



به نام خدا



دانشگاه تهران
دانشکده مهندسی برق و کامپیوتر
شبکه های عصبی و یادگیری عمیق
تمرین سری ۱

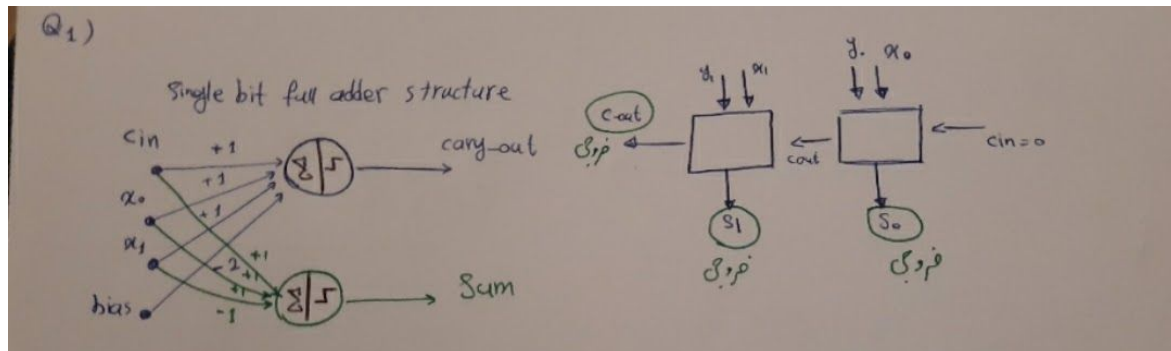
سجاد پاکدامن ساوجی	نام و نام خانوادگی
۸۱۰۱۹۵۵۱۷	شماره دانشجویی
۱۶ اسفند	تاریخ ارسال گزارش

فهرست گزارش سوالات

- 3 سوال 1 – طراحی full-adder با McCulloch-Pitts
- 4 سوال ۲ – به روز رسانی دستی شبکه perceptron
- 5 سوال 3 – Adeline vs Perceptron
- 7 سوال 4 – نحوه اجرای کدها

سوال 1 – طراحی full-adder با McCulloch-Pitts

در این سوال ابتدا یک full adder ساخته شد و مدل ۲ بیتی آن هم با cascade کردن دوتا از این full adder ها بدست می‌آید. شکل full adder ساخته شده با استفاده از نرون مورد نظر در شکل ۱ آمده است. دقت شود که بایاس جزو وزن ها در نظر گرفته شده است و به همین دلیل ورودی نرون ۴ ورودی است. (۲ تا برای بیت های جمع شونده، یکی برای carry که از لایه قبل آمده است و ۱ بعد که همیشه ۱ است و وزن های مربوط به این ورودی نقش بایاس را دارند).



شکل ۱. طراحی شبکه جمع کننده و نحوه اتصال متوالی آن ها

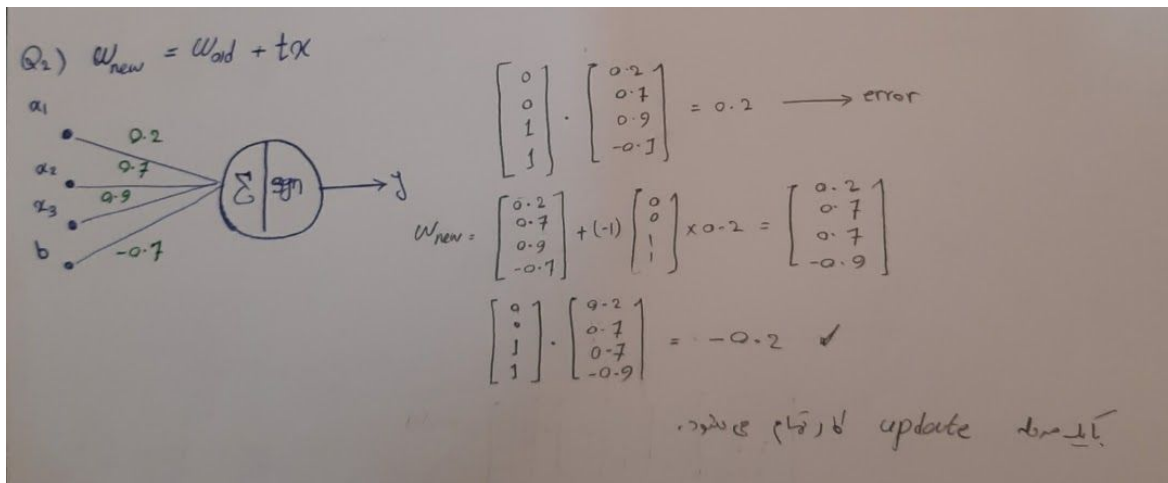
پیاده سازی این شبکه در قسمت ابتدایی نوتیوکی که پیوست شده است وجود دارد و عملکرد صحیح این شبکه در شکل ۲ نشان داده شده است.

