بسمه تعالی

پروژه پایانی درس رایانش عصبی

دکتر هادی ثقفی

لازم است گزارش پروژه حتما بصورت تایپ شده ارسال شود. نتایج اجرای دستورات در MATLAB بصورت اسکرین شات از کل صفحه داخل گزارش قرار گیرد. در مواردی که نوشتن رابطه یا کشیدن شکل خواسته شده، می‌توانید روی کاغذ انجام دهید و عکس واضح آن را داخل گزارش بگذارید.

❶ توابع انتقال

الف- توابع انتقال معرفی شده در درس را به شرح زیر در بازه ورودی [-2,2] رسم کنید.

hardlimit(n)	purelin(n)	logsig(n)	hardlims(n)	poslin(n)
radbas(n)	satlin(n)	satlins(n)	tansig(n)	tribas(n)

ب- در مورد شکل هر یک از توابع فوق و کاربردشان حداکثر در یک سطر توضیح دهید.

یادآوری: ابتدا باید بردار ورودی در محدوده مورد نظر تعریف شود:

```
n = -2:0.01:2;
```

سپس بردار خروجی محاسبه و رسم می شود:

```
a = hardlimit(n);
plot(n,a);
```

❷ تعریف شبکه پرسپترون

یک شبکه پرسپترون جدید با استفاده از دستور newp بصورت زیر تعریف کنید:

```
net = newp(P,T)
```

P ماتریس r×q است که از کنار هم قرار گرفتن q بردار ورودی r عضوی ایجاد می‌شود. بطور متناظر، T ماتریس s×q است که q بردار s عضوی خروجی های مطلوب (متناظر با q بردار ورودی) را تعریف می کند. دستور فوق یک شبکه پرسپترون تک لایه با تعداد r ورودی و s نورون (خروجی) تعریف می‌کند.

شبکه‌ای با یک ورودی و یک نورون بصورت زیر ایجاد کنید و خروجی را نمایش دهید.

```
P = [0 2]
T = [0 1]
net = newp(P,T)
```

❸ اجرا (شبیه سازی) شبکه

یک شبکه پرسپترون جدید با استفاده از دستور newp بصورت زیر تعریف کنید:

```
P = [-2 2;-2 2];
T = [0 1];
net = newp(P,T);
```

الف- شبکه فوق دارای چند ورودی و چند خروجی (نورون) است؟

ب- شبکه فوق دارای چند ضریب وزنی و چند ضریب بایاس است؟

ج- مقدار اولیه ضرایب وزنی را بصورت زیر تنظیم کنید:

```
net.IW{1,1}=[-1 1];
```

مقدار اولیه بایاس را بصورت زیر تنظیم کنید:

```
net.b{1}=[1];
```

جهت چک کردن مقادیر ضرایب وزنی و بایاس از دستورات زیر استفاده کنید و نتیجه را نمایش دهید

```
net.IW{1,1}
net.b{1}
```

د- یک شکل گرافیکی ساده برای شبکه رسم کنید و مقادیر ضرایب وزنی و بایاس را روی آن بنویسید.

ه- دو ورودی زیر را تعریف کنید:

```
p1 = [0;1]
p2 = [-1;1]
```

با توجه به شکل شبکه (بند د) خروجی شبکه درمقابل ورودی های فوق را بصورت دستی محاسبه کنید.

و- جهت محاسبه خروجی توسط متلب از دستور sim بصورت زیر استفاده می‌شود:

```
a1 = sim(net,p1)
a2 = sim(net,p2)
```

خروجی های محاسبه شده را نمایش دهید و با نتایج محاسبه دستی مقایسه کنید.

ز- در بندهای (ه) و (و) ورودی ها بصورت ترتیبی اعمال می شوند یا یکجا؟ درمورد این دو روش اعمال ورودی ها به اختصار توضیح دهید.

ح- جهت اعمال ورودی‌ها به شبکه به صورت ترتیبی از دستورات زیر استفاده کنید و نتیجه را نمایش دهید.

```
p = {p1 p2};
a = sim(net,p)
```

❹ آموزش شبکه پرسپترون به روش گام به گام

الف- درمورد روش گام به گام جهت آموزش شبکه پرسپترون به اختصار توضیح دهید.

ب- یک شبکه پرسپترون جدید با استفاده از دستور newp بصورت زیر تعریف کنید:

```
P = [-2 2;-2 2];
T = [0 1];
net = newp(P,T);
```

چنانچه مقادیر اولیه وزن‌ها و بایاس را تغییر داده باشید، با دستور زیر می‌توانید آنها را به مقادیر پیش فرض (صفر) ریست کنید:

```
net = init(net);
```

جهت تنظیم مقادیر اولیه وزن‌ها و بایاس بصورت تصادفی، می‌توان از دستورات زیر استفاده نمود:

```
net.inputweights{1,1}.initFcn = 'rands';
net.biases{1}.initFcn = 'rands';
net = init(net);
```

نتیجه اجرای دستورات فوق را نمایش دهید.

