

سوال (۱)

پاسخ قسمت عملکرد خوب روی الگوهای پیوسته:

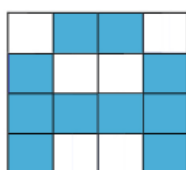
الگوریتم ART1 برای الگوهای پیوسته مناسب نیست. چون با حالت باینری کار میکند و تماماً صفر و یکی است. برای رفع این مشکل الگوریتم ART2 ارائه شده است.

ادامه پاسخ:

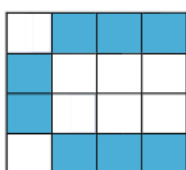
با توجه به شماره دانشجویی داده های زیر انتخاب شده اند:

Train

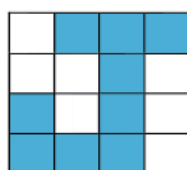
Test



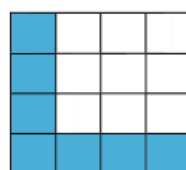
A



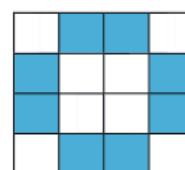
C



J



L



O

در لایه F1 به تعداد ۱۶ نورون و در لایه F2 به تعداد ۴ نورون خواهیم داشت.
مقدار پارامتر p مساوی با ۰.۶ گرفته میشود.

برای تست مشابهت ورودی را با پروتوتایپ برنده اند میکنیم و تعداد یک های نتیجه را بر تعداد یک های ورودی تقسیم میکنیم.

برای آپدیت کردن وزن ها هم وزن قدیمی را با داده جدیدی که میخواهد عضو کلاس شود و تست ویگیلنس را پاس کرده است اند میکنیم.

حال به ترتیب تک تک ورودی ها را به شبکه میدهم و موارد خواسته شده را نشان میدهم:

ورودی اول: A

پس از دادن این نمونه به شبکه وزن های ما به صورت زیر خواهند بود:

```
[ [0. 1. 1. 0. 1. 0. 0. 1. 1. 1. 1. 1. 1. 0. 0. 1.]
  [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
  [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
  [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.] ]
```

در ابتدای کار تمامی پروتوتایپ ها غیر فعال بوده و برنده هم همان اولی خواهد بود. یک کلاس خواهیم داشت که A عضو آن است.

ورودی دوم: C

پس از دادن این نمونه به شبکه وزن های ما به صورت زیر خواهند بود:

```
[ [0. 1. 1. 0. 1. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 1.]  
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]  
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]  
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]]
```

فاصله با پروتوتایپ فعال ۱ مساوی است با ۲.۸۲ و میزان مشابهت ۰.۶۲۵ است. تست پاس میشود و این هم در داخل کلاس یک قرار خواهد گرفت. یعنی A و C اعضای کلاس یک ما هستند.

ورودی سوم: J

پس از دادن این نمونه به شبکه وزن های ما به صورت زیر خواهند بود:

```
[ [0. 1. 1. 0. 1. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 1.]  
[1. 0. 0. 0. 1. 0. 0. 0. 1. 0. 0. 0. 1. 1. 1. 1.]  
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]  
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]]
```

فاصله با پروتوتایپ فعال ۱ مساوی است با ۲.۴۴ و میزان مشابهت ۰.۴۲ است. تست پاس نمیشود و یک کلاس جدید خواهیم داشت. A و C اعضای کلاس یک ما و J عضو کلاس دو ما هستند.

ورودی چهارم: I

پس از دادن این نمونه به شبکه وزن های ما به صورت زیر خواهند بود:

```
[ [0. 1. 1. 0. 1. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 1.]  
[1. 0. 0. 0. 1. 0. 0. 0. 1. 0. 0. 0. 1. 1. 1. 1.]  
[0. 1. 1. 1. 0. 0. 1. 0. 1. 0. 1. 0. 1. 1. 1. 0.]  
[1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]]
```

فاصله با پروتوتایپ فعال ۱ مساوی است با ۲.۸۲ و فاصله با پروتوتایپ فعال ۲ مساوی با ۲.۸۲ است و میزان مشابهت با نزدیک ترین پروتوتایپ (در اینجا که فاصله ها یکی هستند فرقی ندارد کدام را انتخاب کنیم) ۰.۳۳ است. تست پاس نمیشود و یک کلاس جدید خواهیم داشت. A و C اعضای کلاس یک ما و J عضو کلاس دو و I عضو کلاس سه ما هستند.

