به نام خدا



دانشگاه صنعتی امیرکبیر

(پلی‌تکنیک تهران)

درس شبکه‌های عصبی

استاد صفابخش

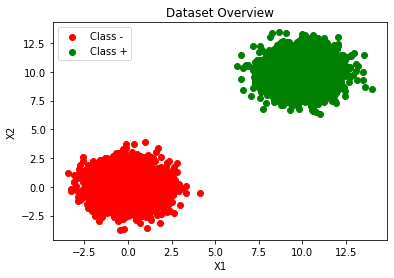
تمرین اول

علیرضا مازوچی

۴۰۰۱۳۱۰۷۵

سوال ۱

برای ایجاد مجموعه‌داده خطی جداپذیر دو توزیع دو بعدی گاوسی با کواریانس یکسان و میانگین‌های و درنظر گرفتم. این مجموعه‌داده با احتمال بسیار بسیار بالا خطی جداپذیر است؛ چراکه فاصله دو مرکز برابر با ده واریانس در هر دو بعد است و در توزیع نرمال 99.9٪ داده‌ها در فاصله‌ای به اندازه چهار انحراف معیار از میانگین قرار گرفته‌اند. مجموعه‌داده می‌تواند چیزی مانند تصویر زیر باشد.



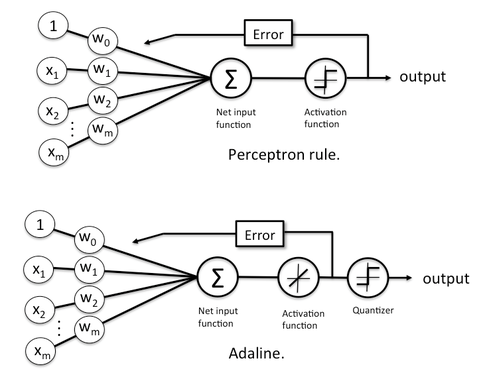
لازم به ذکر است که یک کلاس مثبت با برچسب عددی +۱ و یک کلاس منفی با برچسب عددی ۱- درنظر گرفته شده است.

سوال ۲

باتوجه به آنکه ابعاد مسئله ورودی دوبعدی است، ورودی هر دو واحد پردازشگر هم دوبعدی خواهد بود (). برای مقادیر اولیه وزن‌ها اعداد تصادفی بین 1- تا ۱+ درنظر گرفته‌ام. اما تابع فعالیت دو نورون متفاوت است که جلوتر توضیح داده می‌شود.

تفاوت دو نورون به این شرح است[[1]](#footnote-1)[[2]](#footnote-2):

* پرسپتورون از یک تابع فعالیت استفاده می‌کند که خروجی را باینری (یا به طور مشابه قطبی) می‌کند. این خروجی باینری همان خروجی نهایی نورون است. خطا و درنتیجه بروزرسانی وزن‌ها هم که برای این نورون درنظر گرفته می‌شود برپایه خروجی گسسته خواهد بود. در آدالاین به طور کلاسیک ابتدا یک تابع فعالیت همانی روی ورودی خالص (net input) لحاظ می‌شود که عملا بی‌تاثیر است. خروجی تابع فعال‌سازی این قسمت که خروجی خطی (linear output) نامیده می‌شود برای محاسبه خطا و بروزرسانی وزن‌ها استفاده می‌شود. پس در اینجا تابع خطا برخلاف پرسپتورون پیوسته خواهد بود. نهایتا از یک تابع sign به عنوان تابع آستانه در خروجی خطی نورون استفاده می‌شود تا یک خروجی قطبی (bipolar output) ایجاد شود که این خروجی پیشبینی مدل به ازای داده ورودی است و برای خطا و بروزرسانی وزن استفاده نمی‌شود. خلاصه این تفاوت در تصویر زیر مشخص است:

