



به نام خدا



دانشگاه تهران  
دانشکده مهندسی برق و کامپیوتر  
شبکه های عصبی و یادگیری عمیق

تمرین سری سوم

نام و نام خانوادگی	محمدحسین بدیعی
شماره دانشجویی	810199106
تاریخ ارسال گزارش	3 دی

## فهرست گزارش سوالات

- 3..... Character Recognition using Hebbian Learning Rule – 1 سوال
- 39..... Auto-associative Net – ۲ سوال
- 44..... Discrete Hopfield Net – 3 سوال
- 49..... Bidirectional Associative Memory – 4 سوال

## سوال 1 – Character Recognition using Hebbian Learning Rule

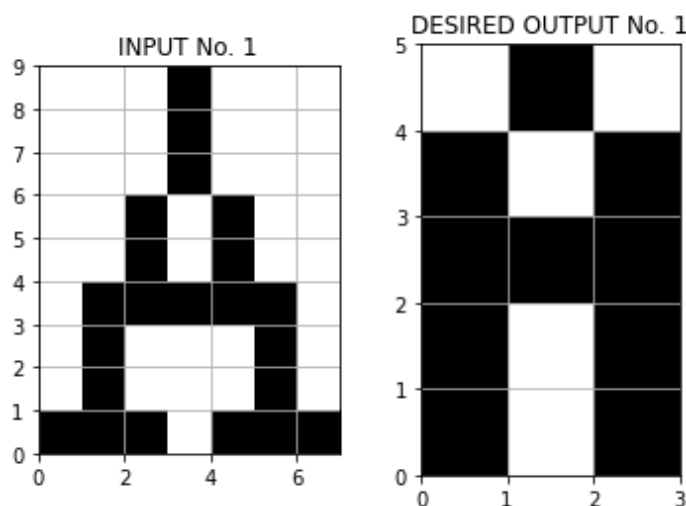
الف) آیا این شبکه می تواند تمام ورودی ها را به خروجی مطلوب برساند؟

پاسخ بخش الف)

بلی، این شبکه می تواند تمامی ورودی ها را به خروجی مطلوب (خروجی مندرج در صورت سوال) برساند. برای درستی صحبت خود شبکه را پیاده سازی کرده و نتایج را به صورت زیر استخراج نمودیم.

**INPUT No.1 with ITS DESIRED OUTPUT:**

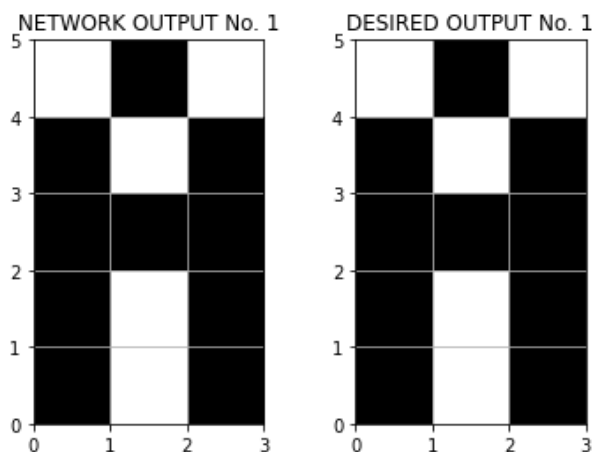
اولین ورودی داده شده و خروجی مطلوب آن به صورت زیر توسط سوال داده شده است:



شکل 1 ورودی شماره یک و خروجی مطلوب برای این ورودی

**OUTPUT No.1 VS ITS DESIRED OUTPUT:**

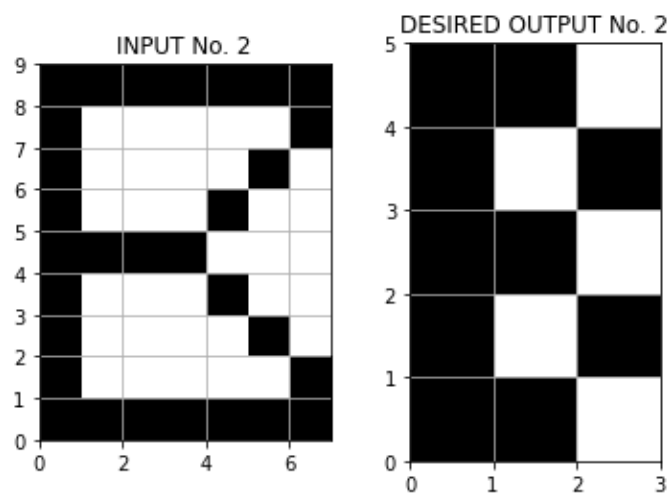
خروجی گرفته شده از شبکه در مقایسه با خروجی مطلوب برای این ورودی به صورت زیر است:



شکل 2 خروجی گرفته شده از شبکه برای ورودی شماره یک و خروجی مطلوب برای این ورودی

### INPUT No.2 with ITS DESIRED OUTPUT:

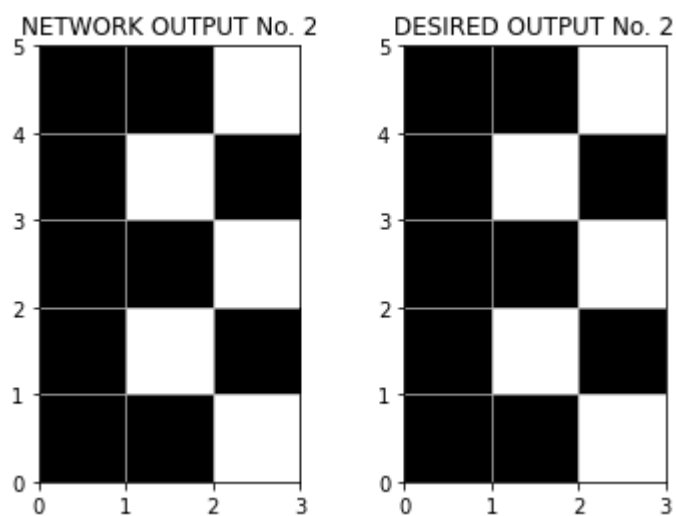
همچنین دومین ورودی داده شده و خروجی مطلوب آن به صورت زیر توسط سوال داده شده است:



شکل 3 ورودی شماره دو و خروجی مطلوب برای این ورودی

### OUTPUT No.2 VS ITS DESIRED OUTPUT:

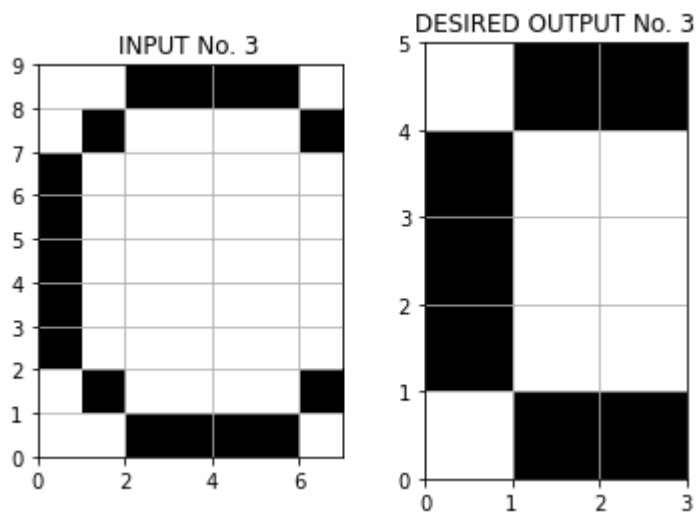
خروجی گرفته شده از شبکه نیز در مقایسه با خروجی مطلوب برای این ورودی به صورت زیر است:



شکل 4 خروجی گرفته شده از شبکه برای ورودی شماره دو و خروجی مطلوب برای این ورودی

### INPUT No.3 with ITS DESIRED OUTPUT:

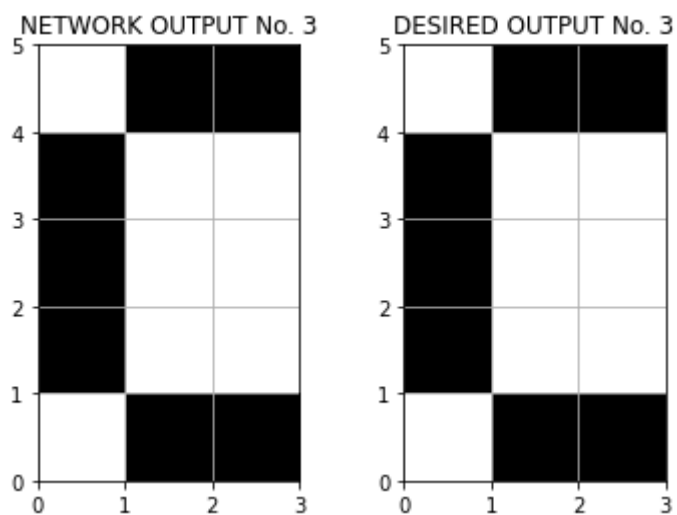
همچنین سومین ورودی داده شده و خروجی مطلوب آن به صورت زیر توسط سوال داده شده است:



شکل 5 ورودی شماره سه و خروجی مطلوب برای این ورودی

### OUTPUT No.3 VS ITS DESIRED OUTPUT:

خروجی گرفته شده از شبکه نیز در مقایسه با خروجی مطلوب برای این ورودی به صورت زیر است:



شکل 6 خروجی گرفته شده از شبکه برای ورودی شماره سه و خروجی مطلوب برای این ورودی

همانطور که مشاهده می‌کنید تمامی خروجی‌های مطلوب از شبکه به ازای ورودی‌های سوال گرفته شده است.

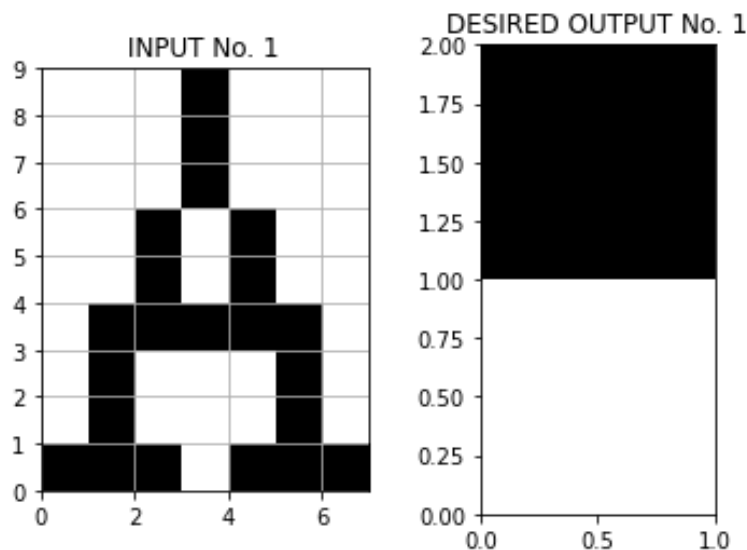
ب) کوچکترین ابعادی که شبکه می تواند ورودی  $9 \times 7$  را به خروجی مطلوب برساند چیست؟

پاسخ بخش ب)

برای این بخش ما تصمیم گرفتیم از کوچکترین ابعادی که می توانیم به عنوان خروجی شبکه استفاده کنیم، شروع نماییم و صحت یادگیری شبکه را بررسی کنیم و سپس سراغ ابعاد دیگر برویم. طبیعتاً کمترین ابعاد به ازای  $1 \times 2$  یا  $2 \times 1$  میسر می شود. با دادن این ابعاد شبکه به درستی خروجی های مطلوب را به ازای ورودی های متناظر تحویل می دهد. لذا چون این تعداد کمترین بعد ممکن بود که به ازای سه ورودی می توانستیم لحاظ کنیم، می توانیم بگوییم که شبکه برای ابعاد  $1 \times 2$  یا  $2 \times 1$  به درستی عمل می کند. نتایج بدست آمده به صورت زیر است:

#### INPUT No.1 with ITS DESIRED OUTPUT:

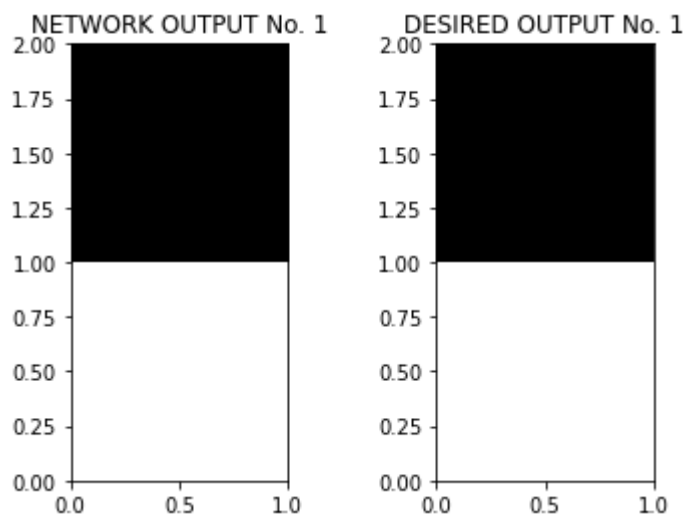
اولین ورودی داده شده و خروجی مطلوب تعریف شده برای آن به صورت زیر است:



شکل 7 ورودی شماره یک و خروجی مطلوب برای این ورودی

### OUTPUT No.1 VS ITS DESIRED OUTPUT:

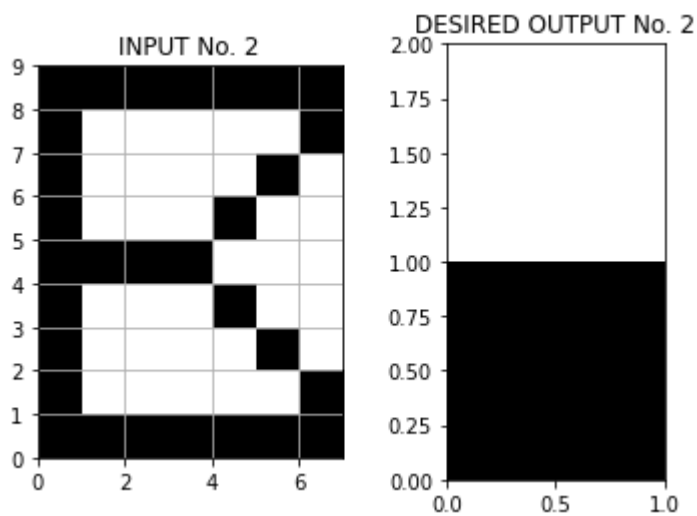
خروجی گرفته شده از شبکه نیز در مقایسه با خروجی مطلوب برای این ورودی به صورت زیر است:



شکل 8 خروجی گرفته شده از شبکه برای ورودی شماره یک و خروجی مطلوب برای این ورودی

### INPUT No.2 with ITS DESIRED OUTPUT:

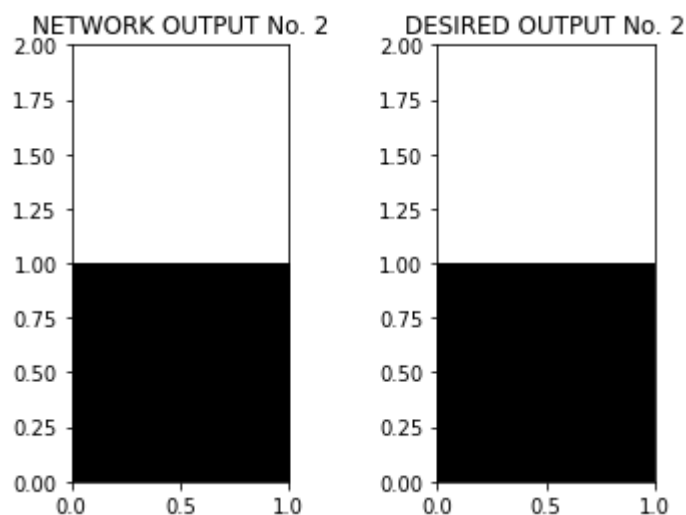
دومین ورودی داده شده و خروجی مطلوب تعریف شده برای آن به صورت زیر است:



شکل 9 ورودی شماره سه و خروجی مطلوب برای این ورودی

### OUTPUT No.2 VS ITS DESIRED OUTPUT:

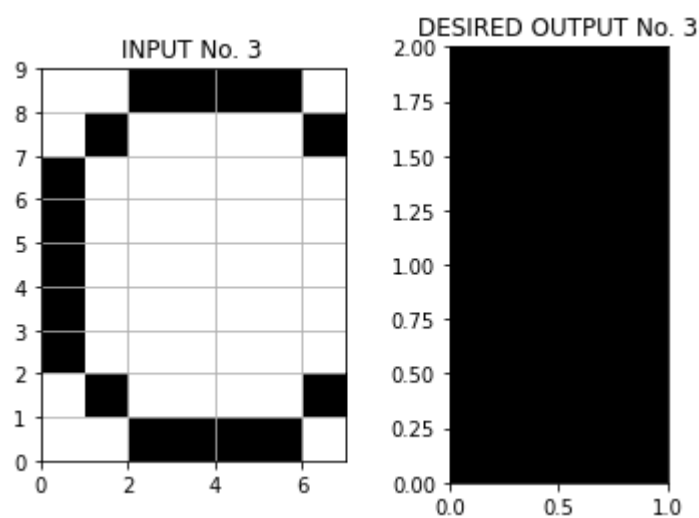
خروجی گرفته شده از شبکه نیز در مقایسه با خروجی مطلوب برای این ورودی به صورت زیر است:



شکل 10 خروجی گرفته شده از شبکه برای ورودی شماره دو و خروجی مطلوب برای این ورودی

### INPUT No.3 with ITS DESIRED OUTPUT:

سومین ورودی داده شده و خروجی مطلوب تعریف شده برای آن به صورت زیر است:

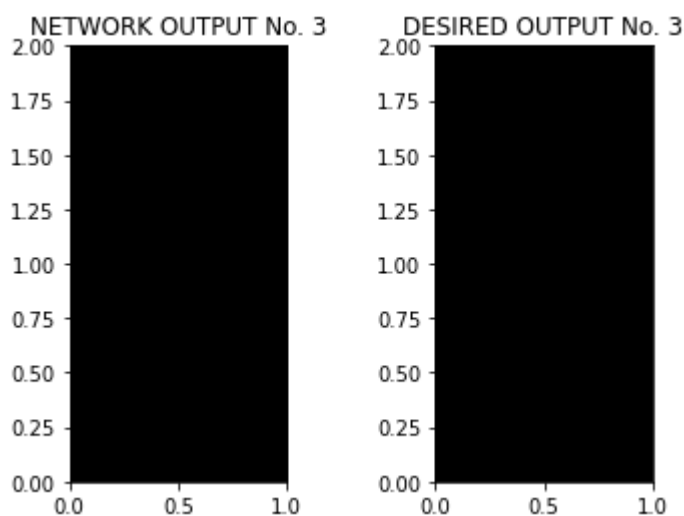


شکل 11 ورودی شماره سه و خروجی مطلوب برای این ورودی



### OUTPUT No.3 VS ITS DESIRED OUTPUT:

خروجی گرفته شده از شبکه نیز در مقایسه با خروجی مطلوب برای این ورودی به صورت زیر است:



شکل 12 خروجی گرفته شده از شبکه برای ورودی شماره سه و خروجی مطلوب برای این ورودی

همانطور که مشاهده می‌نمایید خروجی‌های گرفته شده از شبکه دقیقاً با خروجی‌های مطلوب یکسان هستند. به جهت اینکه خطایی رخ نداده باشد نیز ما چندین بار این تست را انجام دادیم و مجدداً به همین نتیجه رسیدیم. لذا می‌توان گفت که برای ابعاد خروجی  $1 \times 2$  یا  $2 \times 1$  نیز شبکه به خوبی و درستی کار می‌کند. پس پاسخ این سوال ابعاد بدست آمده می‌باشد.

پ) ورودی  $9 \times 7$  را با اضافه کردن ۱۰ و ۴۰ درصد نویز (تبدیل کردن اعداد +۱ و -۱ به صورت تصادفی) به شبکه برای هر دو ابعاد خروجی بخش الف و ب اعمال کنید. خروجی شبکه چیست؟ در چند درصد مواقع خروجی درست تشخیص داده شد؟

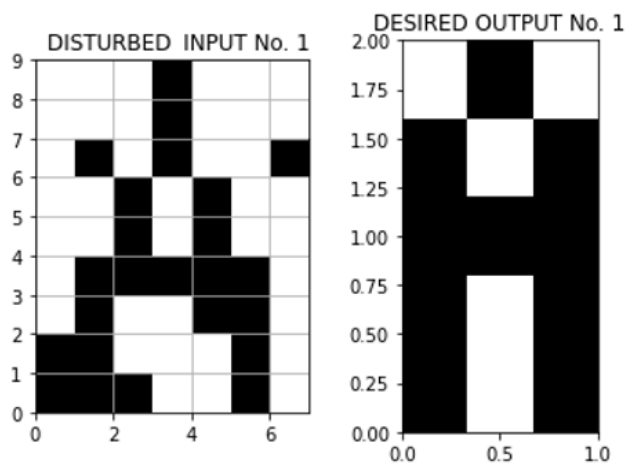
#### پاسخ بخش پ)

مطابق با خواسته‌ی مسأله ما در گام اول اغتشاشی معادل با 10% بر روی ورودی‌ها اعمال کردیم و ورودی‌های شامل اغتشاش (DISTURBED INPUT) را به عنوان ورودی شبکه‌ی نهایی وارد نمودیم تا ببینیم به چه اندازه در تشخیص خروجی درست، موفق عمل خواهد کرد.

