



به نام خدا



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

نام و نام خانوادگی	امیر محمد رنجبر پازکی
شماره دانشجویی	۸۱۰۱۹۹۳۴۰
تاریخ ارسال گزارش	۱۴۰۰/۳/۱۲

فهرست گزارش سوالات

- سوال ۱ - شناسایی حروف با استفاده از روش هب 3
- سوال ۲ - شبکه خودانجمنی 6
- سوال ۳ - شبکه هاپفیلد 8
- سوال ۴ - شبکه BAM 11

سوال ۱ - شناسایی حروف با استفاده از روش هب

الف) بله. همانطور که در تصویر زیر می‌بینید، این شبکه سه حرف را به خوبی شناسایی کرده و خروجی موردنظر را تولید کرده‌است.

```
I remember =))
```

```
Result:
```

```
-#-
```

```
#-#
```

```
###
```

```
#-#
```

```
#-#
```

```
I remember =))
```

```
Result:
```

```
##-
```

```
#-#
```

```
##-
```

```
#-#
```

```
##-
```

```
I remember =))
```

```
Result:
```

```
-##
```

```
#--
```

```
#--
```

```
#--
```

```
-##
```

شکل ۱- خروجی شبکه‌ی ساده برای حروف A، B و C

ب) بله. در بسیاری از موارد نتیجه دلخواه به دست آمد. در زیر برای هر حالت خروجی نشان داده شده‌است. این خروجی‌های برای یک کاراکتر نشان داده شده‌است. کاراکترهای بیشتر در نوت بوک موجود است.

```

-#-#---
---#---
-----
--#-#--
--#-#--
-####--
##-#-#-
-#---#-
###-##-

```

I remember =))

Result:

```

-#-
#-#
###
#-#
#-#

```

```

#--##--
-#-####
---##--
--#-##-
--#-#--
####-#-
-#---#-
-#-----
-----##

```

I remember =))

Result:

```

-#-
#-#
###
#-#
#-#

```

شکل ۲- خروجی کاراکتر A به ازای نویز ۱۰ و ۲۵ درصدی

```

---#---
---#---
---#---
----#--
--#-#--
-#####-
-#---#-
-#---#-
##--###

```

I remember =))

Result:

```

-#-
#-#
###
#-#
#-#

```

```

---#---
---#---
---#---
--#-#--
--#----
-#-##--
-----#-
-#-----
###-###

```

I remember =))

Result:

```

-#-
#-#
###
#-#
#-#

```

شکل ۳- خروجی کاراکتر A به ازای از دست دادن ۱۰ و ۲۵ درصد داده

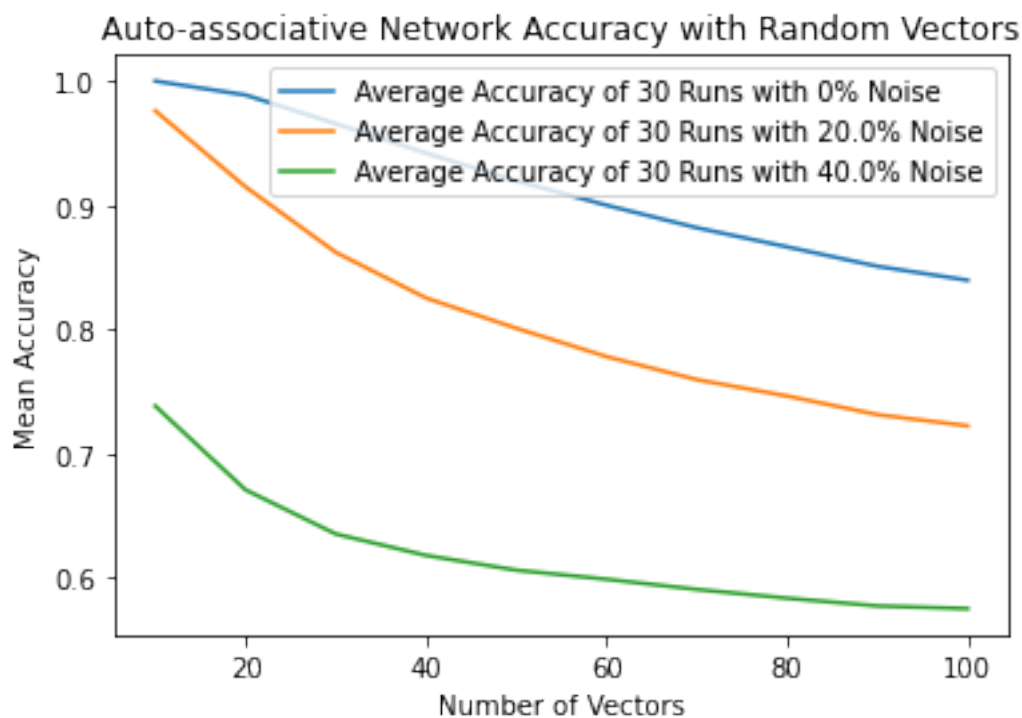
شبکه برای هر کاراکتر در هر حالت ۱۰۰ پیش‌بینی انجام داده‌است که نتایج دقت شبکه در زیر قابل مشاهده‌است.

Ten Percent Noise accuracy: 100.0 %
 Twenty Five Percent Noise accuracy: 97.33333333333333 %
 Ten Percent Missing accuracy: 100.0 %
 Twenty Five Percent Missing accuracy: 99.66666666666667 %

شکل ۴- دقت خروجی شبکه به ازای ۱۰۰ اجرا در هر حالت

سوال ۲ - شبکه خودانجمنی

الف) نمودار زیر به ازای بردارهایی با طول‌هایی از ۱۰ تا ۱۰۰ و هر کدام با ۱۰۰ عدد بین ۱ و -۱ و با خطاهای ۰، ۲۰ و ۴۰ درصد به دست آمده‌اند. رنگ نمایانگر میزان خطا، محور افقی نشان‌دهنده طول بردارها و محور عمودی نمایانگر میانگین دقت است.



شکل ۵- نمودار میانگین دقت بر مبنای طول بردارهای تصادفی به عنوان ورودی

در هر حالت ۳۰ اجرا گرفته شده‌است و مقادیر میانگین و انحراف معیار گزارش شده‌اند. میانگین‌ها در نمودار بالا قابل مشاهده هستند. همانطور که می‌بینید با بالاتر رفتن نویز دقت کاهش یافته‌است.

N: 100 - R: 10 - E: 0
 Accuracy mean: 0.9997333333333334, std: 0.0006289320754704408
 N: 100 - R: 20 - E: 0
 Accuracy mean: 0.9885333333333332, std: 0.00446641790352055
 N: 100 - R: 30 - E: 0
 Accuracy mean: 0.9654555555555555, std: 0.004109143673369775
 N: 100 - R: 40 - E: 0
 Accuracy mean: 0.9417416666666666, std: 0.005969883909163446
 N: 100 - R: 50 - E: 0
 Accuracy mean: 0.9194733333333333, std: 0.005540754060193935
 N: 100 - R: 60 - E: 0
 Accuracy mean: 0.89985, std: 0.0031996382897423307
 N: 100 - R: 70 - E: 0
 Accuracy mean: 0.8817571428571429, std: 0.003447201369069025
 N: 100 - R: 80 - E: 0
 Accuracy mean: 0.8663791666666667, std: 0.00475673681623117
 N: 100 - R: 90 - E: 0
 Accuracy mean: 0.850774074074074, std: 0.004805902007064783
 N: 100 - R: 100 - E: 0
 Accuracy mean: 0.8395433333333333, std: 0.003982602443405823
 N: 100 - R: 10 - E: 0.2
 Accuracy mean: 0.9758666666666668, std: 0.009120428839820111
 N: 100 - R: 20 - E: 0.2
 Accuracy mean: 0.9146500000000001, std: 0.009690330231731036
 N: 100 - R: 30 - E: 0.2
 Accuracy mean: 0.8620111111111111, std: 0.005404033701496837
 N: 100 - R: 40 - E: 0.2
 Accuracy mean: 0.825275, std: 0.0065020349378739385
 N: 100 - R: 50 - E: 0.2
 Accuracy mean: 0.8009466666666668, std: 0.006048676622939004
 N: 100 - R: 60 - E: 0.2
 Accuracy mean: 0.7781277777777778, std: 0.006235157685097023

شکل ۶- مقادیر میانگین و انحراف معیار دقت اجراها

مقادیر بیشتر در نوت بوک قابل مشاهده است.

ب) نویز تاثیر معکوسی رو عملکرد این نوع شبکه‌ها دارد چراکه در الگوها اختلال ایجاد می‌کند و حتی تا حدودی آن‌ها را به یکدیگر نزدیک می‌کند. به همین دلیل، عملکرد شبکه افت می‌کند و دچار اختلال می‌شود.

همچنین، هنگامی که تعداد الگوهای در حافظه این نوع شبکه‌ها بیشتر می‌شود، فاصله الگوها به یکدیگر کمتر می‌شود و به همین دلیل، عملکرد تشخیص شبکه سخت‌تر می‌شود. گویی فضای پوچی ماتریس وزن‌های شبکه‌ی کوچکتر می‌شود و این خطا را بالاتر می‌برد.

ج) قطعا به رابطه‌ی بردارهای ورودی بستگی دارد. زیرا این بردارهای ورودی هستند که ماتریس وزن‌ها را می‌سازند و تعیین کننده فضای پوچی این ماتریس هستند. به عنوان مثال، اگر بردارها با فاصله‌هایی متوسط از هم قرار بگیرند و فضای پوچی را اشغال کنند که در آن می‌توانست بردار دیگری قرار بگیرد ظرفیت شبکه کاهش یافته است.

به طور تجربه روی نمودار مشاهده می‌کنیم که ۲۰ بردار به بعد شکستی در دقت نمودار داشته‌ایم و شیب کاهش میانگین دقت افزایش یافته‌است. پس، با توجه به $N=100$ ، می‌توان گفت تا ۲۰ درصد N می‌توان برای این نوع شبکه ظرفیت در نظر گرفت.

سوال ۳ - شبکه‌ها پفیلد

الف) شبکه‌ی موردنظر طراحی شده‌است و الگوریتم موردنظر پیاده‌سازی شده‌است. خروجی زیر نشان‌دهنده عملکرد شبکه به ازای ورودی دادن این دو عدد است.

```
I remember =))
```

```
Result:
```

```
#####
```

```
#####
```

```
##-----##
```

```
##-----##
```

```
##-----##
```

```
##-----##
```

```
#####
```

```
#####
```

```
I remember =))
```

```
Result:
```

```
--###---
```

```
--###---
```

```
---##---
```

```
---##---
```

```
---##---
```

```
---##---
```

```
---##---
```

```
---##---
```

شکل ۷- خروجی شبکه‌ی هاپفیلد به ازای ورودی صفر و یک

ب) برای دو عدد نویزی به اندازه‌ی ۳۰ درصد اضافه شد. به این معنی که ۳۰ درصد خانه‌هایشان به صورت تصادفی تغییر علامت (رنگ) دادند. عملکرد شبکه بر روی این ورودی‌های نویزی موفق بود. ورودی نویزی و خروجی‌های شبکه به ازای این ورودی در زیر نمایش داده شده‌اند.


```

-#####-
##-#-##-
##---##-
-##-#-##
-#---##-
##--#-##
--#-##-#
#-#####

```

```

|||||||
vvvvvvv
I remember =))
Result:
#####
#####
##----##
##----##
##----##
##----##
#####
#####

```

شکل ۸- خروجی شبکه به ازای صفر با ۳۰ درصد نویز

```

---#-#---
--#-----
--#-#---#
-##-##-#
-----##--
-----#---
-----#---
-----#---
-####---

```

```

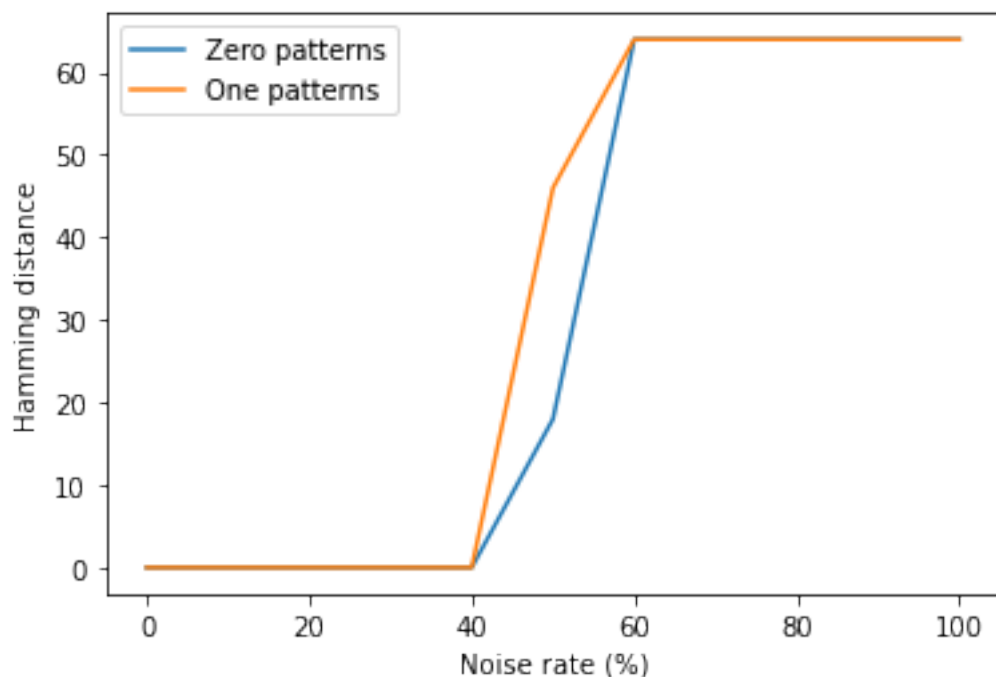
|||||
vvvvvvvv
I remember =))
Result:
--###---
--###---
---##---
---##---
---##---
---##---
---##---
---##---
---##---
---##---

```

شکل ۹- خروجی شبکه به ازای یک با ۳۰ درصد نویز

ج) در زیر نمودار فاصله‌ی hamming خروجی شبکه به ازای ورودی با نویزهای مختلف بر حسب میزان نویز رسم شده است. همانطور که در این نمودار مشاهده می‌کند هر چه میزان نویز در یک الگو بالاتر می‌رود از الگوی اصلی دور می‌شویم و گویی از ناحیه جذب موردنظر در حافظه نیز فاصله می‌گیریم. به عبارت دیگر، فاصله hamming ورودی نویزی از ورودی اصلی بیشتر می‌شود و به همین دلیل، عملکرد شبکه با اختلال روبرو می‌شود.

حال اگر ورودی‌های شبکه وابستگی داشته باشند و فاصله hamming آن‌ها کم باشد، ممکن است با یکدیگر اشتباه گرفته شوند و این مشکل بدتری است که شبکه الگوی اشتباهی را تداعی کند.



شکل ۱۰- نمودار فاصله‌ی hamming خروجی شبکه از خروجی هدف به ازای نویزهای مختلف

همانطور که دیده می‌شود از درصد خطایی بالاتر (۶۰ درصد) خطای شبکه ۱۰۰ درصد می‌شود و کاملاً اشتباه تشخیص می‌دهد.