حال به داده آخر (تست) میرسیم و تست را انجام میدهیم.

ورودی پنجم (تست): O

فاصله با پروتوتایپ ها به صورت زیر است (به ترتیب) :

2.23606797749979, 2.6457513110645907, 2.6457513110645907

نزدیک ترین پروتوتایپ (کمترین فاصله) پروتوتایپ اول بوده یعنی تست ما میگوید که کلاس داده جدید کلاس ۱ است.

سوال ۲

بخش های a و b را با دو فرض میتوان حل کرد. فرض اول اینکه در محیط داده کاملاً مشابه با کلاس متفاوت نداشته باشیم که در این صورت برای هر دو بخش جواب منفی است. فرض دوم اینکه داده کاملاً مشابه با کلاس متفاوت داشته باشیم که در این صورت اصلاً الگوریتم توانایی تفکیک چنین داده هایی را ندارد.

بخش a) پاسخ منفی است. زمانی که کلاس ها یکسان باشند و داده دوم در محدوده کلاس اول باشد نیازی به این کار نیست و دایره ای با چنین شرط ایجاد نمیشود.

بخش b) اگر داده مشابه با کلاس متفاوت داشته باشیم چنین حالتی میتواند پیش بیاید اما الگوریتم به نظر جوابی برای این حالت ندارد و اگر هم چنین فرضی برقرار نباشد پاسخ منفی خواهد بود.

بخش c) دایره های مماس با کلاس مشابه امکان دارد که داشته باشیم. بسته به شعاع ها و مکان مرکز این حالت میتواند پیش بیاید.

بخش d) دایره های مماس با کلاس متفاوت هم کاملاً طبیعی است که وجود داشته باشد. بسته به شعاع ها و مکان مرکز این حالت میتواند پیش بیاید.

بخش e) برای حالتی که دو کلاس یکسان نباشد چنین حالتی پیش نمی آید چون رویکرد الگوریتم کوچکتر کردن شعاع ها تا حد ممکن برای رفع چنین حالتی است. اما زمانی که دو کلاس یکسان باشند این احتمال وجود دارد. که دایره ی جدیدی به وجود آید که شعاع بزرگتری داشته باشد و دایره قبلی را در بر گیرد.

سوال ۳

بخش a)

ضعف شبکه هاپفیلد در مقایسه با ماشین بولتزمن این است که در شبکه هاپفیلد هنگام یادگیری و عمل کمینه سازی احتمال گیر کردن در مینیمم محلی وجود دارد. به دلیل اینکه کمینه سازی انرژی در هنگام جست و جوی حالت بهینه سراسری به صورت ثابت (constant) هست. ماشین بولتزمن با استفاده از الگوریتم تبرید شبیه سازی شده (simulated annealing) که یک تکنیک احتمالاتی برای تخمین نقطه بهینه سراسری یک تابع است این نقطه ضعف را برطرف میکند. بدین صورت که با اضافه کردن کمی احتمال تصادفی (randomness یا نویز) به فرآیند کمینه سازی این شانس را داریم که همانند یک توپ که احتمال پرش را دارد از این نقاط بگذریم و در آنها گیر نکنیم. البته واضح است که نویز باید در راستای مناسب باشد تا بتواند این امکان را فراهم کند.

بخش (b)

هر دو مدل یاد شده مثالی از حافظه انجمنی هستند.

هر دو فیدبک مثبت دارند.

تابع انرژی ای دارند که به یک صورت تکراری کمینه میشوند.

با پیروی از قاعده Hebb به یک حالت خود سامان ده آموزش میبینند.

محاسبات را با استفاده از اترکتور (که به تعبیری میتوان گفت مجموعه ای از حالات (الگوها) در فضای مساله ما هستند) انجام میدهند.

هافیلد برای حافظه قابل آدرس دهی با محتوا استفاده میشود و BSB برای خوشه بندی قابلیت ها و بازنمایی داده و مفهوم استفاده میشود.