### DISTURBED INPUT No.1 (About 10%) with ITS DESIRED OUTPUT

:

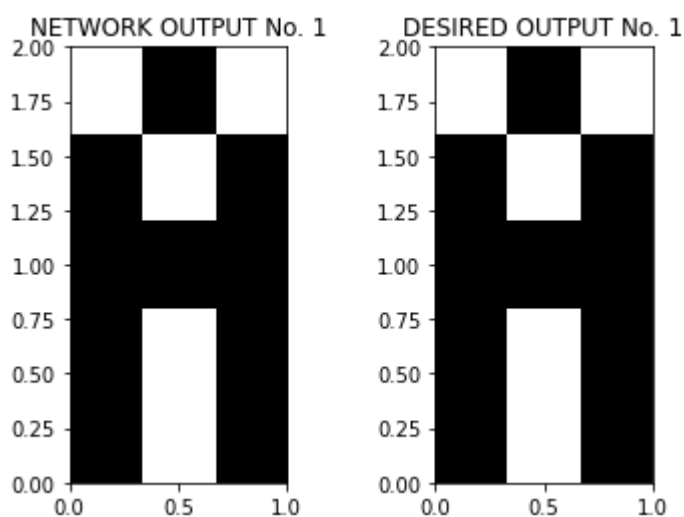
برای اولین ورودی نتایج ورودی شامل اغتشاش و خروجی مطلوب به شرح زیر است:



شکل 13 ورودی شماره یک با اعمال نویز 10 درصد و خروجی مطلوب برای این ورودی

حال اثر این اغتشاش را برای ورودی اول در خروجی شبکه و در مقایسه با خروجی مطلوب ترسیم می‌کنیم که به شرح زیر است:

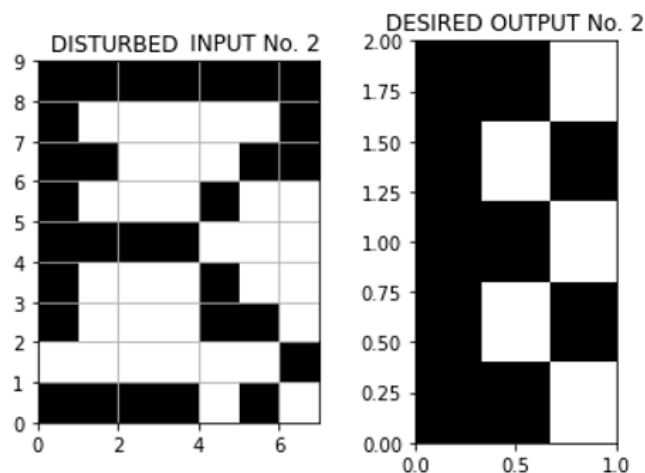
### OUTPUT No.1 (FOR 10% DISTURBED INPUT) VS ITS DESIRED OUTPUT:



شکل 14 خروجی گرفته شده از شبکه برای ورودی شماره یک (دارای نویز 10 درصد) و خروجی مطلوب برای این ورودی

### DISTURBED INPUT No.2 (About 10%) with ITS DESIRED OUTPUT:

برای دومین ورودی نتایج ورودی شامل اغتشاش و خروجی مطلوب به شرح زیر است:



شکل 15 ورودی شماره ۲ دو با اعمال نویز ۱۰ درصد و خروجی مطلوب برای این ورودی

### OUTPUT No.2 (FOR 10% DISTURBED INPUT) VS ITS DESIRED OUTPUT:

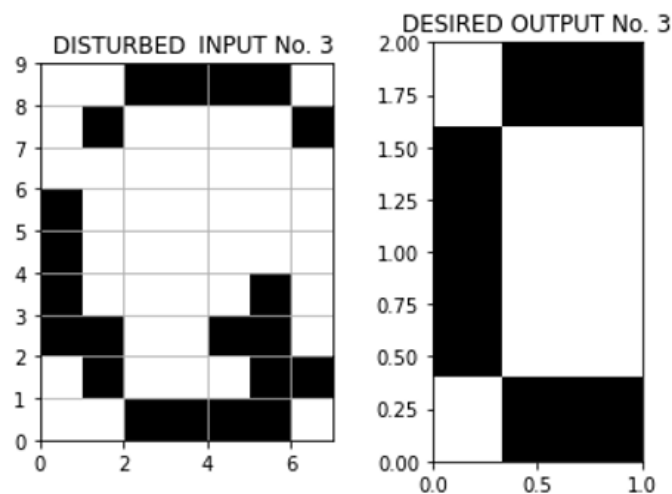
حال اثر این اغتشاش را برای ورودی دوم در خروجی شبکه و در مقایسه با خروجی مطلوب ترسیم می‌کنیم که به شرح زیر است:



شکل 16 خروجی گرفته شده از شبکه برای ورودی شماره ۲ دو (دارای نویز ۱۰ درصد) و خروجی مطلوب برای این ورودی

### DISTURBED INPUT No.3 (About 10%) with ITS DESIRED OUTPUT:

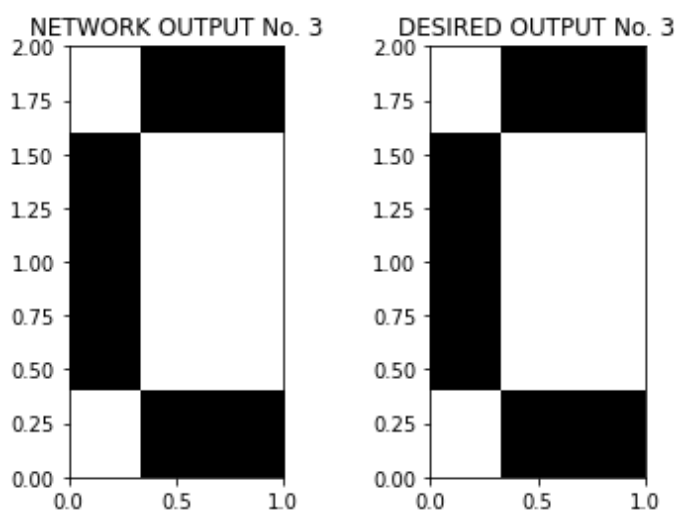
برای سومین ورودی نتایج ورودی شامل اغتشاش و خروجی مطلوب به شرح زیر است:



شکل 17 ورودی شماره سه با اعمال نویز 10 درصد و خروجی مطلوب برای این ورودی

### OUTPUT No.3 (FOR 10% DISTURBED INPUT) VS ITS DESIRED OUTPUT:

حال اثر این اغتشاش را برای ورودی سوم در خروجی شبکه و در مقایسه با خروجی مطلوب ترسیم می کنیم که به شرح زیر است:



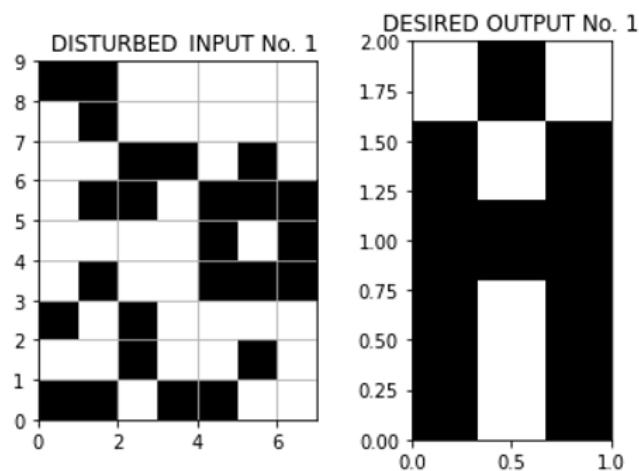
شکل 18 خروجی گرفته شده از شبکه برای ورودی شماره سه (دارای نویز 10 درصد) و خروجی مطلوب برای این ورودی

همانطور که مشاهده می‌فرمایید اغتشاش 10% در ورودی نمی‌تواند در ارائه‌ی نتایج مطلوب توسط شبکه (برای هر سه ورودی) تأثیری بگذارد. حال طبق خواسته‌ی مسأله اغتشاش را بالاتر (40%) می‌بریم و نتایج را مجدداً ارائه می‌کنیم. توجه داشته باشید که صحت میانگین را در آخر سر برای تعداد 1000 آزمایش خدمت شما ارائه خواهیم کرد.

نتایج برای اغتشاش 40% برای مسأله قسمت الف به شرح زیر است:

#### **DISTURBED INPUT No.1 (About 40%) with ITS DESIRED OUTPUT:**

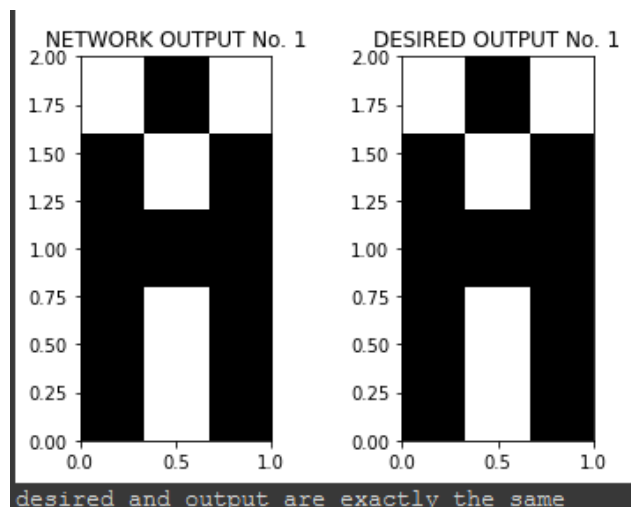
برای اولین ورودی نتایج ورودی شامل اغتشاش و خروجی مطلوب به شرح زیر است:



شکل 19 ورودی شماره یک با اعمال نویز 40 درصد و خروجی مطلوب برای این ورودی

حال اثر این اغتشاش را برای ورودی اول در خروجی شبکه و در مقایسه با خروجی مطلوب ترسیم می‌کنیم که به شرح زیر است:

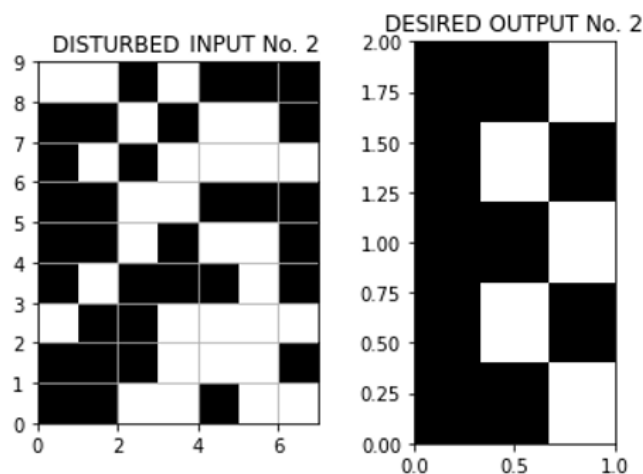
#### **OUTPUT No.1 (FOR 40% DISTURBED INPUT) VS ITS DESIRED OUTPUT:**



شکل 20 خروجی گرفته شده از شبکه برای ورودی شماره یک (دارای نویز 40 درصد) و خروجی مطلوب برای این ورودی

### DISTURBED INPUT No.2 (About 40%) with ITS DESIRED OUTPUT:

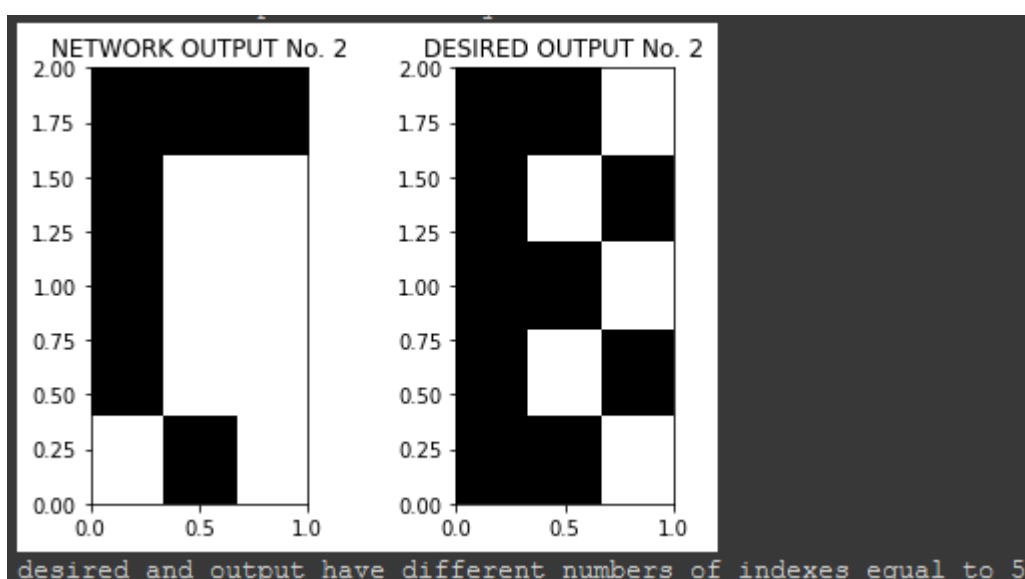
برای دومین ورودی نتایج ورودی شامل اغتشاش و خروجی مطلوب به شرح زیر است:



شکل 21 ورودی شماره دو با اعمال نویز 40 درصد و خروجی مطلوب برای این ورودی

### OUTPUT No.2 (FOR 40% DISTURBED INPUT) VS ITS DESIRED OUTPUT:

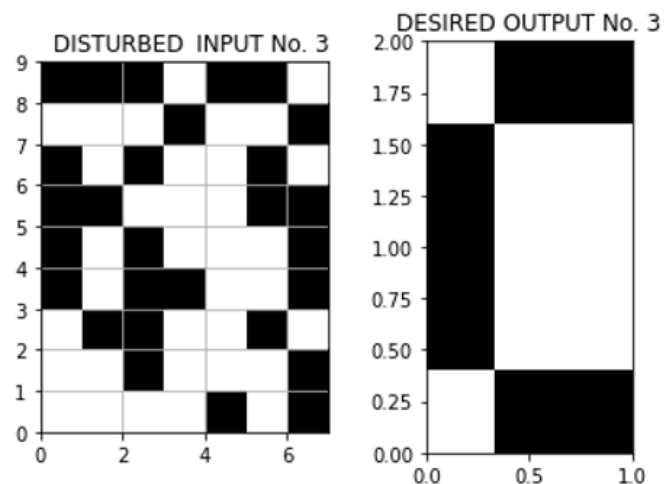
حال اثر این اغتشاش را برای ورودی دوم در خروجی شبکه و در مقایسه با خروجی مطلوب ترسیم می‌کنیم که به شرح زیر است:



شکل 22 خروجی گرفته شده از شبکه برای ورودی شماره دو (دارای نویز 40 درصد) و خروجی مطلوب برای این ورودی

### DISTURBED INPUT No.3 (About 40%) with ITS DESIRED OUTPUT:

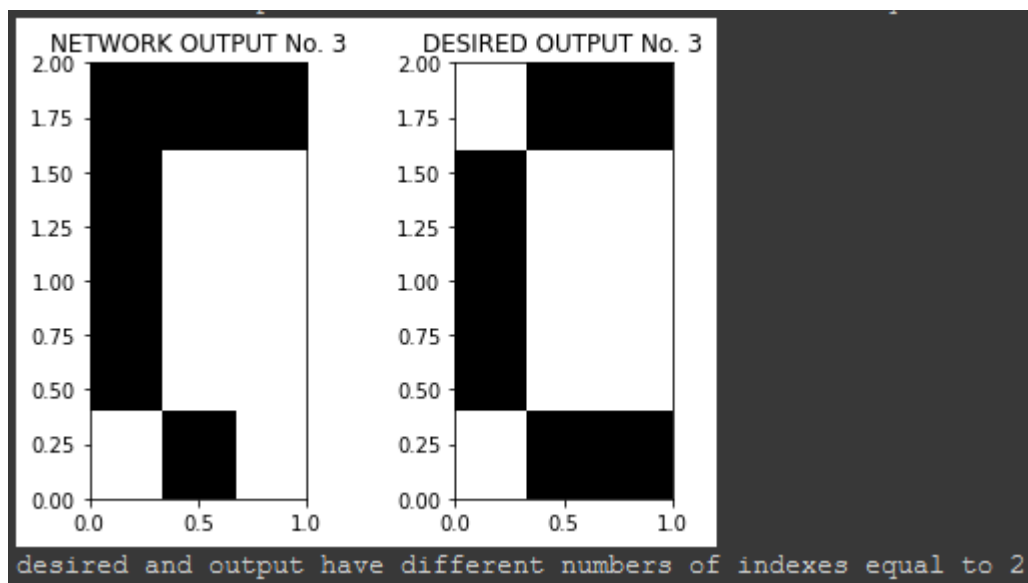
برای سومین ورودی نتایج ورودی شامل اغتشاش و خروجی مطلوب به شرح زیر است:



شکل 23 ورودی شماره سه با اعمال نویز 40 درصد و خروجی مطلوب برای این ورودی

### OUTPUT No.3 (FOR 40% DISTURBED INPUT) VS ITS DESIRED OUTPUT:

حال اثر این اغتشاش را برای ورودی سوم در خروجی شبکه و در مقایسه با خروجی مطلوب ترسیم می کنیم که به شرح زیر است:



شکل 24 خروجی گرفته شده از شبکه برای ورودی شماره سه (دارای نویز 40 درصد) و خروجی مطلوب برای این ورودی

همانطور که مشاهده می‌فرمایید اغتشاش 40% در ورودی در حالت فوق باعث شده است که تنها ورودی 1 به خروجی مطلوب متناظر شود و برای سایر ورودی‌ها شبکه عملکرد خوبی ندارد. به گونه‌ای که برای ورودی دوم، 5 پیکسل (درایه) در مقایسه با خروجی مطلوب دستخوش تغییر شده است و یا برای ورودی سوم نیز 3 پیکسل (درایه) همین مسأله را دارد. این واقعیت دور از انتظار نیست و طبیعتاً هر اندازه که اغتشاش را بالا ببریم و ورودی را از ورودی‌های مورد انتظار شبکه دور تر کنیم، طبیعتاً شبکه نخواهد توانست به درستی ورودی را تشخیص دهد و نتیجتاً خروجی متناظرش را هم ایجاد نخواهد کرد. همانطور که صورت سوال خواسته است میانگین صحت را برای عملکرد شبکه محاسبه کردیم. به ازای 10% اغتشاش نتایج به صورت زیر است.

جدول 1 میانگین صحت بدست آمده برای شبکه با خروجی‌های 3\*5 و نویز 10 درصد

Average Accuracy for Input No.1	100%
Average Accuracy for Input No.2	100%
Average Accuracy for Input No.3	100%
Mean Average Accuracies	100%

حال به ازای اغتشاش 40% نیز جدول بالا را ارائه می‌کنیم:

جدول 2 میانگین صحت بدست آمده برای شبکه با خروجی‌های 3\*5 و نویز 40 درصد

Average Accuracy for Input No.1	68%
Average Accuracy for Input No.2	60%
Average Accuracy for Input No.3	54%
Mean Average Accuracies	60.66%

همانطور که مشاهده می‌نمایید، با افزایش اغتشاش، میانگین صحت تمامی ورودی‌ها به طور متوسط تا عدد 60.66% کاهش یافته است.

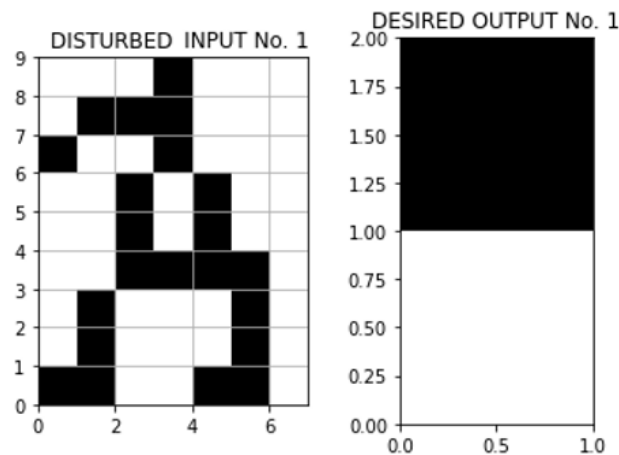
حال مجدداً تمامی مراحل فوق را برای قسمت ب مسأله تکرار می‌کنیم.



