

گزارش تمرین ۴ درس شبکه‌های عصبی

میثم پرویزی

دانشکده مهندسی برق و کامپیوتر

پردیس دانشکده‌های فنی

دانشگاه تهران

است. آموزش را تا جایی ادامه دادیم که تغییر وزن در یک epoch کمتر از ۰,۰۰۰۱ شد.

الف) شعاع مجاورت هر نورون را صفر در نظر گرفتیم.

در این حالت که کد آن در فایل Q1-SOM-Part1.ipynb موجود است. شعاع مجاورت را صفر در نظر گرفتیم. به این ترتیب ۲۰ نورون برنده به صورت زیر به دست آمدند.

Number of Patterns	Cluster
53	128
53	73
49	408
47	510
46	478
43	273
42	554
38	375
37	214
30	122
29	48
17	259
10	318
1	531
1	234
1	486
1	525
1	362
1	334

چکیده- شبکه های عصبی رقابتی شاخه‌ای از شبکه‌های عصبی هستند که در حوزه‌ی یادگیری بدون نظارت مورد استفاده قرار می‌گیرند. الگوریتم‌های مختلفی برای شبکه‌های عصبی رقابتی پیشنهاد شده است که در این تمرین دو مورد از آن‌ها را پیاده‌سازی کرده و مورد بررسی قرار خواهیم داد. کلمات کلیدی- شبکه‌های عصبی رقابتی، SOM، Mexican Hat، یادگیری بدون نظارت.

۱. مقدمه

در این تمرین ابتدا الگوریتم SOM را به زبان پیاده‌سازی کردیم و سپس از آن برای طبقه‌بندی دیتاست مربوط به تعدادی از حروف دستنویس الفبای انگلیسی استفاده کردیم. در بخش بعدی از الگوریتم Mexican Hat برای پیدا کرد ماکزیمم اعداد موجود در یک مجموعه استفاده کردیم.

۲. طبقه‌بندی با الگوریتم SOM

در این بخش ابتدا الگوریتم SOM را به زبان پایتون پیاده‌سازی کردیم. سپس دیتاست Alphabets.npy را که شامل ۵۰۰ تصویر ۲۸*۲۸ دست خط از حروف انگلیسی Z, X, U, S, P, O, I, C می‌باشد را به آن داده و سپس تنظیمات زیر را آزمایش کردیم.

برای این کار ۶۲۵ نورون در نظر گرفتیم که در دو قسمت اول به صورت یک بعدی و در قسمت سوم به صورت دو بعدی کنار هم قرار گرفته‌اند. پس از اجرای الگوریتم، ۲۰ نورونی که تصاویر بیشتری را پوشش دادند را به عنوان برنده انتخاب کردیم. در تمام این سه قسمت نرخ یادگیری را در ابتدا ۰,۳ در نظر گرفتیم که در هر epoch با ضریب ۰,۰۰۰۱ کوچک می‌شود. همچنین برای وزن‌دهی اولیه اعداد تصادفی بین ۰ تا ۰,۱ را انتخاب کردیم که در هر سه قسمت مقدار یکسانی دارد و در فایل initial_weight.npy ذخیره شده

Number of Patterns	Cluster
12	0
12	250
11	6
10	54
10	153
10	3
9	280
9	251
9	259
9	303
9	110
8	130
8	225
7	128
7	281
7	255
7	235
7	231
7	2
7	180

به این ترتیب فقط ۱۷۵ الگو در ۲۰ دسته‌ی برنده قرار گرفتند. همچنین برای رسیدن به شرط توقف الگوریتم، ۳ epoch طی شد.

(د) بررسی نتایج سه قسمت فوق.

در قسمت الف که همسایگی خطی با شعاع صفر بود تعداد نورون‌هایی که در هر مرحله آپدیت می‌شدند فقط یکی بود و همین باعث می‌شد تا تا تفاوت الگوها با همدیگر محسوس نباشد اما در قسمت ب چون شعاع همسایگی ۱ شد، تعداد نورون‌های همسایه به ۳ عدد افزایش یافتند و تا حدی توانستیم تفاوت الگوها از یکدیگر را تشخیص دهیم و در نتیجه تعداد دسته‌ها افزایش یافته و در نتیجه تعداد الگوهای جاگرفته در ۲۰ نورون برتر کاهش یافت. در قسمت ج نیز به همین ترتیب تعداد نورون‌های همسایه به ۹ عدد افزایش یافت و تعداد الگوهای جاگرفته در ۲۰ دسته برتر باز هم کاهش یافت.

۳. الگوریتم Mexican Hat

در این بخش به کمک الگوریتم Mexican Hat مقدار ماکزیمم آرایه‌ی زیر را پیدا کردیم:

[0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1]

0	206
---	-----

به این ترتیب تمام ۵۰۰ الگو در ۲۰ دسته‌ی برنده قرار گرفتند. همچنین برای رسیدن به شرط توقف الگوریتم، ۳ epoch طی شد.