می‌خواهیم آموزش شبکه را با مقادیر اولیه وزن‌ها و بایاس بصورت زیر انجام دهیم:

```
w = [1 -0.8];
net.IW{1,1} = w;
net.b{1} = 0;
```

ورودی آموزش و خروجی مطلوب متناظر آن را بصورت زیر تعریف کنید:

```
p = [1;2];
t = [1];
```

با ورودی فوق، خروجی شبکه را محاسبه کنید و نمایش دهید:

```
a = sim(net,p)
```

آیا خروجی بدست آمده با خروجی مطلوب یکسان است؟

خطای شبکه را بصورت زیر محاسبه کنید و نمایش دهید.

```
e = t-a
```

تغییر مقدار وزن‌ها (Δw) در اثر یک مرحله آموزش شبکه را می توان با استفاده از دستور زیر محاسبه نمود:

```
dW = learnp([],p,[],[],[],[],e,[],[],[],[],[])
```

نتیجه اجرای دستور فوق را نمایش دهید. همچنین Δw را برای وزن‌های شبکه بصورت دستی (با روابط داخل جزوه) محاسبه کنید و با نتیجه فوق مقایسه کنید.

ج- با اجرای دستور زیر درمورد دستور learnpn تحقیق کنید (فقط سطر اول help را بخوانید کافی است!) و با استفاده از جزوه خودتان درمورد عملکرد آن به اختصار توضیح دهید.

```
help learnpn
```

د- جهت تکمیل آموزش شبکه، می‌توان dW بدست آمده مراحل قبل را به w اضافه کرد و با مقادیر جدید وزن‌ها (وهمچنین اصلاح بایاس)، فرایند فوق را برای ورودی بعدی تکرار نمود و این روند را تا آموزش کامل شبکه ادامه داد. با استفاده از تابع adapt می‌توان یک مرحله آموزش شبکه را با استفاده از داده‌های آموزش کامل نمود (مقادیر جدید وزن ها و بایاس جایگزین می‌شود).

خروجی شبکه تعریف شده قبلی را در مقابل ورودی آموزش بدست آورید.

```
a = sim(net,p)
```

شبکه را بصورت زیر آموزش دهید.

```
net = adapt(net,p,t);
```

مجددا خروجی شبکه را در مقابل ورودی آموزش بدست آورید.

```
a = sim(net,p)
```

مقادیر تنظیم شده برای وزن‌ها و بایاس در فرایند آموزش را چک کنید.

```
net.IW{1,1}
net.b{1}
```

ضمن نمایش نتایج بدست آمده، خروجی شبکه را قبل و بعد از آموزش مقایسه کنید.

❺ آموزش شبکه پرسپترون به روش دسته‌ای

جهت آموزش شبکه بصورت دسته‌ای (موازی) از دستور train استفاده می‌شود. این تابع داده‌های آموزش را مکررا به شبکه اعمال می‌کند تا جایی که آموزش شبکه کامل شود و خطا به ازای تمام ورودی‌ها به صفر برسد.

الف- شبکه پرسپترون قبلی را مجددا تعریف کنید.

```
P = [-2 2;-2 2];
T = [0 1];
net = newp(P,T);
```

چهار داده آموزش بصورت زیر تعریف کنید:

```
p1 = [2;2];
t1 = [0];
p2 = [1;-2];
t2 = [1];
p3 = [-2;2];
t3 = [0];
p4 = [-1;1];
t4 = [1];
```

جهت اعمال یکجای داده‌ها، آنها را بصورت زیر داخل ماتریس قرار دهید:

```
p = [p1 p2 p3 p4];
t = [t1 t2 t3 t4];
```

جهت آموزش شبکه دستور زیر را اجرا کنید:

```
net = train(net,p,t);
```

صحت فرایند آموزش را با اعمال ورودی‌های آموزش و مقایسه نتایج با خروجی‌های مطلوب چک کنید.

```
a1 = sim(net,p1)
a2 = sim(net,p2)
a3 = sim(net,p3)
a4 = sim(net,p4)
```

نتایج را نمایش داده و توضیح دهید.

ب- براساس مطالب جزوه درسی توضیح دهید در حالت کلی اگر سیستم تفکیک‌پذیر خطی باشد، کدامیک از دو دستور adapt و train بطور تضمینی قادر به آموزش کامل شبکه هستند؟

❻ نمونه کاربردی شبکه پرسپترون

الف- گیت OR: میخواهیم یک گیت منطقی OR با دو ورودی و یک خروجی را توسط شبکه عصبی مدل کنیم. یعنی یک شبکه پرسپترون تک لایه با دو ورودی و یک خروجی را طوری آموزش دهیم که رفتار گیت OR را تقلید کند. می‌دانیم خروجی گیت OR برابر صفر است اگر هر دو ورودی آن صفر باشد و در بقیه حالات خروجی برابر یک است. لذا داده‌های آموزش بصورت زیر است:

```
P = [0 0 1 1; 0 1 0 1];
T = [0 1 1 1];
```

توضیح دهید چرا ورودی و خروجی‌های آموزش بصورت فوق انتخاب شده؟

شبکه را بصورت زیر ایجاد کنید.

```
or_net = newp(P,T);
```

با شبیه‌سازی شبکه، خروجی آن را در حالت‌های مختلف بدست آورده و با خروجی مطلوب مقایسه کنید.

(مثلا sim(or_net,[0;0]))

ب- شبکه را با استفاده از تابع train آموزش دهید.

```
or_net = train(or_net,P,T);
```

با شبیه‌سازی شبکه، خروجی آن را به ازای ورودی‌های آموزش بدست آورده و با خروجی مطلوب مقایسه کنید.

```
a = sim(or_net,P)
T
```

مقادیر تنظیم شده برای وزن‌ها و بایاس را نمایش دهید.

```
net.IW{1,1}
net.b{1}
```

ج- گیت AND: یک گیت منطقی AND با دو ورودی و یک خروجی را توسط شبکه عصبی مدل کنید. می‌دانیم خروجی گیت AND برابر یک است اگر هر دو ورودی آن یک باشد و در بقیه حالات خروجی برابر صفر است. بر این اساس داده‌های آموزش را بصورت مناسب انتخاب کنید و مراحل بند (الف) و (ب) را برای این شبکه تکرار کنید.

❼ تعریف شبکه خطی

یک شبکه خطی جدید با استفاده از دستور newlin بصورت زیر تعریف کنید:

```
net = newlin(P,T)
```

P ماتریس r×q است که از کنار هم قرار گرفتن q بردار ورودی r عضوی ایجاد می‌شود. بطور متناظر، T ماتریس s×q است که q بردار s عضوی خروجی های مطلوب (متناظر با q بردار ورودی) را تعریف می کند. دستور فوق یک شبکه خطی تک لایه با تعداد r ورودی و s نورون (خروجی) تعریف می‌کند.

شبکه‌ای با یک ورودی و یک نورون بصورت زیر ایجاد کنید و خروجی را نمایش دهید.

```
P = [-1 1;-1 1]
T = [-1 1]
net = newlin(P,T)
```

مقادیر پیش فرض وزن‌ها و بایاس صفر است که می‌توان آنها را مثلا بصورت زیر تغییر داد:

```
net.IW{1,1}=[-1 2];
net.b{1}=[3];
```

جهت چک کردن مقادیر ضرایب وزنی و بایاس از دستورات زیر استفاده کنید و نتیجه را نمایش دهید:

```
net.IW{1,1}
net.b{1}
```

ورودی p را بصورت زیر تعریف کنید:

```
p = [4;5];
```

خروجی شبکه درمقابل ورودی فوق را بصورت دستی محاسبه کنید.

جهت محاسبه خروجی توسط متلب از دستور زیر استفاده کنید. نتیجه را نمایش دهید و با محاسبه دستی مقایسه کنید.

```
a = sim(net,p)
```

❽ ایجاد مستقیم شبکه خطی آموزش دیده

همان طور که در درس گفته شد، شبکه خطی را می‌توان با نوشتن معادله خط گذرنده از نقاط متناظر با داده‌های آموزش (یا حل دستگاه چندمعادله چند مجهول خطی) بصورت مستقیم بدست آورد. بدین منظور کافی است بجای دستور newlin از دستور newlind استفاده شود. به عنوان مثال بردار ورودی و هدف را بصورت زیر درنظر بگیرید:

```
P = [1 2 3];
T = [2 4.1 5.9];
net = newlind(P,T);
```

net حاوی شبکه آموزش دیده است. مقادیر ضرایب وزنی و بایاس تنظیم شده را ببینید:

```
net.IW{1,1}
net.b{1}
```

شبکه را با داده‌های آموزش شبیه‌سازی کنید و ضمن نمایش نتایج، آن را با خروجی‌های مطلوب مقایسه کنید. دلیل وجود اختلاف‌های احتمالی چیست؟

```
a = sim(net,P)
```

❾ آموزش شبکه خطی

مشابه شبکه پرسپترون، شبکه‌های خطی را نیز می‌توان با استفاده از دستورات adapt و train آموزش داد. به عنوان مثال شبکه را بصورت زیر تعریف کنید.

```
P = [2 1 -2 -1;2 -2 2 1];
T = [0 1 0 1];
net = newlin(P,T);
```

سپس شبکه را با استفاده از تابع train آموزش دهید.

```
net = train(net,P,T);
```

مقادیر تنظیم شده برای وزن‌ها و بایاس در فرایند آموزش را چک کنید.

```
net.IW{1,1}
net.b{1}
```

خروجی شبکه را در مقابل ورودی آموزش بدست آورید.

```
a = sim(net,P)
```

ضمن نمایش نتایج، آن را با خروجی‌های مطلوب مقایسه کنید. دلیل وجود اختلاف‌های احتمالی چیست؟ (دقت کنید ممکن است این سوال جواب متفاوتی با سوال مشابه قسمت قبل داشته باشد)

موفق باشید

ثقفی