مجددا در این مرحله نیز مطابق با خواسته‌ی مسأله ما در گام اول اغتشاشی معادل با 10% بر روی ورودی‌ها اعمال کردیم و ورودی‌های شامل اغتشاش (DISTURBED INPUT) را به عنوان ورودی شبکه‌ی نهایی وارد نمودیم تا ببینیم به چه اندازه در تشخیص خروجی درست، موفق عمل خواهد کرد.

#### **DISTURBED INPUT No.1 (About 10%) with ITS DESIRED OUTPUT:**

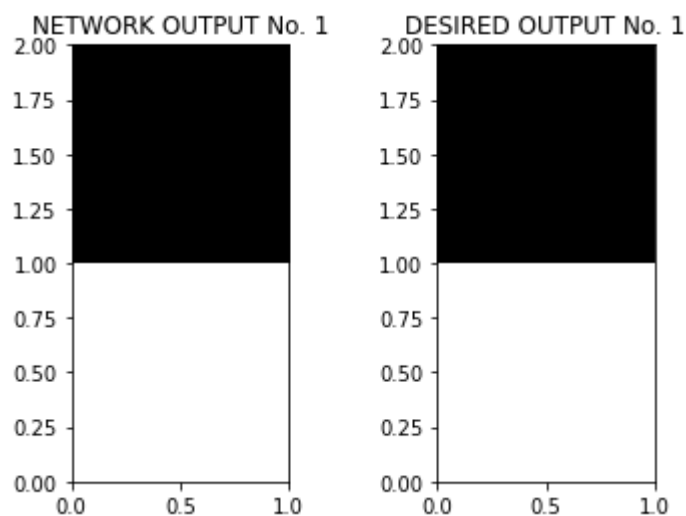
برای اولین ورودی نتایج ورودی شامل اغتشاش و خروجی مطلوب به شرح زیر است:



شکل 25 ورودی شماره یک با اعمال نویز 10 درصد و خروجی مطلوب برای این ورودی

حال اثر این اغتشاش را برای ورودی اول در خروجی شبکه و در مقایسه با خروجی مطلوب ترسیم می‌کنیم که به شرح زیر است:

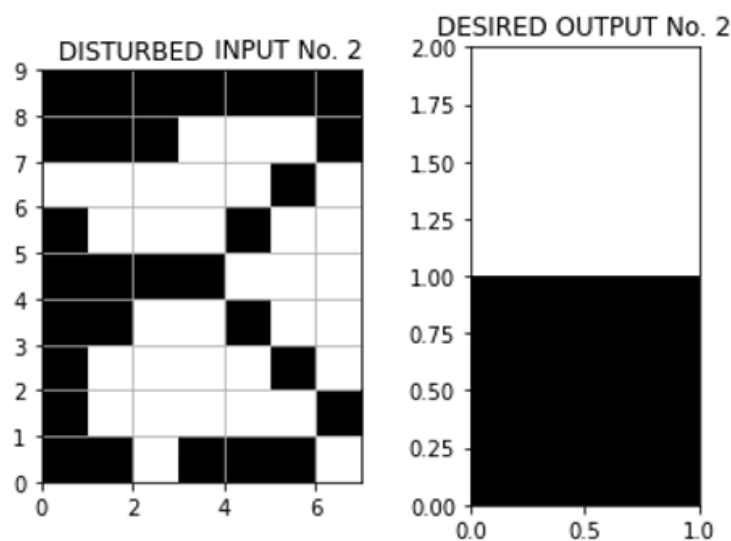
#### **OUTPUT No.1 (FOR 10% DISTURBED INPUT) VS ITS DESIRED OUTPUT:**



شکل 26 خروجی گرفته شده از شبکه برای ورودی شماره یک (دارای نویز 10 درصد) و خروجی مطلوب برای این ورودی

### **DISTURBED INPUT No.2 (About 10%) with ITS DESIRED OUTPUT:**

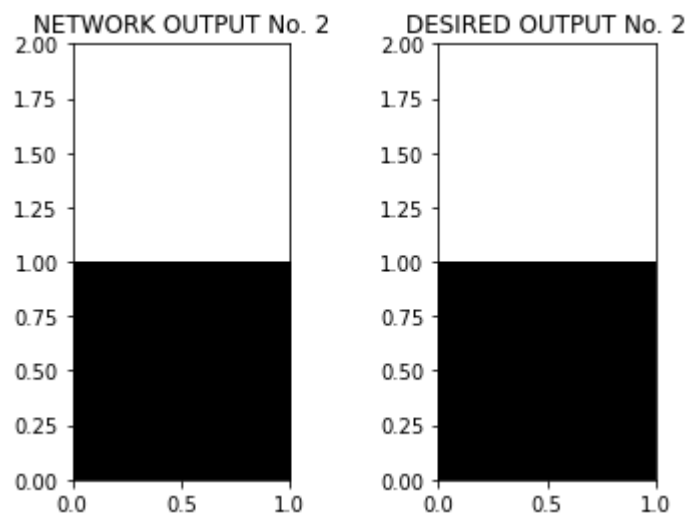
برای دومین ورودی نتایج ورودی شامل اغتشاش و خروجی مطلوب به شرح زیر است:



شکل 27 ورودی شماره دو با اعمال نویز 10 درصد و خروجی مطلوب برای این ورودی

### **OUTPUT No.2 (FOR 10% DISTURBED INPUT) VS ITS DESIRED OUTPUT:**

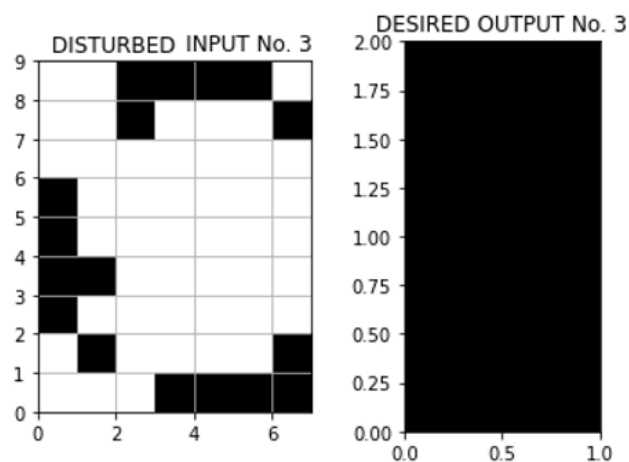
حال اثر این اغتشاش را برای ورودی دوم در خروجی شبکه و در مقایسه با خروجی مطلوب ترسیم می‌کنیم که به شرح زیر است:



شکل 28 خروجی گرفته شده از شبکه برای ورودی شماره دو (دارای نویز 10 درصد) و خروجی مطلوب برای این ورودی

### **DISTURBED INPUT No.3 (About 10%) with ITS DESIRED OUTPUT:**

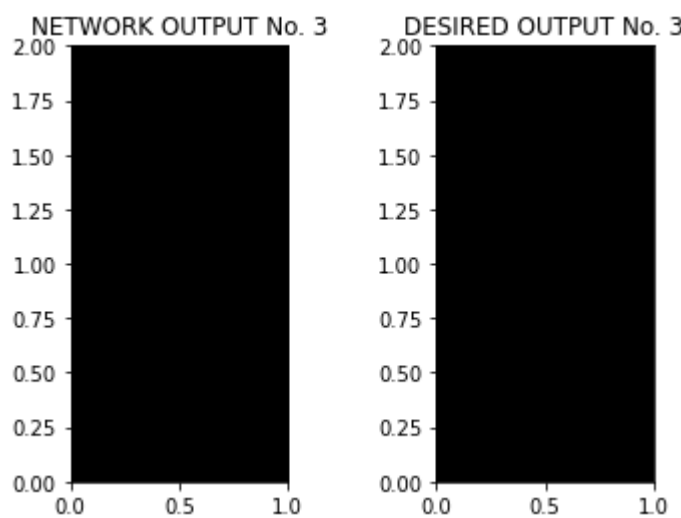
برای سومین ورودی نتایج ورودی شامل اغتشاش و خروجی مطلوب به شرح زیر است:



شکل 29 ورودی شماره سه با اعمال نویز 10 درصد و خروجی مطلوب برای این ورودی

### **OUTPUT No.3 (FOR 10% DISTURBED INPUT) VS ITS DESIRED OUTPUT:**

حال اثر این اغتشاش را برای ورودی سوم در خروجی شبکه و در مقایسه با خروجی مطلوب ترسیم می کنیم که به شرح زیر است:



شکل 30 خروجی گرفته شده از شبکه برای ورودی شماره سه (دارای نویز 10 درصد) و خروجی مطلوب برای این ورودی

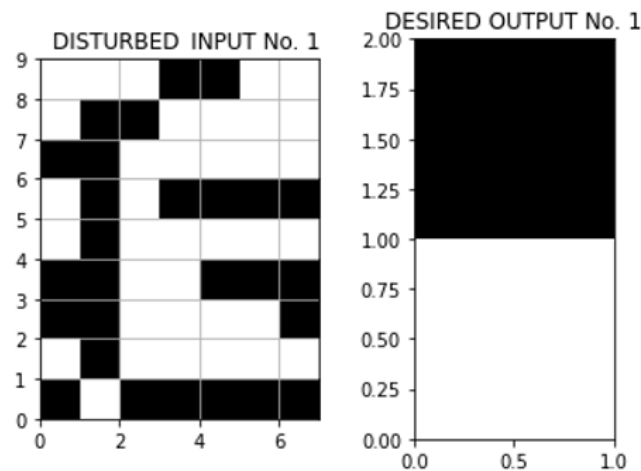
همانطور که مشاهده می فرمایید اغتشاش 10% در ورودی نمی تواند در ارائه ی نتایج مطلوب توسط شبکه (برای هر سه ورودی) تاثیری بگذارد. حال طبق خواسته ی مسأله اغتشاش را بالاتر (40%) می بریم و نتایج

را مجددا ارائه می کنیم. توجه داشته باشید که صحت میانگین را در آخر سر برای تعداد 1000 آزمایش خدمت شما ارائه خواهیم کرد.

نتایج برای اغتشاش 40% برای مسأله قسمت الف به شرح زیر است:

#### **DISTURBED INPUT No.1 (About 40%) with ITS DESIRED OUTPUT:**

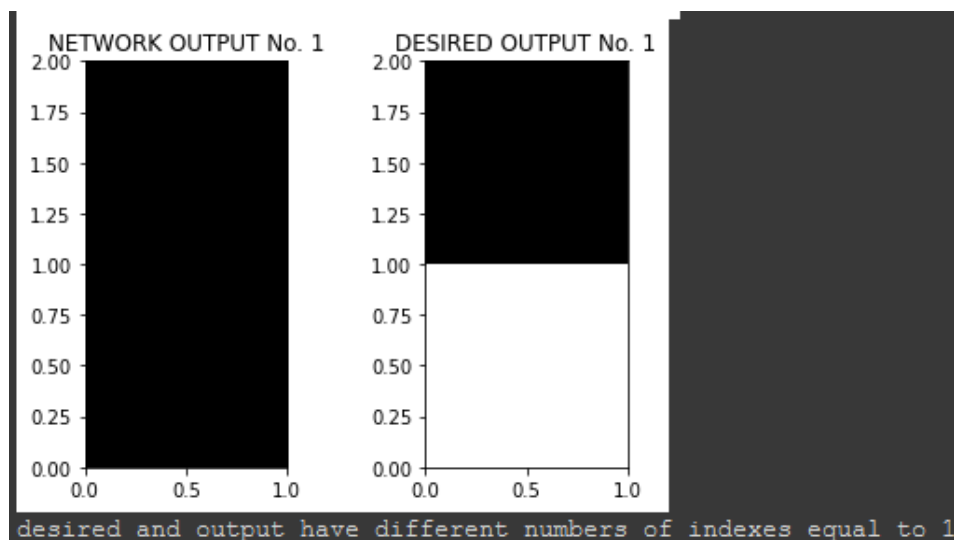
برای اولین ورودی نتایج ورودی شامل اغتشاش و خروجی مطلوب به شرح زیر است:



شکل 31 ورودی شماره یک با اعمال نویز 40 درصد و خروجی مطلوب برای این ورودی

حال اثر این اغتشاش را برای ورودی اول در خروجی شبکه و در مقایسه با خروجی مطلوب ترسیم می کنیم که به شرح زیر است:

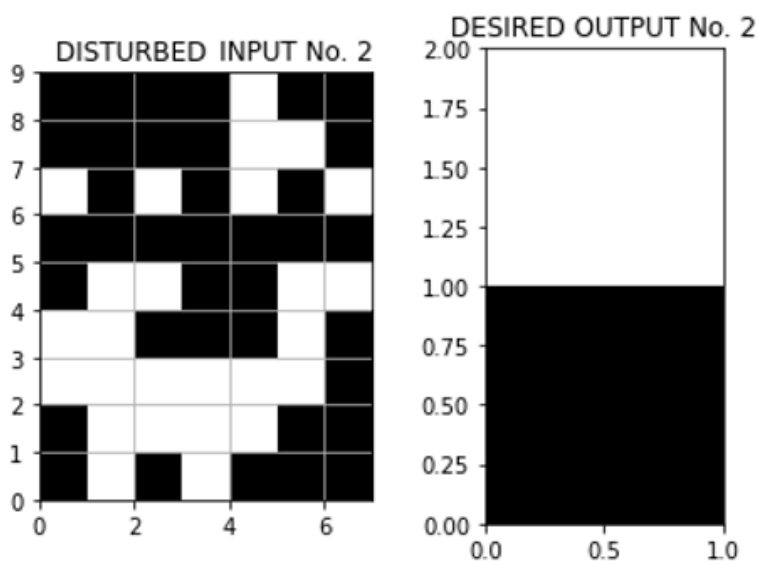
#### **OUTPUT No.1 (FOR 40% DISTURBED INPUT) VS ITS DESIRED OUTPUT:**



شکل 32 خروجی گرفته شده از شبکه برای ورودی شماره یک (دارای نویز 40 درصد) و خروجی مطلوب برای این ورودی

### **DISTURBED INPUT No.2 (About 40%) with ITS DESIRED OUTPUT:**

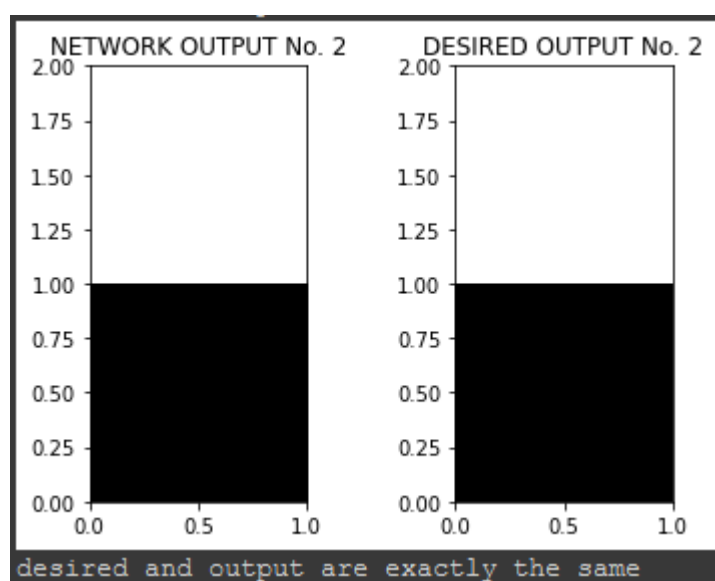
برای دومین ورودی نتایج ورودی شامل اغتشاش و خروجی مطلوب به شرح زیر است:



شکل 33 ورودی شماره دو با اعمال نویز 40 درصد و خروجی مطلوب برای این ورودی

### **OUTPUT No.2 (FOR 40% DISTURBED INPUT) VS ITS DESIRED OUTPUT:**

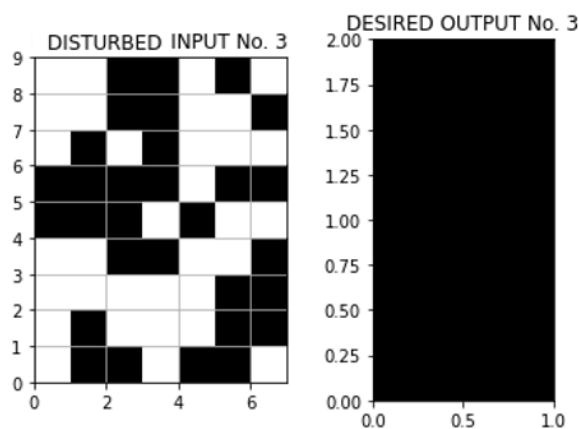
حال اثر این اغتشاش را برای ورودی دوم در خروجی شبکه و در مقایسه با خروجی مطلوب ترسیم می‌کنیم که به شرح زیر است:



شکل 34 خروجی گرفته شده از شبکه برای ورودی شماره دو (دارای نویز 40 درصد) و خروجی مطلوب برای این ورودی

### DISTURBED INPUT No.3 (About 40%) with ITS DESIRED OUTPUT:

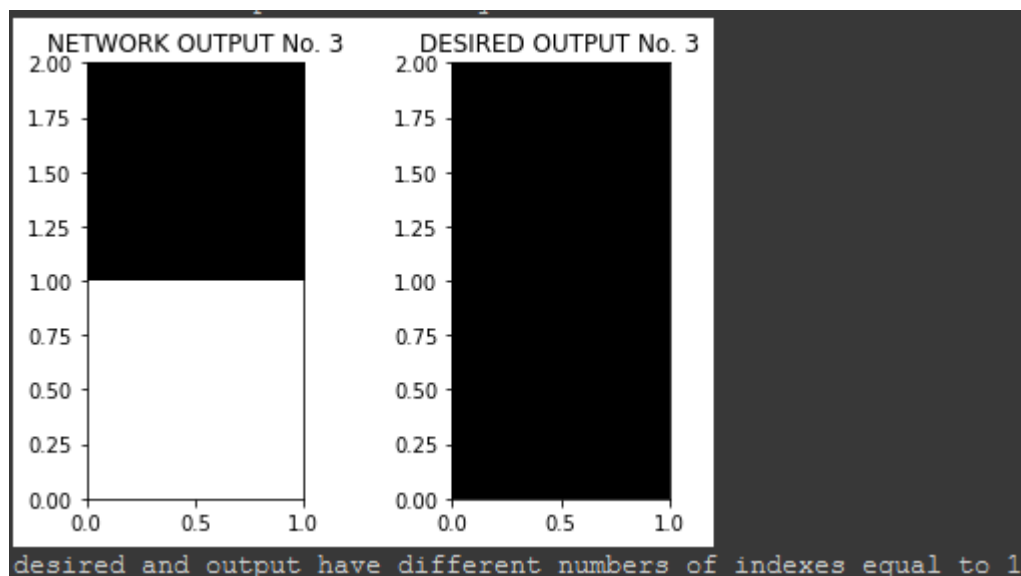
برای سومین ورودی نتایج ورودی شامل اغتشاش و خروجی مطلوب به شرح زیر است:



شکل 35 ورودی شماره سه با اعمال نویز 40 درصد و خروجی مطلوب برای این ورودی

### OUTPUT No.3 (FOR 40% DISTURBED INPUT) VS ITS DESIRED OUTPUT:

حال اثر این اغتشاش را برای ورودی سوم در خروجی شبکه و در مقایسه با خروجی مطلوب ترسیم می‌کنیم که به شرح زیر است:



شکل 36 خروجی گرفته شده از شبکه برای ورودی شماره سه (دارای نویز 40 درصد) و خروجی مطلوب برای این ورودی

همانطور که مشاهده می‌فرمایید اغتشاش 40% در ورودی در حالت فوق باعث شده است که تنها ورودی 1 به خروجی مطلوب متناظر شود و برای سایر ورودی‌ها شبکه عملکرد خوبی ندارد. به‌گونه‌ای که برای

ورودی دوم، 5 پیکسل (درایه) در مقایسه با خروجی مطلوب دستخوش تغییر شده است و یا برای ورودی سوم نیز 3 پیکسل (درایه) همین مسأله را دارد. این واقعیت دور از انتظار نیست و طبیعتاً هر اندازه که اغتشاش را بالا ببریم و ورودی را از ورودی‌های مورد انتظار شبکه دور تر کنیم، طبیعتاً شبکه نخواهد توانست به درستی ورودی را تشخیص دهد و نتیجتاً خروجی متناظرش را هم ایجاد نخواهد کرد. همانطور که صورت سوال خواسته است میانگین صحت را برای عملکرد شبکه محاسبه کردیم. به ازای 10% اغتشاش نتایج به صورت زیر است.

جدول 3 میانگین صحت بدست آمده برای شبکه با خروجی‌های 1\*2 و نویز 10 درصد

Average Accuracy for Input No.1	100%
Average Accuracy for Input No.2	100%
Average Accuracy for Input No.3	100%
Mean Average Accuracies	100%

حال به ازای اغتشاش 40% نیز جدول بالا را ارائه می‌کنیم:

جدول 4 میانگین صحت بدست آمده برای شبکه با خروجی‌های 1\*2 و نویز 40 درصد

Average Accuracy for Input No.1	89%
Average Accuracy for Input No.2	81%
Average Accuracy for Input No.3	77%
Mean Average Accuracies	82.33%

همانطور که مشاهده می‌نمایید، با افزایش اغتشاش، میانگین صحت تمامی ورودی‌ها به طور متوسط تا عدد 82.33% کاهش یافته است. البته توجه داشته باشید که این عدد برای خروجی 1\*2 (مانند اینجا) از خروجی 3\*5 (مانند نمونه‌ی قبلی) بیشتر است و این یعنی نتایج دارای اغتشاش ورودی برای خروجی با بعد کمتر دارای صحت بالاتری است. (البته توجه داریم که ما حتی در صورتیکه 1 پیکسل (درایه) هم متفاوت با خروجی مطلوب باشد، آن خروجی را نادرست تلقی می‌کنیم. و لذا شاید بتوان با کوریلیشن گیری یا محاسبه‌ی فاصله بین خروجی بدست آمده و خروجی مطلوب و نهایتاً نگاشت خروجی بدست آمده به نزدیکترین خروجی مطلوب، به نتایج بهتری و با صحت بیشتری رسید ولی با توجه به اینکه تغییر یک پیکسل از خروجی را نیز نتیجه‌ی نادرستی تلقی می‌کنیم، به نتایج فوق رسیدیم.)

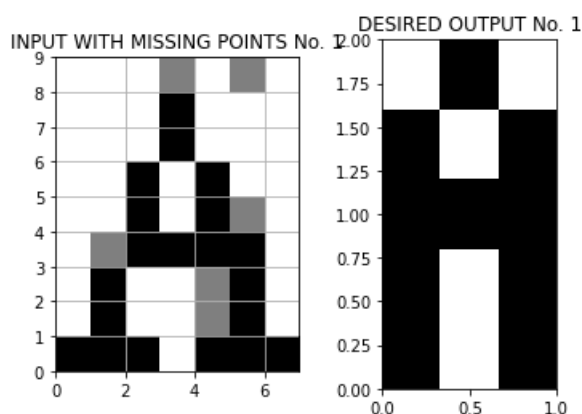
ت) ورودی ۹\*۷ را با از بین بردن ۱۰ و ۴۰ درصد از اطلاعات ( به جای ۱+ و ۱- مقدار صفر قرار دهید ) به شبکه برای هر دو ابعاد خروجی بخش الف و ب اعمال کنید. خروجی شبکه چیست؟ در چند درصد مواقع خروجی درست تشخیص داده شد؟

### پاسخ بخش ت)

مطابق با خواسته‌ی مسأله ما در گام اول missing point هایی معادل با 10% بر روی ورودی‌ها اعمال کردیم و ورودی‌های شامل missing points را به عنوان ورودی شبکه‌ی نهایی وارد نمودیم تا ببینیم به چه اندازه در تشخیص خروجی درست، موفق عمل خواهد کرد.

### INPUT No.1 with MISSING POINTS (About 10%) with ITS DESIRED OUTPUT:

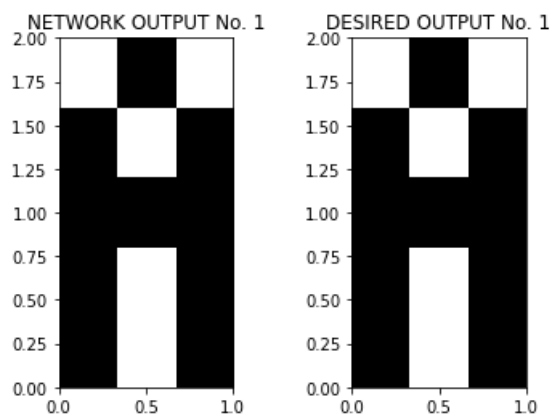
برای اولین ورودی نتایج ورودی شامل missing points و خروجی مطلوب به شرح زیر است:



شکل 37 ورودی شماره یک با اعمال حذف اطلاعات 10 درصد و خروجی مطلوب برای این ورودی

حال اثر این missing points را برای ورودی اول در خروجی شبکه و در مقایسه با خروجی مطلوب ترسیم می‌کنیم که به شرح زیر است:

### OUTPUT No.1 (FOR 10% MISSING POINTS) VS ITS DESIRED OUTPUT:

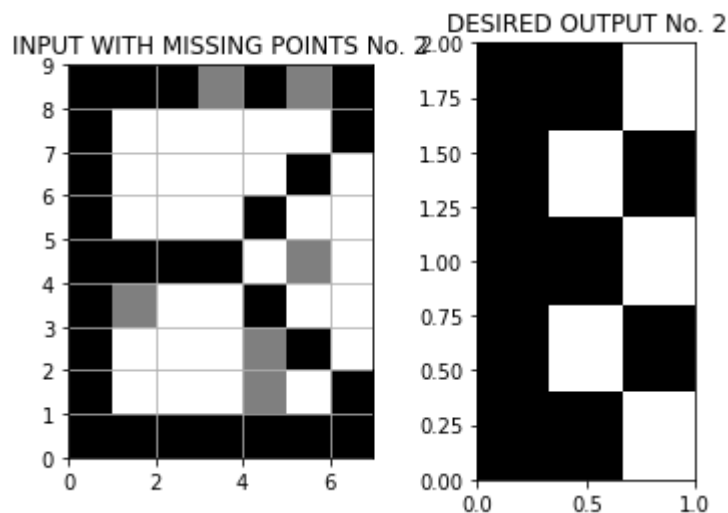


شکل 38 خروجی گرفته شده از شبکه برای ورودی شماره یک (اعمال 10 درصد حذف اطلاعات) و خروجی مطلوب برای این ورودی



### INPUT No.2 with MISSING POINTS (About 10%) with ITS DESIRED OUTPUT:

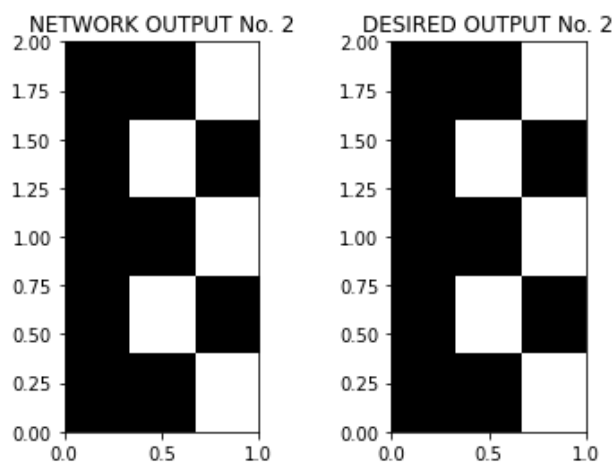
برای دومین ورودی نتایج ورودی شامل missing points و خروجی مطلوب به شرح زیر است:



شکل 39 ورودی شماره دو با اعمال حذف اطلاعات 10 درصد و خروجی مطلوب برای این ورودی

### OUTPUT No.2 (FOR 10% MISSING POINTS) VS ITS DESIRED OUTPUT:

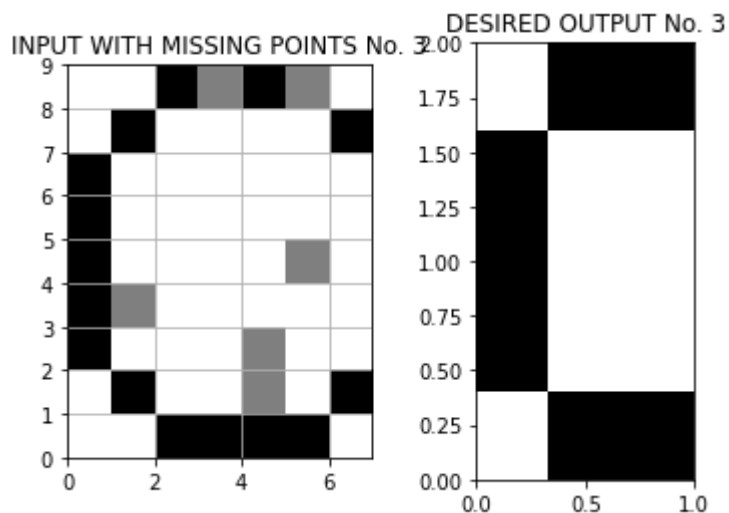
حال اثر این missing points را برای ورودی دوم در خروجی شبکه و در مقایسه با خروجی مطلوب ترسیم می‌کنیم که به شرح زیر است:



شکل 40 خروجی گرفته شده از شبکه برای ورودی شماره دو (اعمال 10 درصد حذف اطلاعات) و خروجی مطلوب برای این ورودی

### INPUT No.3 with MISSING POINTS (About 10%) with ITS DESIRED OUTPUT:

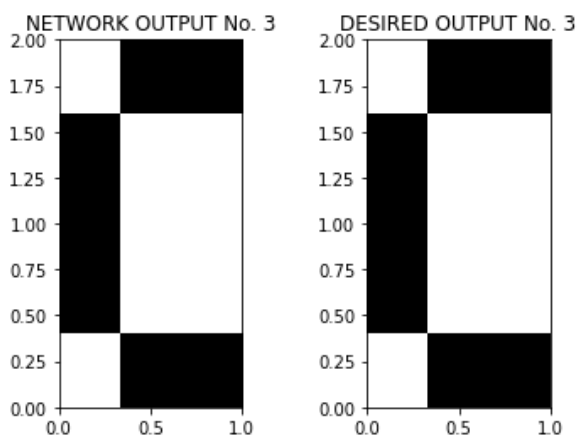
برای سومین ورودی نتایج ورودی شامل missing points و خروجی مطلوب به شرح زیر است:



شکل 41 ورودی شماره سه با اعمال حذف اطلاعات 10 درصد و خروجی مطلوب برای این ورودی

### OUTPUT No.3 (FOR 10% MISSING POINTS) VS ITS DESIRED OUTPUT:

حال اثر این missing points را برای ورودی سوم در خروجی شبکه و در مقایسه با خروجی مطلوب ترسیم می‌کنیم که به شرح زیر است:



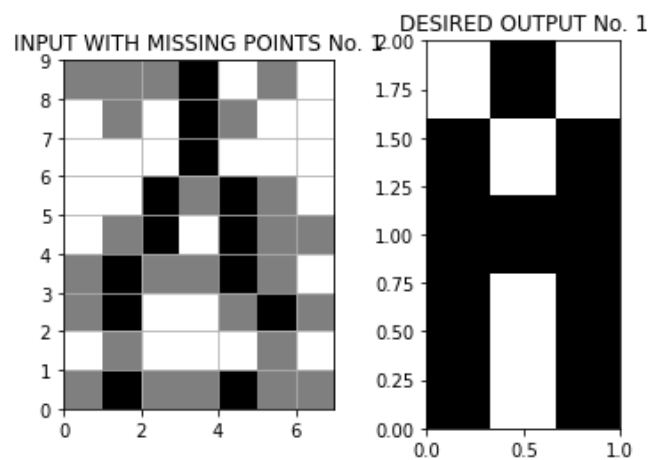
شکل 42 خروجی گرفته شده از شبکه برای ورودی شماره سه (اعمال 10 درصد حذف اطلاعات) و خروجی مطلوب برای این ورودی

همانطور که مشاهده می‌فرمایید 10% missing points در ورودی نمی‌تواند در ارائه‌ی نتایج مطلوب توسط شبکه (برای هر سه ورودی) تاثیری بگذارد. حال طبق خواسته‌ی مسأله missing points را بالاتر (40%) می‌بریم و نتایج را مجدداً ارائه می‌کنیم. توجه داشته باشید که صحت میانگین را در آخر سر برای تعداد 100 آزمایش خدمت شما ارائه خواهیم کرد.

نتایج برای 40% missing points برای مسأله قسمت الف به شرح زیر است:

#### INPUT No.1 with MISSING POINTS (About 40%) with ITS DESIRED OUTPUT:

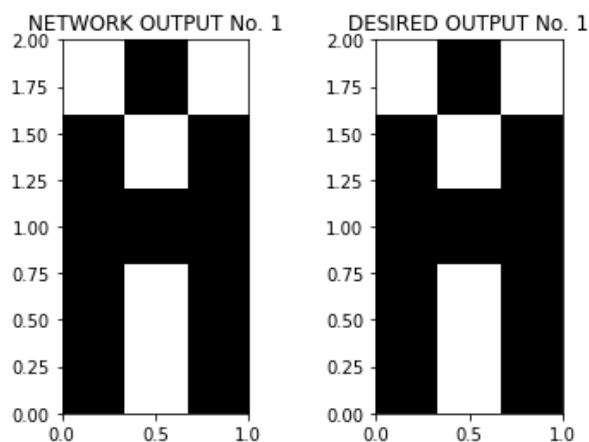
برای اولین ورودی نتایج ورودی شامل missing points و خروجی مطلوب به شرح زیر است:



شکل 43 ورودی شماره یک با اعمال حذف اطلاعات 40 درصد و خروجی مطلوب برای این ورودی

حال اثر این missing points را برای ورودی اول در خروجی شبکه و در مقایسه با خروجی مطلوب ترسیم می‌کنیم که به شرح زیر است:

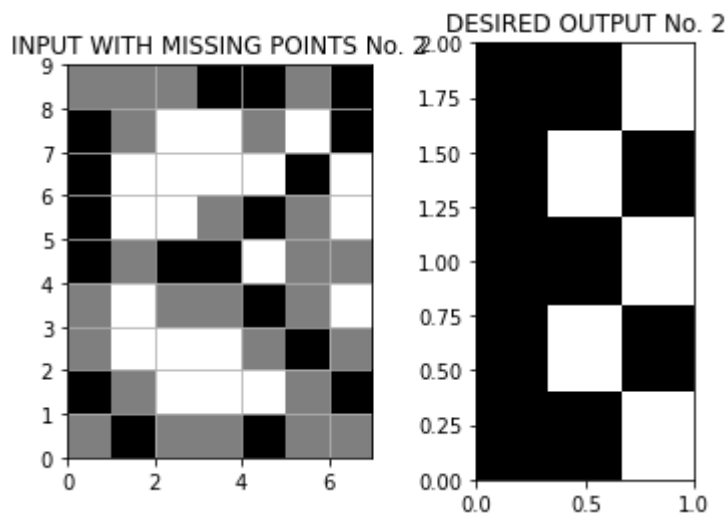
#### OUTPUT No.1 (FOR 40% MISSING POINTS) VS ITS DESIRED OUTPUT:



شکل 44 خروجی گرفته شده از شبکه برای ورودی شماره یک (اعمال 40 درصد حذف اطلاعات) و خروجی مطلوب برای این ورودی

### INPUT No.2 with MISSING POINTS (About 40%) with ITS DESIRED OUTPUT:

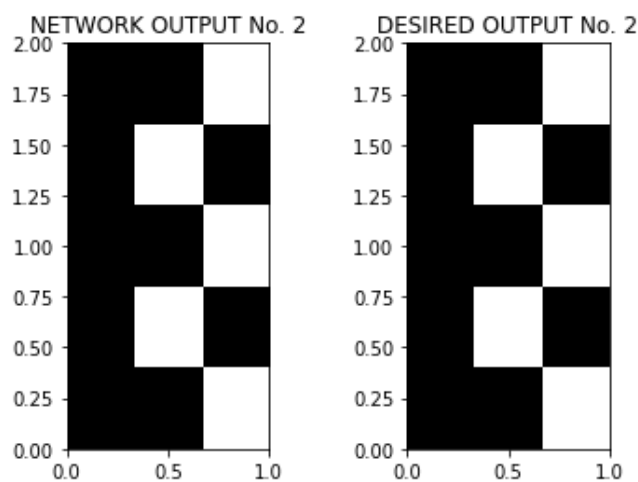
برای دومین ورودی نتایج ورودی شامل missing points و خروجی مطلوب به شرح زیر است:



شکل 45 ورودی شماره دو با اعمال حذف اطلاعات 40 درصد و خروجی مطلوب برای این ورودی

### OUTPUT No.2 (FOR 40% MISSING POINTS) VS ITS DESIRED OUTPUT:

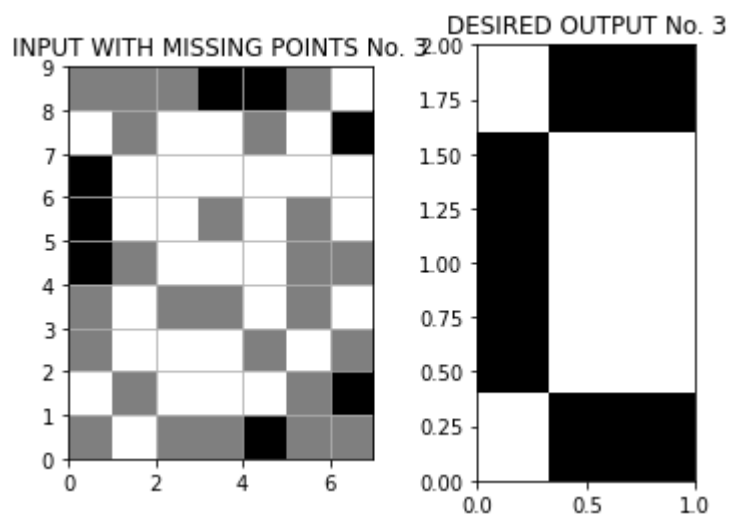
حال اثر این missing points را برای ورودی دوم در خروجی شبکه و در مقایسه با خروجی مطلوب ترسیم می‌کنیم که به شرح زیر است:



شکل 46 خروجی گرفته شده از شبکه برای ورودی شماره دو (اعمال 40 درصد حذف اطلاعات) و خروجی مطلوب برای این ورودی

### INPUT No.3 with MISSING POINTS (About 40%) with ITS DESIRED OUTPUT:

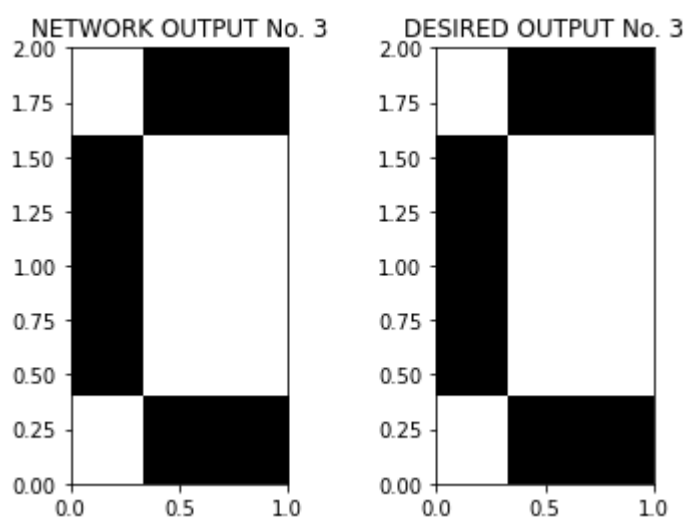
برای سومین ورودی نتایج ورودی شامل missing points و خروجی مطلوب به شرح زیر است:



شکل 47 ورودی شماره سه با اعمال حذف اطلاعات 40 درصد و خروجی مطلوب برای این ورودی

### OUTPUT No.3 (FOR 40% with MISSING POINTS) VS ITS DESIRED OUTPUT:

حال اثر این missing points را برای ورودی سوم در خروجی شبکه و در مقایسه با خروجی مطلوب ترسیم می‌کنیم که به شرح زیر است:



شکل 48 خروجی گرفته شده از شبکه برای ورودی شماره سه (اعمال 40 درصد حذف اطلاعات) و خروجی مطلوب برای این ورودی

همانطور که مشاهده می‌فرمایید 40% missing points در ورودی نمی‌تواند در ارائه‌ی نتایج مطلوب توسط شبکه (برای هر سه ورودی) تاثیری بگذارد و شبکه به خوبی عمل می‌کند.

همانطور که صورت سوال خواسته است میانگین صحت را برای عملکرد شبکه محاسبه کردیم.

به ازای 10% missing points نتایج به صورت زیر است.