[0 0]	+	[0 0]	=	0	[0. 0.]
[0 0]	+	[0 1]	=	0	[0. 1.]
[0 0]	+	[1 0]	=	0	[1. 0.]
[0 0]	+	[1 1]	=	0	[1. 1.]
[0 1]	+	[0 0]	=	0	[0. 1.]
[0 1]	+	[0 1]	=	0	[1. 0.]
[0 1]	+	[1 0]	=	0	[1. 1.]
[0 1]	+	[1 1]	=	1	[0. 0.]
[1 0]	+	[0 0]	=	0	[1. 0.]
[1 0]	+	[0 1]	=	0	[1. 1.]
[1 0]	+	[1 0]	=	1	[0. 0.]
[1 0]	+	[1 1]	=	1	[0. 1.]
[1 1]	+	[0 0]	=	0	[1. 1.]
[1 1]	+	[0 1]	=	1	[0. 0.]
[1 1]	+	[1 0]	=	1	[0. 1.]
[1 1]	+	[1 1]	=	1	[1. 0.]

شکل ۲. عملکرد شبکه جمع کننده بر روی تمامی اعداد ممکن

سوال ۲ - به روز رسانی دستی شبکه perceptron

حل تشریحی (بروز رسانی دستی شبکه یاد شده) در شکل ۳ آمده است و قانون بروز رسانی و نرخ یادگیری نیز ذکر شده است.

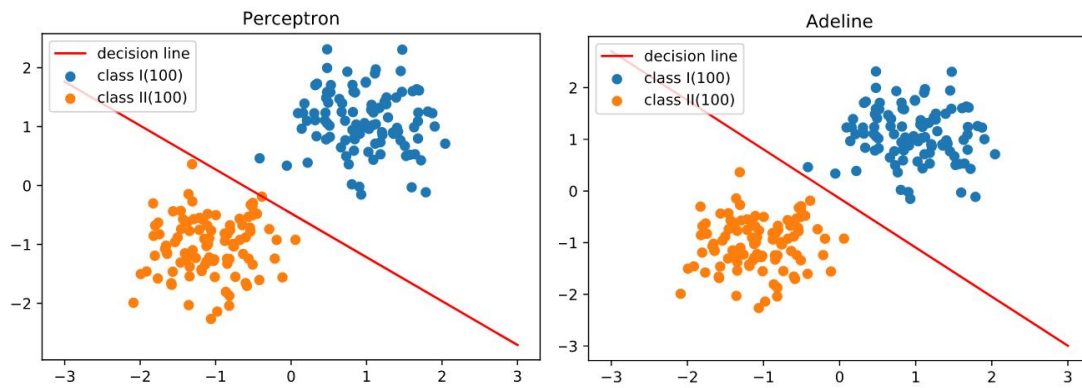


شکل ۳. بروز رسانی دستی شبکه perceptron

سوال 3 – Adeline vs Perceptron

مجموعه داده اول

آ) خطوط جداساز در شکل ۴ نمایش داده شده اند.

