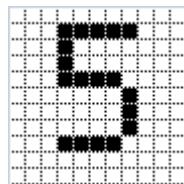


## گزارش پروژه درس شبکه عصبی - شناسایی اعداد با استفاده از شبکه عصبی هاپفیلد

مهندی محمدی

در ابتدا به بررسی ماتریس اعداد می‌پردازیم. نشانه اعداد ۰ الی ۹ در ماتریس‌هایی با ابعاد  $11 \times 11$  طراحی می‌گردند. به عنوان مثال نشانه عدد ۵ به شکل زیر در یک ماتریس با ابعاد ذکر شده، در نرم‌افزار Paint طراحی گردیده است :



لازم به ذکر است که نقاط توپر ارزش "۰" و نقاط خالی ارزش "۱" دارند و تصاویر ایجاد شده به فرمت Monochrome Bitmap ذخیره می‌شوند. در ادامه کدهای مطلب آورده شده است و بعد از آن به تحلیل کدها می‌پردازیم.

```
clc;
close all;
clear all;

output=zeros(11,11);

N0=imread('0.bmp');
N0=(N0==0)+(N0==1)*-1;

N1=imread('1.bmp');
N1=(N1==0)+(N1==1)*-1;

N2=imread('2.bmp');
N2=(N2==0)+(N2==1)*-1;

N3=imread('3.bmp');
N3=(N3==0)+(N3==1)*-1;
```

```

N4=~imread('4.bmp');
N4=(N4==0)+(N4==1)*-1;

N5=~imread('5.bmp');
N5=(N5==0)+(N5==1)*-1;

N6=~imread('6.bmp');
N6=(N6==0)+(N6==1)*-1;

N7=~imread('7.bmp');
N7=(N7==0)+(N7==1)*-1;

N8=~imread('8.bmp');
N8=(N8==0)+(N8==1)*-1;

N9=~imread('9.bmp');
N9=(N9==0)+(N9==1)*-1;

T=cat(2,N0(:),N1(:),N2(:),N3(:),N4(:),N5(:),N6(:),
      N7(:),N8(:),N9(:));
net = newhop(T);

pause on;
for d=0.01:0.02:0.45
    Noisy_N0=imnoise(N0,'salt & pepper',d);
    Noisy_N1=imnoise(N1,'salt & pepper',d);
    Noisy_N2=imnoise(N2,'salt & pepper',d);
    Noisy_N3=imnoise(N3,'salt & pepper',d);
    Noisy_N4=imnoise(N4,'salt & pepper',d);
    Noisy_N5=imnoise(N5,'salt & pepper',d);
    Noisy_N6=imnoise(N6,'salt & pepper',d);
    Noisy_N7=imnoise(N7,'salt & pepper',d);
    Noisy_N8=imnoise(N8,'salt & pepper',d);
    Noisy_N9=imnoise(N9,'salt & pepper',d);
    for i=0:9
        Ai =
        {eval(strcat('Noisy_N',int2str(i),':'))};
        [Y,~,~] = sim(net,{1 59},[],Ai);

```

```

Y=cell2mat(Y);
N=size(Y,2);
figure;
output(:)=Ai{1};
subplot(6,10,1);
imshow(output);
title(strcat('d=',num2str(d)));
for j=1:N
    output(:)=Y(:,j);
    subplot(6,10,j+1);
    imshow(output);
    title(strcat('#',int2str(j)));
    if Y(:,j) == N0(:)
        break;
    elseif Y(:,j) == N1(:)
        break
    elseif Y(:,j) == N2(:)
        break
    elseif Y(:,j) == N3(:)
        break
    elseif Y(:,j) == N4(:)
        break
    elseif Y(:,j) == N5(:)
        break
    elseif Y(:,j) == N6(:)
        break
    elseif Y(:,j) == N7(:)
        break
    elseif Y(:,j) == N8(:)
        break
    elseif Y(:,j) == N9(:)
        break
    else
        end
    end
end
display('press any key to continue...');
pause;

```

```

    close all;
end

```

ابدا با دستور `clc` محیط command window پاک می‌گردد، سپس با دستور `all` تمامی تصاویر ایجادشده توسط برنامه‌های اجراشده قبلی، بسته می‌شوند و بعد از آن با استفاده از دستور `clear all` تمامی متغیرهای ایجادشده توسط برنامه‌های اجراشده قبلی، از حافظه پاک می‌شوند. با استفاده از دستور `output=zeros(11,11)` یک ماتریس  $11 \times 11$  با درایه‌های ۰ ایجاد می‌شود. حال توسط دستور ترکیبی `N=imread('.\*.bmp');` تصویر مربوط به نشانه ۰ خوانده شده و پس از اعمال نقیض منطقی در ماتریس  $N$  ذخیره می‌شود. در قدم بعدی با استفاده از دستور ترکیبی `(N==0)+(N==1)=N` مقادیر ۱- بجای ۰ و مقادیر ۰ بجای ۱ قرار می‌گیرند. روند کلی این دستورات به شکل گرافیکی در ذیل ترسیم گشته است :

۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱
۱	۱	۱	۱	۱	۰	۰	۰	۱	۱	۱	۱
۱	۱	۱	۱	۰	۱	۱	۱	۰	۱	۱	۱
۱	۱	۱	۱	۰	۱	۱	۱	۰	۱	۱	۱
۱	۱	۱	۱	۰	۱	۱	۱	۰	۱	۱	۱
۱	۱	۱	۱	۰	۱	۱	۱	۰	۱	۱	۱
۱	۱	۱	۱	۰	۱	۱	۱	۰	۱	۱	۱
۱	۱	۱	۱	۰	۱	۱	۱	۰	۱	۱	۱
۱	۱	۱	۱	۱	۰	۰	۰	۱	۱	۱	۱
۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱
۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱
۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱

تصویر اصلی `.bmp`

.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	1	1	1	.	.	.	.
.	.	.	.	1	.	.	.	1	.	.	.
.	.	.	.	1	.	.	.	1	.	.	.
.	.	.	.	1	.	.	.	1	.	.	.
.	.	.	.	1	.	.	.	1	.	.	.
.	.	.	.	1	.	.	.	1	.	.	.
.	.	.	.	1	.	.	.	1	.	.	.
.	.	.	.	1	.	.	.	1	.	.	.
.	.	.	.	1	.	.	.	1	.	.	.
.	.	.	.	1	.	.	.	1	.	.	.
.	.	.	.	1	.	.	.	1	.	.	.

ماتریس  $N$  پس از اجرای دستور ترکیبی؛  $N = \sim\text{imread}('..\'.bmp')$

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	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	-1	1	1	1
1	1	1	-1	1	1	1	1	-1	1	1	1

ماتریس  $N$  پس از اعمال دستور ترکیبی؛  $(N == 0) + (N == 1)^*$

این روند برای تمامی نشانه‌های اعداد ۱ تا ۹ نیز تکرار می‌گردد. در ادامه با استفاده از دستور ترکیبی زیر ماتریس داده‌های آموزشی شبکه هاپفیلد آماده می‌شود :

```
T=cat(۲,N۰(:),N۱(:),N۲(:),N۳(:),N۴(:),N۵(:),N۶(:),N۷(:),N۸(:),N۹(:));
```

در این دستور ترکیبی، هر ماتریس تبدیل یافته از تصویر اصلی، به یک بردار  $121 \times 1$  تبدیل می‌گردد و سپس این  $10$  بردار با هم ترکیب شده و یک ماتریس  $121 \times 10$  تشکیل می‌شود. حال این ماتریس به شکل زیر به شبکه هاپفیلد جهت آموزش داده می‌شود :

```
net = newhop(T);
```

بعد از این دستور به اولین حلقه for برنامه می‌رسیم که درصد اعمال نویز فلفل - نمکی به هر یک از تصاویر اعداد ۰ تا ۹ را مشخص می‌نماید. حلقه دوم برنامه مشخص می‌کند که تصویر نویزی کدامیک از اعداد، جهت تست به شبکه هاپفیلد داده شود. با دستورات متوالی زیر تصویر نویزی مورد نظر به شبکه هاپفیلد جهت تست داده می‌شود :

```
Ai = {eval(strcat('Noisy_N',int2str(i),':'))};
```

```
[Y,~,~] = sim(net,{1 59},[],Ai);
```

با استفاده از مجموعه دستورات اول، تصویر نویزی عدد مورد نظر، به صورت یک بردار  $121 \times 1$  در بردار Ai ذخیره می‌شود و با استفاده از دستور دوم بردار Ai به شبکه عصبی هاپفیلد جهت تست اعمال می‌گردد. در این مثال تعداد تکرار جهت همگرا شدن ۵۹ می‌باشد. دستورات بعدی جهت نمایش تصویر هر مرحله از تکرار در زیر تصویرهای مربوطه، با استفاده از ماتریس output و تشخیص همگرا شدن به یکی از بردارهای ورودی آموزشی بکار می‌رود. به عنوان مثال اگر عدد  $0$  نویزی به شبکه هاپفیلد جهت تست اعمال شود و بردار  $\{y\}_9$  حاصل از تکرار نهم دستور sim، با بردار ورودی عدد  $0$  بدون نویز یکی باشد، برنامه تشخیص می‌دهد که بردار  $0$  نویزی در واقع عدد  $0$  بدون نویز است. لازم به ذکر است که اگر نویز اعمال شده به تصاویر کم باشد، نرخ تشخیص صحیح زیاد می‌شود،

در صورتی که اگر نویز اعمال شده به تصاویر زیاد باشد، نرخ تشخیص صحیح کم می‌شود. همچنین یک تصویر نویز بالا ممکن است به تصویر دیگری غیر از تصویر واقعی همگرا شود و یا اصلاً به بردار خاصی همگرا نشود و ۵۹ تکرار انجام شود و نهایتاً عددی تشخیص داده نشود.

بنابراین در کاربردهای واقعی مانند تشخیص پلاک خودرو، بهتر است با استفاده از روش‌های پردازش تصاویر دیجیتال شامل الگوریتم‌های حذف نویز و الگوریتم‌های مورفولوژی تا حد امکان نویز تصاویر و یا درهم‌رفتگی اعداد (ناشی از مسائل محیطی) اصلاح شود و پس از استخراج هر عدد از پس‌زمینه اصلی، اعداد شناسایی شوند؛ با این روش نرخ تشخیص صحیح افزایش پیدا می‌کند.