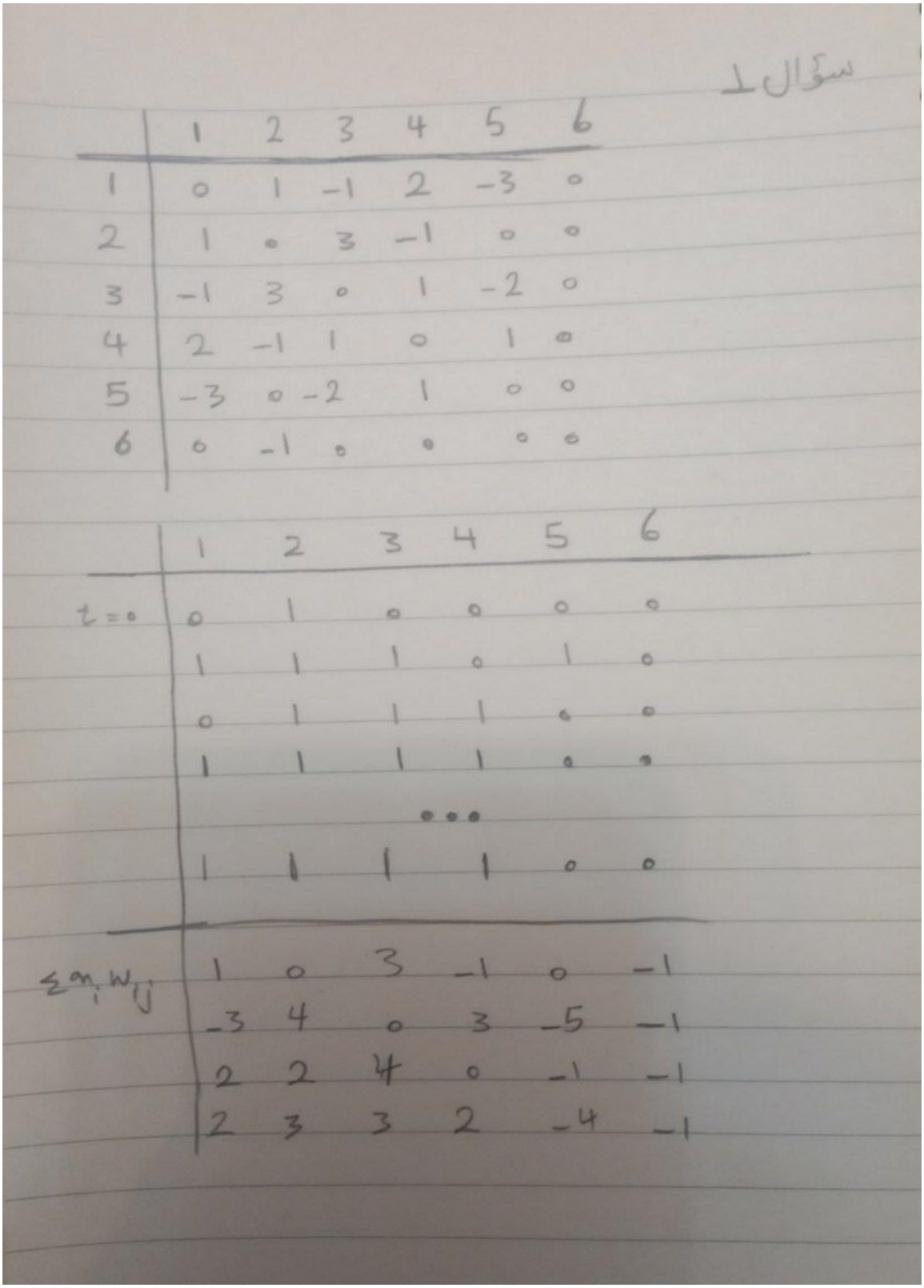
گزارش تمرین ۳
هوش محاسباتی

یاسمین مدنی
۹۷۵۳۲۲۶۵

سوال۱



سؤال ۱

	1	2	3	4	5	6
1	0	1	-1	2	-3	0
2	1	0	3	-1	0	0
3	-1	3	0	1	-2	0
4	2	-1	1	0	1	0
5	-3	0	-2	1	0	0
6	0	-1	0	0	0	0

	1	2	3	4	5	6
$t=0$	0	1	0	0	0	0
	1	1	1	0	1	0
	0	1	1	1	0	0
	1	1	1	1	0	0
			...			
	1	1	1	1	0	0
$\sum x_i w_{ij}$	1	0	3	-1	0	-1
	-3	4	0	3	-5	-1
	2	2	4	0	-1	-1
	2	3	3	2	-4	-1

سوال ۲:

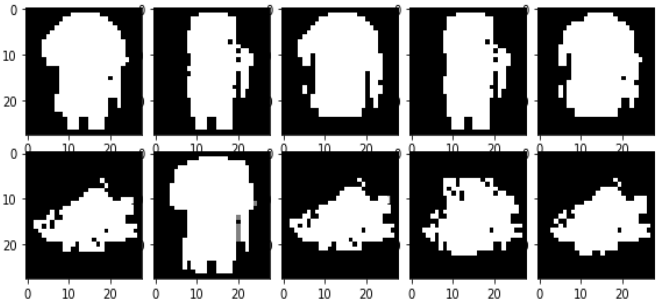
در تابع accمقدار هر پیکسل تصویر اصلی و نهایی را مقایسه و براساس آن دقت را گزارش می کنیم.