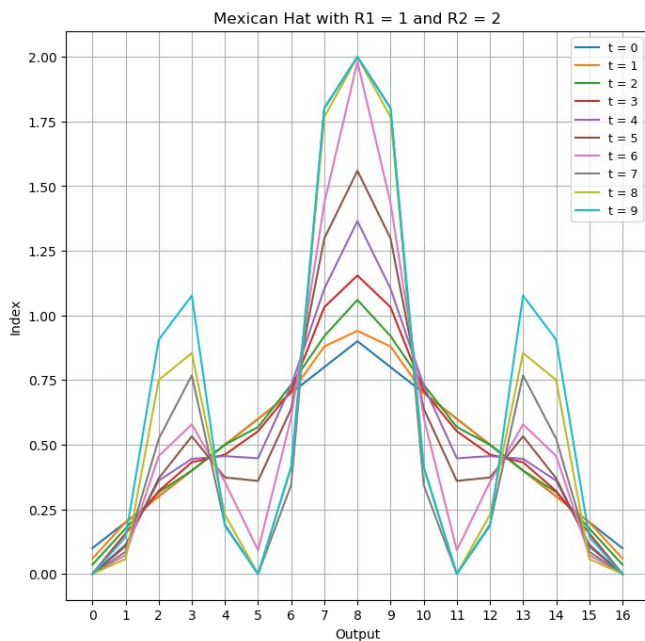
ب) همسایگی نورون‌ها را خطی فرض کرده و با $R=1$ وزن‌ها را بروز کردیم. در این حالت که کد آن در فایل Q1-SOM-Part2.ipynb موجود است. شعاع مجاورت را ۱ در نظر گرفتیم. به این ترتیب ۲۰ نورون برنده به صورت زیر به دست آمدند.

Number of Patterns	Cluster
27	123
25	134
22	130
19	136
19	131
19	378
18	126
17	377
17	125
16	376
16	121
15	135
15	137
14	129
14	375
12	138
12	476
12	477
12	118
12	139

به این ترتیب فقط ۳۳۳ الگو در ۲۰ دسته‌ی برنده قرار گرفتند. همچنین برای رسیدن به شرط توقف الگوریتم، ۳ epoch طی شد.

ج) نورون‌ها را روی یک شبکه‌ی 25×25 فرض کرده و با $R=1$ وزن‌ها را بروز کردیم.

در این حالت که کد آن در فایل Q1-SOM-Part3.ipynb موجود است. در این قسمت همسایگی نورون‌ها را به صورت دوبعدی در نظر گرفته و شعاع مجاورت را ۱ در نظر گرفتیم. به این ترتیب ۲۰ نورون برنده به صورت زیر به دست آمدند.



(ج) مقایسه نتایج قسمت‌های فوق.

در قسمت الف چون $R2=\infty$ در نظر گرفته شده است تقریباً تمام درایه‌ها باید در هر epoch بروز شوند و همین با توجه به مقادیر $C1$ و $C2$ باعث شد تا خروجی‌ها صفر شوند اما در قسمت ب چون مقادیر $R1$ و $R2$ منطقی‌تر انتخاب شده‌اند نتایج بهتری به دست دادند.

پس از پیاده‌سازی الگوریتم در زبان پایتون تنظیمات زیر را روی آن بررسی کردیم.

الف) $R1=0$ و $R2=\infty$

در این حالت سایر پارامترهای شبکه را به صورت زیر در نظر گرفتیم:

T_MAX = 10

R1 = 0

R2 = 100

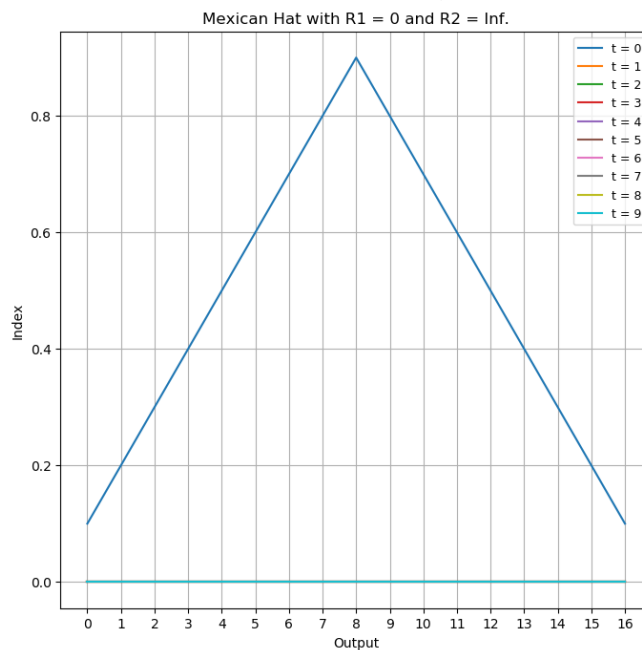
C1 = 0.6

C2 = -0.4

X_MAX = 2

لازم به ذکر است در اینجا چون طول بردار ورودی ۱۷ است، مقدار $R2=100$ به نوعی نقش بی‌نهایت را بازی می‌کند.

پس از اجرای الگوریتم نمودار زیر به دست آمد که نتیجه‌ی مطلوبی نیست چرا که از $t=1$ به بعد تمام آرایه صفر شده است و داده‌ها از بین رفته‌اند.



ب) $R1=1$ و $R2=2$

در این حالت سایر پارامترهای شبکه را به صورت زیر در نظر گرفتیم:

T_MAX = 10

R1 = 1

R2 = 2

C1 = 0.6

C2 = -0.4

X_MAX = 2

پس از اجرای الگوریتم نمودار زیر به دست آمد و همانطور که ملاحظه می‌شود با افزایش t خروجی شبکه رفته رفته آشکار شده است: