

به نام خدا



دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده برق و کامپیوتر



شبکه‌های عصبی و یادگیری عمیق

تمرین شماره ۳

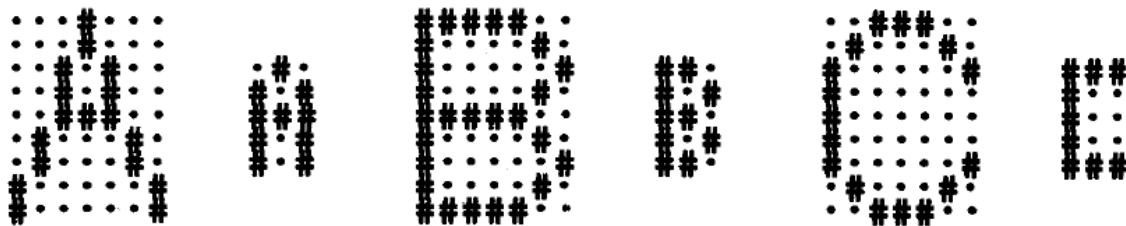
اردیبهشت ۹۹

فهرست سوالات

- سوال ۱ – Character Recognition using Hebbian Learning Rule ۳
- سوال ۲ – Storage Capacity in an Auto-associative Net ۴
- سوال ۳ – Iterative Auto-associative Net ۵
- سوال ۴ – Recurrent Hetero-Associative Network ۶

سوال ۱ – Character Recognition using Hebbian Learning Rule

طبق آنچه در درس آموخته شد، یکی از کاربردهای قانون یادگیری هب شناسایی حروف است. در این سوال می‌خواهیم شبکه‌ای را طراحی کنیم که با دادن حروف $7*9$ بتواند خروجی $3*5$ آنها را تولید کند. به منظور تعریف این حروف در Python، به نقطه‌ها عدد -1 و به $\#$ ها عدد $+1$ اختصاص دهید.



(۱) آیا این شبکه قادر خواهد بود تا تمامی ورودی‌ها را به خروجی مطلوب برساند؟

در ادامه، می‌خواهیم میزان مقاومت این شبکه را در برابر نویز (جایگزین کردن اعداد $+1$ و -1) و از دست رفتن اطلاعات (جایگزین کردن صفر به جای $+1$ و -1) بسنجیم.

(۲) در گام اول، 20% و 40% درصد نویز بصورت تصادفی به شبکه اضافه کنید. در چند درصد مواقع شبکه موفق خواهد شد تا خروجی را به درستی تشخیص دهد؟ (راهنمایی: برای محاسبه کد خود را چندین بار اجرا کرده و درصد دفعاتی را که شبکه توانسته است به درستی به هر سه خروجی برسد محاسبه کنید).

(۳) در گام بعد، 20% و 40% درصد از اطلاعات تصویر را از بین ببرید (جای مقادیر $+1$ و -1 ، صفر قرار دهید). در چند درصد مواقع شبکه موفق خواهد شد تا خروجی را به درستی تشخیص دهد؟

(۴) مقاومت شبکه در برابر کدامیک بیشتر است؟ نویز یا از دست رفتن اطلاعات؟ نشان دهید.

(۵) میزان حداکثر مقاومت شبکه در برابر نویز را بیان کنید (راهنمایی: با درصدهای مختلف آزمایشات را آن قدر تکرار کنید تا به نتیجه قابل اعتماد برسید).

سوال ۲- Storage Capacity in an Auto-associative Net

یک نکته مهم در کار با *associative memory neural networks* تعداد بردارهای قابل ذخیره‌سازی است. در این سوال می‌خواهیم به این موضوع بپردازیم.

(۱) با استفاده از *Modified Hebbian Learning Method* بردار $S = [1, 1, 1, -1]$ را ذخیره کنید.

(۲) حال، بردار **دوم** دلخواهی را به گونه‌ای ذخیره کنید که شبکه قادر به بازیابی آن باشد.

(۳) بیان کنید تحت چه شرایطی شبکه نخواهد توانست بردار جدید را ذخیره کند؟ مثال بزنید.

(۴) همین کار را تا جایی انجام دهید که بردار جدیدی را نتوان در شبکه ذخیره کرد. تا چند بردار می‌توان در شبکه ذخیره نمود به صورتیکه شبکه قادر به بازیابی آن‌ها باشد؟

(۵) قانونی کلی برای تعداد بردارهای قابل ذخیره در یک شبکه با n ورودی را بیان کنید.

جواب بخش‌های متفاوت را به زبان Python پیاده‌سازی کرده و در گزارش خود بطور کامل راجع به جواب هر بخش توضیح دهید.

سوال ۳ – Iterative Auto-associative Net

در شبکه‌های خود انجمنی داده‌های ورودی و خروجی یکسان هستند. عملکرد این شبکه‌ها از روی توانایی آن‌ها برای بازتولید یک الگوی ذخیره شده از روی ورودی نویزی سنجیده می‌شود.

(۱) با استفاده از *Modified Hebbian Learning Method* بردار $S = [1, 1, 1, -1]$ را ذخیره کنید.

(۲) عملکرد شبکه را در حالتیکه اطلاعات سه تا از چهار مقدار ورودی از بین رفته و صفر شده باشد، بررسی کنید (هر جایگشت و حالت ممکن بررسی شود). نتایج خود را گزارش کنید.

(۳) عملکرد شبکه را در حالتیکه سه اشتباه و جابه‌جایی در بردار ورودی رخ داده باشد، بررسی کنید (هر جایگشت و حالت ممکن بررسی شود). نتایج خود را گزارش کنید.

(۴) سه مرحله بالا را به ترتیب با استفاده از شبکه *Discrete Hopfield Net* انجام دهید.