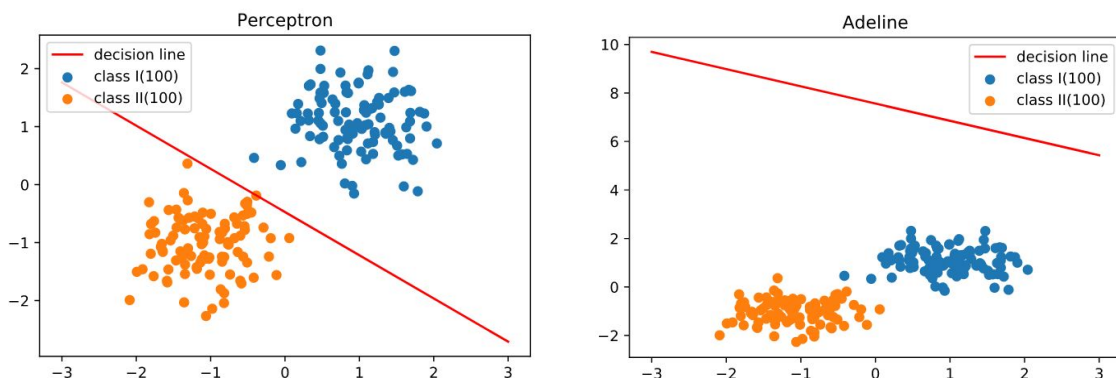


شکل ۴. خطوط تصمیم برای مجموعه داده اول

ب) در مقایسه نتایج بدست آمده این نکته قابل اشاره است که الگوریتم یادگیری perceptron به گونه‌ای است که با پیدا کردن یک جواب متوقف می‌شود اما از آنجایی که الگوریتم یادگیری Adeline بر مننای SSE بین هدف و net است اگر با شرط مناسبی خاتمه پیدا کند در این صورت جدا سازی با مارجین بیشتر (همانند SVM) ایجاد می‌کند. این margin بیشتر به robust شدن دسته بندی کمک می‌کند و شبکه را در برابر خطای واریانس حفظ می‌کند.

ج) در ابتدا $\text{learning rate} = 0.2$ قرار گرفت و مشاهده شد که اگر lr به اندازه ی کافی کوچک نباشد، شبکه Adeline قادر به همگرایی و یافتن یک جواب نخواهد بود. (این مورد در کتاب نیز ذکر شده است) در مقابل شبکه Perceptron یک جواب پیدا می‌کند.

زمان یادگیری در این حالت برای Adeline بی نهایت است و برای Perceptron (همان طور که در کتاب ذکر شده است) کران بالا دارد.



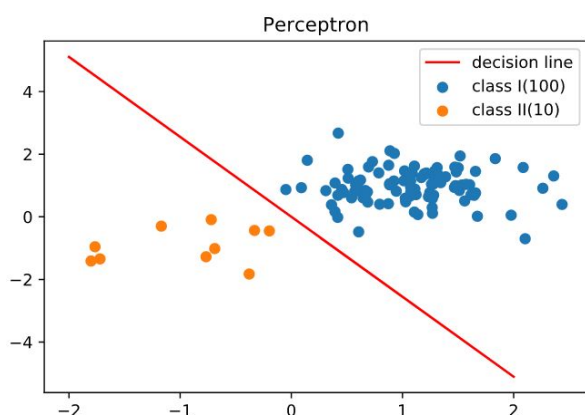
شکل ۵. مقایسه عملکرد دو شبکه برای ترخ یادگیری زیاد

سپس $\text{learning rate} = 0.01$ قرار گرفت و شبکه Adeline نیز همان‌طور که انتظار میرفت جواب مطلوبی پیدا کرد. این جواب در شکل ۴ آورده شده است.

در مورد زمان همگرایی باید به این نکته اشاره کرد که زمان همگرایی در شبکه adeline (و شبکه perceptron) به learning rate و cost threshold بستگی دارد اما در حالت کلی با learning rate یکسان شبه Adeline زمان بیشتری برای آموزش نیاز دارد، زیرا به دنبال بهترین پاسخ است در حالی که شبکه perceptron تنها به دنبال یافتن یک پاسخ است.

مجموعه داده دوم

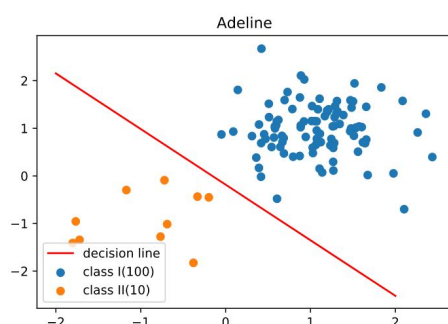
آ) خطوط جداساز در شکل ۶ آمده اند.