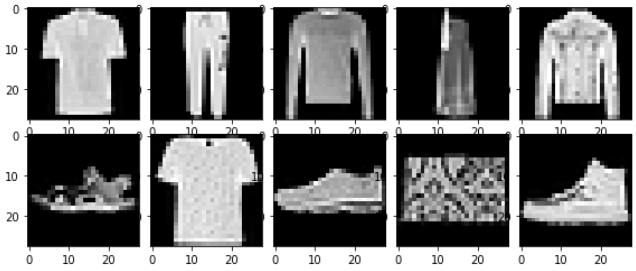
در کلاس Hopfield یک شبکه با تعداد نورون های دلخواه و ماتریس وزن ها را داریم که تابع create_w_matrixاین ماتریس را مقدار دهی می کند می دانیم عناصر قطر اصلی ۰ و سایردرایه ها متقارن و برابر حاصل ضرب خانه iام و jآم تصویر اند به شرط آنکه تصویر را به صورت درایه های 1و 1- نمایش داده باشیم.

که این کار و فلت سازی تصویر در تابع mat2vecصورت میپذیرد.به این صورت که درایه های کمتر از میانگین تصویر با -۱ و سایر با ۱ جایگزین می شوند.

در تابع Predict عمل اصلی بازسازی تصویر به کمک محاسبه انرژی هر نقطه صورت می گیرد که برای این کار تابع energy به صورت جداگانه تعریف شده است.

ئس آماده سازی دیتا ها و انتخاب دیتاها از کلاس های مختلف شبکه را ابتدا ترین و سپس تست میکنیم. در شکل های زیر دیتای ترین و نمونه ای از بازسازی داده های نویزی قابل مشاهده اند.





جدول زیر خلاصه دقت خواسته شد در سوال برای هر یک از شبکه های زیر و داده های نویزی را نمایش می دهد.

Network size / Noise	15	28	32
10%	0.8800000000000001	0.8807397959183673	0.88154296875
30%	0.8706666666666667	0.8807397959183673	0.88212890625
60%	0.8773333333333333	0.874234693877551	0.88486328125

کد های مربوطه در نت بوک کامنت گذاری شده اند.

از جدول می توان نتیجه گرفت افزایش تعداد نورون حالات پایدار و درنتیجه دقت شبکه را با وجود نوبز بالا افزایش می دهد.

سوال ۳:

مسئله فروشنده دوره گرد یک چالش شناخته شده در علوم کامپیوتر است: این مسئله شامل یافتن کوتاه ترین مسیر ممکن است که همه شهرها را در یک نقشه معین فقط یک بار طی می کند. اگرچه توضیح ساده آن، این مشکل در واقع NP-Complete است. این نشان می دهد که دشواری حل آن با تعداد شهرها به سرعت افزایش می یابد و ما در واقع راه حل کلی برای حل مشکل نمی دانیم. به همین دلیل، ما در حال حاضر در نظر داریم که هر روشی که بتواند راه‌حلی کمتر از بهینه را پیدا کند، عموماً به اندازه کافی خوب است (ما نمی‌توانیم بررسی کنیم که آیا راه‌حلی که برگردانده می‌شود، در اکثر مواقع بهینه است یا خیر).

از آنجا که با یک سوال unsupervised طرف هستیم mlp راه حل مناسبی برای سوال نیست.

این مورد با شبکه های عصبی recurrent قابل حل است که از جمله آنها می توان به هاپفیلد اشاره کرد.با توجه به این مقاله این سوال با شبکه هاپفیلد قابلیت حل شدن را داراست.

از آنجا که somداده ها با ویژگی ها شبیه هم را در نزدیکی هم قرار می دهد می توان از این روش برای مینیمم کردن فاصله ها استفاده کرد.برای حل آن، می‌توانیم تغییراتی در تکنیک نقشه خودسازماندهی (SOM)نیز اعمال کنیم.به طور مشابه somاین سوال با RBFنیز قابل حل است .(لینک مقاله)

برای استفاده از این شبکه برای حل TSP، مفهوم اصلی درک نحوه تغییر تابع همسایگی است. اگر به جای یک شبکه، یک آرایه دایره ای از نورون ها را اعلام کنیم، هر گره فقط از نورون های جلو و پشت خود آگاه خواهد بود. یعنی شباهت درونی فقط در یک بعد کار خواهد کرد. با انجام این اصلاح جزئی، نقشه خودسازماندهی مانند یک حلقه الاستیک رفتار می‌کند و به شهرها نزدیک‌تر می‌شود اما به لطف عملکرد همسایگی تلاش می‌کند تا محیط آن را به حداقل برساند.

اگرچه این اصلاح ایده اصلی پشت این تکنیک است، اما آنطور که هست کار نخواهد کرد: الگوریتم به سختی همگرا می شود. برای اطمینان از همگرایی آن، می‌توانیم نرخ یادگیری α را برای کنترل

کاوش و بهره‌برداری از الگوریتم در نظر بگیریم. برای به دست آوردن اکتشاف بالا در ابتدا، و پس از آن بهره برداری بالا در اجرا، ما باید هم در تابع همسایگی و هم در نرخ یادگیری یک کاهش را لحاظ کنیم. کاهش سرعت یادگیری، جابجایی تهاجمی کمتری از نورون‌های اطراف مدل را تضمین می‌کند و تضعیف همسایگی منجر به بهره‌برداری متوسط‌تر از مینیمم محلی هر بخش از مدل می‌شود.

پس از خواندن و نرمالایز دیتا عملیات شبکه را روی مختصات شهر ها پیاده کرده و در epochها مختلف تصویر شبکه را در فولدر دیاگرام ها ذخیره کرده ایم که پیوست فایل تمرین است.

برای مثال ۶ تصویر زیر در حالت های ۱۰۰۰ – ۵۰۰۰-۷۰۰۰-۱۲۰۰۰- ۲۴۰۰۰ و نهایی را نمایش می دهند .(تصاویر بیشتر در فولدر diagrams)