



به نام خدا



دانشگاه تهران
دانشکده مهندسی برق و کامپیوتر
درس شبکه‌های عصبی و یادگیری عمیق

تمرین سری ۳

سیاوش شمس	نام و نام خانوادگی
810197644	شماره دانشجویی
۰۱/۰۲/۳۱	تاریخ ارسال گزارش

فهرست گزارش سوالات

سوال ۱ – Pattern Association using Hebbian Learning Rule ۳

(۱) ۳

(۲) ۴

(۳) ۵

(۴) ۶

(۵) ۸

(۶) ۱۲

(۷) ۱۵

سوال ۲ – Auto-associative Net ۱۶

(۱) ۱۶

(۲) ۱۷

(۳) ۱۸

(الف) ۱۸

(ب) ۲۱

(۴) ۲۱

(۵) ۲۴

سوال ۳ – Discrete Hopfield Network ۲۸

(۱) ۲۸

(۲) ۲۹

(۳) ۳۰

(۴) ۳۰

(۵) ۳۱

سوال ۴ – Bidirectional Associative Memory ۳۳

۳۳..... (۱)

۳۴..... (۲)

۳۴..... (۳)

۳۴..... (۴)

سوال ۱ – Pattern Association using Hebbian Learning Rule

(۱)

در یک شبکه خروجی های هدف به صورت زیر به دست می آیند

$$t(p) = f(s(p)^T W)$$

که W همان بردار وزن های ما می باشد، هدف ما تعیین W به طوری است که شبکه ورودی های $s(p)$ را به خروجی متناظر نگاشت کند.

ماتریس وزن ها توسط الگوریتم زیر که به قاعده هب معروف است به دست می آید:

۱. دادن مقدار اولیه صفر به تمام وزن ها

۲. برای زوج ورودی و خروجی s, t مراحل زیر را انجام می دهیم

۳. $x_i = s_i$

۴. $y_j = t_j$

۵. $w_{i,j}^{new} = w_{i,j}^{old} + x_i y_j$

البته الگوریتم فوق به صورت ساده تر با ضرب دو ماتریس ورودی و خروجی به صورت زیر خلاصه می شود:

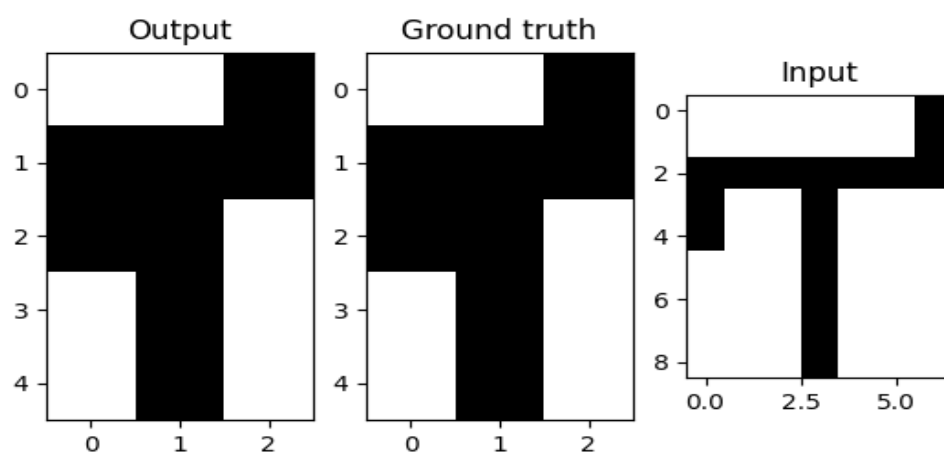
$$W = \sum_{p=1}^P s(p)t(p)^T$$

جدول ۱-۱ ماتریس وزن های به دست آمده به کمک قاعده هب را نشان می دهد.

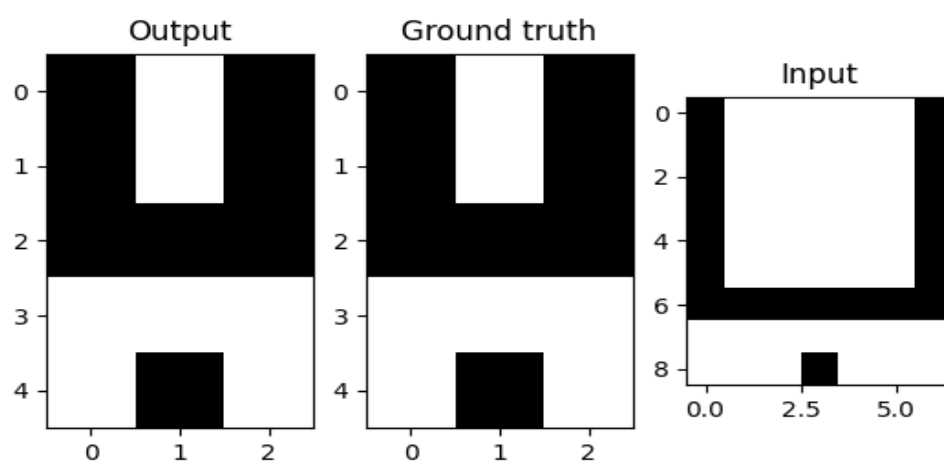
جدول ۱-۱- وزن های به دست آمده به کمک قاعده هب

3	-1	1	1	-3	1	1	1	3	1	-3	1	1	1	1
-1	3	-3	-3	1	-3	-3	-3	-1	1	1	1	1	-3	1
-1	3	-3	-3	1	-3	-3	-3	-1	1	1	1	1	-3	1
-1	3	-3	-3	1	-3	-3	-3	-1	1	1	1	1	-3	1
-1	3	-3	-3	1	-3	-3	-3	-1	1	1	1	1	-3	1
1	-3	3	3	-1	3	3	3	1	-1	-1	-1	-1	3	-1
3	-1	1	1	-3	1	1	1	3	1	-3	1	1	1	1
-1	3	-3	-3	1	-3	-3	-3	-1	1	1	1	1	-3	1
-1	3	-3	-3	1	-3	-3	-3	-1	1	1	1	1	-3	1
-1	3	-3	-3	1	-3	-3	-3	-1	1	1	1	1	-3	1
-1	3	-3	-3	1	-3	-3	-3	-1	1	1	1	1	-3	1
-1	3	-3	-3	1	-3	-3	-3	-1	1	1	1	1	-3	1
1	-3	3	3	-1	3	3	3	1	-1	-1	-1	-1	3	-1
1	-3	3	3	-1	3	3	3	1	-1	-1	-1	-1	3	-1
-3	1	-1	-1	3	-1	-1	-1	-3	-1	3	-1	-1	-1	-1
-3	1	-1	-1	3	-1	-1	-1	-3	-1	3	-1	-1	-1	-1
-3	1	-1	-1	3	-1	-1	-1	-3	-1	3	-1	-1	-1	-1
-3	1	-1	-1	3	-1	-1	-1	-3	-1	3	-1	-1	-1	-1
1	-3	3	3	-1	3	3	3	1	-1	-1	-1	-1	3	-1
1	-3	3	3	-1	3	3	3	1	-1	-1	-1	-1	3	-1
-1	3	-3	-3	1	-3	-3	-3	-1	1	1	1	1	-3	1
-1	3	-3	-3	1	-3	-3	-3	-1	1	1	1	1	-3	1
-3	1	-1	-1	3	-1	-1	-1	-3	-1	3	-1	-1	-1	-1
-3	1	-1	-1	3	-1	-1	-1	-3	-1	3	-1	-1	-1	-1
-3	1	-1	-1	3	-1	-1	-1	-3	-1	3	-1	-1	-1	-1
1	-3	3	3	-1	3	3	3	1	-1	-1	-1	-1	3	-1
1	-3	3	3	-1	3	3	3	1	-1	-1	-1	-1	3	-1
-1	3	-3	-3	1	-3	-3	-3	-1	1	1	1	1	-3	1
-1	3	-3	-3	1	-3	-3	-3	-1	1	1	1	1	-3	1
-3	1	-1	-1	3	-1	-1	-1	-3	-1	3	-1	-1	-1	-1
-1	3	-3	-3	1	-3	-3	-3	-1	1	1	1	1	-3	1
-1	3	-3	-3	1	-3	-3	-3	-1	1	1	1	1	-3	1
3	-1	1	1	-3	1	1	1	3	1	-3	1	1	1	1
1	-3	3	3	-1	3	3	3	1	-1	-1	-1	-1	3	-1
-1	3	-3	-3	1	-3	-3	-3	-1	1	1	1	1	-3	1
-1	3	-3	-3	1	-3	-3	-3	-1	1	1	1	1	-3	1
-1	-1	1	1	1	1	1	1	-1	1	1	1	1	1	1
-1	3	-3	-3	1	-3	-3	-3	-1	1	1	1	1	-3	1
-1	3	-3	-3	1	-3	-3	-3	-1	1	1	1	1	-3	1
3	-1	1	1	-3	1	1	1	3	1	-3	1	1	1	1
3	-1	1	1	-3	1	1	1	3	1	-3	1	1	1	1
-1	3	-3	-3	1	-3	-3	-3	-1	1	1	1	1	-3	1
-1	3	-3	-3	1	-3	-3	-3	-1	1	1	1	1	-3	1
-3	1	-1	-1	3	-1	-1	-1	-3	-1	3	-1	-1	-1	-1
-1	3	-3	-3	1	-3	-3	-3	-1	1	1	1	1	-3	1
-1	3	-3	-3	1	-3	-3	-3	-1	1	1	1	1	-3	1
3	-1	1	1	-3	1	1	1	3	1	-3	1	1	1	1
3	-1	1	1	-3	1	1	1	3	1	-3	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	-1	-1	-1
1	1	-1	-1	-1	-1	-1	-1	1	-1	-1	-1	-1	-1	-1
3	-1	1	1	-3	1	1	1	3	1	-3	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	-1	-1	-1
3	-1	1	1	-3	1	1	1	3	1	-3	1	1	1	1
-1	3	-3	-3	1	-3	-3	-3	-1	1	1	1	1	-3	1
-1	3	-3	-3	1	-3	-3	-3	-1	1	1	1	1	-3	1
-3	1	-1	-1	3	-1	-1	-1	-3	-1	3	-1	-1	-1	-1
-1	3	-3	-3	1	-3	-3	-3	-1	1	1	1	1	-3	1
-1	3	-3	-3	1	-3	-3	-3	-1	1	1	1	1	-3	1
1	1	-1	-1	-1	-1	-1	-1	1	3	-1	3	3	-1	3
1	1	-1	-1	-1	-1	-1	-1	1	3	-1	3	3	-1	3
1	1	-1	-1	-1	-1	-1	-1	1	3	-1	3	3	-1	3
1	-3	3	3	-1	3	3	3	1	-1	-1	-1	-1	3	-1
1	1	-1	-1	-1	-1	-1	-1	1	3	-1	3	3	-1	3
1	1	-1	-1	-1	-1	-1	-1	1	3	-1	3	3	-1	3
1	1	-1	-1	-1	-1	-1	-1	1	3	-1	3	3	-1	3
1	1	-1	-1	-1	-1	-1	-1	1	3	-1	3	3	-1	3

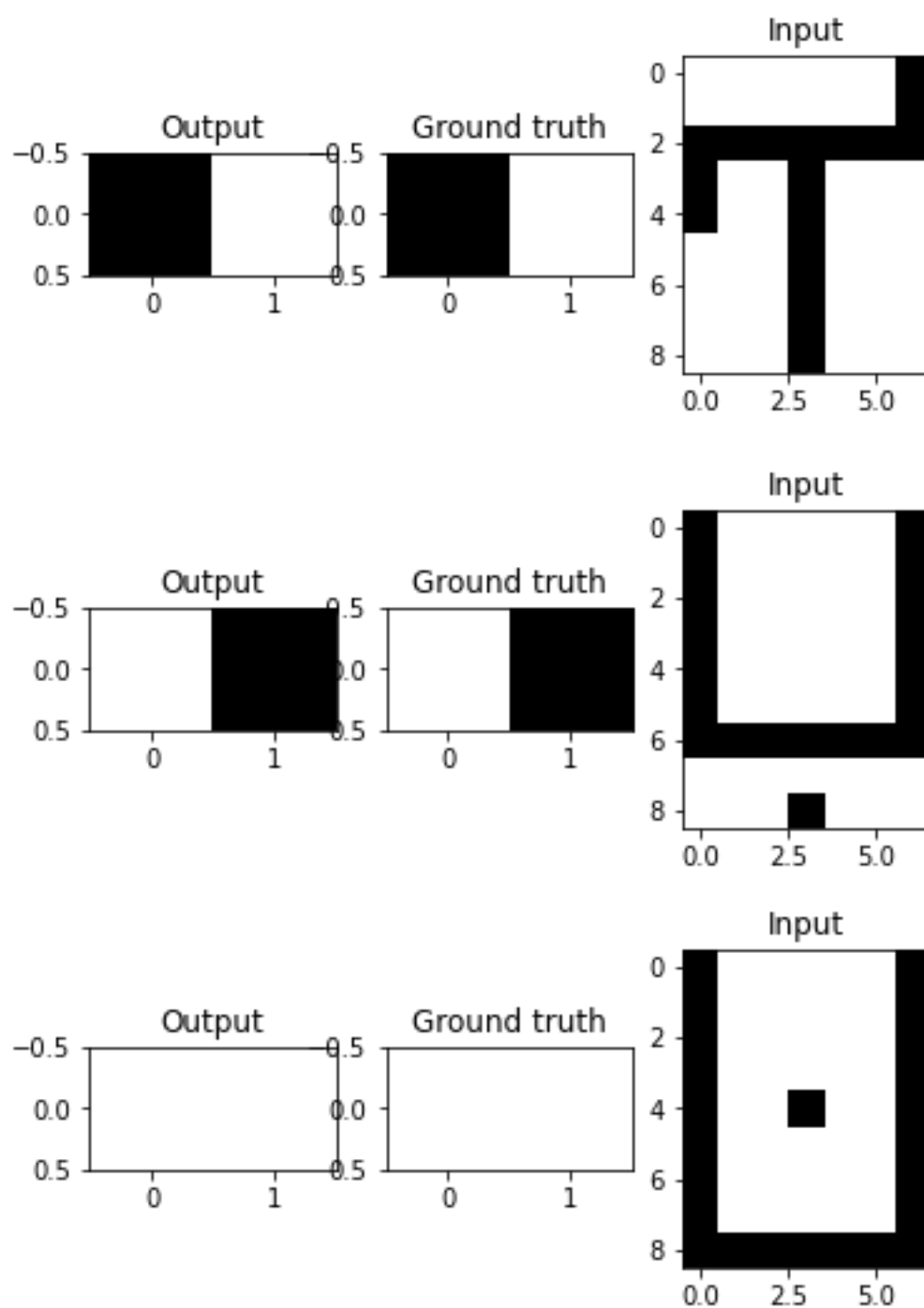
بله شبکه قابلیت تولید خروجی را دارد، شکل اول از راست ورودی شبکه، شکل وسط خروجی مد نظر ما و شکل چپ خروجی شبکه ما می باشد



شکل ۱-۱- ورودی، خروجی مد نظر، و خروجی شبکه از راست به چپ

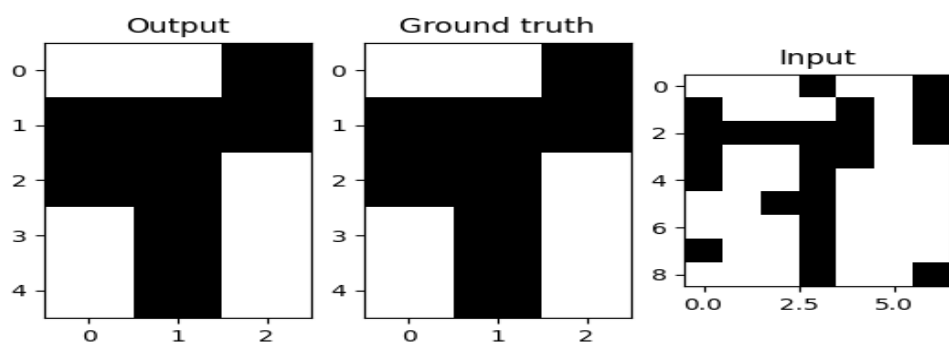


شکل ۲-۱- ورودی، خروجی مد نظر، و خروجی شبکه از راست به چپ

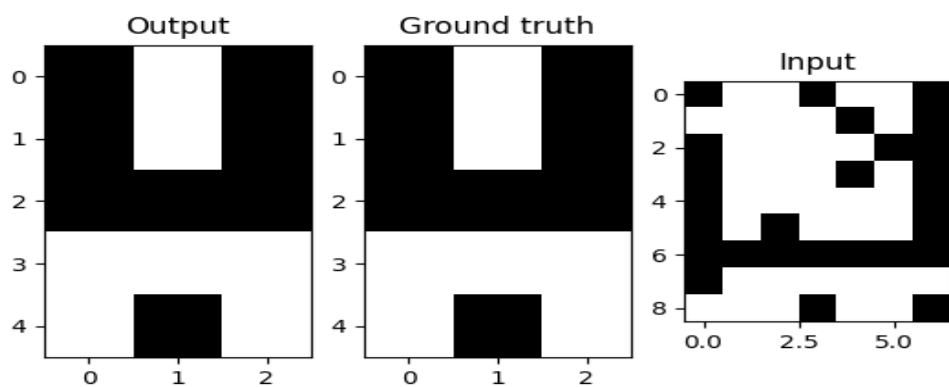


شکل ۱-۴- ورودی، خروجی مد نظر، و خروجی شبکه از راست به چپ برای خروجی با سایز ۱*۲

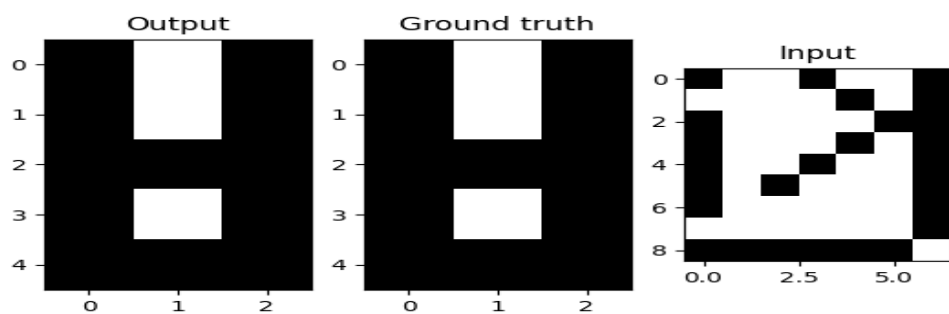
With 20% noise



With 20% noise

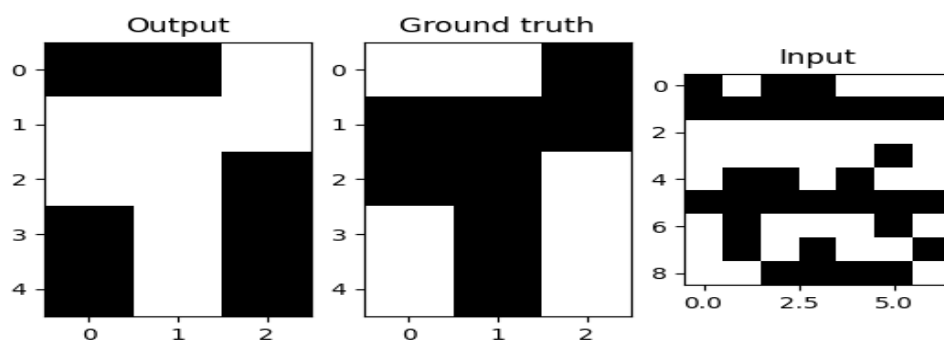


With 20% noise

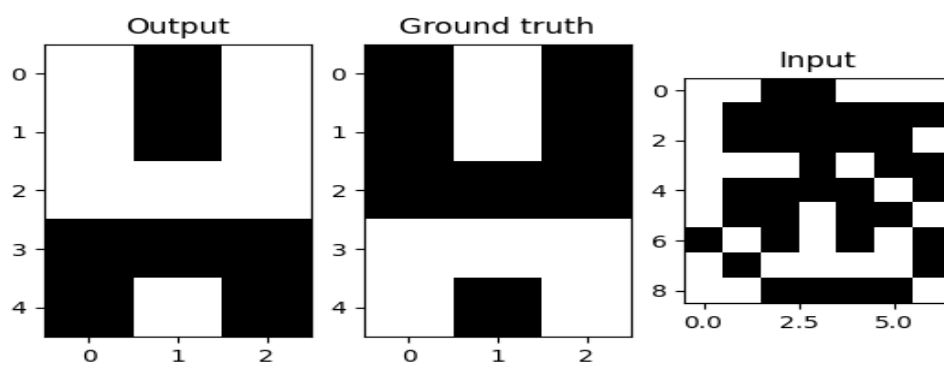


شکل ۱-۵- ورودی با نویز ۲۰٪، خروجی مطلوب، و خروجی شبکه از راست به چپ

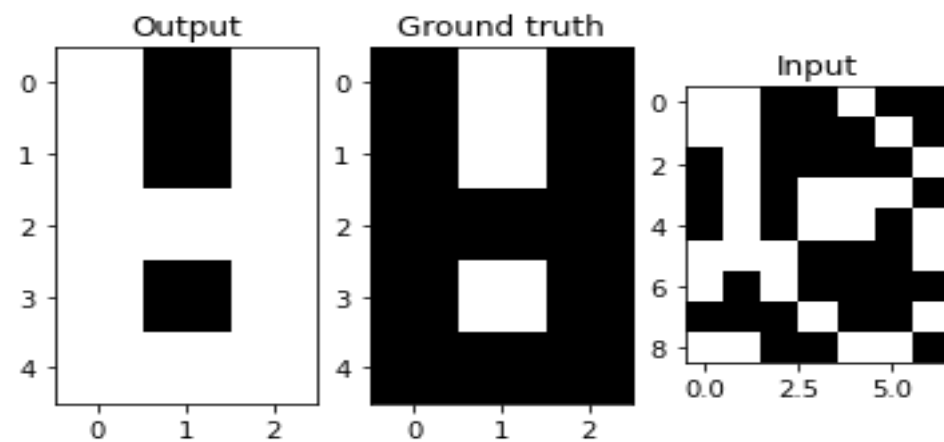
With 60% noise



With 60% noise

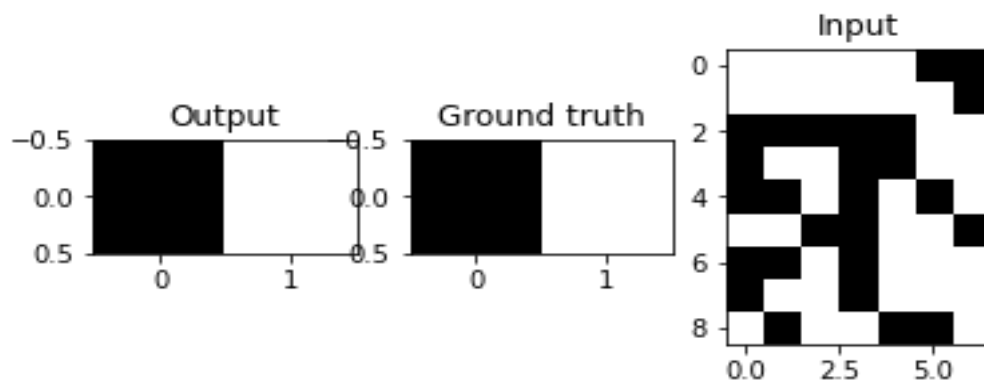


With 60% noise

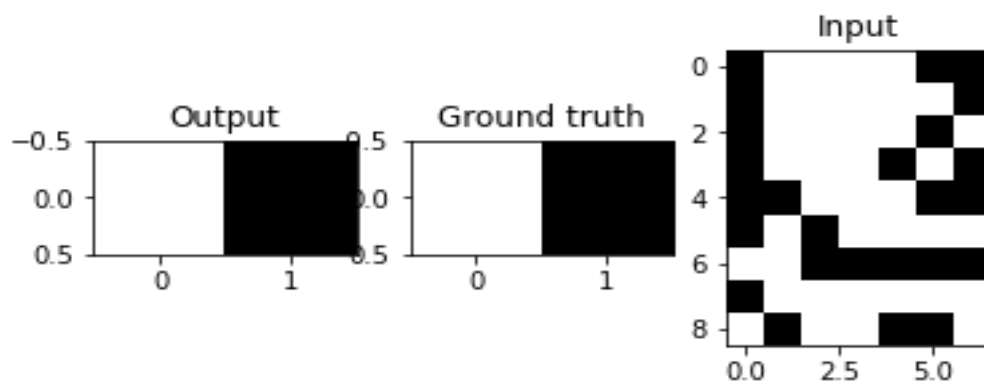


شکل ۱-۶- ورودی با نویز ۶۰٪، خروجی مطلوب، و خروجی شبکه از راست به چپ

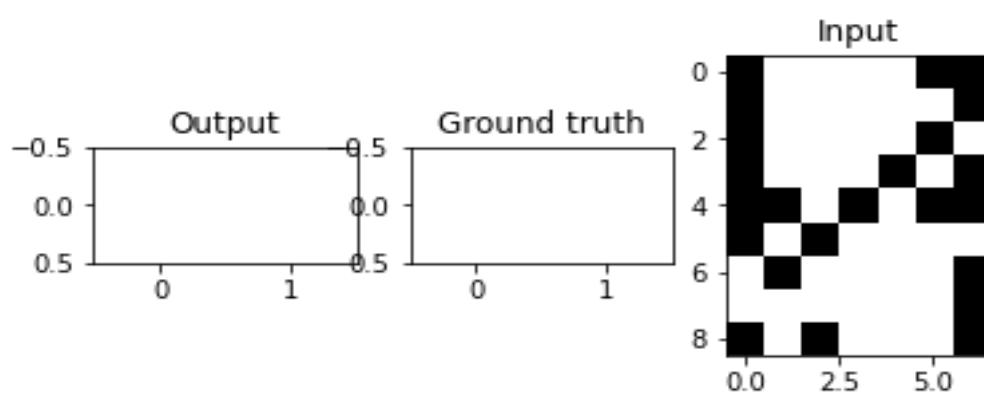
With 20% noise



With 20% noise

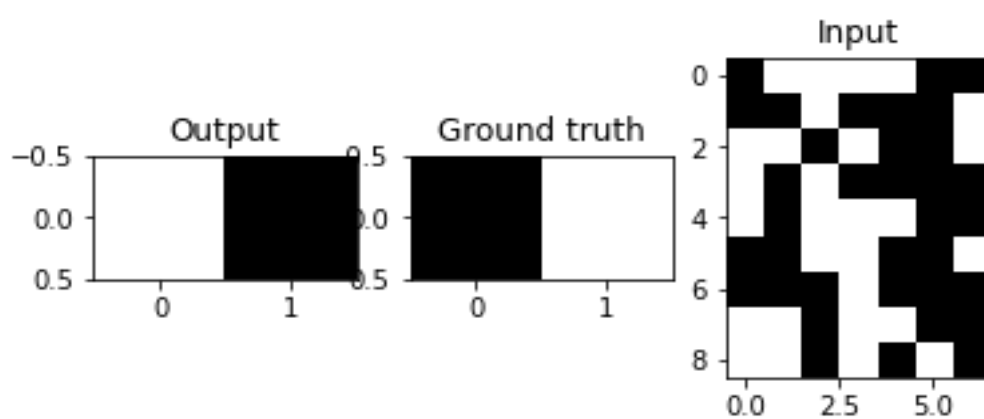


With 20% noise

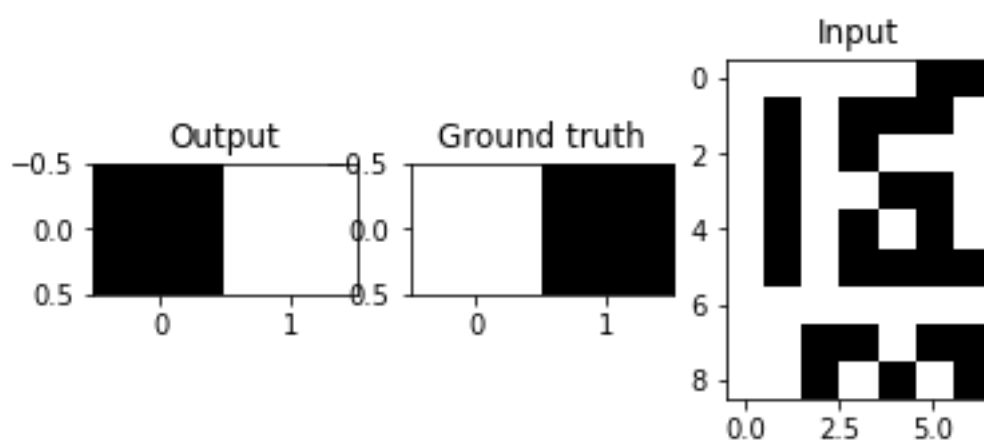


شکل ۱-۶- ورودی با نویز ۲۰٪، خروجی مطلوب، و خروجی شبکه از راست به چپ برای شبکه با سایز خروجی ۲*۱

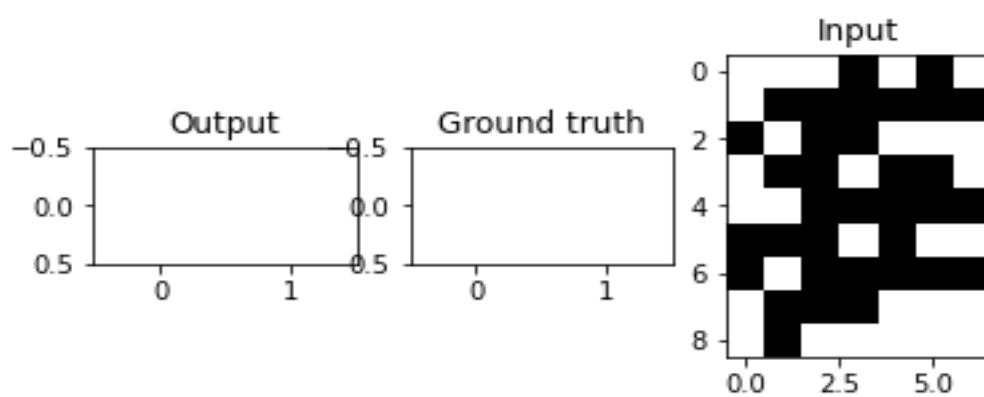
With 60% noise



With 60% noise



With 60% noise



شکل ۱-۷- ورودی با نویز ۶۰٪، خروجی مطلوب، و خروجی شبکه از راست به چپ برای شبکه با سایز خروجی ۲*۱

```

Our model gives the correct output(5*3) for input1 with 20% noise with accuracy = 95 %
Our model gives the correct output(5*3) for input2 with 20% noise with accuracy = 100 %
Our model gives the correct output(5*3) for input3 with 20% noise with accuracy = 87 %
Our model gives the correct output(5*3) for input1 with 60% noise with accuracy = 1 %
Our model gives the correct output(5*3) for input2 with 60% noise with accuracy = 2 %
Our model gives the correct output(5*3) for input3 with 60% noise with accuracy = 1 %

```

شکل ۸-۱- دقت خروجی با ابعاد ۵*۳ شبکه به ازای مقادیر مختلف نویز برای ورودی ها

```

Our model gives the correct output(1*2) for input1 with 20% noise with accuracy = 93 %
Our model gives the correct output(1*2) for input2 with 20% noise with accuracy = 72 %
Our model gives the correct output(1*2) for input3 with 20% noise with accuracy = 100 %
Our model gives the correct output(1*2) for input1 with 60% noise with accuracy = 7 %
Our model gives the correct output(1*2) for input2 with 60% noise with accuracy = 12 %
Our model gives the correct output(1*2) for input3 with 60% noise with accuracy = 1 %

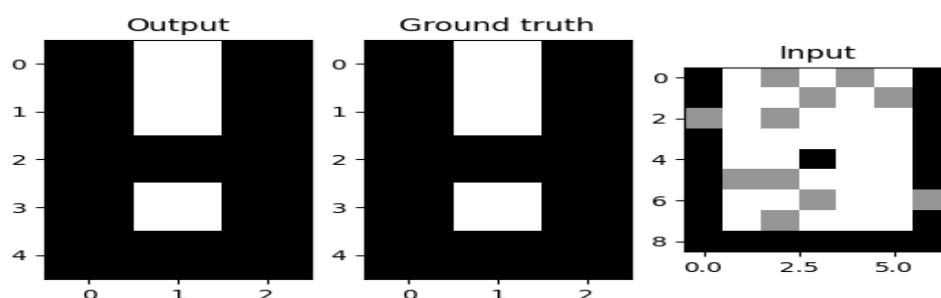
```

شکل ۹-۱- دقت خروجی با ابعاد ۱*۲ شبکه به ازای مقادیر مختلف نویز برای ورودی ها

(۶

خانه های خاکستری، اطلاعات از دست رفته را نشان می دهند

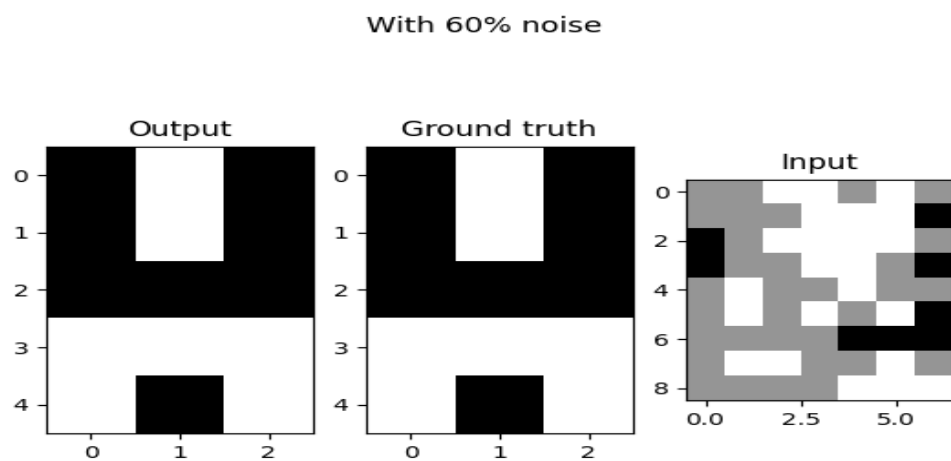
With 20% noise



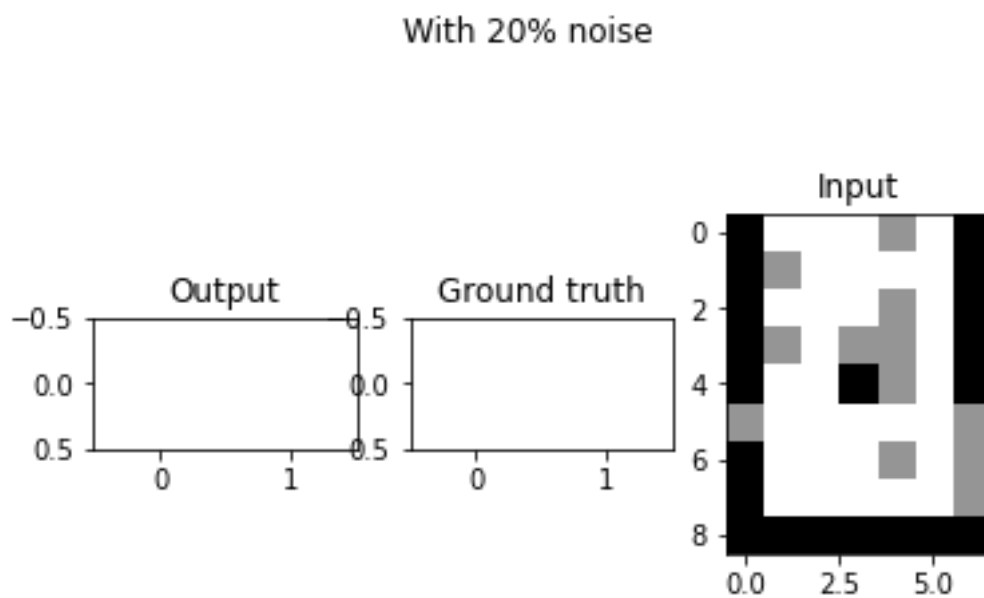
With 60% noise



شکل ۱۰-۱- ورودی با ۶۰٪ اطلاعات از دست رفته، خروجی مد نظر، خروجی شبکه به ترتیب از راست به چپ



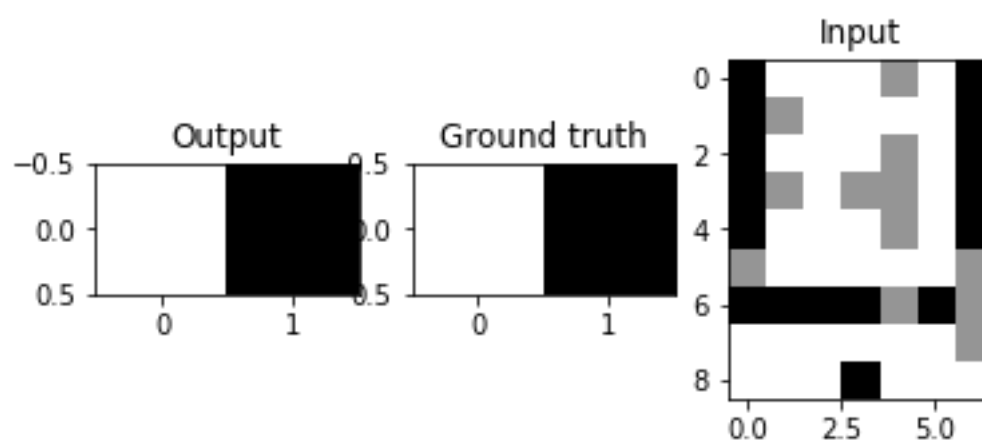
شکل ۱-۱۱- ورودی با ۶۰٪ اطلاعات از دست رفته، خروجی مد نظر، خروجی شبکه به ترتیب از راست به چپ



شکل ۱-۱۲- ورودی با ۲۰٪ اطلاعات از دست رفته، خروجی مد نظر، خروجی شبکه به ترتیب از راست به چپ با ساین