سوال ۴ - شبکه BAM

الف) شبکه‌ی موردنظر طراحی شده‌است و الگوریتم موردنظر پیاده‌سازی شده‌است. حروف A، B و C به آن داده شدند و ماتریس وزن‌های زیر به دست آمد.

$$\begin{bmatrix} [1. & -1. & 3.] \\ [-3. & -1. & -1.] \\ [1. & 3. & -1.] \\ [-3. & -1. & -1.] \\ [3. & 1. & 1.] \\ [-1. & -3. & 1.] \\ [-3. & -1. & -1.] \\ [-1. & -3. & 1.] \\ [1. & -1. & -1.] \\ [-3. & -1. & -1.] \\ [3. & 1. & 1.] \\ [-1. & -3. & 1.] \\ [-1. & -3. & 1.] \\ [-1. & 1. & 1.] \\ [-1. & 1. & -3.] \end{bmatrix}$$

شکل ۱۱- ماتریس وزن‌های شبکه به ازای ورودی A، B و C

ب) به شبکه‌ی موردنظر حروف A، B و C و خروجی‌های آن‌ها داده‌شد که در زیر نتیجه پیش‌بینی شبکه را مشاهده می‌کنید که به درستی تشخیص داده‌شده‌است.

I remember =))

-#-

#-#

###

#-#

#-#

[-1 -1 -1]

I remember =))

##-

#-#

##-

#-#

##-

[-1 -1 1]

I remember =))

-##

#--

#--

#--

-##

[-1 1 -1]

شکل ۱۲- خروجی شبکه به ازای ورودی دادن کاراکترها (x)

```
I remember =))
```

```
-#-
```

```
#-#
```

```
###
```

```
#-#
```

```
#-#
```

```
[-1 -1 -1]
```

```
*****
```

```
I remember =))
```

```
##-
```

```
#-#
```

```
##-
```

```
#-#
```

```
##-
```

```
[-1 -1 1]
```

```
*****
```

```
I remember =))
```

```
-##
```

```
#--
```

```
#--
```

```
#--
```

```
-##
```

```
[-1 1 -1]
```

```
*****
```

شکل ۱۳ - خروجی شبکه به ازای ورودی دادن خروجی کاراکترها (y)

ج) به هر کدام از ورودی‌ها ۱۰ و ۴۰ درصد نویز اعمال شد و برای هر کاراکتر ۱۰۰ بار این کار تکرار شد. تعداد دفعات درست تشخیص داده شدن هر در زیر آمده است.

Ten percent noise accuracy:

A: 100 - B: 100 - C: 100

Fourthy percent noise accuracy:

A: 43 - B: 33 - C: 44

شکل ۱۴ - تعداد درست تشخیص داده شدن هر کاراکتر در ۱۰۰ بار اجرا

همانطور که مشاهده می‌کنید با اعمال نویز ۴۰ درصدی تشخیص شبکه مختل می‌شود و دلیل این موضوع فاصله گرفتن کاراکتر از حالت اصلی خوب و نزدیک تر شدن آن به فضای پوچی ماتریس وزن‌های شبکه می‌باشد.

(د) بله. این ورودی به درستی A تشخیص داده شده است. چرا که دو عدد ۱- در جایگاه دوم و سوم y ویژگی منحصر به فرد A نسبت به B و C است و شبکه در این مورد نشان داده است که نسبت به این missing مقاوم است. خروجی شبکه در زیر قابل مشاهده است.