جدول 5 میانگین صحت بدست آمده برای شبکه با خروجی‌های 3\*5 و حذف اطلاعات 10 درصد

Average Accuracy for Input No.1	100%
Average Accuracy for Input No.2	100%
Average Accuracy for Input No.3	100%
Mean Average Accuracies	100%

حال به ازای 40% missing points نیز جدول بالا را ارائه می‌کنیم:

جدول 6 میانگین صحت بدست آمده برای شبکه با خروجی‌های 3\*5 و حذف اطلاعات 40 درصد

Average Accuracy for Input No.1	100%
Average Accuracy for Input No.2	100%
Average Accuracy for Input No.3	100%
Mean Average Accuracies	100%

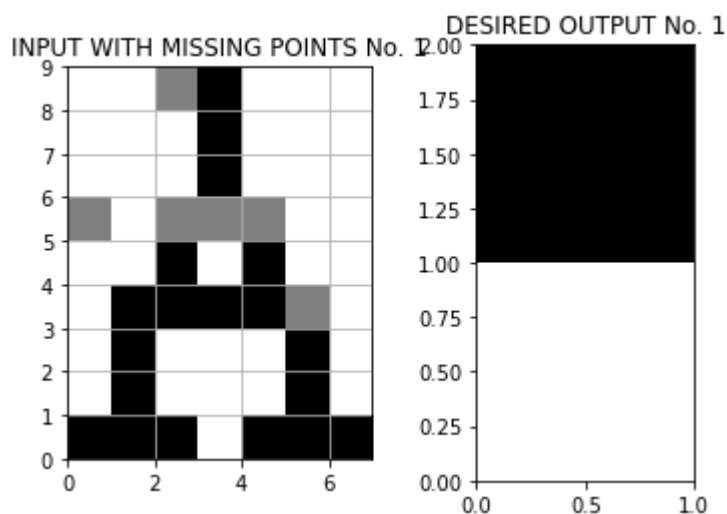
همانطور که مشاهده می‌نمایید، missing points برای این دو مقدار نتوانست عملکرد خوب شبکه را بهم بریزد که نتیجتاً می‌توانیم بگوییم که شبکه نسبت به missing points ها بسیار مقاوم است.

حال مجدداً تمامی مراحل فوق را برای قسمت ب مسأله تکرار می‌کنیم.

مجدداً در این مرحله نیز مطابق با خواسته‌ی مسأله ما در گام اول missing points معادل با 10% بر روی ورودی‌ها اعمال کردیم و ورودی‌های شامل missing points را به عنوان ورودی شبکه‌ی نهایی وارد نمودیم تا ببینیم به چه اندازه در تشخیص خروجی درست، موفق عمل خواهد کرد.

### INPUT No.1 with MISSING POINTS (About 10%) with ITS DESIRED OUTPUT:

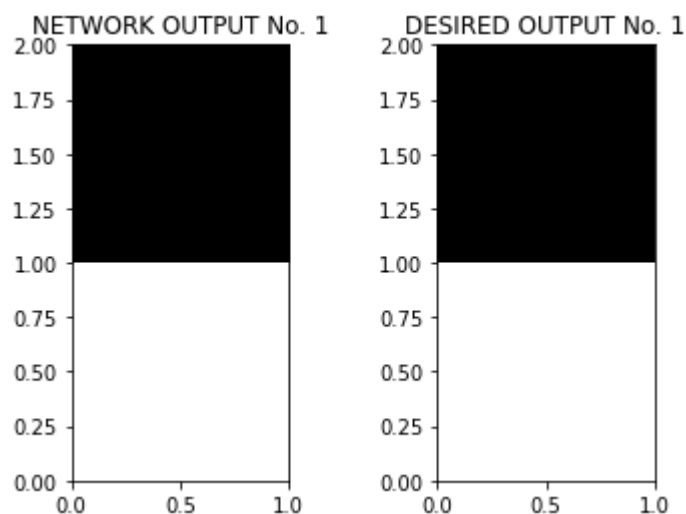
برای اولین ورودی نتایج ورودی شامل missing points و خروجی مطلوب به شرح زیر است:



شکل 49 ورودی شماره یک با اعمال حذف اطلاعات 10 درصد و خروجی مطلوب برای این ورودی

حال اثر این missing points را برای ورودی اول در خروجی شبکه و در مقایسه با خروجی مطلوب ترسیم می‌کنیم که به شرح زیر است:

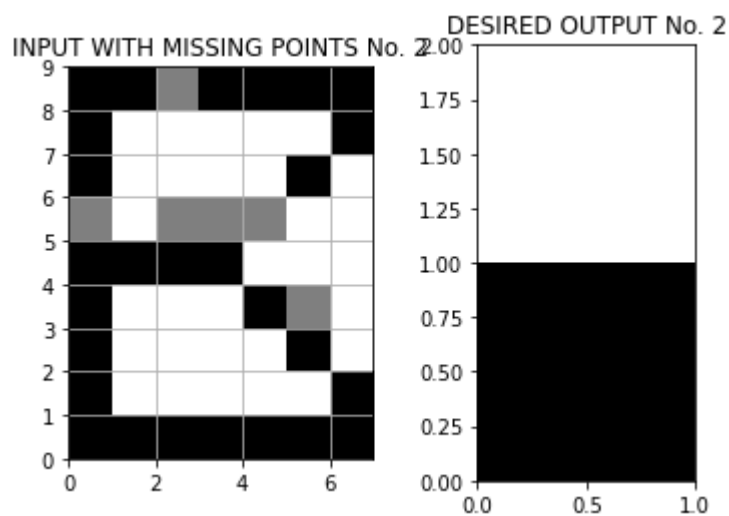
### OUTPUT No.1 (FOR 10% MISSING POINTS) VS ITS DESIRED OUTPUT:



شکل 50 خروجی گرفته شده از شبکه برای ورودی شماره یک (اعمال 10 درصد حذف اطلاعات) و خروجی مطلوب برای این ورودی

### INPUT No.2 with MISSING POINTS (About 10%) with ITS DESIRED OUTPUT:

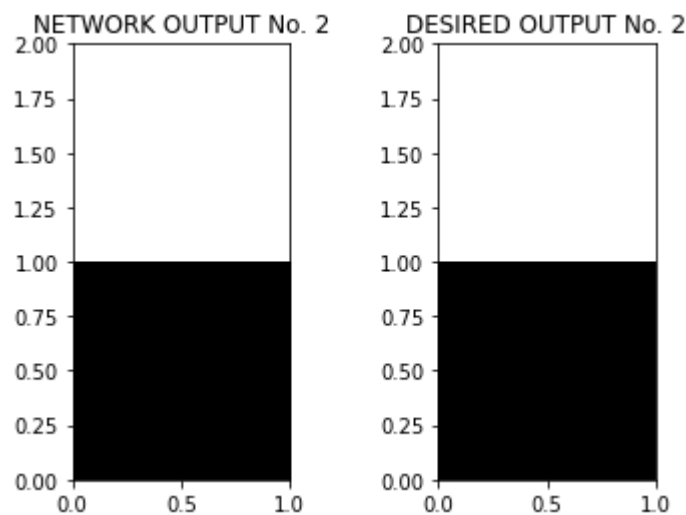
برای دومین ورودی نتایج ورودی شامل missing points و خروجی مطلوب به شرح زیر است:



شکل 51 ورودی شماره دو با اعمال حذف اطلاعات 10 درصد و خروجی مطلوب برای این ورودی

### OUTPUT No.2 (FOR 10% MISSING POINTS) VS ITS DESIRED OUTPUT:

حال اثر این missing points را برای ورودی دوم در خروجی شبکه و در مقایسه با خروجی مطلوب ترسیم می‌کنیم که به شرح زیر است:

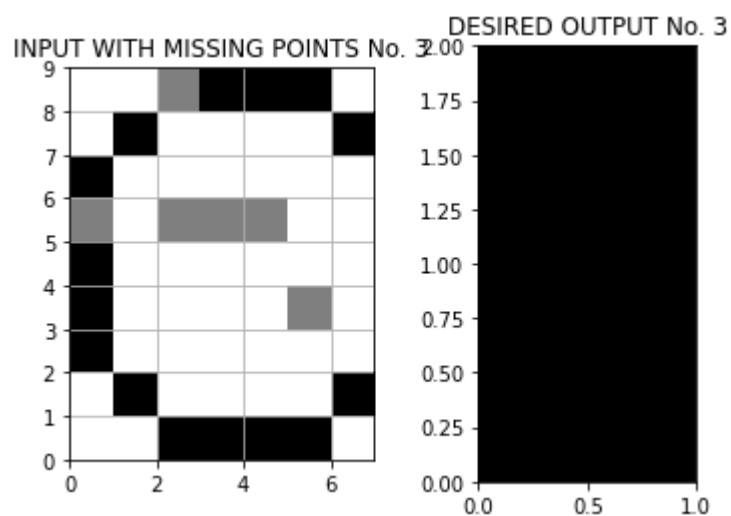


شکل 52 خروجی گرفته شده از شبکه برای ورودی شماره دو (اعمال 10 درصد حذف اطلاعات) و خروجی مطلوب برای این ورودی



### INPUT No.3 with MISSING POINTS (About 10%) with ITS DESIRED OUTPUT:

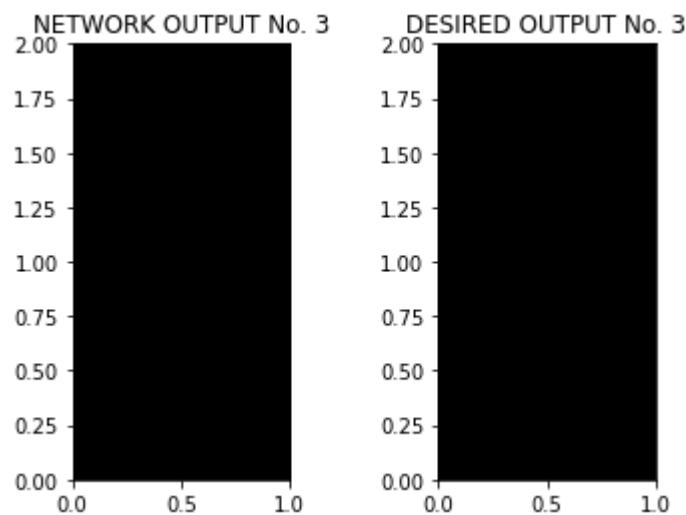
برای سومین ورودی نتایج ورودی شامل missing points و خروجی مطلوب به شرح زیر است:



شکل 53 ورودی شماره سه با اعمال حذف اطلاعات 10 درصد و خروجی مطلوب برای این ورودی

### OUTPUT No.3 (FOR 10% MISSING POINTS) VS ITS DESIRED OUTPUT:

حال اثر این missing points را برای ورودی سوم در خروجی شبکه و در مقایسه با خروجی مطلوب ترسیم می‌کنیم که به شرح زیر است:



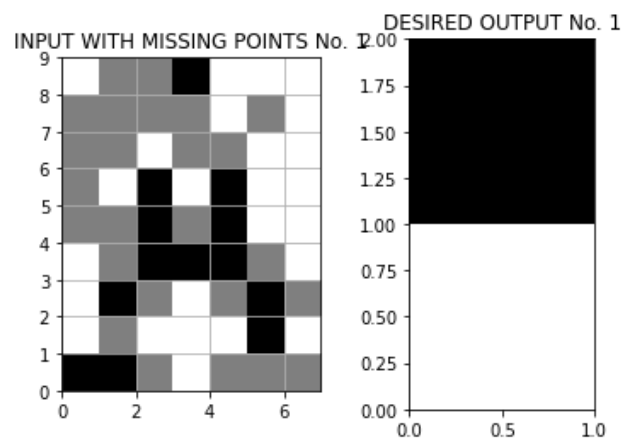
شکل 54 خروجی گرفته شده از شبکه برای ورودی شماره سه (اعمال 10 درصد حذف اطلاعات) و خروجی مطلوب برای این ورودی

همانطور که مشاهده می‌فرمایید 10% missing points در ورودی نمی‌تواند در ارائه‌ی نتایج مطلوب توسط شبکه (برای هر سه ورودی) تاثیری بگذارد. حال طبق خواسته‌ی مسأله missing points را بالاتر (40%) می‌بریم و نتایج را مجدداً ارائه می‌کنیم. توجه داشته باشید که صحت میانگین را در آخر سر برای تعداد 100 آزمایش خدمت شما ارائه خواهیم کرد.

نتایج برای 40% missing points برای مسأله قسمت الف به شرح زیر است:

#### INPUT No.1 with MISSING POINTS (About 40%) with ITS DESIRED OUTPUT:

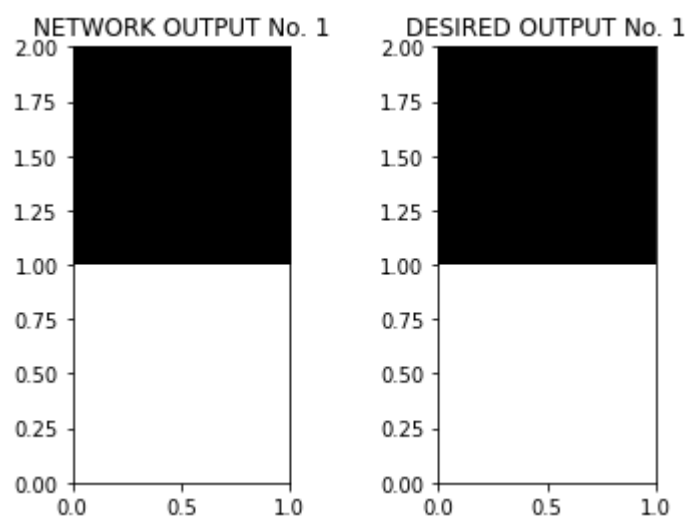
برای اولین ورودی نتایج ورودی شامل missing points و خروجی مطلوب به شرح زیر است:



شکل 55 ورودی شماره یک با اعمال حذف اطلاعات 40 درصد و خروجی مطلوب برای این ورودی

حال اثر این missing points را برای ورودی اول در خروجی شبکه و در مقایسه با خروجی مطلوب ترسیم می‌کنیم که به شرح زیر است:

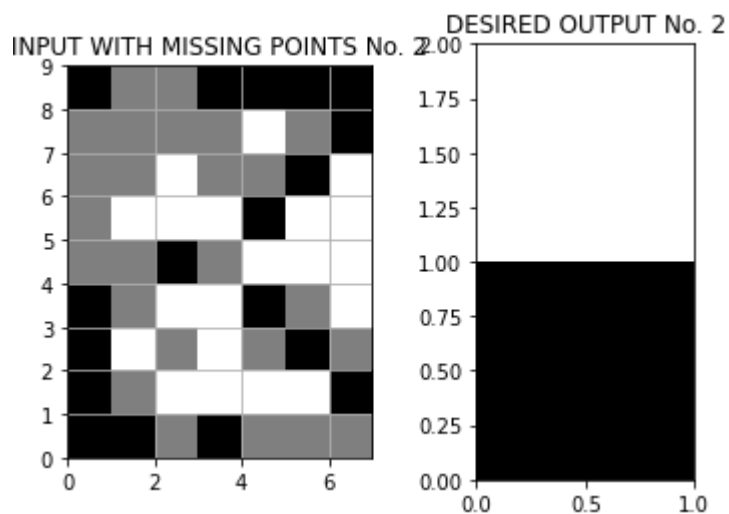
#### OUTPUT No.1 (FOR 40% MISSING POINTS) VS ITS DESIRED OUTPUT:



شکل 56 خروجی گرفته شده از شبکه برای ورودی شماره یک (اعمال 40 درصد حذف اطلاعات) و خروجی مطلوب برای این ورودی

### INPUT No.2 with MISSING POINTS (About 40%) with ITS DESIRED OUTPUT:

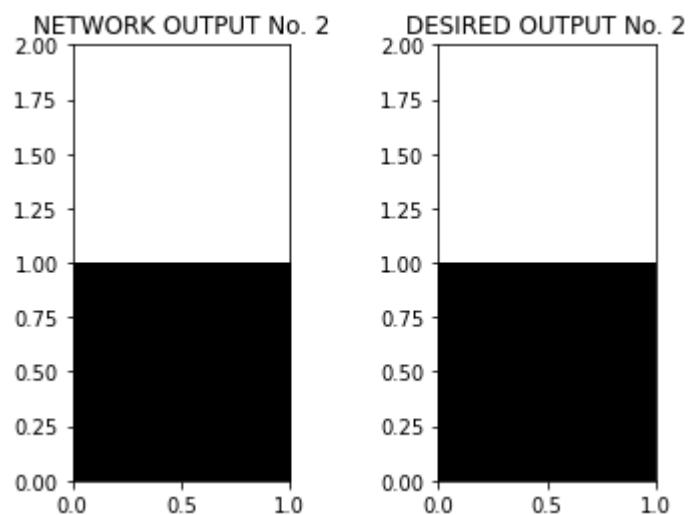
برای دومین ورودی نتایج ورودی شامل missing points و خروجی مطلوب به شرح زیر است:



شکل 57 ورودی شماره دو با اعمال حذف اطلاعات 40 درصد و خروجی مطلوب برای این ورودی

### OUTPUT No.2 (FOR 40% MISSING POINTS) VS ITS DESIRED OUTPUT:

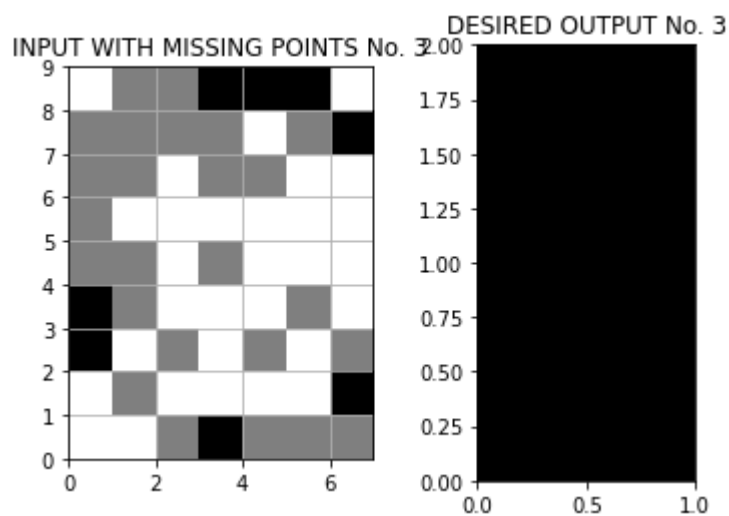
حال اثر این missing points را برای ورودی دوم در خروجی شبکه و در مقایسه با خروجی مطلوب ترسیم می‌کنیم که به شرح زیر است:



شکل 58 خروجی گرفته شده از شبکه برای ورودی شماره دو (اعمال 40 درصد حذف اطلاعات) و خروجی مطلوب برای این ورودی

### INPUT No.3 with MISSING POINTS (About 40%) with ITS DESIRED OUTPUT:

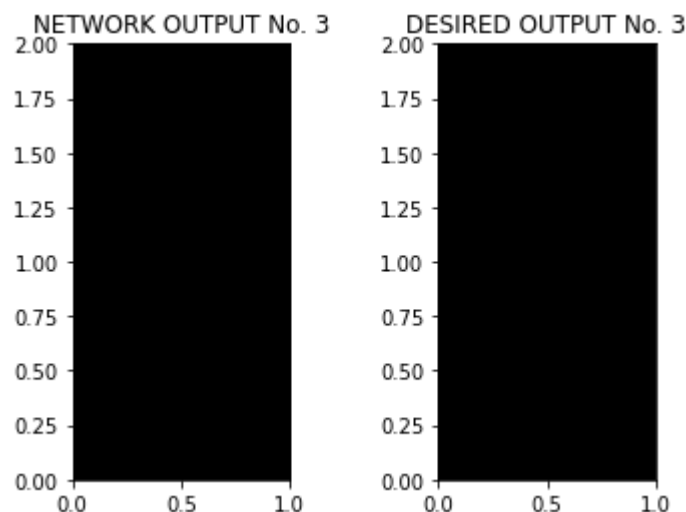
برای سومین ورودی نتایج ورودی شامل missing points و خروجی مطلوب به شرح زیر است:



شکل 59 ورودی شماره سه با اعمال حذف اطلاعات 40 درصد و خروجی مطلوب برای این ورودی

### OUTPUT No.3 (FOR 40% MISSING POINTS) VS ITS DESIRED OUTPUT:

حال اثر این missing points را برای ورودی سوم در خروجی شبکه و در مقایسه با خروجی مطلوب ترسیم می‌کنیم که به شرح زیر است:



شکل 60 خروجی گرفته شده از شبکه برای ورودی شماره سه (اعمال 40 درصد حذف اطلاعات) و خروجی مطلوب برای این ورودی

همانطور که مشاهده می‌فرمایید 40% missing points در ورودی نمی‌تواند در ارائه‌ی نتایج مطلوب توسط شبکه (برای هر سه ورودی) تاثیری بگذارد و شبکه به خوبی عملکرد خود را حفظ می‌کند. همانطور که صورت سوال خواسته است میانگین صحت را برای عملکرد شبکه محاسبه کردیم. به ازای 10% missing points نتایج به صورت زیر است.

جدول 7 میانگین صحت بدست آمده برای شبکه با خروجی‌های 1\*2 و حذف اطلاعات 10 درصد

Average Accuracy for Input No.1	100%
Average Accuracy for Input No.2	100%
Average Accuracy for Input No.3	100%
Mean Average Accuracies	100%

حال به ازای 40% missing points نیز جدول را ارائه می‌کنیم:

جدول 8 میانگین صحت بدست آمده برای شبکه با خروجی‌های 1\*2 و حذف اطلاعات 40 درصد

Average Accuracy for Input No.1	100%
Average Accuracy for Input No.2	100%
Average Accuracy for Input No.3	100%
Mean Average Accuracies	100%

همانطور که مشاهده می‌نمایید، missing points برای این دو مقدار و برای مسأله‌ی بخش ب نیز نتوانست عملکرد خوب شبکه را بهم بریزد که نتیجتاً می‌توانیم بگوییم که شبکه نسبت به missing points ها بسیار مقاوم است.

د) مقاومت شبکه در برابر نویز بیشتر است یا از دست دادن اطلاعات؟ تاثیر ابعاد خروجی بر مقاومت شبکه چیست؟

پاسخ بخش د)

مقاوم بودن شبکه برای خروجی با ابعاد کمتر (1\*2) و نیز به ازای missing points ها از نتایج بدست آمد.

همانطور که مشاهده کردیم، متوسط صحت‌ها برای ورودی‌هایی دارای missing points بسیار بالاتر و در واقع مقاوم‌تر از داده‌های نویزی بود. ما برای اطمینان از صحت حرف خود تعداد missing points را تا 80% نیز بالاتر بردیم و نتایج با متوسط میانگین صحت 96% و 94% برای خروجی‌های به ترتیب 1\*2 و 3\*5 به ازای ورودی‌ها بدست آمدند. این مقاومت شبکه‌ی ما را نسبت به از دست دادن اطلاعات در مقابل نویزی بودن اطلاعات نشان می‌دهد. همچنین مشاهده کردیم که نتایج برای ابعاد خروجی کمتر بهتر بودند چه از نظر نویزی کردن و چه از نظر از دست دادن برخی اطلاعات. مجدداً نتایج را ذکر می‌نماییم تا به صحت گفته‌ها اطمینان حاصل کنیم.

برای بعد 3\*5

جدول 9 مقایسه تمامی نتایج برای خروجی 3\*5

	Average Accuracy for Input No.1	Average Accuracy for Input No.2	Average Accuracy for Input No.3	Mean Average Acc
10% noise	100%	100%	100%	100%
10% missing info	100%	100%	100%	100%
40% noise	68%	60%	54%	60.66%
40% missing info	100%	100%	100%	100%
80% noise	0%	0%	0%	0%
80% missing info	98%	92%	92%	94%

برای بعد 1\*2

جدول 10 مقایسه تمامی نتایج برای خروجی 1\*2

	Average Accuracy for Input No.1	Average Accuracy for Input No.2	Average Accuracy for Input No.3	Mean Average Acc
10% noise	100%	100%	100%	100%
10% missing info	100%	100%	100%	100%
40% noise	89%	81%	77%	82.33%
40% missing info	100%	100%	100%	100%
80% noise	0%	0%	0%	0%
80% missing info	98%	94%	98%	96.66%

## سوال ۲ – Auto-associative Net

(۱) وزنهای شبکه را با استفاده از Modified Hebbian Learning Rule بیابید.

پاسخ بخش 1

نحوه آپدیت وزن‌ها در Modified Hebbian Rule به صورت زیر است:

Modified Hebbian Rule

$$W = \sum_{p=1}^P s(p)s(p)^T - PI_n$$

و همچنین activation function مورد استفاده در این روش  $\text{sign}()$  می‌باشد. اکنون طبق خواسته‌ی مسأله با استفاده از فومول فوق و تعریف ورودی‌های مورد نظر صورت سوال اقدام به محاسبه‌ی ماتریس وزن شبکه می‌کنیم که نتیجه به صورت زیر خواهد بود:

وزن بدست آمده با استفاده از قانون Modified Hebbian:

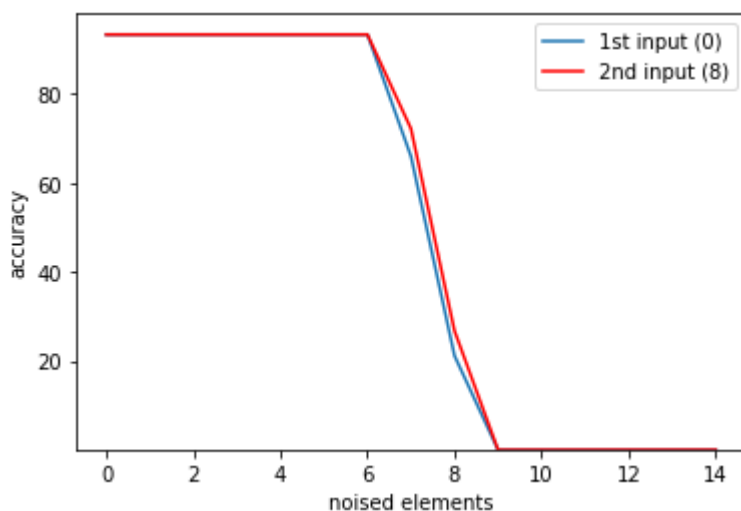
```
Modified Hebbian Rule
[[ 0.  2.  2.  2. -2.  2.  2.  0.  2.  2. -2.  2.  2.  2.  2.]
 [ 2.  0.  2.  2. -2.  2.  2.  0.  2.  2. -2.  2.  2.  2.  2.]
 [ 2.  2.  0.  2. -2.  2.  2.  0.  2.  2. -2.  2.  2.  2.  2.]
 [ 2.  2.  2.  0. -2.  2.  2.  0.  2.  2. -2.  2.  2.  2.  2.]
 [-2. -2. -2. -2.  0. -2. -2.  0. -2. -2.  2. -2. -2. -2. -2.]
 [ 2.  2.  2.  2. -2.  0.  2.  0.  2.  2. -2.  2.  2.  2.  2.]
 [ 2.  2.  2.  2. -2.  2.  0.  0.  2.  2. -2.  2.  2.  2.  2.]
 [ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [ 2.  2.  2.  2. -2.  2.  2.  0.  0.  2. -2.  2.  2.  2.  2.]
 [ 2.  2.  2.  2. -2.  2.  2.  0.  2.  0. -2.  2.  2.  2.  2.]
 [-2. -2. -2. -2.  2. -2. -2.  0. -2. -2.  0. -2. -2. -2. -2.]
 [ 2.  2.  2.  2. -2.  2.  2.  0.  2.  2. -2.  0.  2.  2.  2.]
 [ 2.  2.  2.  2. -2.  2.  2.  0.  2.  2. -2.  2.  0.  2.  2.]
 [ 2.  2.  2.  2. -2.  2.  2.  0.  2.  2. -2.  2.  2.  0.  2.]
 [ 2.  2.  2.  2. -2.  2.  2.  0.  2.  2. -2.  2.  2.  2.  0.]]
```

شکل 61 وزن بدست آمده برای دو ورودی مورد نظر به جهت به حافظه سپاری مطابق با روش Modified Hebbian Rule

۲) درصد موفقیت تست قوام شبکه (به جای ۱ و ۱- و به جای ۱- و ۱ قرار دهید) با یک اشتباه در ورودی را به دست آورید. در ادامه نموداری به صورت زیر برای تمامی حالات تعداد اشتباه در دو تصویر نمایش دهید.

## پاسخ بخش ۲)

شبکه بدست آمده برای یک اشتباه در ورودی، 14 عنصر را بدرستی پیش‌بینی کرده و تنها یک خطا خواهد داشت. بدین صورت درصد موفقیت برای اعمال نویز بر روی تنها یک المان برابر با 93.33% خواهد بود. حال ما این روند را تکرار کرده و با اعمال نویز بر روی تعداد بیشتری المان ورودی، نتایج را بدست آوردیم. به جهت اینکه تمامی ترکیب‌های ممکن برای انتخاب هر چندتایی از المان در نتایج ظاهر شوند و تاثیر خود را بر روی صحت نهایی بگذارند، ما از قوانین احتمالات حساب کرده و میانگین را برای تعداد بسیار زیادی از یک‌تایی (انتخاب یک المان رندوم برای اعمال نویز)، دو تایی، سه تایی و... تا 15 تایی را محاسبه کردیم. بدین صورت می‌توانستیم با انتخاب تعداد iteration های زیاد، سهم تغییر هر المان از انتخاب چندتایی را برای بررسی تاثیرگذاری بر روی صحت یکسان کنیم. در نهایت نتایج به صورت زیر درآمد.



شکل 62 صحت برآورد شده بر اثر اعمال نویز بر روی المان‌ها به ازای تعداد المان نویزی

همانطور که مشاهده می‌فرمایید، عملاً شبکه در حالت بدون نویز نیست دچار خطا شده و تفاوت دو ورودی را از یکدیگر تشخیص نمی‌دهد. این مسأله را از جنبه‌های گوناگون بررسی می‌کنیم. اولین مسأله اینکه دو ورودی شرط کافی را از نقطه نظر عمود بودن ندارند و طبیعتاً نباید انتظار داشته باشیم که شبکه

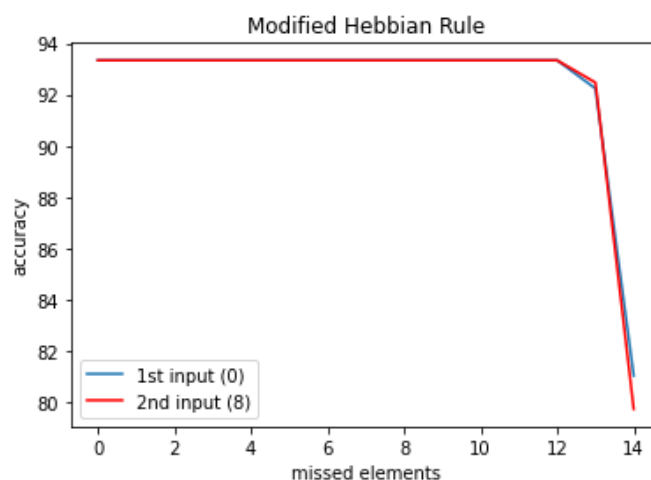


صد در صد بتواند این دو ورودی را به خوبی به حافظه بسپارد که البته از نتایج نیز این موضوع را مشاهده می‌کنیم. دومین موردی که از مقایسه‌ی این دو ورودی مشهود است، شباهت بالای دو ورودی با یکدیگر است (اختلاف تنها در یک بیت است) و لذا ممکن است شبکه نسبت به این میزان شباهت، robustness خوبی را از خود نشان ندهد. سومین نکته تعداد المان‌های زیاد هر تصویر است (با لحاظ شباهتی است که بین دو ورودی وجود دارد). طبیعتاً اگر دو المان داشتیم و ورودی‌ها با یکدیگر تنها در یک المان متفاوت بودند، می‌توانستیم انتظار این را داشته باشیم که شبکه قدرت به خاطر سپاری خوبی داشته باشد، ولی در اینجا که تفاوت در یک المان در مقیاس 15 المان از ابعاد ورودی مطرح است، طبیعتاً شبکه قدرت به خاطر سپاری صد در صدی را شاید نتواند از خود نشان دهد که از نتایج نیز کاملاً این موضوع مشهود است. با این حال شبکه باز هم توانسته است با این میزان شباهت، صحت بالایی و نیز مقاومت خوبی در برابر نویز نشان دهد. (مجدداً در بخش 4م این سوال تاثیر شباهت را در کنار ورودی جدیدی که سوال ذکر کرده است، بررسی می‌کنیم).

۳) مراحل فوق را برای حالت داده‌ها از بین رفته باشند (به جای مقادیر ۱ و -۱ صفر قرار گیرد) تکرار کنید. تفاوت موجود در نمودار به چه معناست؟ آیا نتیجه‌گیری شما قابل تعمیم است؟

### پاسخ بخش 3

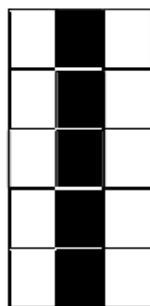
در این مرحله از پیاده‌سازی مرحله‌ی قبل استفاده کرده و تنها برای مقادیر انتخاب شده برای حذف اطلاعات، صفر قرار می‌دهیم. روند مشابه قبل است و نتیجه‌ی بدست آمده به صورت زیر می‌باشد.



شکل 63 صحت برآورد شده بر اثر حذف اطلاعات بر روی المان‌های ورودی به ازای تعداد المان حذف شده

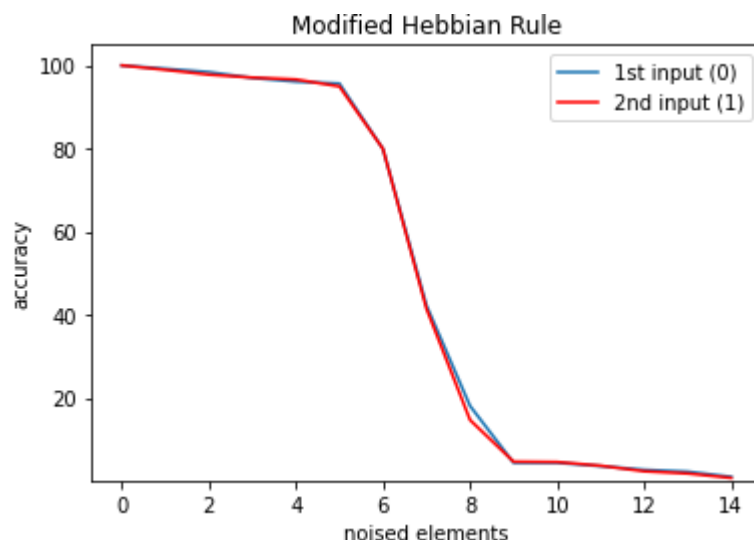
همانطور که از نتیجه بدست آمده مشاهده می‌فرمایید، این شبکه نسبت به حذف اطلاعات بسیار robustتر است نسبت به نویزی بونِ اطلاعات. طبق این نتایج شبکه با حذف 12 المان از هر کدام از ورودی‌ها توانسته است که صحت خود را بر روی 93.33% حفظ کند و سپس اگرچه کاهش صحت رخ می‌دهد ولی در مقایسه با نویزی بودن اطلاعات می‌بینیم که در اینجا بسیار مقاوم‌تر عمل کرده است در مقایسه با قسمت قبل. در مورد اینکه این نتیجه قابل تعمیم است یا خیر باید نکته‌ای را یادآوری کنیم و آن این است که اطلاعات موجود در یک آرایه ممکن است سهم برابری در تشخیص ورودی‌ها نداشته باشند و به همین دلیل است که فلسفه‌ی وزن‌ها و آموزش وزن‌ها شکل گرفته است. فلذا اینکه کدام بعد از داده از مجموعه ریکوردهای یک دیتاست حذف شده طبیعتاً با یک دیتاست دیگر فرق دارد و همین تعداد بُدهای دارای اهمیت بالا (و طبیعتاً وزن‌های اختصاصی زیاد) و نیز کوریلیشن بین مقادیر بُدهای (ستون‌های) ریکوردهای موجود در یک دیتاست و عواملی از این قبیل بر مقاومت شبکه در برابر حذف اطلاعات تاثیرگذار است و قبل از تعمیم این موضوع باید شبکه جدید را از نظر تمامی فاکتورهای ذکر شده بین ریکوردهایش در مقایسه با شبکه فعلی خود بررسی کنیم و سپس این نتایج را تعمیم دهیم. ولی به طور کلی شبکه‌های حافظه دار با ریکوردهای باینری و با تعداد ابعاد بالا دارای robustness بیشتری نسبت به noise هستند. این موضوع را می‌توان در ماهیت نویز و حذف اطلاعات نشان داد. در واقع با اعمال نویز به یک سیستم بویژه یک سیستم باینری (در سیستم باینری، یک داده‌ی نویزی ماهیت قطب مخالف را کسب می‌کند) در صورتی که در هنگام از دست دادن اطلاعات، بُدهای متفاوتی از ریکورها بررسی نمی‌شوند و این موضوع می‌تواند توسط سایر ریکوردهای با اهمیت که توانایی تشخیص شبکه با وزن‌دهی زیاد بر روی این ابعاد است، جبران شود و نهایتاً نتیجه‌ی مورد انتظار خروجی بدست آید.

۴) شبکه جدیدی با ماتریس سمت چپ (عدد ۰) و ماتریس زیر بسازید. مراحل ۲ را تکرار کنید. شهود نمودار جدید و تفاوت آن با حالت قبل را بیان کنید.



#### پاسخ بخش 4)

در این بخش مجدداً مراحل بخش دوم را بر روی ماتریس سمت چپ و ماتریس جدید موجود در این بخش اجرا می‌کنیم و مطابق با گام‌های طی شده در بخش دوم به خروجی زیر برای صحت در ازای تعداد اطلاعات (بُدهای) نویزی می‌رسیم.



شکل 64 صحت برآورد شده بر اثر اعمال نویز بر روی المان‌ها به ازای تعداد المان نویزی برای بخش 4 سوال دوم تمرین

در نتایج فوق، بهبودی شبکه در تشخیص را مشاهده می‌فرمایید. این موضوع دور از انتظار نیست و می‌توان به میزان شباهت دو ورودی در این بخش در مقایسه با بخش 2 ارجاع داد. همانگونه که مشاهده می‌فرمایید در این بخش، ورودی‌ها دارای شباهت کمتری نسبت به بخش دوم هستند و طبیعتاً شبکه توانسته قدرت خوبی در به خاطر سپاری ورودی‌ها از خود نشان دهد. همچنین مشاهده می‌کنیم که برای زمانی که هیچگونه نویزی به شبکه وارد نشده است، میانگین صحت بدست آمده برابر با 100% می‌باشد و این بهبودی شبکه را برای به خاطر سپاری داده‌گان با شباهت کمتر را نشان می‌دهد. همچنین در برابر نویز هم مقاومت بیشتری نسب به بخش دوم دارد که باز دلیل آن را به همین شباهت کمتر بین دو نمونه‌ی ورودی می‌یابیم.

### سوال 3 – Discrete Hopfield Net

۱) تصویر ورودی (شکل سمت چپ) را با استفاده از کد ارسال شده به حالت باینری تبدیل کرده و سائز آن را به ۱۰۰ در ۱۰۰ کاهش دهید.

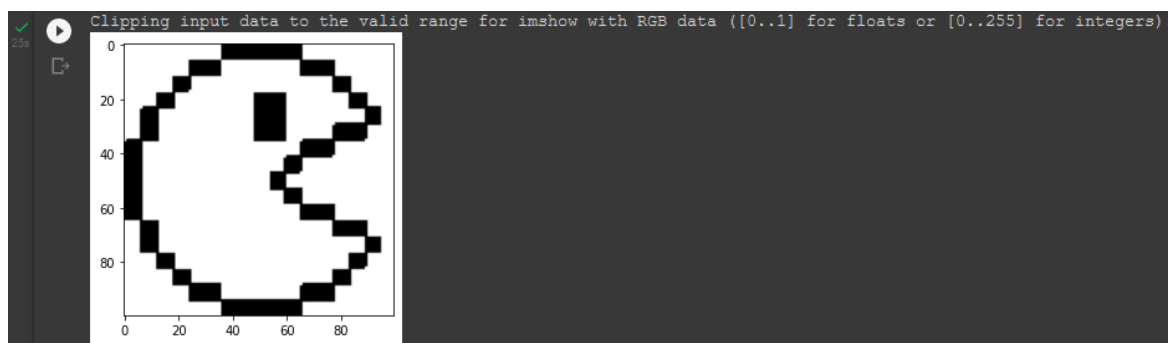
**نکته:** در این مرحله نیازی به کد زدن نمی باشد و تنها با Run کردن کدی که برای شما قرار داده شده است، به تصویر سیاه و سفیدی که قرار است با آن کار کنید می رسید.

#### پاسخ بخش 1

مطابق با خواسته‌ی سوال کد را در colab ران کرده و فایل‌های تصویر را در google drive قرار دادیم.

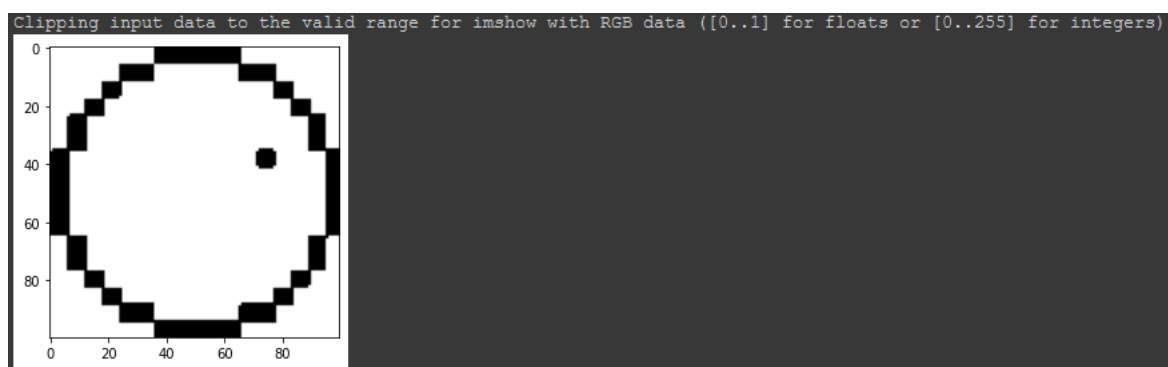
خروجی مطابق زیر بدست آمد:

نتیجه استخراج تصویر آموزش به صورت زیر است:



شکل 65 اجرای کد قرار گرفته در قسمت الف سوال 3 و استخراج تصویر آموزش

و نتیجه استخراج تصویر تست نیز به صورت زیر است:

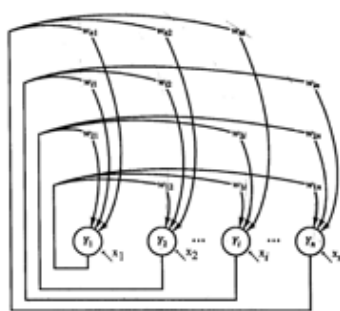


شکل 66 اجرای کد قرار گرفته در قسمت الف سوال 3 و استخراج تصویر آموزش

۲) با استفاده از تصویر باینری ورودی، ماتریس وزن ها را بسازید.

