آن که داده ها به صورت تصادفی قرار گرفته اند در نتیجه انتخاب هفتصد داده اول به عنوان داده های آموزش مشکلی برای پخش شدگی داده های آموزش بر روی کل داده ها ایجاد نمی کند.

۳.۱ ج

همانطور که مشاهده میشود خطای میانگین مربعات نسبتا بالا هست که به نظر میرسد امکان فیت شدن مدل خطی مناسبی بر روی داده ها وجود ندارد.

```
Linear_md =

  RegressionLinear
         ResponseName: 'Y'
    ResponseTransform: 'none'
                 Beta: [2×1 double]
                 Bias: 135.0999
               Lambda: 0.0014
              Learner: 'svm'


  Properties, Methods

Accuracy on train data is 5.836043e+03
Accuracy on validation data is 6.448950e+03
```

شکل ۲: خروجی رگرسیون خطی

۴.۱ د

برای این بخش از توابع fitnlm و mnrfit میتوانستیم استفاده کنیم اما به پیاده سازی دستی و بر مبنای تئوری میپردازیم. ابتدا داریم :

$$z = \frac{Z}{۱ + e^{a+bx+cy}} \rightarrow$$

$$ln\frac{Z-z}{z} = a + bx + cy$$

در نتیجه با گرفتن Z برابر با عددی در نزدیکی ماکسیمم داده های آموزش و سپس حل مساله رگرسیون خطی مشابه مساله قبل داریم :

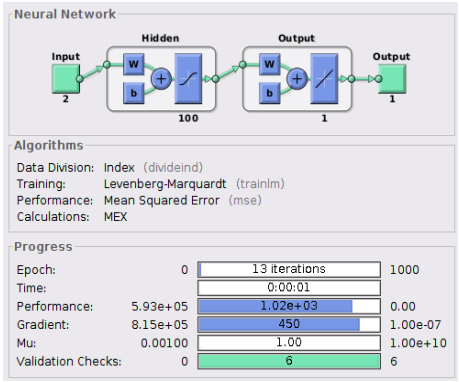
```
Z = max(Y_train);
Z_train = log((Z-Y_train)./Y_train);
Linear_md = fitrlinear(X_train, Z_train)
logestic_valid_out = sum(Linear_md.Beta .* X_valid') + Linear_md.Bias;
logestic_valid_out = Z ./ (1 + exp(logestic_valid_out));
fprintf('Accuracy on validation data is %i\n',immse(logestic_valid_out, Y_valid'));
```

شکل ۳: رگرسیون لاجستیک

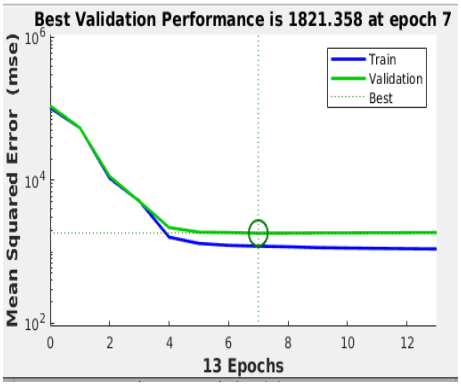
که خطای میانگین مربعات این روش نیز در حدود هشت هزار به دست می آید.

۵.۱ ه

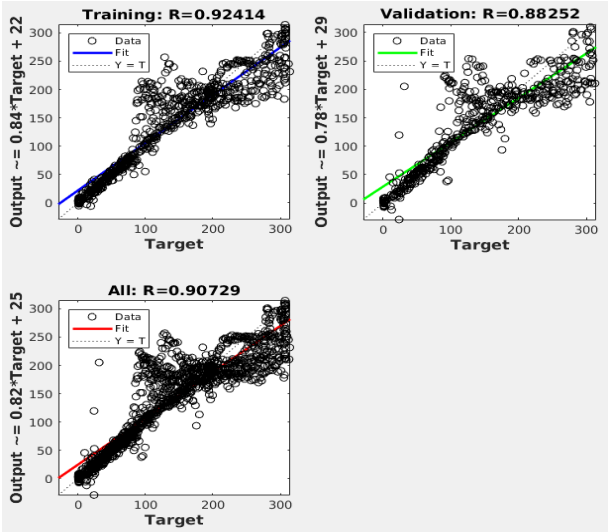
برای آموزش این بخش از fitnet با یک لایه ی پنهان شامل صد نورون استفاده میکنیم. همچنین چون در بخش ب خواسته شده بود که هفتصد داده اول را برای آموزش استفاده کنیم از divideind به عنوان divideFcn شبکه استفاده میکنیم. خروجی آموزش شبکه و میزان خطای آن به شرح زیر است.



شکل ۴: خروجی شبکه



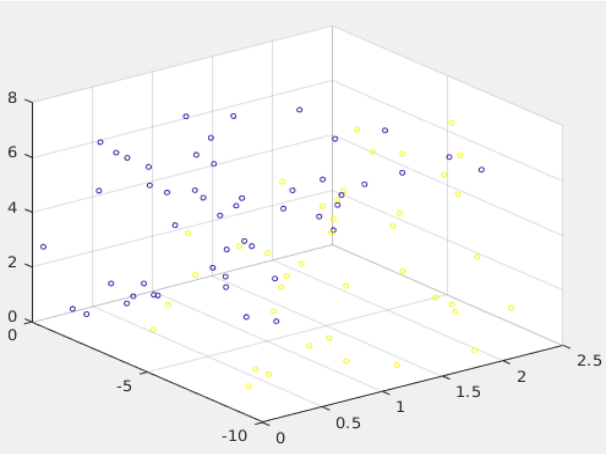
شکل ۵: عملکرد شبکه



شکل ۶: رگراسیون شبکه

۲ سوال دوم

در ابتدا ساختار کد و شبکه هایی که تست و بررسی شده اند را بررسی میکنیم و در ادامه نتایج هر بخش را معرفی میکنیم. در مورد ساختار کد، کل سوال یک زیر بخش به عنوان سوال دارد که شامل لود کردن داده ها و ترسیم داده های آموزش در فضای سه بعدی می باشد.



شکل ۷: داده های لیبل دار در فضای سه بعدی

در ادامه هر زیر بخش یک سکشن دارد که احیانا اگر تغییراتی در ساختار ذخیره سازی داده ها برای ورودی دادن به بخش شبکه ی آن زیر بخش هست در آن انجام شده است. در حقیقت این موضوع برای بخش دو انجام شده است که بتوانیم از خروجی داده های آموزشی که یک لیبل یک یا صفر هست دو خروجی بسازیم. این کار را به اسن صورت انجام میدهیم که یک خروجی دقیقا مانند همان خروجی اصلی میشود و یک خروجی نیز طبیعتا معکوس این خروجی خواهد بود.

```
%% B (prepare data for this part)
Train_out(1,:) = (TrainData(4,:)==0); % generate first output
Train_out(2,:) = (TrainData(4,:)==1); % generate second output
Train_out = double(Train_out); % ouput size is a 2*N
Train_in = TrainData(1:3,:); % input size is a 3*N
```

شکل ۸: ساخت خروجی برای شبکه ی بخش ب از روی خروجی های اصلی

حال در هر بخش ما شش طرح مختلف برای شبکه را امتحان کرده ایم.

- شبکه با یک لایه ی مخفی شامل دو نرون
- شبکه با یک لایه مخفی شامل ۲۵ نرون
- شبکه با دو لایه مخفی شامل ۱۰ نرون در هرلایه مخفی

و این طرح برای دو شبکه feedforwardnet و patternnet طراحی شده اند که در مجموع دوازده شبکه برای هر بخش سوال میباشد که متناسب با شبکه ای که میخواهیم مورد مطالعه قرار دهیم سکشن مربوطه را ران و در ادامه بخش آموزش ، محاسبات دقت و محاسبه ماتریس خروجی را می توان اجرا کرد.