شکل ۶. مقایسه عملکرد شبکه‌ها برای تعداد داده‌های غیر همسان

ب) در مقایسه عملکرد و همگرایی شبکه Adeline می‌توان گفت که اگر توزیع‌ها همسان باشند در این صورت Adeline با انتخاب learning rate مناسب همگرا می‌شود. اما در این مثال که تعداد داده‌ها با یکدیگر برابر نیست (اگر از روش‌های تخمین چگالی استفاده کنیم با توجه به این که تعداد داده‌ها بسیار متفاوت هستند، توزیع‌های متفاوتی بدست می‌آید) تضمینی برای همگرایی وجود ندارد.

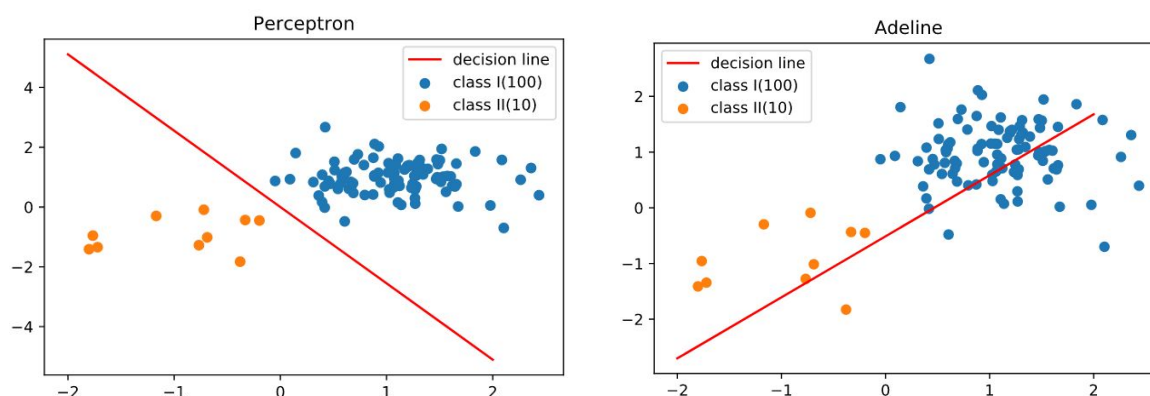
در مرحله بعدی $\text{learning rate} = 0.5$ قرار گرفت و $\text{max_epoch} = 20000$ قرار داده شد و شکل ۷ نشان می‌دهد که با انتخاب هاپیر پارامترهای مناسب شبکه Adeline نیز برای این مسئله همگرا می‌شود.



شکل ۷. همگرایی شبکه Adeline برای داده‌های غیر هم توزیع

ج) در ابتدا $\text{learning rate} = 1$ قرار داده شد. در این حالت همان‌طور که انتظار می‌رفت، شبکه Adeline به علت بزرگ بودن learning rate نتوانست hyper plane مناسب را پیدا

کند. در حالی که (همان طور که قبلا اشاره شد فرایند یادگیری در perceptron دارای کران بالا است و لزوما همگرا می‌شود اگر داده ها جدا پذیر خطی باشند) در شبکه perceptron به یک پاسخ همگرایی اتفاق افتاد. شکل ۸ عملکرد این دو شبکه را در شرایط توضیح داده شده نشان می‌دهد.



شکل ۸. مقایسه عملکرد شبکه‌ها برای توزیع‌های ناهمسان و نرخ یادگیری بزرگ

در مورد زمان همگرایی باید به این نکته اشاره کرد که زمان همگرایی در شبکه adeline (و شبکه perceptron) به learning rate و cost threshold بستگی دارد اما در حالت کلی با learning rate یکسان شبه Adeline زمان بیشتری برای آموزش نیاز دارد، زیرا به دنبال بهترین پاسخ است در حالی که شبکه perceptron تنها به دنبال یافتن یک پاسخ است.

اجرای کد

تمامی پیاده سازی ها در jupyter notebook ای که پیوست شده است قرار دارد. نتایج نیز در notebook بدون نیاز به اجرا وجود دارند. در صورتی که نیاز به اجرا است، سلول ها باید به ترتیب اجرا شوند.