پاسخ بخش 2)

شبکه *Discrete Hopfield Net* دارای ساختار زیر است و نحوه محاسبه وزن هایش برای ورودی bipolar بدین صورت می باشد:



To store a set of bipolar patterns  $s(p)$ ,  $p = 1, \dots, P$ , where  
 $s(p) = (s_1(p), \dots, s_i(p), \dots, s_n(p))$ ,  
the weight matrix  $W = \{w_{ij}\}$  is given by  

$$w_{ij} = \sum_p s_i(p)s_j(p) \quad \text{for } i \neq j$$
and  

$$w_{ii} = 0.$$

The application algorithm is stated for binary patterns; the activation function can be modified easily to accommodate bipolar patterns.

شکل 67 شبکه *Discrete Hopfield Net* و نحوه محاسبه وزن ها برای ورودی bipolar (برگرفته از جزوه دکتر کلهر-دانشگاه تهران)

مطابق با روند ذکر شده، ماتریس وزن و نیز سایز این ماتریس به صورت زیر بدست آمدند.

```

✓ [52] weights are equal:
2s [[0. 1. 1. ... 1. 1. 1.]
    [1. 0. 1. ... 1. 1. 1.]
    [1. 1. 0. ... 1. 1. 1.]
    ...
    [1. 1. 1. ... 0. 1. 1.]
    [1. 1. 1. ... 1. 0. 1.]
    [1. 1. 1. ... 1. 1. 0.]]
size of weight matrix is:
(10000, 10000)

```

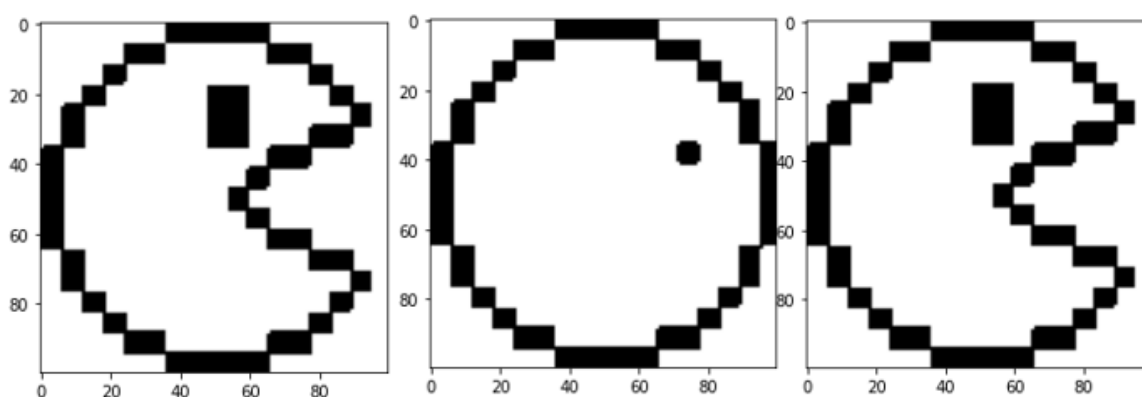
شکل 68 ماتریس وزن و سایز این ماتریس برای بخش 3 سوال دوم و برای ورودی آموزش موجود در تصویر 65

با توجه به اینکه ورودی آموزش  $100 \times 100$  و در واقع ریکورد برابر با 10000 بُعد است لذا ماتریس وزن مربوط به این شبکه می بایست  $10000 \times 10000$  باشد. همانطور که مشاهده می کنید قطر دارای درایه های صفر و بقیه عناصر از -1 و 1 نیز تشکیل شده اند که نشان از درستی محاسبات مربوط به این قسمت را می دهد. با این حال برای بررسی صحت ماتریس وزن به بخش بعدی مراجعه می کنیم تا ببینیم خروجی مطلوب در ازای ورودی ذکر شده استخراج می شود یا خیر که خواهیم دید شبکه به خوبی برای ورودی داده ی تست این عملکرد را از خود نشان می دهد.

۳) با کمک ماتریس وزنها و با استفاده از تصویر سمت چپ سعی کنید تا تصویر اصلی را بازیابی کنید. برای این کار نیاز است که در  $iteration$  های پیاپی به تصویر سالم نزدیک شده تا اینکه در نهایت به تصویری واضح برسید. بعد از  $iteration$  ۱۰۰۰ بدست آمده را رسم کنید تا همگرایی تصویر سمت راست به تصویر چپ دیده شود.

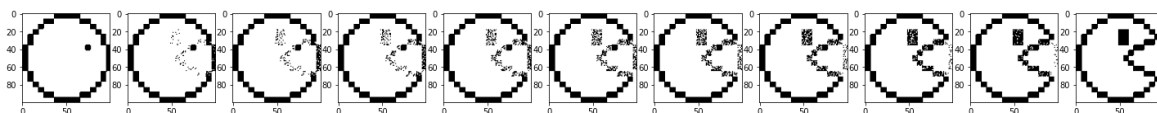
### پاسخ بخش ۳

طبق خواسته‌ی سوال ورودی تست را به شبکه دادیم تا عملیات prediction انجام شود. ابتدا ورودی و خروجی مطلوب و خروجی شبکه را برای ورودی تصویر تست بررسی می‌کنیم.



شکل ۶۹ تصویر آموزش (سمت چپ) و تصویر تست (وسط) و خروجی شبکه (سمت راست)

همانطور که مشاهده می‌کنید پس از  $10000$  iteration یا به عبارتی یک دور کامل آموزش خروجی‌ها (مساوی با بعد خروجی یا المان‌ها تصویر ورودی) به تصویر به خاطر سپاری شده رسیدیم. حال طبق صورت سوال برای هر  $1000$  iteration اقدام به رسم تصویر می‌کنیم. نتیجه به صورت زیر شده است:



شکل ۷۰ مراحل بازسازی تصویر به خاطر سپرده شده توسط شبکه برای ورودی تست در  $10000$  گام

همانطور که مشاهده می‌فرمایید برای  $iteration$  های متوالی همگرایی بیشتر و بیشتر و به عبارتی فاصله همینگ کمتر شده است تا نهایتاً به تصویر آموزش که توسط شبکه به خاطر سپرده بودیم، رسیدیم.

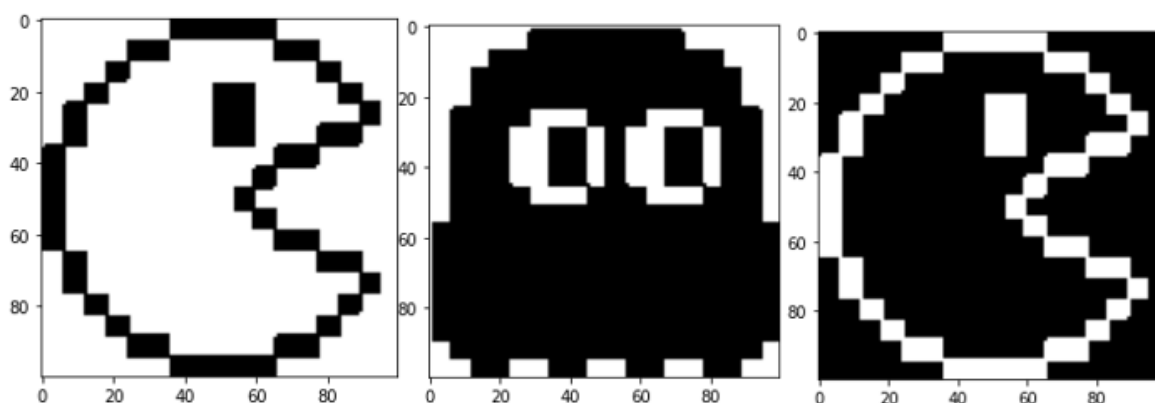
(۴) مراحل ۳ را با تصویر ورودی زیر تکرار کنید. مشاهدات خود را توضیح دهید و علت را بیان کنید.



#### پاسخ بخش 4

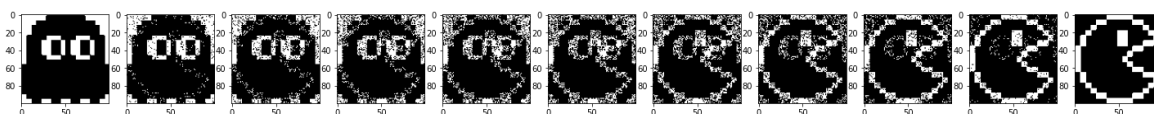
طبق خواسته‌ی بخش چهارم ورودی تستی مطابق با شکل فوق را به شبکه دادیم تا عملیات prediction انجام شود.

ابتدا ورودی و خروجی مطلوب و خروجی شبکه را برای ورودی تصویر تست جدید را بررسی می‌کنیم.



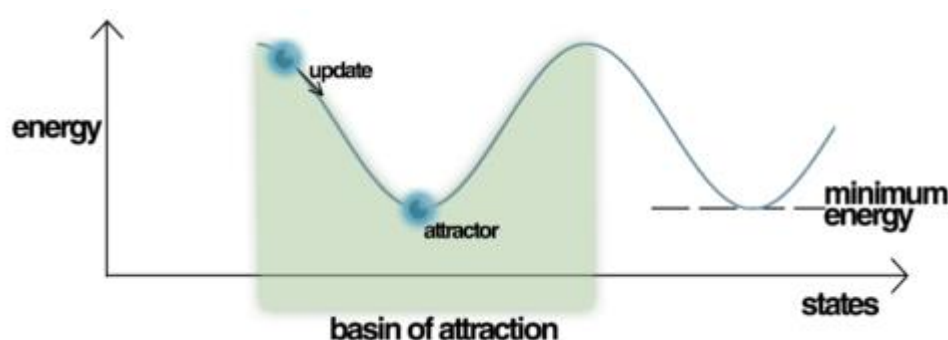
شکل 71 تصویر مطلوب (سمت چپ) و تصویر تست (وسط) و خروجی شبکه (سمت راست)

همانطور که مشاهده می‌کنید پس از 10000 iteration یا به عبارتی یک دور کامل آموزش خروجی‌ها (مساوی با بُعد خروجی یا المان‌ها تصویر ورودی) به قرینه‌ی بیتی تصویر به خاطر سپاری شده رسیدیم. حال طبق صورت سوال برای هر 1000 iteration اقدام به رسم تصویر می‌کنیم. نتیجه به صورت زیر شده می‌باشد:



شکل 72 مراحل بازسازی تصویر به خاطر سپرده شده توسط شبکه برای ورودی تست قسمت چهارم در 10000 گام

همانطور که مشاهده می‌فرمایید قرینه‌ی بیتی (بجای یک منفی یک و بالعکس) تصویر به خاطر سپرده شده، برای ورودی ghost توسط شبکه پیش‌بینی شده است. دلیل این امر را با توجه به مشاهداتی که در سوال دوم تمرین داشتیم توضیح می‌دهیم. در سوال دوم مشاهده کردیم که اعمال نویز به ورودی پس از اینکه حدود نصفی از بیت‌ها را تحت تاثیر قرار داد باعث شد که میانگین صحت با شیب نسبتاً تندی کاهش یابد و به عبارتی در این حالات، تصاویر تست ورودی در ناحیه جذب تصویر ذخیره شده قرار ندارد و نمی‌تواند به آنها میل کند. در این بخش نیز همین اتفاق رخ داده است. در واقع تصویر تست این بخش (ghost.jpg) دقیقاً مانند همان تصاویر نویزی است که با مشاهده چشمی نیز کاملاً مشهود است که نویزی حدوداً یا حتی بالغ بر 50 درصد نسبت به تصویر به حافظه سپرده شده دارد. لذا این عامل باعث شده که در ناحیه جذب مورد انتظار ما قرار نگیرد و طبیعتاً به نقطه بهینه‌ی 10000 بُعدی مطلوب ما همگرا نشود که نتایج هم این موضوع را نشان داده و نقطه‌ی همگرایی ظاهراً تصویری با قرینه‌ی بیتی تصویر آموزش (همان تصویری که توسط شبکه به خاطر سپرده شده است) می‌باشد. مثال شهودی استدلال ما می‌تواند مشابه شکل زیر باشد.



شکل 73 نمونه‌ای از وجود **local min** های متعدد و ناحیه جذب (تابع انرژی را به عنوان مثال می‌توانیم تابعی از فاصله همینگ خروجی شبکه و ریکورد آموزش بگیریم)

با دقت در منطق موجود در شکل فوق و مطابق با خروجی شبکه می‌توانیم استنباط کنید که قرینه‌ی بیتی تصویر آموزش نیز یکی از لوکال مین‌های دیگر این شبکه است که این تصویر ورودی در ناحیه جذب آن قرار گرفته و به آن همگرا شده است.



## سوال 4 – Bidirectional Associative Memory

الف) ماتریس وزن مربوط به ۳ پترن اول (C,E,R) را بدست آورید.

پاسخ بخش الف)

طبق خواسته‌ی سوال ماتریس وزن را محاسبه می‌کنیم. نحوه محاسبه به صورت زیر است.

**Setting the Weights.** The weight matrix to store a set of input and target vectors  $s(p):t(p)$ ,  $p = 1, \dots, P$ , where

$$s(p) = (s_1(p), \dots, s_i(p), \dots, s_n(p))$$

and

$$t(p) = (t_1(p), \dots, t_j(p), \dots, t_m(p)),$$

can be determined by the Hebb rule. The formulas for the entries depend on whether the training vectors are binary or bipolar. For binary input vectors, the weight matrix  $W = \{w_{ij}\}$  is given by

$$w_{ij} = \sum_p (2s_i(p) - 1) (2t_j(p) - 1).$$

For bipolar input vectors, the weight matrix  $W = \{w_{ij}\}$  is given by

$$w_{ij} = \sum_p s_i(p)t_j(p).$$

شکل 74 نحوه محاسبه وزن‌ها برای شبکه BAM (برگرفته از جزوه دکتر کلهر-دانشگاه تهران)

در نهایت وزن‌های محاسبه شده را پرینت نمودیم و نتیجه به صورت زیر می‌باشد:

```
weights are equal to
[[-1  1  1]
 [-3 -1 -1]
 [-3 -1 -1]
 [-3 -1 -1]
 [ 3  1  1]
 [ 1  3 -1]
 [-3 -1 -1]
 [-1  1  1]
 [ 1  3 -1]
 [-3 -1 -1]
 [ 1  3 -1]
 [ 3  1  1]
 [-1  1  1]
 [-1 -3  1]
 [-3 -1 -1]]
```

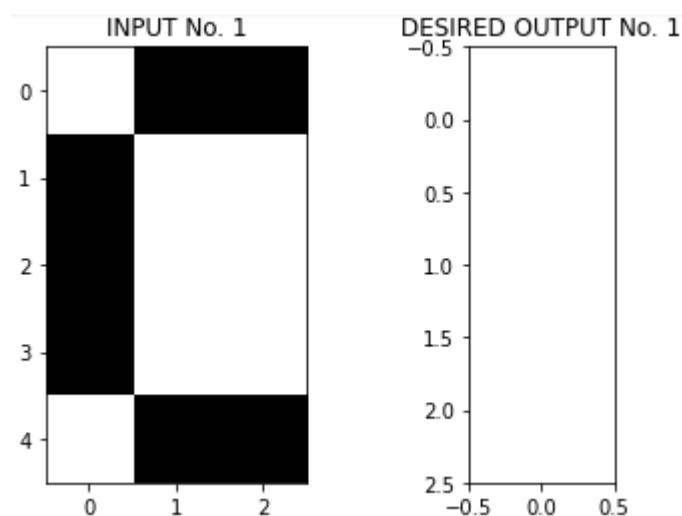
شکل 75 ماتریس وزن بدست آمده برای قسمت الف سوال 4 (شبکه BAM)

ب) توانایی شبکه در بازیابی اطلاعات از هر دو جهت را بررسی کنید و نتایج کامل را گزارش کنید.

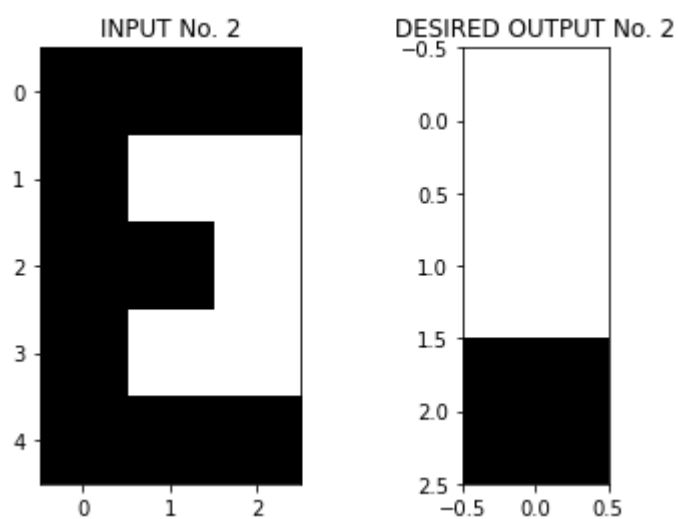
### پاسخ بخش ب)

در ابتدا ورودی و خروجی مطلوب متناظر با ورودی‌ها را رسم می‌کنیم.

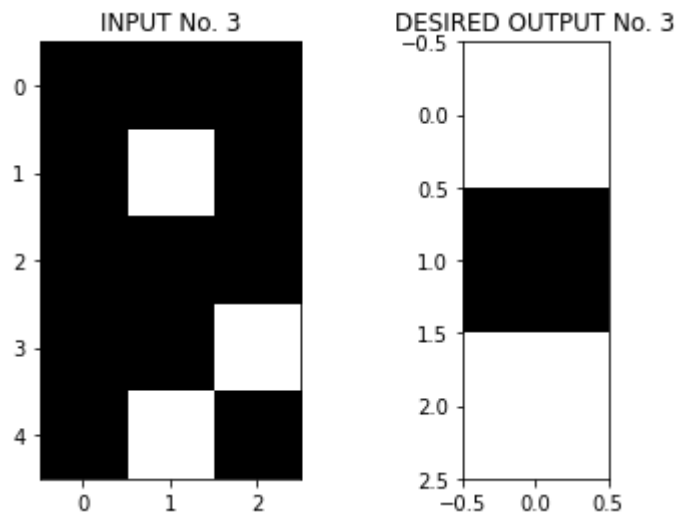
ورودی اول و خروجی متناظر آن در جهت رفت به صورت زیر است. (طبیعتاً در جهت برگشت جای این دو داده به عنوان ورودی و خروجی عکس می‌شود و در واقع ورودی رفت، خروجی برگشت است و خروجی رفت، ورودی برگشت خواهد شد.



شکل 76 ورودی اول و خروجی متناظر آن در جهت رفت شبکه forward



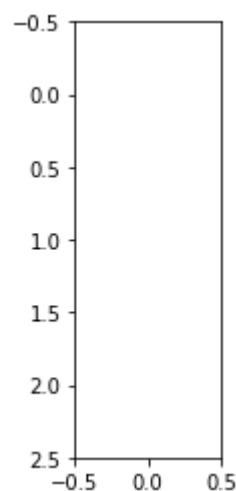
شکل 77 ورودی دوم و خروجی متناظر آن در جهت رفت شبکه forward



شکل 78 ورودی سوم و خروجی متناظر آن در جهت رفت شبکه forward

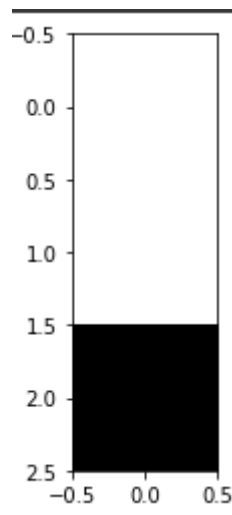
اکنون با استفاده از وزن محاسبه شده در قسمت الف اقدام به پیش‌بینی ورودی‌ها در جهت جلو می‌کنیم. طبیعتاً تصاویر سمت چپ در شکل‌های فوق ورودی و تصاویر سمت راست‌شان، خروجی‌های متناظر هر یک می‌باشد.

نتایج بدست آمده به شرح زیر است.



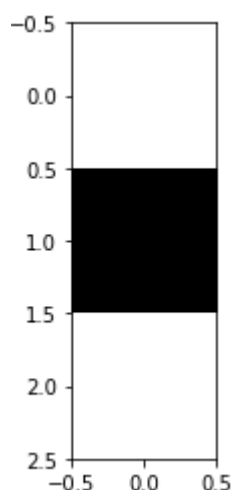
شکل 79 خروجی استخراج شده از شبکه در جهت رفت برای حرف C

همانطور که مشاهده می‌فرمایید خروجی گرفته شده از شبکه با خروجی مطلوبی که مشاهده کرده بودیم برای ورودی حرف C یکی است. حال سراغ ورودی بعدی می‌رویم.