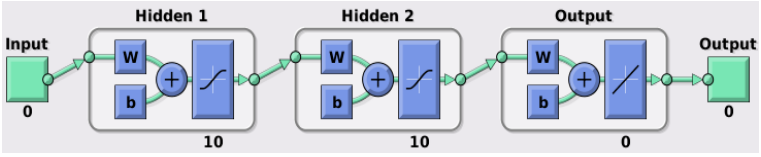
در مورد بخش آموزش شبکه این نکته حائز اهمیت است که همانطور که در صورت سوال گفته شده است برای ارزیابی شبکه در هر بخش و جلوگیری از رخ دادن overfitting ما ۲۰ درصد از داده های لیبل دار سوال را که به عنوان داده های آموزش داده شده اند را به عنوان داده های ارزیابی قرار میدهیم. در عین حال میدانیم که در اینجا با توجه به این که واقعا یک سری هم داده تست داریم که احتمالا یک شخص ثالثی بعدا خروجی شبکه را با توجه به لیبل ها این داده ها ارزیابی خواهد کرد، سوزاندن داده بخشی از داده های لیبل دارمان به عنوان داده های تست بی معنی است و در نتیجه از ترکیب بیست درصد برای داده های ارزیابی و هشتاد درصد برای آموزش استفاده میکنیم.

```
%% Setup and train network for part B
% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 80/100;
net.divideParam.valRatio = 20/100;
net.divideParam.testRatio = 0/100;
```

شکل ۹: تنظیم درصد داده های آموزش، ارزیابی و تست از روی داده های لیبل دار

۱.۲ مقدمه

حال که ساختار شبکه هایی که برای حل این سوال استفاذه کرده ایم را مطرح کردیم، بد نیست که نگاهی هم به توابع فعال سازی نرون های هر لایه بیندازیم.



شکل ۱۰: ساختار یک شبکه feedforward با دو لایه ی مخفی و قبل از آموزش

یک نمونه از شبکه feedforward را در بالا مشاهده میکنید. طبیعتا چون شبکه از روی داده ها آموزش ندیده است فعلا ابعاد ورودی و خروجی شبکه مشخص نیست اما پس از آموزش شبکه متناسب با ابعاد ورودی داده تعداد نورون های ورودی شبکه و نورون خروجی مشخص میشود. به طور مثال به شکل خروجی زیر دقت کنید :

```
Neural Network

            name: 'Feed-Forward Neural Network'
        userdata: (your custom info)

    dimensions:

        numInputs: 1
        numLayers: 3
       numOutputs: 1
   numInputDelays: 0
   numLayerDelays: 0
numFeedbackDelays: 0
numWeightElements: 161
      sampleTime: 1
```

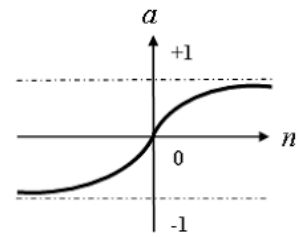
شکل ۱۱: اطلاعات شبکه شکل قبل پس از آموزش بر روی داده ها با ساختار بخش الف

همانطور که مشاهده میشود ابعاد ورودی ما سه بوده که در نتیجه یک بردار ۳تایی از ورودی داشته و در ادامه ده نرون در لایه پنهان اول، ده نورون در لایه پنهان دوم و یک نرون در خروجی داشته و در نتیجه برای عدد NWE داریم:

$$NWE = M * N۱ + N۱ * N۲ + N۱ + N۲ + N۲ * O + ۱ = ۳ * ۱۰ + ۱۰ * ۱۰ + ۲۰ + ۱۰ + ۱ = ۱۶۱$$

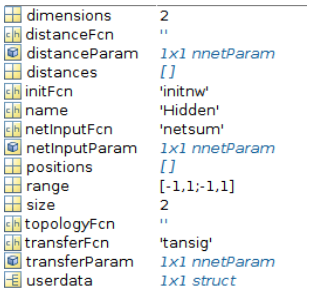
همچنین طبیعتا شبکه ساختار فیدبک در اتصالاتش ندارد که در اطلاعات بخش connections شبکه نیز قابل مشاهده است.