خروجی ۱*۲

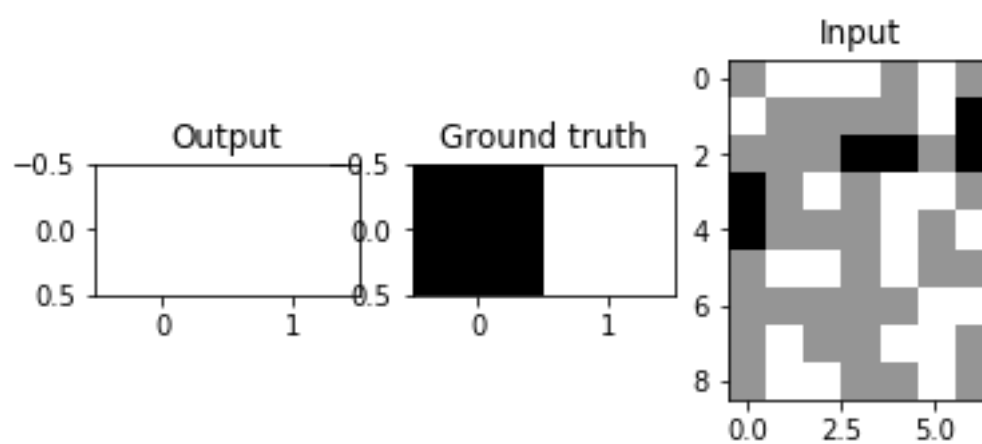
With 20% noise



شکل ۱-۱۳- ورودی با ۲۰٪ اطلاعات از دست رفته، خروجی مد نظر، خروجی شبکه به ترتیب از راست به چپ با سایز خروجی

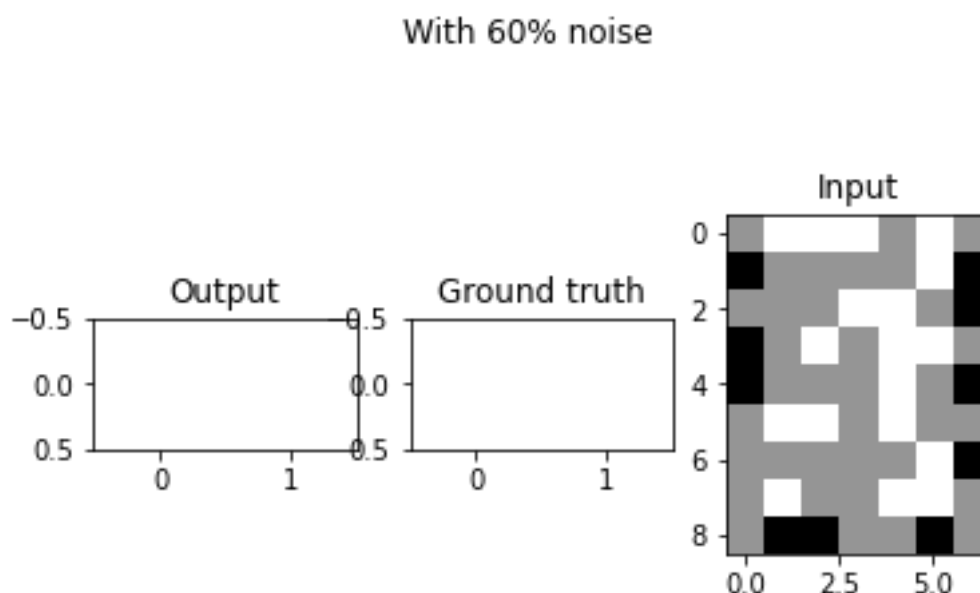
۱*۲

With 60% noise



شکل ۱-۱۴- ورودی با ۶۰٪ اطلاعات از دست رفته، خروجی مد نظر، خروجی شبکه به ترتیب از راست به چپ با سایز خروجی

۱*۲



شکل ۱-۱۵- ورودی با ۶۰٪ اطلاعات از دست رفته، خروجی مد نظر، خروجی شبکه به ترتیب از راست به چپ با سایز خروجی ۱*۲

```
Our model gives the correct output(5*3) for input1 with 20% noise with accuracy = 100 %
Our model gives the correct output(5*3) for input2 with 20% noise with accuracy = 100 %
Our model gives the correct output(5*3) for input3 with 20% noise with accuracy = 98 %
Our model gives the correct output(5*3) for input1 with 60% noise with accuracy = 94 %
Our model gives the correct output(5*3) for input2 with 60% noise with accuracy = 100 %
Our model gives the correct output(5*3) for input3 with 60% noise with accuracy = 92 %
```

شکل ۱-۱۶- دقت خروجی با ابعاد ۵*۳ شبکه به ازای مقادیر مختلف اطلاعات از دست رفته

```
Our model gives the correct output(1*2) for input1 with 20% noise with accuracy = 100 %
Our model gives the correct output(1*2) for input2 with 20% noise with accuracy = 92 %
Our model gives the correct output(1*2) for input3 with 20% noise with accuracy = 100 %
Our model gives the correct output(1*2) for input1 with 60% noise with accuracy = 93 %
Our model gives the correct output(1*2) for input2 with 60% noise with accuracy = 73 %
Our model gives the correct output(1*2) for input3 with 60% noise with accuracy = 100 %
```

شکل ۱-۱۷- دقت خروجی با ابعاد ۱*۲ شبکه به ازای مقادیر مختلف اطلاعات از دست رفته

(۷)

با توجه به نتایج قسمت های پیشین می بینیم که شبکه با دقت خیلی بهتری ورودی های با اطلاعات از دست رفته را به خروجی درست تبدیل می کند در حالی که ورودی شامل نویز را با دقت کمتری می تواند تشخیص دهد. مخصوصا اگر نویز زیاد باشد. زیرا نویز باعث اطلاعات غلط در تصویر می شود در حالی که وقتی اطلاعات از دست رفته باشد، اطلاعات به جا مانده معمولا برای تمایز کافی هستند.

اگر از نظر ابعاد بررسی کنیم متوجه می شویم که شبکه با سایز ۱۲×۱۲ در مواقع نوپز زیاد عملکرد بهتری دارد، اما در مجموع شبکه با سایز خروجی ۳×۵ دقت بهتری را از خود نشان می دهد

سوال ۲- Auto-associative Net

(۱)

وزن ها را به کمک قاعده هب و قاعده هب اصلاح^۱ شده به دست می آوریم

جدول ۲-۱- وزن های به دست آمده به کمک قاعده هب

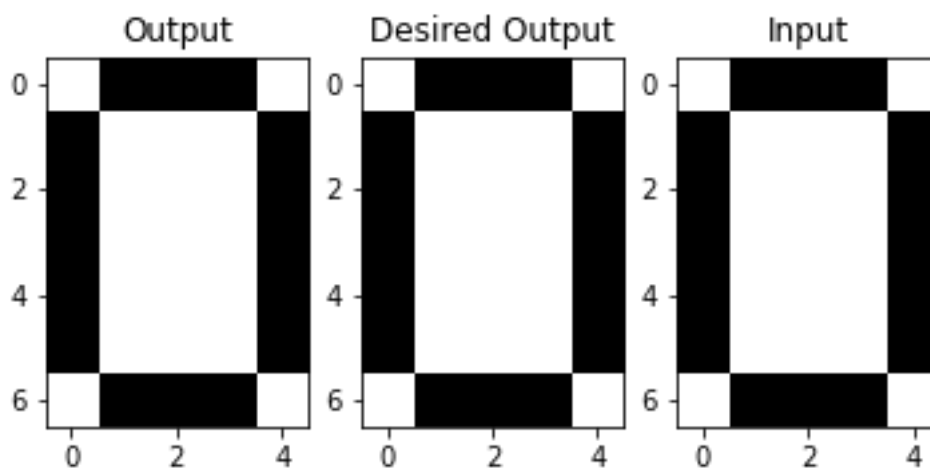
3	-1	-3	-1	3	-1	1	1	3	-1	1	3	1	3	-1	1	1	-1	1	1	-1	3	1	3	1	1	-3	-3	-3	1
-1	3	1	3	-1	3	-3	-3	-1	3	1	-1	-3	-1	3	1	1	-1	1	1	3	-1	-3	-1	1	3	-1	1	1	1
-3	1	3	1	-3	1	-1	-1	-3	1	-1	-3	-1	-3	1	-1	-1	1	-1	-1	1	-3	-1	-3	-1	1	-3	-1	3	3
-1	3	1	3	-1	3	-3	-3	-1	3	1	-1	-3	-1	3	1	1	-1	1	1	3	-1	-3	-1	1	3	-1	1	1	1
3	-1	-3	-1	3	-1	1	1	3	-1	1	3	1	3	-1	1	1	-1	1	1	-1	3	1	3	1	1	-3	1	1	1
-1	3	1	3	-1	3	-3	-3	-1	3	1	-1	-3	-1	3	1	1	-1	1	1	3	-1	-3	-1	1	3	-1	1	1	1
1	-3	-1	-3	1	-3	3	3	1	-3	-1	1	3	1	-3	-1	-1	-1	-1	-1	-3	1	3	1	-1	-3	1	1	-1	-1
1	-3	-1	-3	1	-3	3	3	1	-3	-1	1	3	1	-3	-1	-1	-1	-1	-1	-3	1	3	1	-1	-3	1	1	-1	-1
3	-1	-3	-1	3	-1	1	1	3	-1	1	3	1	3	-1	1	1	-1	1	1	-1	3	1	3	1	1	-3	1	1	1
-1	3	1	3	-1	3	-3	-3	-1	3	1	-1	-3	-1	3	1	1	-1	1	1	3	-1	-3	-1	1	3	-1	1	1	1
1	1	-1	1	1	1	-1	-1	1	1	3	1	-1	1	1	3	-1	-3	-1	3	1	1	-1	1	3	1	1	-1	-1	-1
3	-1	-3	-1	3	-1	1	1	3	-1	1	3	1	3	-1	1	1	-1	1	1	-1	3	1	3	1	1	1	3	3	1
1	-3	-1	-3	1	-3	3	3	1	-3	-1	1	3	1	-3	-1	-1	-1	-1	-1	-3	1	3	1	-1	-3	1	1	-1	-1
1	-3	-1	-3	1	-3	3	3	1	-3	-1	1	3	1	-3	-1	-1	-1	-1	-1	-3	1	3	1	-1	-3	1	1	-1	-1
3	-1	-3	-1	3	-1	1	1	3	-1	1	3	1	3	-1	1	1	-1	1	1	3	-1	-3	-1	1	3	-1	1	1	1
1	1	-1	1	1	1	-1	-1	1	1	3	1	-1	1	1	3	-1	-3	-1	3	1	1	-1	1	3	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	-3	1	3	1	-1	-3	1	1	-1	-1
1	1	-1	1	1	1	-1	-1	1	1	3	1	-1	1	1	3	-1	-3	-1	3	1	1	-1	1	3	1	1	-1	-1	-1
-1	3	1	3	-1	3	-3	-3	-1	3	1	-1	-3	-1	3	1	1	-1	1	1	3	-1	-3	-1	1	3	-1	1	1	1
3	-1	-3	-1	3	-1	1	1	3	-1	1	3	1	3	-1	1	1	-1	1	1	-1	3	1	3	1	1	-3	1	1	1
1	-3	-1	-3	1	-3	3	3	1	-3	-1	1	3	1	-3	-1	-1	-1	-1	-1	-3	1	3	1	-1	-3	1	1	-1	-1
3	-1	-3	-1	3	-1	1	1	3	-1	1	3	1	3	-1	1	1	-1	1	1	-1	3	1	3	1	1	-3	1	1	1
1	-3	-1	-3	1	-3	3	3	1	-3	-1	1	3	1	-3	-1	-1	-1	-1	-1	-3	1	3	1	-1	-3	1	1	-1	-1
3	-1	-3	-1	3	-1	1	1	3	-1	1	3	1	3	-1	1	1	-1	1	1	-1	3	1	3	1	1	-3	1	1	1
1	1	-1	1	1	1	-1	-1	1	1	3	1	-1	1	1	3	-1	-3	-1	3	1	1	-1	1	3	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	-3	1	3	1	-1	-3	1	1	-1	3
-3	1	3	1	-3	1	-1	-1	-3	1	-1	-3	-1	-3	1	-1	-1	-1	-1	-1	-3	-1	-3	-1	1	-3	-1	-1	3	3
-3	1	3	1	-3	1	-1	-1	-3	1	-1	-3	-1	-3	1	-1	-1	-1	-1	-1	-3	-1	-3	-1	1	-3	-1	-1	3	3
-3	1	3	1	-3	1	-1	-1	-3	1	-1	-3	-1	-3	1	-1	-1	-1	-1	-1	-3	-1	-3	-1	1	-3	-1	-1	3	3
1	1	-1	1	1	1	-1	-1	1	1	-1	-1	-1	-1	1	1	-1	-1	-1	-1	-3	1	3	-1	1	1	-1	-1	-1	3