شکل 80 خروجی استخراج شده از شبکه در جهت رفت برای حرف E

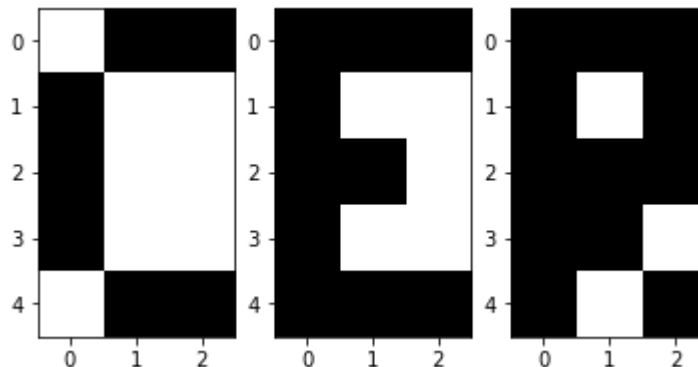
همانطور که مشاهده می‌فرمایید در اینجا نیز به درستی پیش‌بینی انجام شده است و خروجی متناظر با ورودی حرف E از شبکه استخراج شده است. (با صحبتی که میان اینجانب و سایر دانشجویان بود گویا بقیه این خروجی را اشتباه محاسبه کردند که بدلیل مشکلات deep copy در پایتون به نظرم به این مشکل خوردند. من تصحیحات لازم برای جلوگیری از این مشکل را انجام دادم و نتیجه درستی برای اینجانب حاصل شد.)



شکل 81 خروجی استخراج شده از شبکه در جهت رفت برای حرف R

برای آخرین حرف که R می‌باشد نیز شبکه به درستی پیش‌بینی کرده است.

حال جهت برگشت را بررسی می‌کنیم و جای ورودی و خروجی‌ها را جابجا و شبکه را معکوس می‌کنیم.



شکل 82 خروجی‌های گرفته شده از شبکه برای جهت رفت (به ترتیب از چپ به راست متناظر با  $(-1,-1,1)$  و  $(-1,1,-1)$  و  $(1,-1,-1)$ )

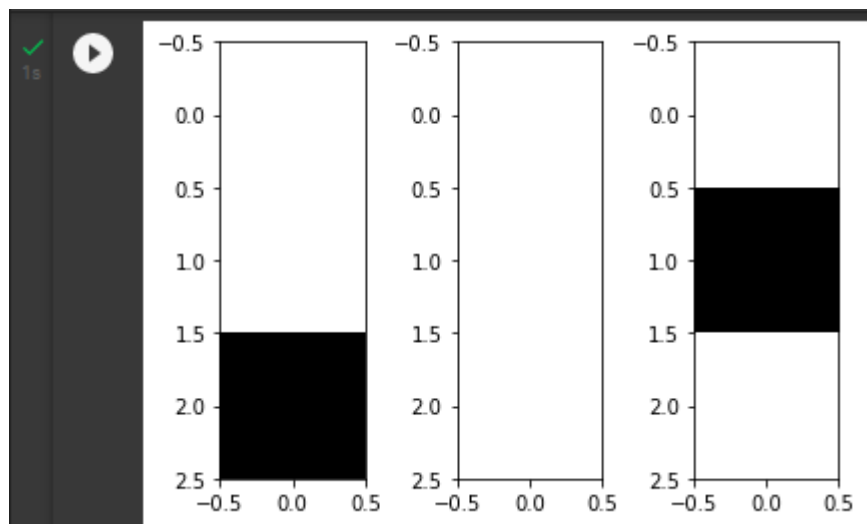
همانطور که مشاهده می‌فرمایید شبکه BAM ما توانسته به خوبی خروجی‌های متناظر با ورودی‌ها را در جهت معکوس پیش‌بینی کند.

لذا به عنوان یک گزارش نهایی از این شبکه BAM به ازای ورودی‌های وارد شده به شبکه 100% صحت برای جهت رفت و 100% صحت برای جهت برگشت داشتیم. (البته در این مرحله نویز وارد نکردیم و برای تاثیر نویز به بخش بعدی مراجعه می‌کنیم).

پ) به ورودی‌ها ۴۰ درصد نویز اعمال کنید (تبدیل کردن اعداد +۱ و -۱ به صورت تصادفی) کد را ۱۰۰ بار اجرا کرده و درصد خروجی درست شبکه را بیان نمایید.

#### پاسخ بخش پ)

در ابتدا نتایج را برای یک بار اجرا بررسی می‌کنیم. اینگونه با 40% خطا احتمال می‌دهم که خطاهایی در نتایج مشهود باشد. لذا برای یک بار اجرا که بررسی کردیم نتایج به شرح زیر است.



شکل 83 خروجی‌های متناظر با اعمال خطای 40 درصد به هر یک از ورودی‌ها در جهت رفت (از چپ به راست به ازای ورودی‌های حرف C و E و R)

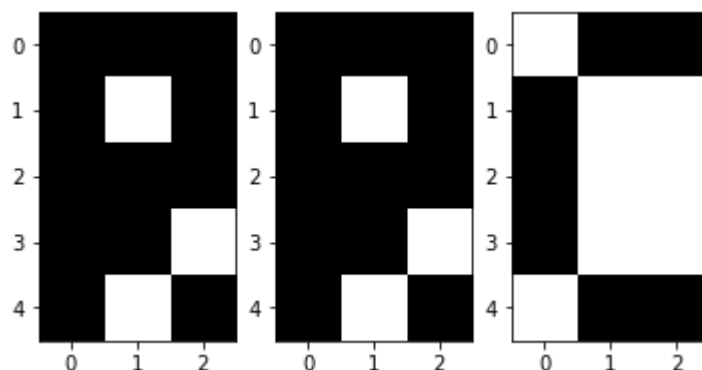
سه خروجی بدست آمده مطابق با تصاویر فوق است و همانطور که مشاهده می‌فرمایید، خروجی ورودی حرف E و نیز خروجی متناظر حرف C اشتباه تشخیص داده شده است که مربوط به همان اعمال نویز 40 درصد در ورودی‌هاست ولی خروجی مطلوب متناظر با حرف R به درستی پیش‌بینی شده است. حال طبق خواسته‌ی سوال این فرایند را برای 100 بار تکرار انجام می‌دهیم. نهایتاً نتایج بدست آمده به صورت زیر می‌باشند.

```
mean prediction accuracy of 1st input(C) [41.]
mean prediction accuracy of 2nd input(E) [22.]
mean prediction accuracy of 3rd input(R) [11.]
mean prediction accuracy of all inputs 24.666666666666668
```

شکل 84 صحت بدست آمده از خروجی‌های شبکه با اعمال نویز 40 درصد به ورودی‌ها در جهت رفت

همانطور که مشاهده می‌فرمایید نویز به طور متوسط توانسته است صحت را به طور میانگین تا 24% کاهش دهد. ما برای اطمینان از نتیجه‌ی فوق تصمیم گرفتیم که نویز را 20 درصد اعمال کنیم که ببینیم صحت بالاتر می‌رود یا خیر و مشاهده کردیم که برای نویز 20% صحت تا 80% نیز می‌رسد. لذا به نظر می‌رسد شبکه طراحی شده طبق خواسته‌ی صورت سوال، بسیار به نویز حساس بوده و عملاً نویز 40% باعث کاهش بسیار زیاد در میانگین صحت پیش‌بینی‌ها می‌شود.

حال شبکه برگشت را مورد بررسی قرار می‌دهیم. روندی که برای جهت رفت پیش گرفتیم را برای جهت برگشت تکرار می‌کنیم و نتایج زیر را بدست می‌آوریم.



شکل 85 خروجی‌های متناظر با اعمال خطای 40 درصد به هر یک از ورودی‌ها در جهت برگشت

همانطور که مشاهده می‌فرمایید، هر سه پیش‌بینی اشتباه هستند چون ما از تصویر سمت چپ حرف C، وسط حرف E و سمت راست حرف R را انتظار داشتیم که همگی را اشتباه تشخیص داده است. دقت کنید که نویز 40٪ در جهت برگشت یعنی تغییر یک بیت از سه بیت ورودی (این ورودی‌ها همان خروجی‌های مطلوب جهت رفت هستند که نویزی شده‌اند).

در نهایت برای این شبکه در جهت برگشت نیز 100 بار مرحله‌ی اعمال نویز به ورودی و استخراج خروجی را انجام دادیم که نهایتاً نتایج زیر برای مسیر برگشت حاصل شد.

```

mean prediction accuracy of 1st input([-1, -1, -1]) [0.]
mean prediction accuracy of 2nd input([-1, -1, 1]) [0.]
mean prediction accuracy of 3rd input([-1, 1, -1]) [45.]
mean prediction accuracy of all inputs 15.0

```

شکل 86 صحت بدست آمده از خروجی‌های شبکه با اعمال نویز 40 درصد به ورودی‌ها در جهت برگشت

همانطور که مشاهده می‌فرمایید در جهت برگشت نیز پیش‌بینی‌ها مطلوب نبوده و نویز تاثیر بسیار زیادی در کاهش صحت شبکه در مسیر برگشت نیز داشته است.

ت) حداکثر تعداد پترن که می توان در این شبکه ذخیره کرد چه تعداد است؟

پاسخ بخش ت)

این بخش را با توجه به اسلاید 30م استاد پاسخ می دهیم. در ابتدا باید عرض کنیم که چون این روش در واقع بر پایه یادگیری و به حافظه سپاری روش hebbian rule می باشد. لذا طبق این اسلاید استاد، در صورتیکه  $n$  بعد ورودی داشته باشیم، حداکثر  $n$  تصویر (الگو) را می توانیم به خاطر بسپاریم. لذا با توجه به آنکه BAM به صورت دو جهت عمل می کند و ما عملاً یک طرف رکوردهای 15 بُعدی و یک طرف رکوردهای 3 بُعدی داریم، به همین جهت تعداد بُد کمتر تعیین کننده شده و طبیعتاً ما حداکثر به تعداد 3 پترن را با این روش می توانیم در شبکه ذخیره کنیم.

ث) تمام پترن ها را در نظر بگیرید و شبکه را پیاده سازی کنید. کدام پترن ها احتمال خطای بیشتری دارند ؟ با دلیل نشان دهید

پاسخ بخش ث)

طبق خواسته ی مسئله اکنون تمامی پترن ها را وارد کرده و شبکه را مجدد پیاده و تمامی مراحل را تکرار کردیم.

در ابتدا ماتریس وزن را محاسبه نمودیم که مطابق با ماتریس زیر است.

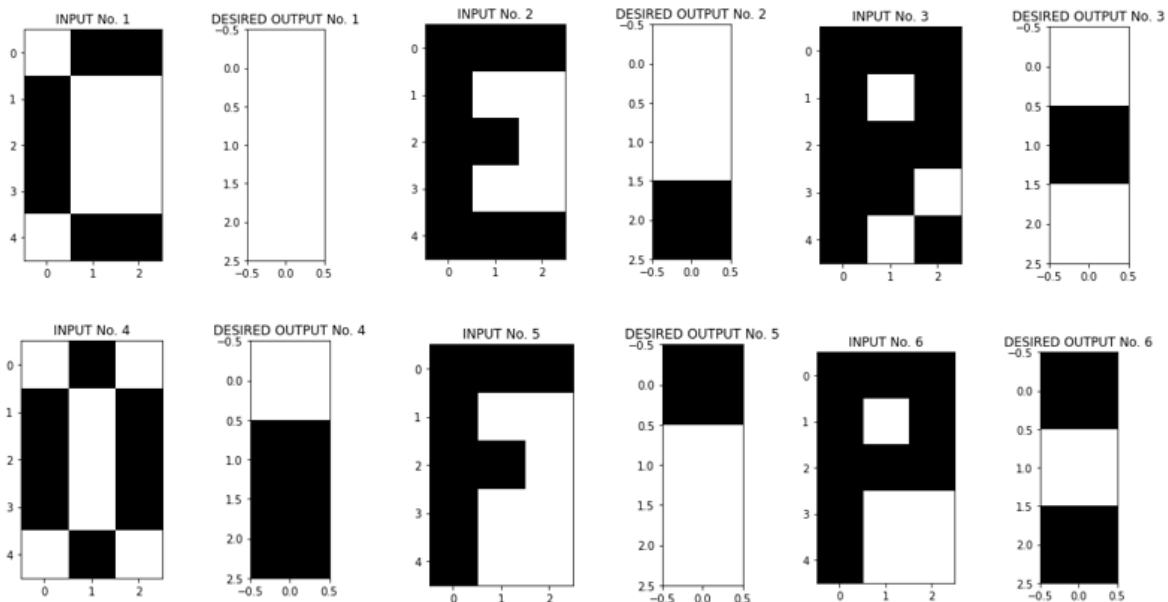
```
weights are equal to
[ [ 2 -2  0]
  [-2 -2  0]
  [ 0 -4 -2]
  [-2 -2  0]
  [ 2  2  0]
  [ 0  4  2]
  [-2 -2  0]
  [ 2 -2  0]
  [ 0  4  2]
  [-2 -2  0]
  [ 0  4 -2]
  [ 0  4  2]
  [ 2 -2  0]
  [-4  0  2]
  [-4  0 -2] ]
```

شکل 87 ماتریس وزن بدست آمده برای قسمت ث سوال 4 (شبکه BAM)



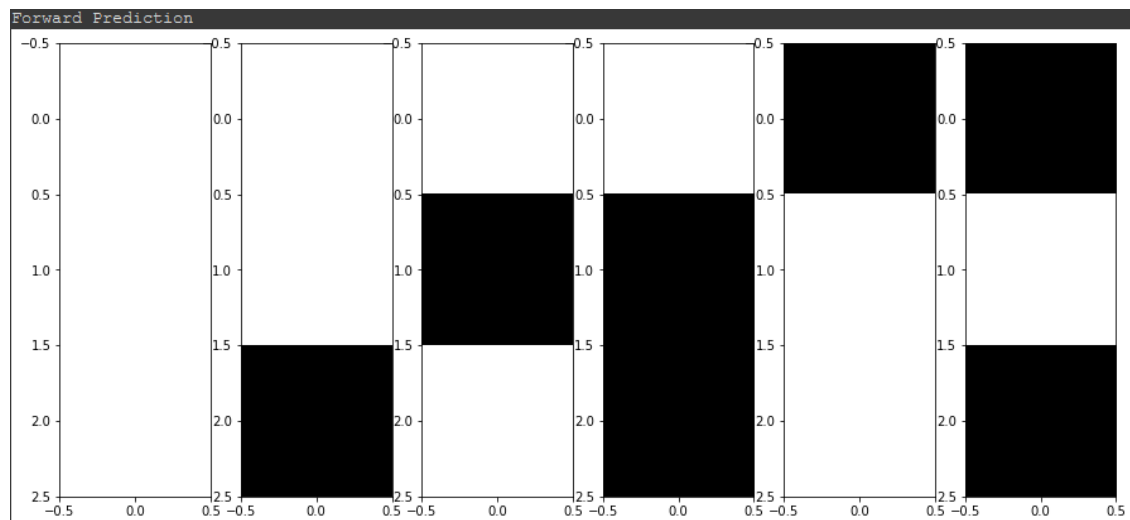
سپس اقدام به پیش‌بینی در هر یک از جهت‌های رفت و برگشت نمودیم که نتایج این قسمت نیز به صورت زیر است.

ابتدا ورودی‌ها و خروجی متناظرشان در جهت رفت را نشان می‌دهیم و سپس نتیجه پیش‌بینی را بررسی می‌کنیم.



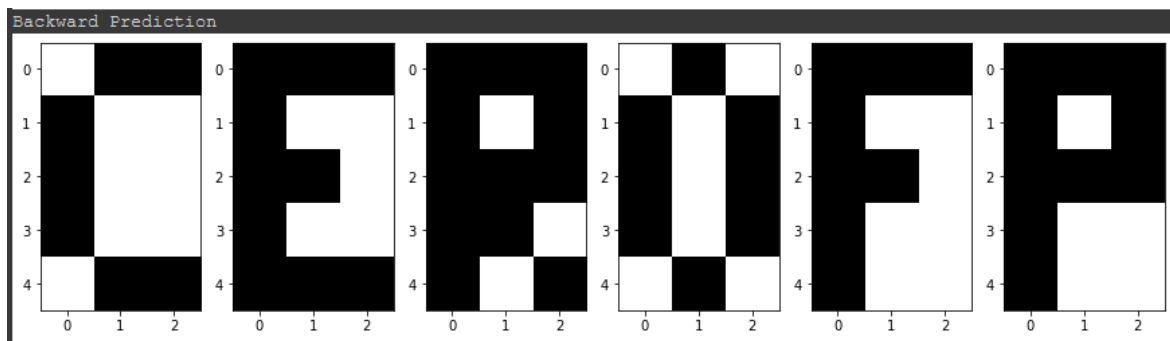
شکل 88 ورودی و خروجی‌های مطلوب متناظر در شبکه BAM و جهت رفت

حال نتایج پیش‌بینی شبکه در مسیر رفت را بررسی می‌کنیم. ترتیب خروجی‌ها از چپ به راست متناظر با ورودی اول تا ششم است.



شکل 89 خروجی‌های پیش‌بینی شده از شبکه BAM در مسیر رفت که کاملاً منطبق بر خروجی‌های مطلوب است

عملکرد شبکه در مسیر رفت عالی است. حال عملکرد را در جهت برگشت هم مشاهده می‌کنیمو نتایج پیش‌بینی شبکه به صورت زیر است.



شکل 90 خروجی‌های پیش‌بینی شده از شبکه BAM در مسیر برگشت که کاملاً منطبق بر خروجی‌های مطلوب است

پس تا اینجا مشاهده می‌فرمایید که پیش‌بینی‌ها بدرستی صورت گرفته است و عملکرد شبکه در دو جهت مطلوب و در پیش‌بینی ورودی‌ها از کارایی خوبی برخوردار است.

در باب اینکه کدام الگوها احتمال خطای بیشتری دارند باید ذکر کنم که به نظر می‌رسد الگوهایی که فاصله همینگ کمتری نسبت به یکدیگر دارند، احتمال خطای بیشتری را به دنبال خواهند داشت. لذا حدس اولیه این است که P و R که بیشترین شباهت و کمترین فاصله همینگ را دارند باید احتمال خطای بیشتری در تشخیص آنها رخ دهد. به همین جهت روند زیر را برای ارائه یک استدلال پی می‌گیریم.

حال نتایج صحت را برای 100 ایتريشن بررسی می‌کنیم تا ببینیم کدام ورودی‌ها به خوبی تشخیص داده شده و کدام ورودی‌ها، شبکه در تشخیصشان دچار مشکل و خطای بیشتری شده است. فلذا یک نویز 40 درصد وارد ورودی‌ها می‌کنیم و نتایج را بررسی می‌نماییم.

```

0s ▶ mean prediction accuracy of 1st input(C) [21.]
      mean prediction accuracy of 2nd input(E) [23.]
      mean prediction accuracy of 3rd input(R) [8.]
      mean prediction accuracy of 4th input(O) [22.]
      mean prediction accuracy of 5th input(F) [20.]
      mean prediction accuracy of 6th input(P) [14.]
      mean prediction accuracy of all inputs 18.0

```

شکل 91 مقایسه بیشترین احتمال خطا برای تشخیص ورودی‌ها با تکنیک اعمال نویز و بررسی صحت شبکه در تشخیص (نویز ورودی: 40٪)

همانطور که پیش‌بینی کرده بودیم،  $P$  و  $R$  کمترین صحت را در تشخیص‌شان مشاهده می‌کنیم. طبیعتاً این بواسطه‌ی همان شباهتی است که ذکر شد و بدلیل این شباهت بسیار بالا، شبکه دچار خطا شده و نمی‌تواند خروجی مناسب را تشخیص دهد. پس از این دو ورودی،  $F$  و  $C$  نیز دارای شباهت بالایی هستند که نتیجتاً شبکه برای این ورودی‌ها هم دچار خطا شده است. لازم به ذکر است که ما خروجی برگشت را هم بررسی کردیم لکن با اعمال یک نویز 30٪ یا 40٪ و در واقع تغییر یک بیت در ورودی شبکه برگشت، تشخیص دچار مشکل شده و نتایج مطلوبی بدست نمی‌آید. لذا به عنوان یک نتیجه‌گیری نهایی برای این سوال آخر باید بگوییم که هراندازه شباهت بین داده‌گان بالاتر باشد، عملاً بُدهای کمتری از ریکوردها دارای اهمیت شده و کوچکترین نویز باعث کاهش کارایی شبکه در تشخیص می‌شود و اصطلاحاً حساسیت شبکه به نویز بسیار بالا رفته و robustness خوبی از خود نشان نخواهد داد.

با تشکر

بدیعی