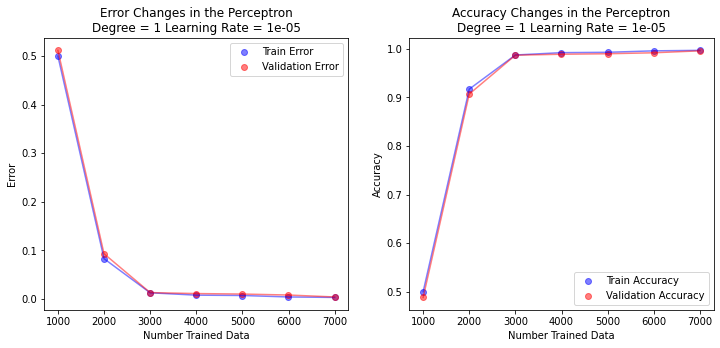


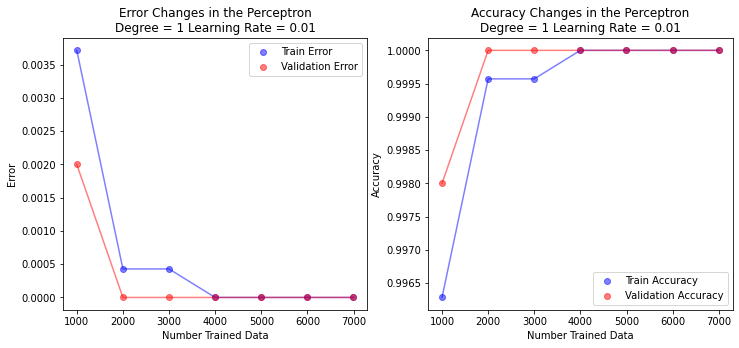
* تفاوت دیگری هم که بین دو نورون وجود دارد برای حالت درجه دو است (بر اساس تدریس کلاس). در نورون پرسپتورون درجه دوم از و استفاده می‌شود. درحالی که در آدالاین مرتبه دوم علاوه بر دو جمله مذکور از هم کمک گرفته می‌شود.

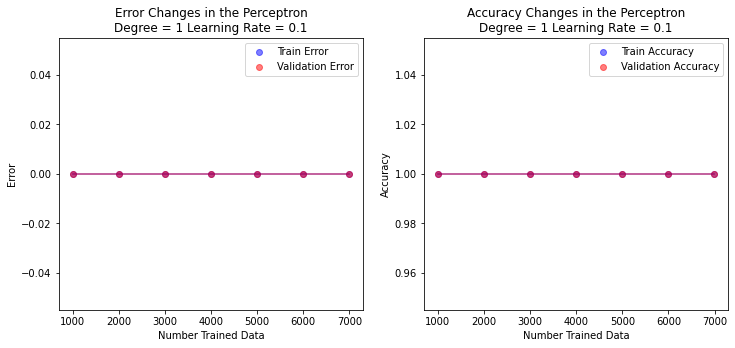
سوال ۳

برای نورون پرسپترون پنج نرخ یادگیری 0.00001، 0.01، 0.1، 1 و 100 را تست کردم که نتایج به ترتیب در ادامه آورده شده است. باتوجه به سرعت همگرایی بالای پرسپترون برای این مجموعه‌داده حتی یک گام هم برای پایین‌ترین نرخ‌های یادگیری کافی به نظر می‌آید. با توجه به نمودار‌ها بهتر است نرخ یادگیری حداقل 0.1 باشد چراکه برای نرخ‌های یادگیری کمتر از آن همگرایی کمی کندتر انجام شده است. لذا برای آزمایش بعدی نرخ یادگیری را روی مقدار 0.1 تنظیم می‌کنم. نهایتا آنکه دقت نهایی تمام نرخ‌های یادگیری در این آزمایش مطلوب است.

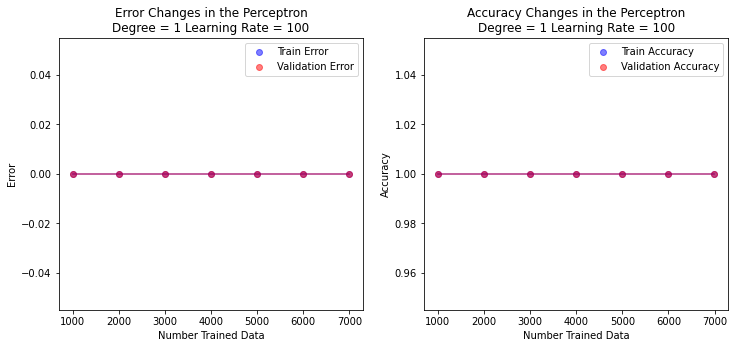
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *نرخ یادگیری* | 0.00001 | 0.01 | 0.1 | 1 | 100 |
| *صحت مجموعه آموزشی* | *0.997* | *1* | *1* | *0.999* | *1* |
| *صحت مجموعه اعتبارسنجی* | *0.996* | *1* | *1* | *1* | *1* |