I remember =))

-#-

#-#

###

#-#

#-#

[-1 -1 -1]

شکل ۱۵- خروجی شبکه به ازای ورودی y با داده missing

(ه) مابقی حروف نیز به حروف قبلی اضافه شده اند و در فرآیند آموزش شبکه مورد استفاده قرار گرفته اند و ماتریس وزن ها را ساخته اند. ماتریس وزن ها به صورت زیر به دست آمده است.

```
[ [ 2. -2. 6. ]
  [-2. -2. -2. ]
  [ 6. 2. -2. ]
  [ 0. 0. 0. ]
  [ 0. 0. 0. ]
  [-4. 0. 4. ]
  [ 0. 0. 0. ]
  [ 2. -6. 2. ]
  [ 0. 4. 0. ]
  [ 0. 0. 0. ]
  [ 0. 0. 0. ]
  [-2. 2. 2. ]
  [ 0. -4. 4. ]
  [-2. 2. -2. ]
  [ 2. 2. -6. ] ]
```

شکل ۱۶- ماتریس وزن ها با استفاده از تمامی حروف

(و) خیر. زیرا شبکه در قسمت y کلا ۸ حالت دارد. زیاد شدن کاراکترهای باعث نزدیکی آنها و کاهش فضای پوچی ماتریس وزن های شبکه می شود. این اتفاق عملکرد شبکه را مختل می کند و شبکه نمی تواند زوج X و y درست (یا حتی معتبری) را تشخیص دهد. همانطور که در تصویر زیر می بینید، عملکرد شبکه در این حالت با اختلال روبرو شده است و توانایی شناسایی کاراکتر مورد نظر (A) را نداشته است.

See old pattern... No memory :(

##-

--#

-#-

--#

#--

[-1. -1. 1.]

شکل ۱۷- عدم توانایی شبکه در تشخیص کاراکتر A بعد از زیادتیر شدن الگوها

حداکثر تعداد الگویی که می‌توان ذخیره کرد، سه عدد است چراکه شبکه با ۴ الگو نیز تست شد ولی نتوانست الگوها را به درستی تشخیص بدهد. خروجی به ورودی موردنظر نزدیک تر شد ولی نتوانست به درستی و کام زوج ورودی و خروجی را تشخیص دهد.

```
[ [ 0. 0. 4.]
[-4. 0. 0.]
[ 2. 2. -2.]
[-4. 0. 0.]
[ 4. 0. 0.]
[-2. -2. 2.]
[-4. 0. 0.]
[ 0. -4. 0.]
[ 0. 0. 0.]
[-4. 0. 0.]
[ 4. 0. 0.]
[-2. -2. 2.]
[-2. -2. 2.]
[-2. 2. 2.]
[ 0. 0. -4.]
```

See old pattern... No memory :(

-#-

##-

##-

-#-

##-

[-1. -1. -1.]

See old pattern... No memory :(

##-

-#-

##-

-#-

##-

[-1. -1. 1.]

See old pattern... No memory :(

-##

#--

#--

#--

-##

شکل ۱۸- عدم توانایی شبکه در تشخیص کاراکترها با چهار الگو

ز) فاصله‌ی hamming بین حروف به صورت زیر به دست آمد.

Hamming distances:

A - B: 4
A - C: 7
A - D: 4
A - E: 6
A - F: 6
A - G: 5
A - H: 3
B - C: 7
B - D: 2
B - E: 4
B - F: 4
B - G: 7
B - H: 5
C - D: 7
C - E: 3
C - F: 5
C - G: 2
C - H: 8
D - E: 6
D - F: 6
D - G: 5
D - H: 5
E - F: 2
E - G: 5
E - H: 5
F - G: 7
F - H: 5
G - H: 6

شکل ۱۸ - فاصله‌ی hamming بین کاراکترها

با توجه به فاصله‌های hamming بین حروف، حرف H با احتمال زیاد با موفقیت تشخیص داده می‌شود. دلیل این موضوع آن است که حداقل فاصله H با کاراکتر دیگری ۳ است که در کنار A بیشترین حداقل فاصله را دارند و به همین دلیل، احتمال اشتباه گرفته شدن کمتری دارند. از طرفی میانگین فاصله‌ی hamming کاراکتر h با دیگر کاراکترها از A بیشتر است. به همین دلیل، به نظر H شانس بیشتری برای تشخیص درست دارد.