(۵) تعداد الگوهای قابل ذخیره سازی با شبکه *Discrete Hopfield Net* را محاسبه کنید.

(۶) با ذکر دلیل گزارش کنید که کدام یک از دو شبکه‌ای که به کار بردید و بررسی کردید، برای این کار مناسب‌تر عمل می‌کند.

سوال ۴ - Recurrent Hetero-Associative Network

نکته قابل توجه در شبکه های مربوط به *Bidirectional Associative Memory* این است که از دوطرف دارای خاصیت *mapping* هستند.

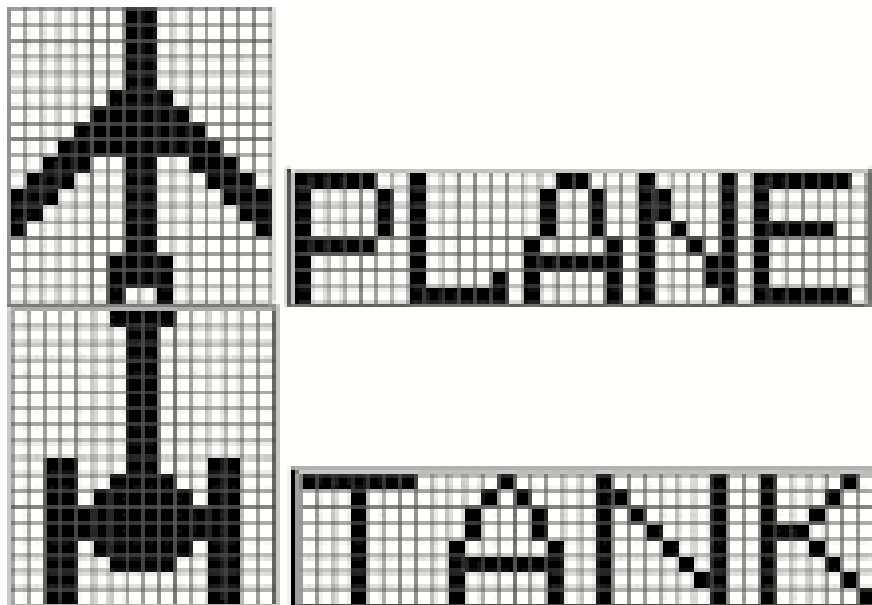
(۱) از دو الگوی زیر در شبکه BAM استفاده کنید تا *mapping* دوسویه مناسب ایجاد شود، سپس

برای صحت عملکرد شبکه، خروجی را از دوطرف تولید کنید و نمایش دهید (رنگ مشکی را با عدد +۱ و سفید را -۱ کدگذاری کنید).

(۲) اغتشاش را طوری وارد الگوهای تصویری کنید که فاصله *hamming* آن با تصویر اصلی حدود

۳۰ درصد باشد، خروجی را نمایش داده (که در حالت بدون اغتشاش برابر با نام تصویر است) و عملکرد شبکه را مقایسه و تحلیل کنید.

توجه: منظور از نمایش تصویر یا خروجی در هر دو بخش سوال، یک تصویر سیاه و سفید متناسب با ماتریس داده است.



تصویر ۱ - دو الگوی ارائه شده برای ذخیره سازی توسط الگوریتم BAM

نکات:

- مهلت تحویل این تمرین ۲۳ اردیبهشت است.
- گزارش را در قالب تهیه شده که روی صفحه درس در CECM بارگذاری شده، بنویسید.
- گزارش، فایل کدها و سایر ضmann مورد نیاز را با فرمت زیر در سامانه مدیریت دروس بارگذاری نمائید.
HW3_[Lastname]_[StudentNumber].zip
- گزارش شما در فرآیند تصحیح از اهمیت ویژه‌ای برخوردار است. لطفاً تمامی نکات و فرض‌هایی که برای پیاده‌سازی‌ها و محاسبات خود در نظر می‌گیرید را در گزارش ذکر کنید.
- در گزارش خود برای تصاویر زیرنویس و برای جداول هم بالانویس اضافه کنید.
- الزامی به ارائه توضیح جزئیات کد در گزارش نیست. اما باید نتایج بدست آمده را گزارش و تحلیل کنید.
- دستیاران آموزشی ملزم به اجرا کردن کدهای شما نیستند. بنابراین هرگونه نتیجه و یا تحلیلی که در شرح سوال از شما خواسته شده است را به طور واضح و کامل در گزارش بیاورید. در صورت عدم رعایت این مورد، بدیهی است که از نمره تمرین کسر می‌شود.
- در صورت مشاهده تقلب نمرات تمامی افراد شرکت‌کننده در آن صفر لحاظ می‌شود.
- در انجام همه تکالیف، تنها مجاز به استفاده از زبان برنامه‌نویسی Python هستید.
- استفاده از کدهای آماده برای تمرین‌ها به هیچ وجه مجاز نیست. در صورت مشاهده نمره کسر می‌شود.
- نحوه محاسبه تاخیر به این شکل است: پس از مهلت اصلی ذکر شده، به مدت هفت روز و با کسر جریمه سی درصدی تا ۳۰ اردیبهشت بارگذاری ممکن است و در نهایت، پس از بازه ارسال با تاخیر نمره تمرین صفر خواهد شد.
- در صورت وجود هرگونه ابهام یا مشکل در سوالات، می‌توانید از طریق رایانامه‌های زیر با دستیاران آموزشی، آقای سامان ستوده پیما (سوالات ۱ و ۲ و ۳) و خانم عارفه فرهمندی (سوالات ۳ و ۴) در تماس باشید:

s.sotoudeh@ut.ac.ir

arefe00.farahmandi@gmail.com