همچنین اگر بخواهیم نورون های شبکه را با مدل یک نرون کلی تطبیق دهیم به عنوان تابع ورودی هر نورون حاصل ضرب بردار وزن در ورودی مربوط به آن به اضافه ی یک بایاس به عنوان ورودی هر نورون به آن وارد خواهد شد. در ادامه در مورد نورون های لایه های پنهان توابع فعال سازی پیش فرض تابع tansig بوده هست که ما نیز از همین تابع استفاده میکنیم.

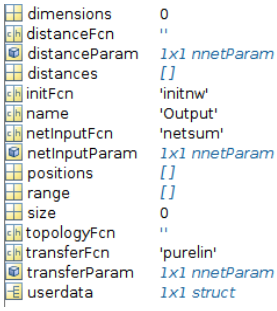


$$a = \text{Tansig}\,(n) = \frac{2}{1+e^{-2n}} - 1$$

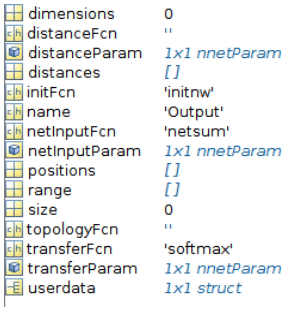
در مورد لایه خروجی نیز در شبکه ی feedforward این تابع به صورت یک تابع خطی ساده و در شبکه ی patternnet این تابع softmax بوده که ما نیز در حل این مساله از همین توابع استفاده میکنیم. در مورد بخش آموزش نیز الگوریتم پیش فرض الگوریتم Levenberg-Marquardt میباشد که ما نیز در این مساله از همان الگوریتم استفاده میکنیم.



شکل ۱۲: اطلاعات نورون های پنهان



شکل ۱۳: اطلاعات نورون های خروجی در شبکه feedforward



شکل ۱۴: اطلاعات نورون های خروجی در شبکه patternnet

در ادامه به ارائه و تحلیل خروجی های این شبکه ها بر روی داده هایمان میپردازیم. توجه شود که در هنگام آموزش شبکه ها معیار خطای ما و خورجی شبکه های ما نه یک لیبل دقیق صفر و یک، بلکه یک عدد تقریبا نزدیک به یک یا صفر برای هر ورودی بوده و معیار خطای ما برای داده های ارزیابی و آموزش در حین فرآیند آموزش کمترین مربعات روی این داده های پیوسته بوده است. در نتیجه برای آن که بخواهیم واقعا دقت را محاسبه کنیم، ابتدا برای تعیین خروجی در بخش الف که یک عدد برای خروجی داریم از رابطه زیر استفاده میکنیم :

$$Label_{partA} = round(out_{net}) > ۰$$

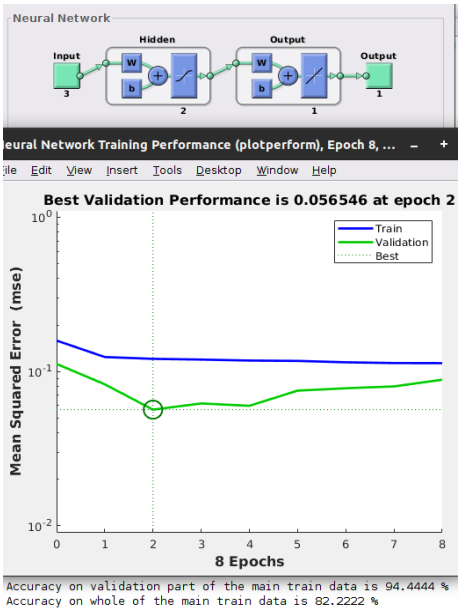
که در نتیجه لیبیل خروجی به این صورت یک یا صفر خواهد بود. توجه شود که شرط بزرگتر از صفر برای آن گذاشته شده است که مطمئن باشیم لیبل خروجی شبکه از رنج مجاز یک یا صفر دسته ها تعدی نکند. در حقیقت به صورت معمول خروجی شبکه عددی نزدیک به صفر یا یک خواهد بود و در نتیجه تابع round پاسخ مدنظر را منطقا خواهد داد اما اگر احیانا به شبکه ورودی عجیبی داده شود که شبکه رفتار غیرقایل پیش بینی و خروجی خیلی دوری دهد، طبیعتا ما به هیچ وجه علاقه نداریم که خروجی ما از قوانین لیبل دسته ها تعدی کند و در نتیجه با گذاشتن شرط مربوطه این اطمینان را حاصل میکنیم. همچنین در مورد بخش ب نیز که دو خروجی داریم (با فرض آن که خروجی دوم مربوط به یک بودن لیبل ها هست) داریم :