¹ Modified Hebbian Learning Rule

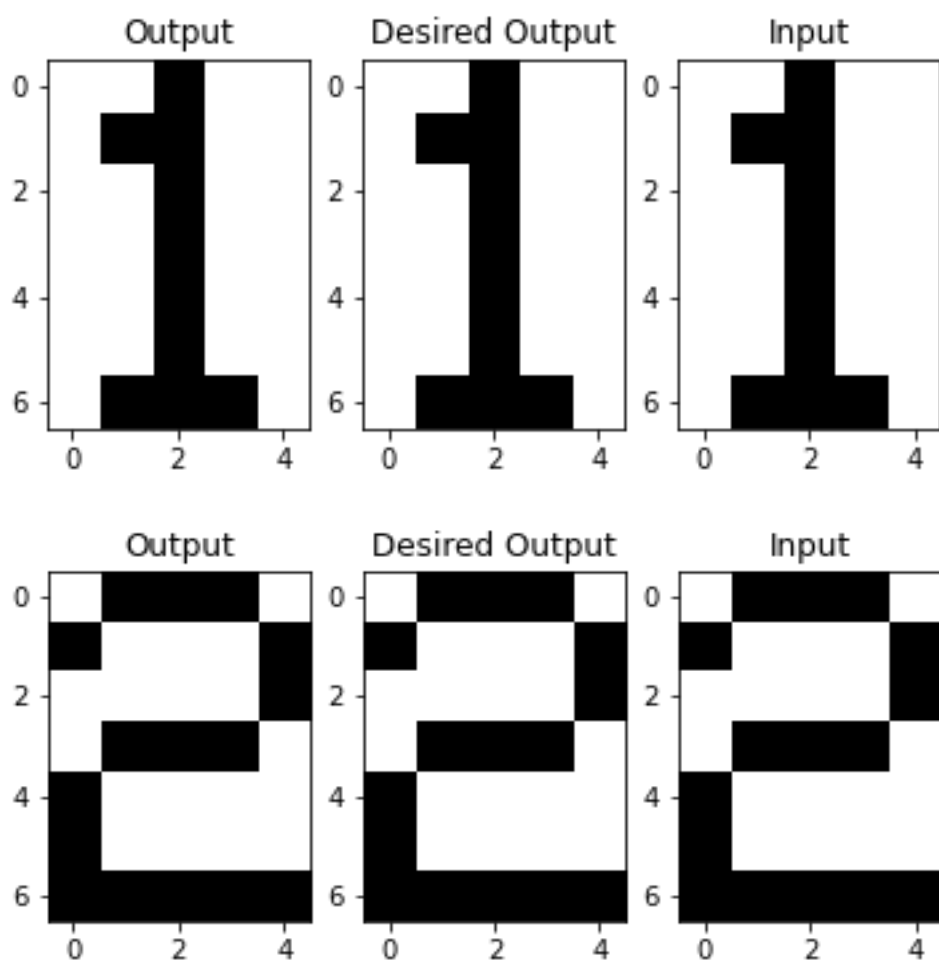
جدول ۲-۲- وزن های به دست آمده به کمک قاعده هب اصلاح شده

[illegible]

(۲)



شکل ۲-۱- ورودی، خروجی مطلوب و خروجی شبکه از راست به چپ

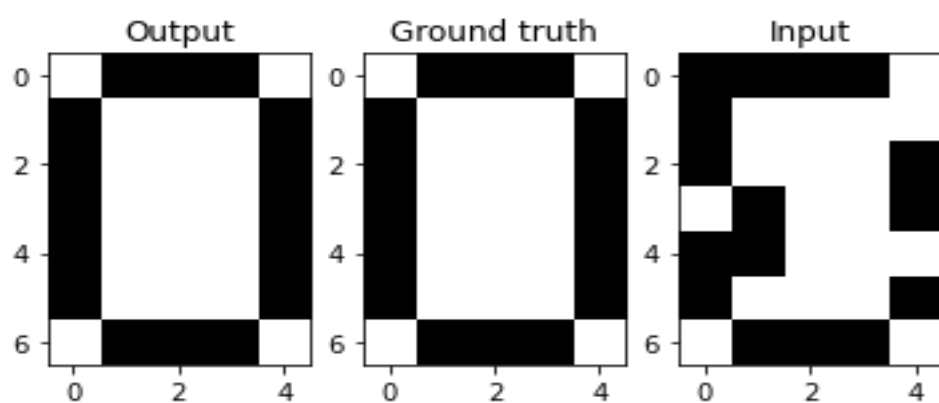


شکل ۲-۲- ورودی، خروجی مطلوب و خروجی شبکه از راست به چپ

۳

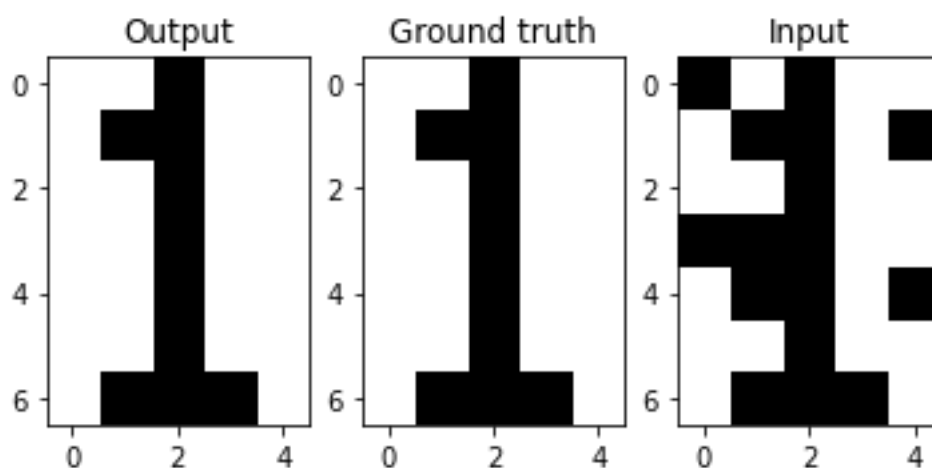
(الف)

With 20% noise

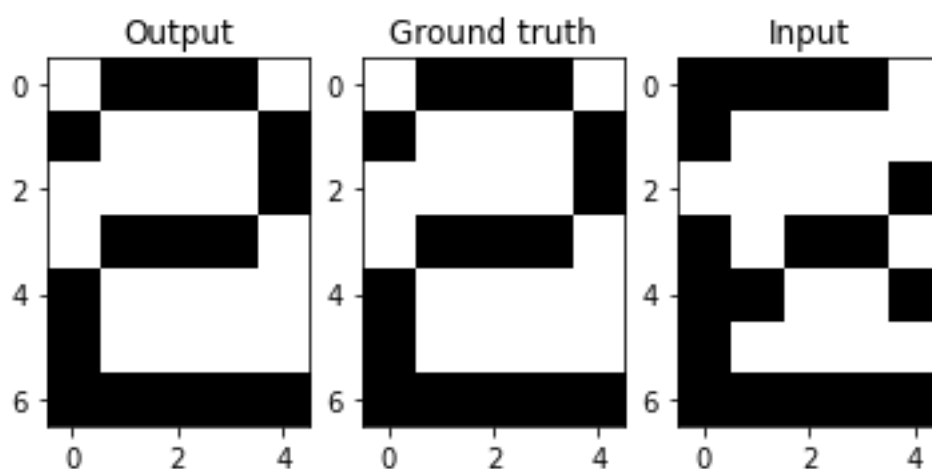


شکل ۲-۳- ورودی با ۲۰٪ نویز، خروجی مطلوب و خروجی شبکه از راست به چپ

With 20% noise



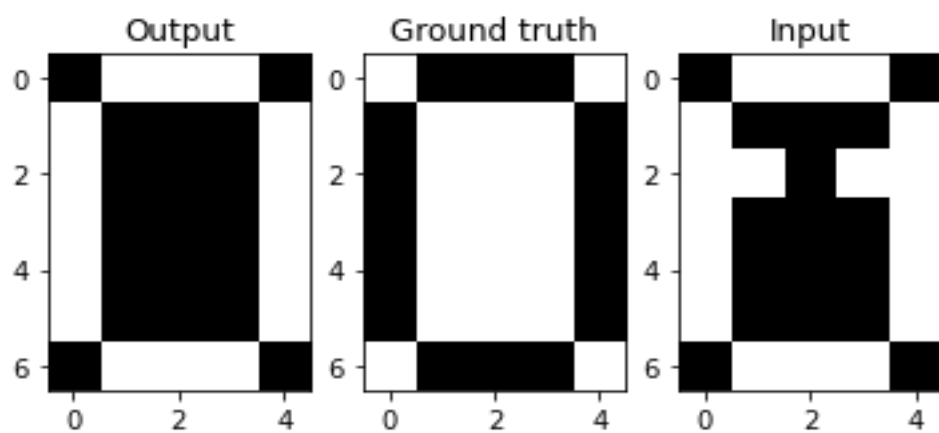
With 20% noise



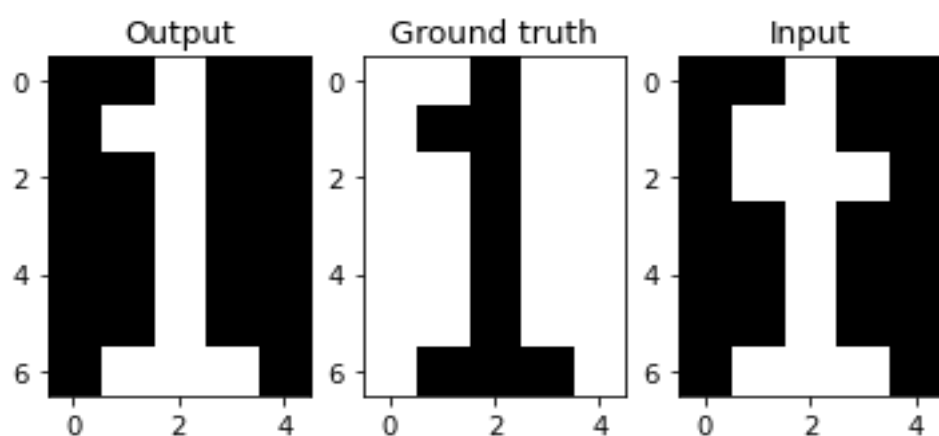
شکل ۲-۴- ورودی با ۲۰٪ نویز، خروجی مطلوب و خروجی شبکه از راست به چپ

می بینیم شبکه در مقابل نویز ۲۰٪ مقاوم است.

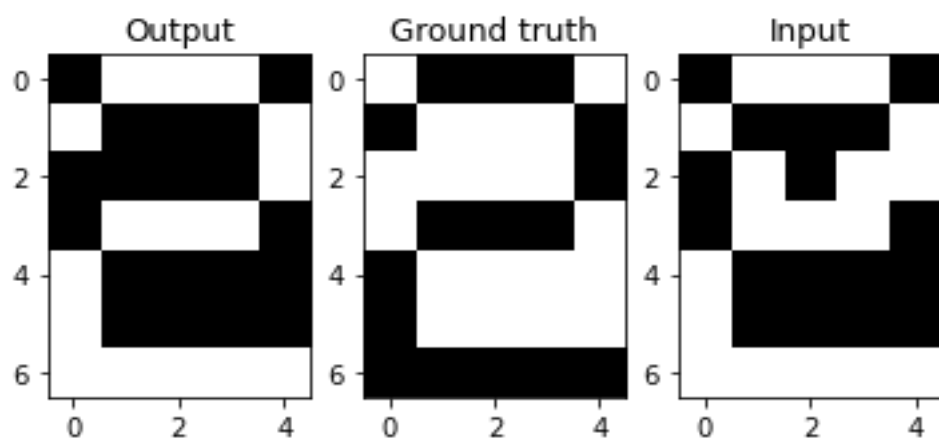
With 80% noise



With 80% noise



With 80% noise



شکل ۲-۵- ورودی با ۸۰٪ نویز، خروجی مطلوب و خروجی شبکه از راست به چپ

می بینیم شبکه در مقابل نویز ۸۰٪ مقاوم نیست، اما چون اطلاعات با احتمال ۸۰٪ تغییر علامت می دهند، تصویر خروجی شبکه قرینه تصویر خروجی مطلوب است.

(ب)

برای اینکه بفهمیم کدام عدد به نویز حساس تر است میانگین دقت در ۱۰۰ بار تکرار را حساب می کنیم

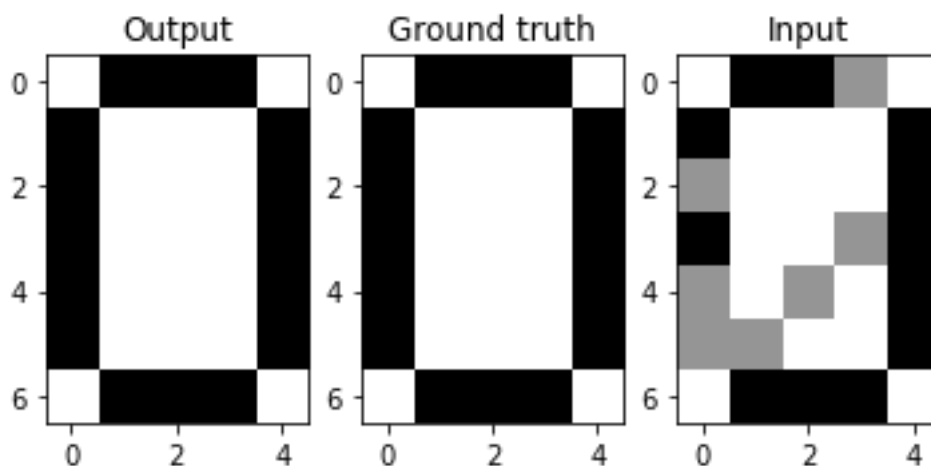
```
Our model gives the correct output for input0 with 20% noise with accuracy = 90 %
Our model gives the correct output for input1 with 20% noise with accuracy = 95 %
Our model gives the correct output for input2 with 20% noise with accuracy = 88 %
Our model gives the correct output for input0 with 80% noise with accuracy = 0 %
Our model gives the correct output for input1 with 80% noise with accuracy = 0 %
Our model gives the correct output for input2 with 80% noise with accuracy = 0 %
```

شکل ۲-۶- دقت شبکه به ازای ورودی ها و نویزهای مختلف

با توجه به شکل ۲-۴ نتیجه می گیریم که عدد ۱ نسبت به ۰ و ۲ کمتر به نویز حساس است، احتمالا دلیل این است که عدد ۱ شباهت کمتری به اعداد ۰ و ۲ لاتین دارد

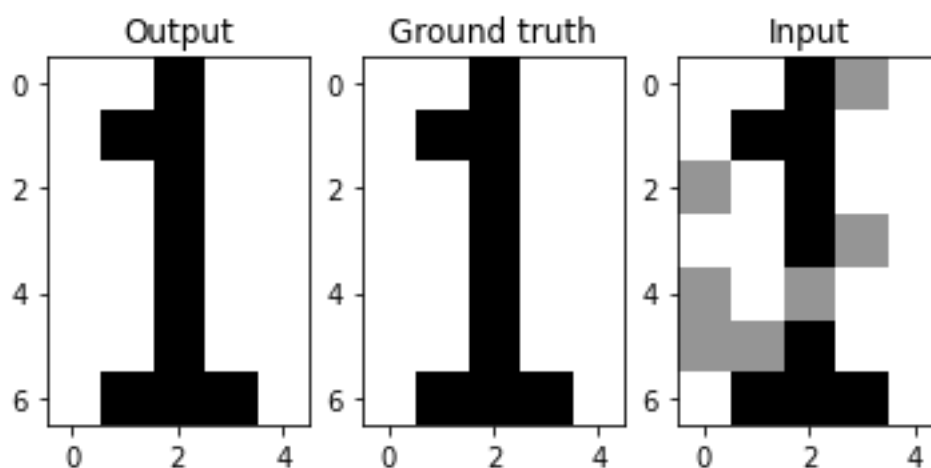
(۴)

With 20% noise

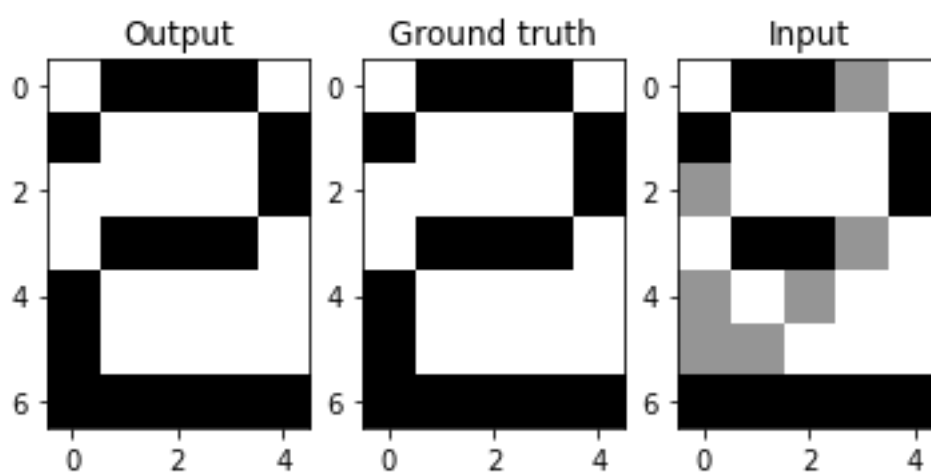


شکل ۲-۷- ورودی با ۲۰٪ اطلاعات از دست رفته، خروجی مطلوب و خروجی شبکه از راست به چپ

With 20% noise

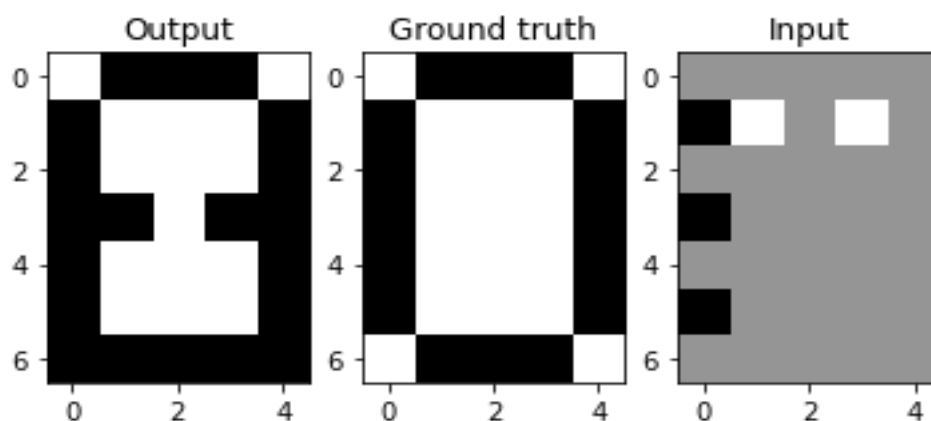


With 20% noise

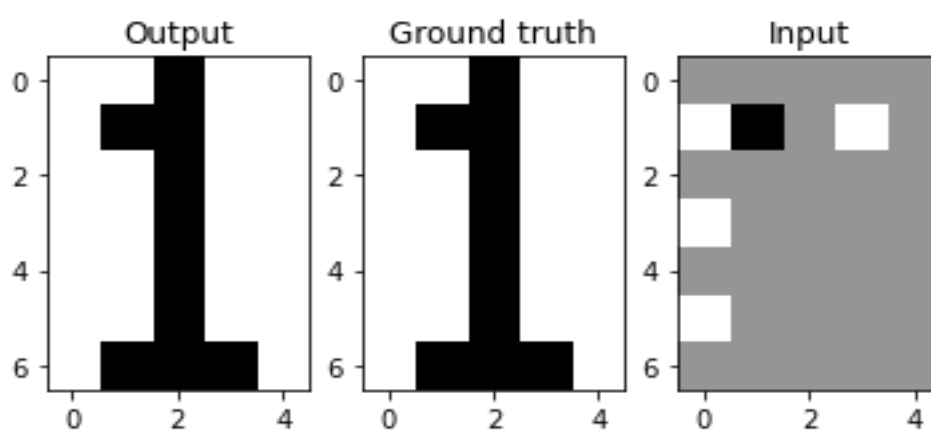


شکل ۲-۸- ورودی با ۲۰٪ اطلاعات از دست رفته، خروجی مطلوب و خروجی شبکه از راست به چپ

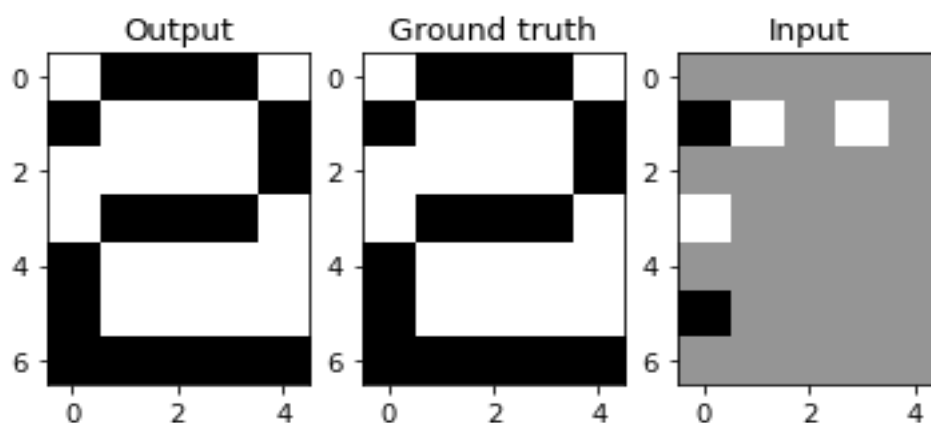
With 80% noise



With 80% noise



With 80% noise



شکل ۲-۹- ورودی با ۸۰٪ اطلاعات از دست رفته، خروجی مطلوب و خروجی شبکه از راست به چپ

می بینیم که شبکه نسبت به از دست رفتن اطلاعات مقاومت بیشتری دارد

(۵)

همانگونه که در شکل ۲-۱۰ می بینیم شبکه با تصاویر بیشتر دارای عملکرد مطلوبی نیست هنگامی که الگوهای ورودی شبکه متعامد نیستند قانون هب خطاهایی را تولید می کند یکی از راهکارها برای مقابله با این مشکل استفاده از شبه معکوس است که به توضیح آن می پردازیم، طبق شبکه داریم:

$$\text{Equation 1: } Wp_q = t_q$$

که t_q خروجی متانظر با ورودی p_q می باشد

خطای شبکه در این حالت از رابطه زیر به دست می آید

$$F(W) = \sum_{q=1}^Q ||t_q - Wp_q||^2$$

اگر معادله ۱ را به فرم ماتریسی بنویسیم داریم

$$WP = T$$

$$T = [t_1, \dots, t_Q], \quad P = [p_1, \dots, p_Q]$$

خطا به صورت زیر در می آید

$$||E||^2 = \sum \sum e_{ij}^2$$

اثبات می شود که اگر W را به صورت زیر پیدا کنیم خطا برابر صفر می شود

$$W = TP^{-1}$$

اما در مواقعی که ماتریس P دارای معکوس نیست می توانیم از عملگر شبه معکوس استفاده کنیم،

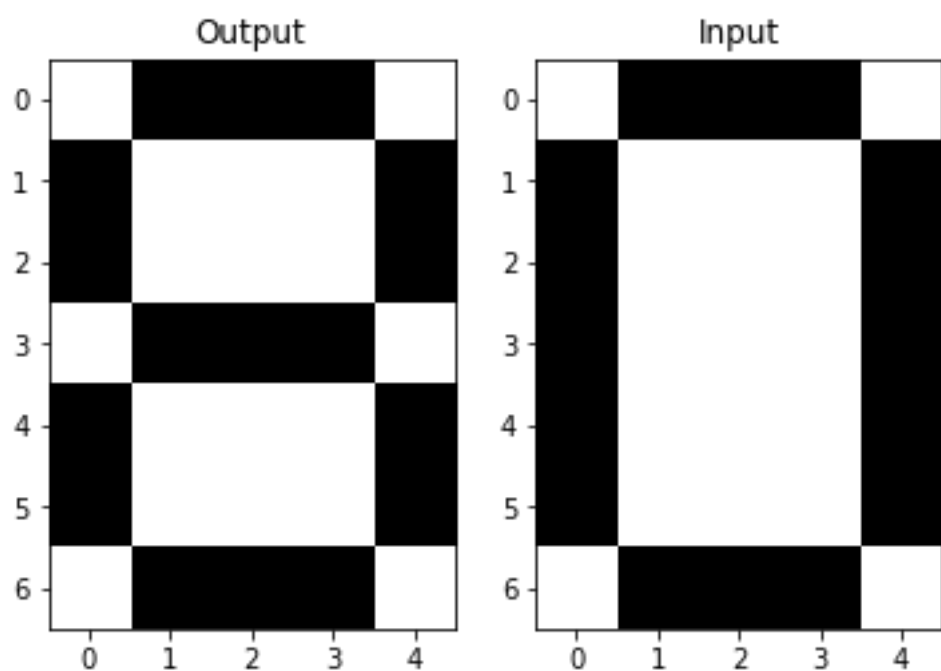
یعنی

$$W = TP^{\dagger}$$

که شبه معکوس به صورت زیر به دست می آید

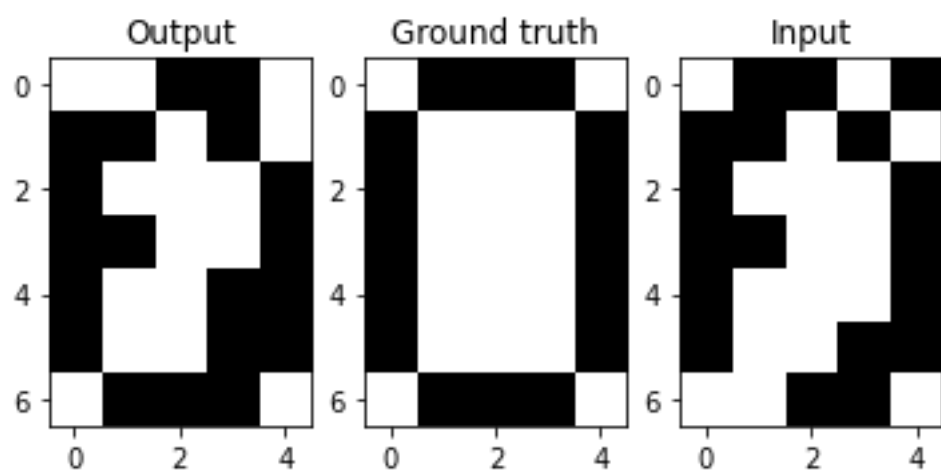
$$P^{\dagger} = (P^T P)^{-1} P^T$$

در ادامه به کمک این روش مقاومت شبکه با ورودی های متعدد را در برابر نویز بررسی می کنیم.



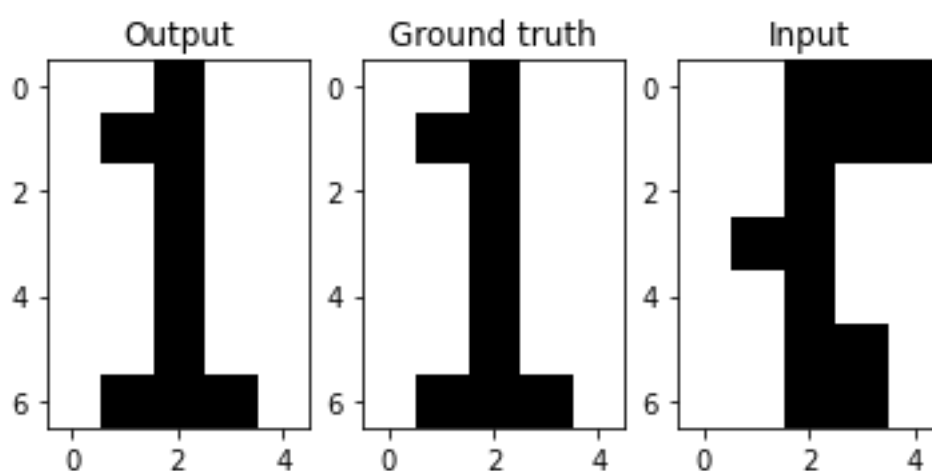
شکل ۲-۱۰- ورودی، خروجی شبکه هنگامی که اعداد بیشتری را به شبکه می دهیم

With 20% noise

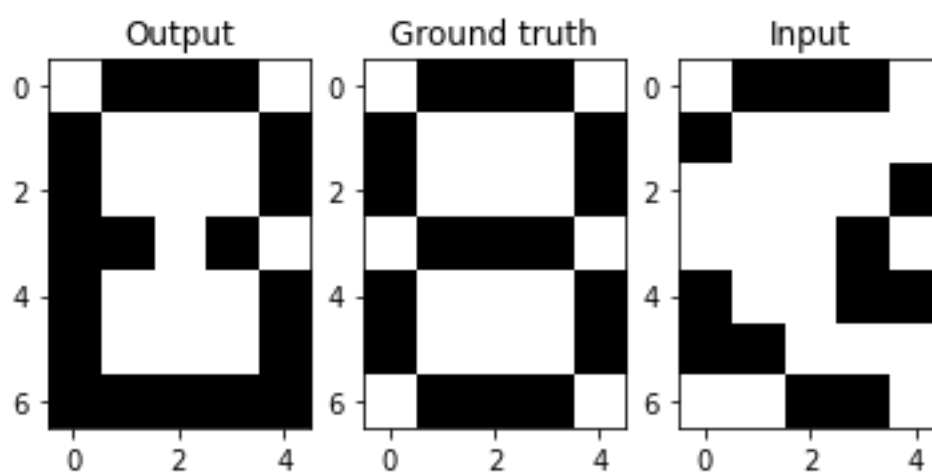


شکل ۲-۱۱- ورودی با ۲۰٪ نویز، خروجی مطلوب و خروجی شبکه از راست به چپ

With 20% noise

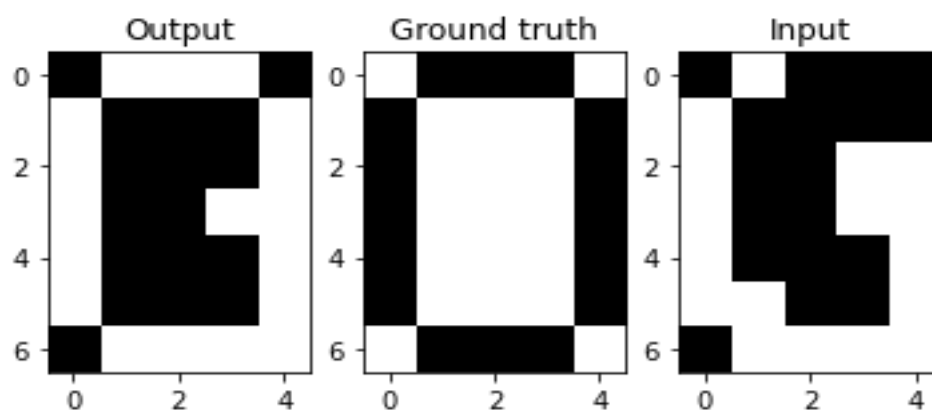


With 20% noise

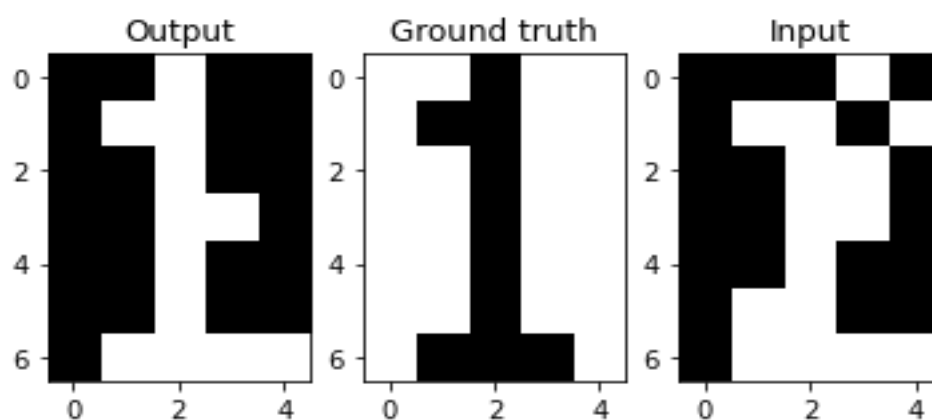


شکل ۲-۱۲- ورودی با ۲۰٪ نویز، خروجی مطلوب و خروجی شبکه از راست به چپ

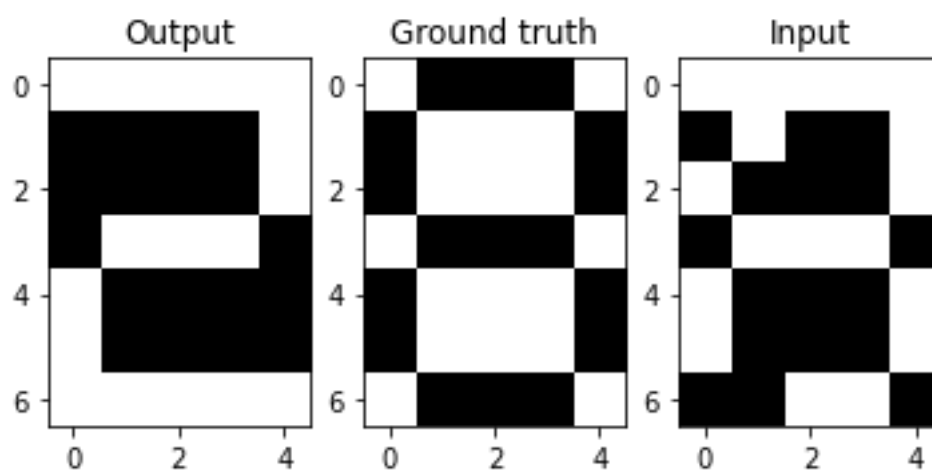
With 80% noise



With 80% noise



With 80% noise



شکل ۲-۱۳- ورودی با ۸۰٪ نویز، خروجی مطلوب و خروجی شبکه از راست به چپ

با توجه به نتایج فوق می بینیم که شبکه ای که وزن های آن با روش شبه معکوس به دست آمده نسبت به شبکه ای که به روش هب وزن های آن به دست آمده هنگامی که شبکه قرار است ورودی های بیشتری را به خاطر بسپارد عملکرد بهتری دارد.

سوال ۳- Discrete Hopfield Network

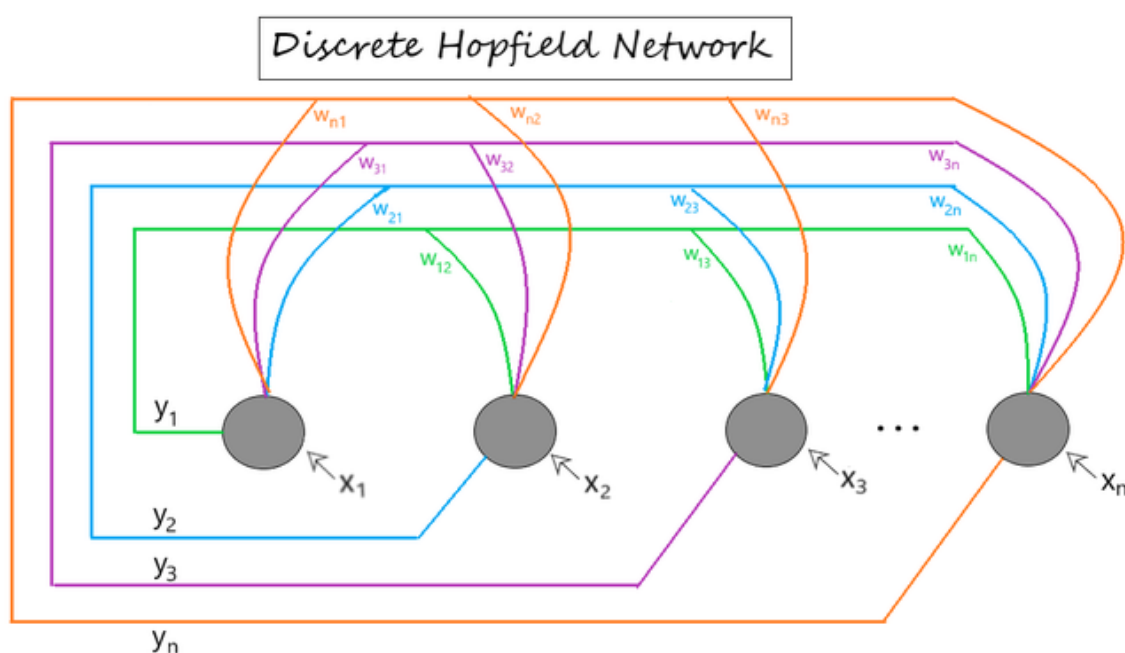
(۱)

شبکه هاپفیلد، گسسته است به این معنا که تنها با بردار های دو قطبی کار می کند که فقط مقادیر ۱ و -۱ را می توانند داشته باشند، این شبکه کاملاً درون متصل^۱ می باشد به این معنی که هر واحد به تمام واحدهای دیگر متصل است، در این شبکه ماتریس وزن ها متقارن است و اتصال خودی نداریم

$$w_{ij} = w_{ji}$$

$$w_{ii} = 0$$

معماری این شبکه به صورت شکل؟ است



شکل ۳-۱- معماری شبکه هاپفیلد

الگوریتم این شبکه به صورت زیر است:

۱. مقدار دهی اولیه ماتریس وزن ها (با استفاده از قانون هب)

۲. برای هر بردار ورودی x مراحل زیر را انجام بده

¹ Interconnected

$$y_i = x_i \quad ۳.$$

$$y_{in_i} = x_i + \sum_j y_j w_{ji} \quad ۴.$$

$$y_i = \begin{cases} 1 & \text{if } y_{in_i} > \theta \\ y_i & \text{if } y_{in_i} = \theta \\ 0 & \text{if } y_{in_i} < \theta \end{cases} \quad ۵.$$

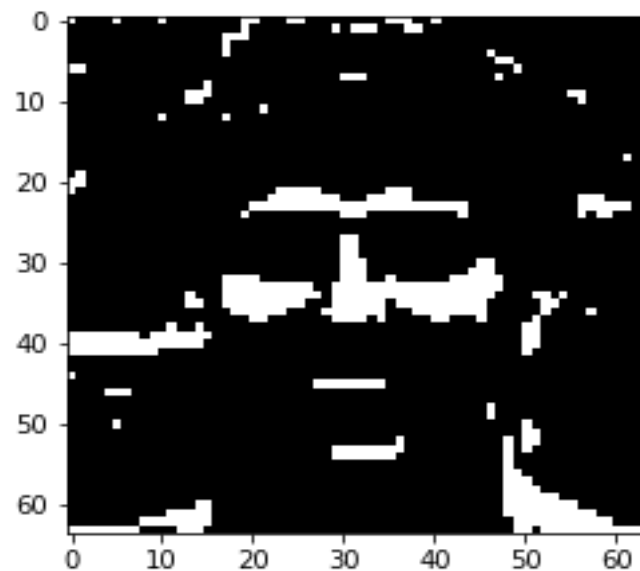
۶. ارسال y_i به تمام واحد های دیگر (آپدیت کردن اکتیویشن ها)

۷. بررسی همگرایی (اگر همگرا نشدیم به مرحله ۲ برو)

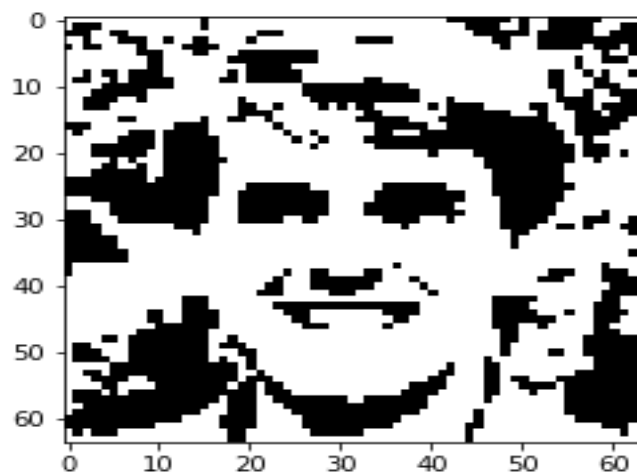
اثبات می شود که خروجی این شبکه همگرا می شود

(۲)

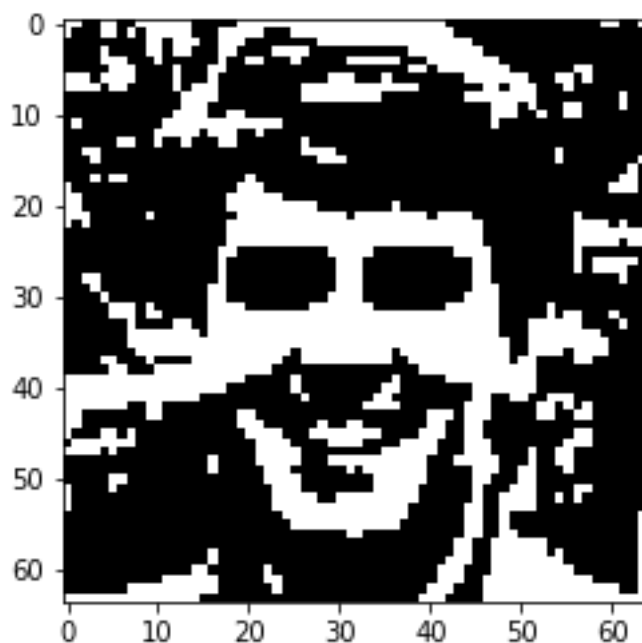
پیش پردازش های خواسته شده را با آستانه های مختلف روی تصاویر اعمال می کنیم



شکل ۳-۲- تصویر پیش پردازش شده با آستانه ۱۵۰



شکل ۳-۳- تصویر پیش پردازش شده با آستانه ۱۰۰



شکل ۳-۴- تصویر پیش پردازش شده با آستانه ۱۲۵

تصویری که با قرار دادن آستانه برابر ۱۲۵ تولید می شود واضح تر از سایر آستانه ها است، لذا در ادامه با آستانه ۱۲۵ کار خواهیم کرد.

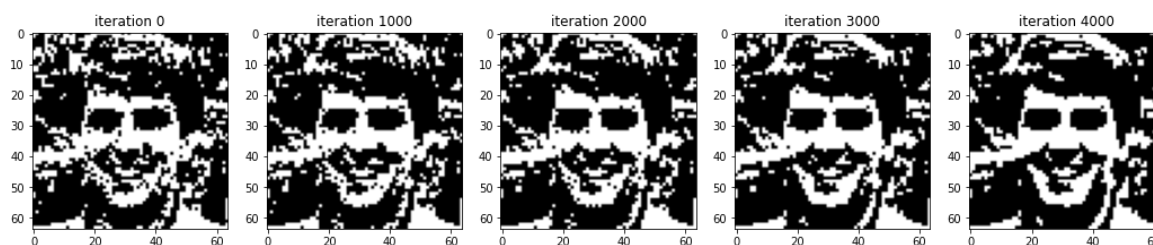
(۳)

ماتریس وزن ها را بر اسا قاعده هب اصلاح شده می سازیم

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

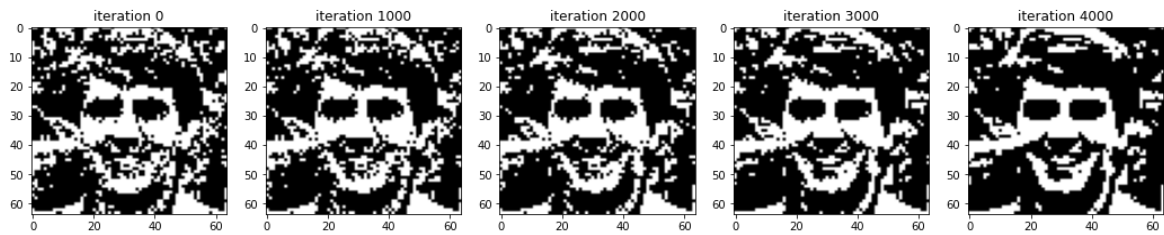
(۴)

Trying to reconstruct test1



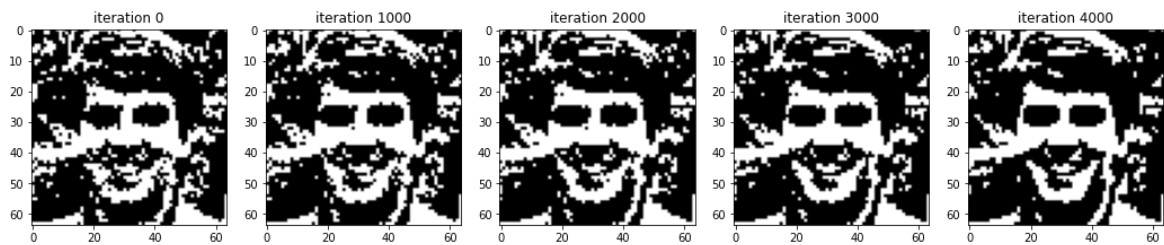
شکل ۳-۵- تصاویر ورودی اول پس از هر ۱۰۰۰ بار تکرار

Trying to reconstruct test2



شکل ۳-۶- تصاویر ورودی دوم پس از هر ۱۰۰۰ بار تکرار

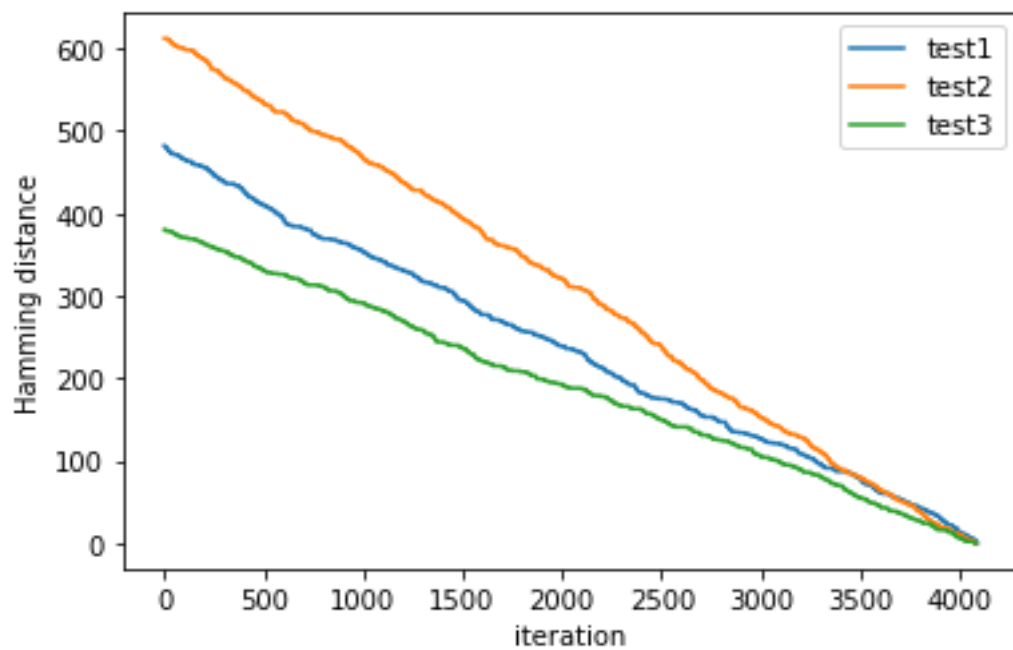
Trying to reconstruct test3



شکل ۳-۷- تصاویر ورودی سوم پس از هر ۱۰۰۰ بار تکرار

همانطور که می بینیم شبکه ما قابلیت بازیابی تصویر اصلی را از روی تصاویر نویزی دارد.

(۵)



شکل ۳-۸- فاصله Hamming به ازای تکرار برای ۳ ورودی مختلف

از روی شکل ۳-۸ متوجه می شویم که تصویر تست ۲ دارای بیشترین نویز و تصویر تست ۳ دارای کمترین نویز هستند و فاصله Hamming همگی در حدود ۴۰۷۰ تکرار به صفر می رسد و پاسخ شبکه همگرا می شود.

سوال ۴ - Bidirectional Associative Memory

(\)

جدول ۴-۱- وزن های به دست آمده برای BAM

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84															

(۲)

```
Similarity between predicted output and ground truth given Hillary: 100.0
Similarity between predicted output and ground truth given Clinton: 100.0
Similarity between predicted output and ground truth given Kenstar: 100.0
Similarity between predicted output and ground truth given FirstLady: 100.0
Similarity between predicted output and ground truth given President: 100.0
Similarity between predicted output and ground truth given Gentleman: 100.0
```

شکل ۴-۱- شباهت بین خروجی شبکه و خروجی واقعی، یکبار با ورودی نام و یکبار با ورودی ویژگی

(۳)

معیار شباهت را مساوی بودن بیت های خروجی شبکه و خروجی واقعی در نظر گرفتیم، مثلاً با توجه به شکل ۴-۲ با دادن ورودی Clinton به همراه ۲۰٪ نویز، در ۱۰۰ بار تکرار به صورت میانگین ۹۸.۶۸٪ بیت های خروجی با خروجی واقعی مطابقت دارد.

```
Average accuracy of model given input Clinton with 10% noise: 99.7936507936508
Average accuracy of model given input Hillary with 10% noise: 100.0
Average accuracy of model given input Kenster with 10% noise: 99.80952380952381
Average accuracy of model given input Clinton with 20% noise: 96.93650793650795
Average accuracy of model given input Hillary with 20% noise: 98.68253968253968
Average accuracy of model given input Kenster with 20% noise: 96.71428571428574
```

شکل ۴-۲- شباهت بین خروجی شبکه و خروجی واقعی با ورودی اسم ها همراه نویز ۱۰٪ و ۲۰٪

```
Average accuracy of model given input President with 10% noise: 100.0
Average accuracy of model given input FirstLady with 10% noise: 100.0
Average accuracy of model given input Gentleman with 10% noise: 100.0
Average accuracy of model given input President with 20% noise: 99.6734693877551
Average accuracy of model given input FirstLady with 20% noise: 98.87755102040815
Average accuracy of model given input Gentleman with 20% noise: 99.51020408163265
```

شکل ۴-۳- شباهت بین خروجی شبکه و خروجی واقعی با ورودی ویژگی ها همراه نویز ۱۰٪ و ۲۰٪

(۴)

همانطور که می بینیم دقت شبکه کاهش یافته و قادر به بازیابی نام از روی ویژگی SweetGirl نیست دلیل این است که بردار های ذخیره شده در شبکه باعث کاهش فضای پوچی و کاهش قابلیت بازیابی الگو می شوند

```
Similarity between predicted output and ground truth given Hillary: 100.0
Similarity between predicted output and ground truth given Clinton: 100.0
Similarity between predicted output and ground truth given Kenstar: 100.0
Similarity between predicted output and ground truth given Lewisky: 100.0
Similarity between predicted output and ground truth given FirstLady: 100.0
Similarity between predicted output and ground truth given President: 100.0
Similarity between predicted output and ground truth given Gentleman: 100.0
Similarity between predicted output and ground truth given SweetGirl: 91.83673469387756
```

شکل ۴-۴- شباهت بین خروجی شبکه و خروجی واقعی ورودی ویژگی ها