$$Label_{partB} = out۲_{net} > out۱_{net}$$

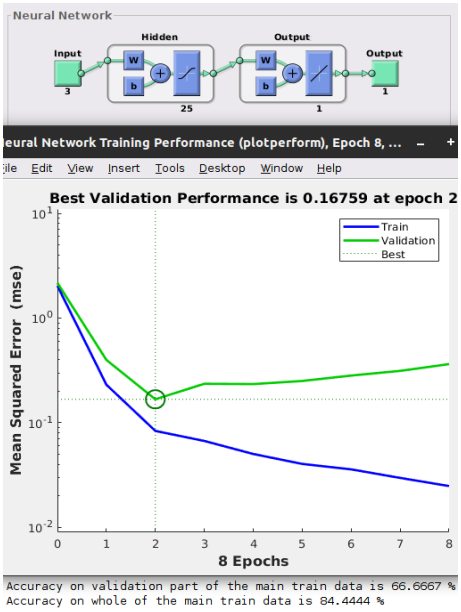
در نتیجه در ادامه ی کار ما برای هر بخش شش نمودار خواهیم داشت که شامل نمودار perfomanace روی دو بخش آموزش و ارزیابی که از روی داده های لیبل دار جدا کرده ایم میباشد. همچنین دو عدد دیگر نیز برای هر شبکه گذارش میشود که یکی خطای شبکه بر روی داده های ارزیابی پس از آن که لیبل خروجی شبکه را توسط روابط بالا تعیین کردیم و با مقایسه نسبت به لیبل های اصلی میباشد و هچمچین یک عدد دیگر نیز همین درصد روی کل داده ها میباشد که چون در اینجا کل داده های آموزش ما شامل دو دسته ی آموزش و ارزیابی بوده، منطقا انتظار داریم که این درصد دقت از درصد دقت نخست بزرگتر باشد.

در انتها اشاره به یک نکته مهم است و آن هم که اعدادی که در اینجا گزارش میشوند طبیعتا چون داده های آموزش و ارزیابی به صورت رندوم انتخاب شده اند، درصد های دقت متفاوتی را در هر اجرا میگیریم فلذا اتکا به این درصد ها به صورت کلی کار صحیحی نمی باشد. در مورد خروجی گزارش شده نیز با توجه به آن که دقت خاصی در مساله از ما خواسته نشده است، در هر بخش یکی از خروجی های شبکه ها با دقت قابل قبول را به عنوان بردار خروجی آن بخش در پوشه ضمیمه گزارش کرده ایم و طبیعتا با اجراهای متعدد همین کد یا ایجاد تغییرات کوچکی در ساختار شبکه ها میتوان دقت های بالاتری هم گرفت که خب چون حداقلی برای دقت شبکه روی داده های تست تعیین نشده است و دیتاست لیبل دار ما نیز کلا ۹۰ داده دارد، ما به همان خروجی های شبکه با یک دقت قابل قبول اکتفا میکنیم.

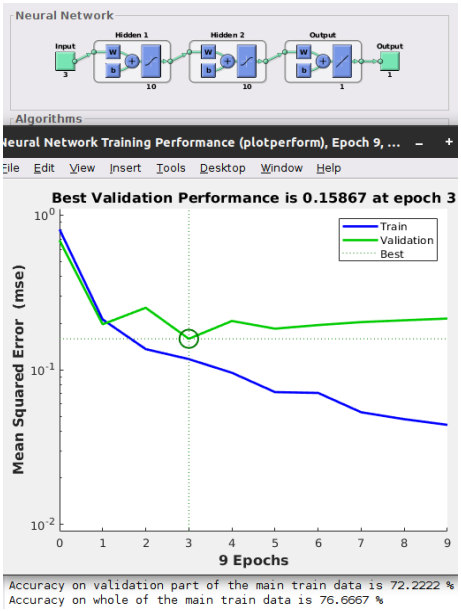
۲.۲ نتایج بخش الف



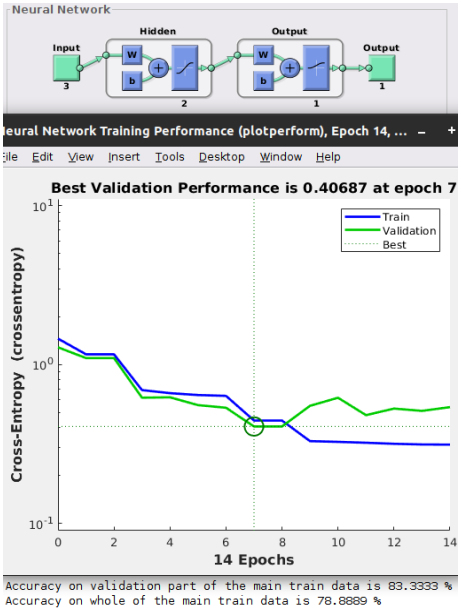
شکل ۱۵: خروجی آموزش شبکه feedforward با یک لایه پنهان دو نورونه



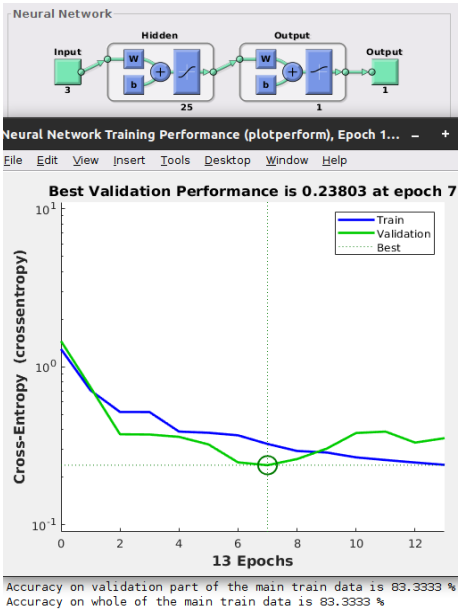
شکل ۱۶: خروجی آموزش شبکه feedforward با یک لایه پنهان بیست و پنج نورونه



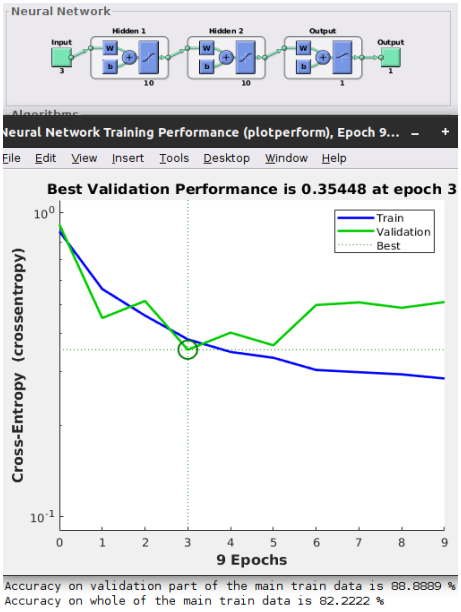
شکل ۱۷: خروجی آموزش شبکه feedforward با دو لایه پنهان ده نورونه



شکل ۱۸: خروجی آموزش شبکه patternnet با یک لایه پنهان دو نورونه

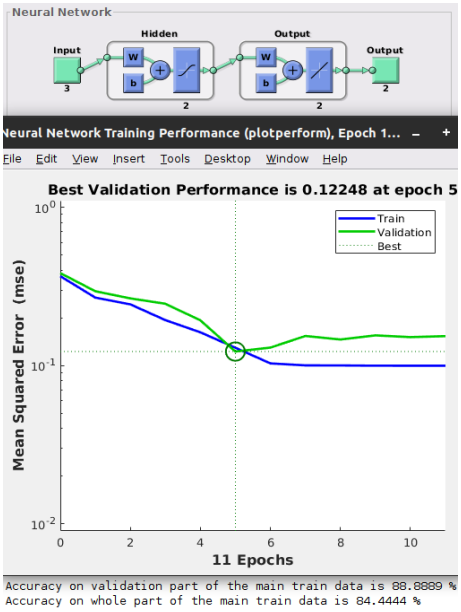


شکل ۱۹: خروجی آموزش شبکه patternnet با یک لایه پنهان بیست و پنج نورونه

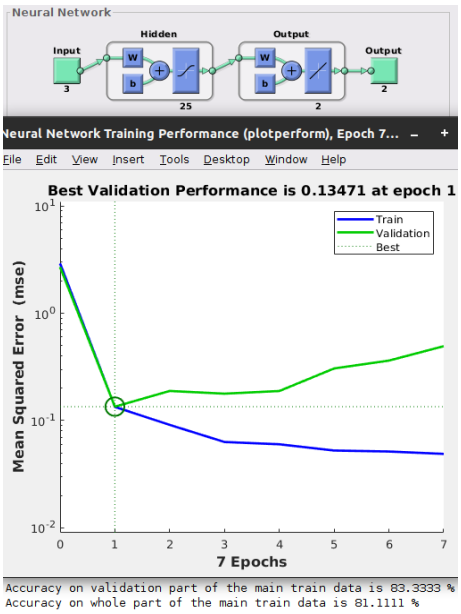


شکل ۲۰: خروجی آموزش شبکه patternnet با دو لایه پنهان ده نورونه

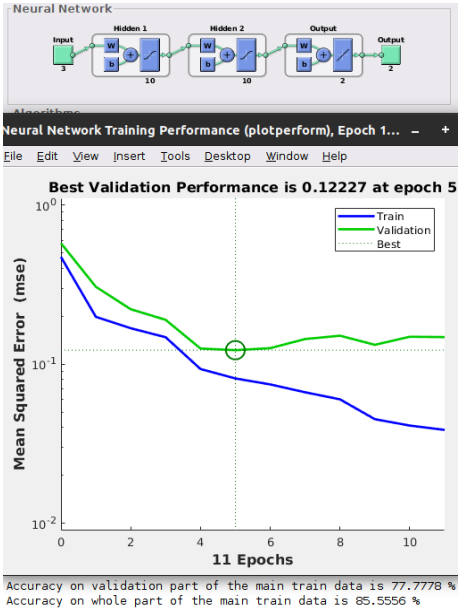
۳.۲ نتایج بخش ب



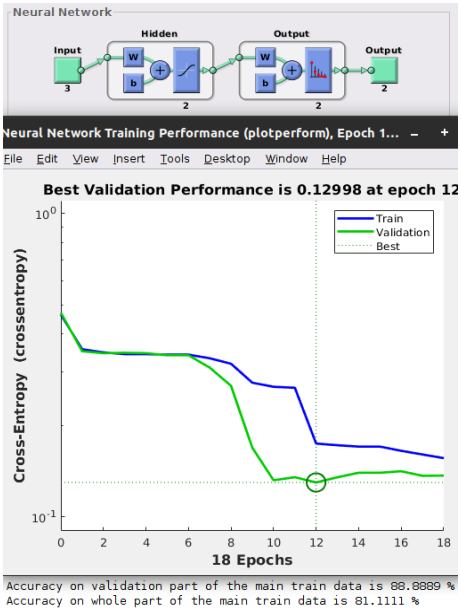
شکل ۲۱: خروجی آموزش شبکه feedforward با یک لایه پنهان دو نورونه



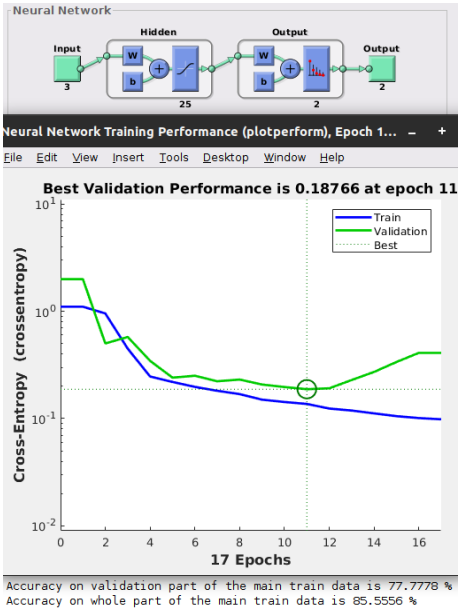
شکل ۲۲: خروجی آموزش شبکه feedforward با یک لایه پنهان بیست و پنج نورونه



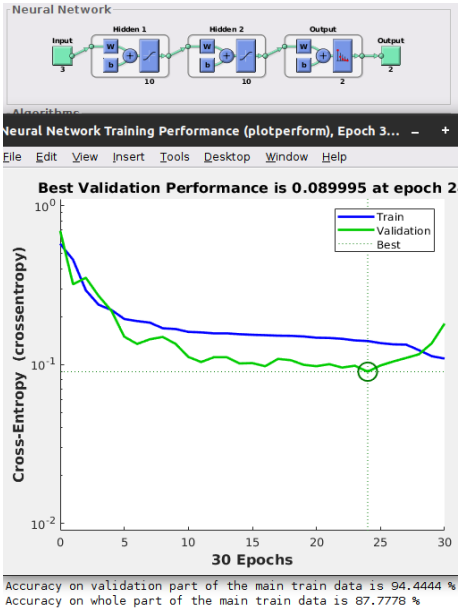
شکل ۲۳: خروجی آموزش شبکه feedforward با دو لایه پنهان ده نورونه



شکل ۲۴: خروجی آموزش شبکه patternnet با یک لایه پنهان دو نورونه



شکل ۲۵: خروجی آموزش شبکه patternnet با یک لایه پنهان بیست و پنج نورونه



شکل ۲۶: خروجی آموزش شبکه patternnet با دو لایه پنهان ده نورونه

۴.۲ نتیجه گیری

همانطور که مشاهده میشود نتایج و دقت های مختلفی با طراحی های مختلف به دست می آید. از طرفی با آموزش های مجدد نتایج عملا ممکن است تغییر کند و شاید نتوان نتیجه گیری کلی در مورد پرفرمنس شبکه ها گرفت اما چیزی که مشخص است چه در بخش الف و چه در بخش ب به صورت کلی وقتی تعداد نرون های لایه پنهان را بسیار بالا بردیم عملا پرفنمنس ما نه تنها بهتر نشده و بعضا کاهش یافته است. در حقیقت شاید بتوان گفت پیچیدگی داده ها و از طرفی تعداد داده ها نیز به اندازه ای بزرگ نیست که این تعداد زیاد نورون بخواهد به ما کمک کند و عملا با این که با افزایش تعداد نرون ها قدرت شبکه در یادگیری بالا رفته ولی این تعداد نیز خیلی کاربردی در این مساله ندارد.

از طرف دیگر مخصوصا در شبکه feedforward در بخش الف به نظر میرسه که همان دو نورون در لایه مخفی نیز عملکرد مناسسب دارد و فرم کلی مساله را به خوبی یاد گرفته و درصد دقت خوبی دارد. البته با ران های متعدد دقت های حاصل شده متفاوت بود اما میانگین قابل قبول بود.

در نهایت به نظر میرسه که طراحی شبکه های دولایه نیز به طور میانگین پرفرمنس مناسبی داشته و با تغییر ساختار مساله از بخش الف به بخش ب و استفاده از دو نرون خروجی نیز پرفرمنس ها خیلی بهبود چشم گیر زیادی پیدا نکرده اند. البته که بستگی به طرح شبکه هم دارد که میتوان جفت به جفت طرح های مشابه را مقایسه نمود.