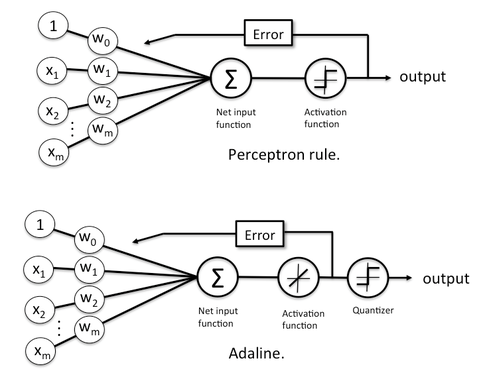








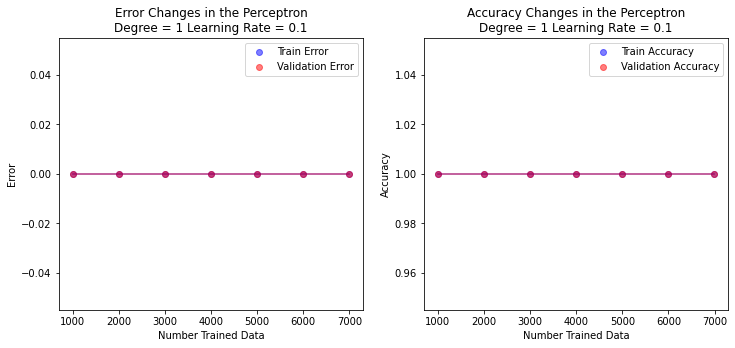
برای تابع فعال‌سازی پرسپترون باید توجه کرد مطابق معماری زیر خروجی تابع فعال‌سازی مستقیما به عنوان خروجی استفاده می‌شود؛ پس حتما باید از تابع فعال‌سازی‌ای کمک گرفت که خروجی قطبی یا باینری داشته باشد.

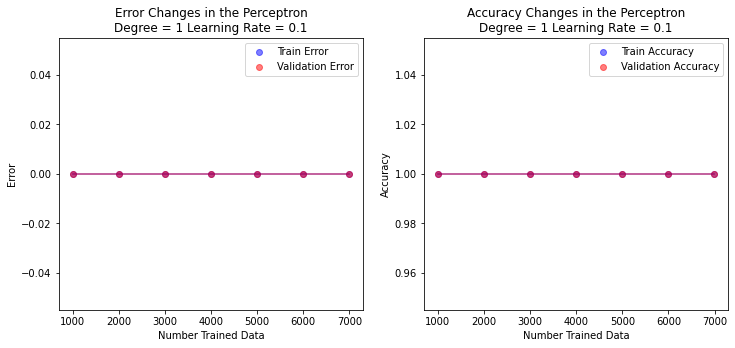


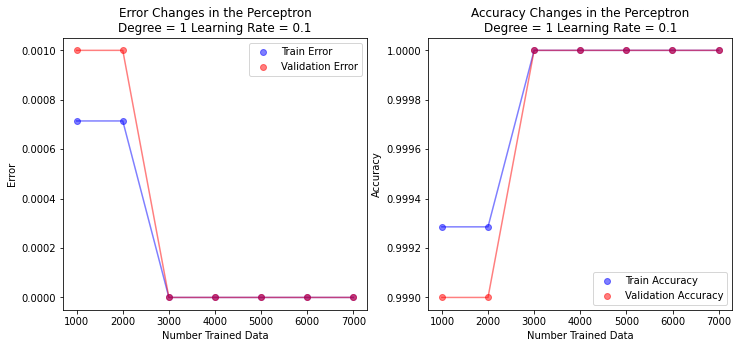
برای این کار از تابع پله زیر با سه مقدار t برابر با -۵، ۰ و +۵ استفاده کردم و نتایج این سه حالت به ترتیب در ادامه آورده شده است. لازم به ذکر است برای بررسی نرخ یادگیری از مقدار t=0 کمک گرفتم.

نتایج عملی تفاوت چشمگیری میان سه مقدار نشان نمی‌دهد. از نظر تئوری هم تفاوت زیادی بین این سه تابع فعال‌سازی وجود ندارد چراکه عملا تفاوت این سه مقدار در بایاس ورودی خالص نمود پیدا می‌کند و تاثیر مهمی در فرآیند آموزش ندارد.

|  |  |  |  |
| --- | --- | --- | --- |
| *تابع فعال‌سازی* | t = -5 | t = 0 | t = +5 |
| *صحت مجموعه آموزشی* | *۱* | *۱* | *۱* |
| *صحت مجموعه اعتبارسنجی* | *۱* | *۱* | *۱* |





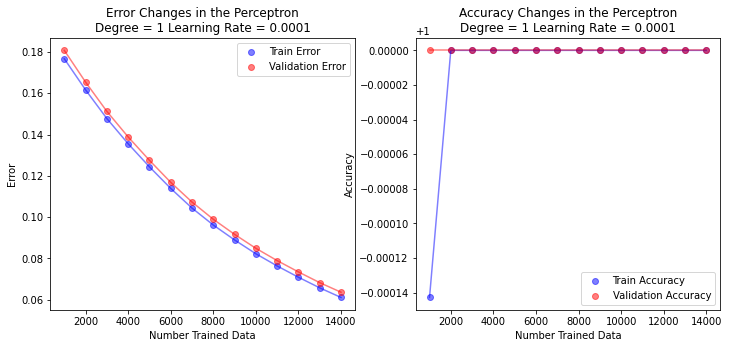


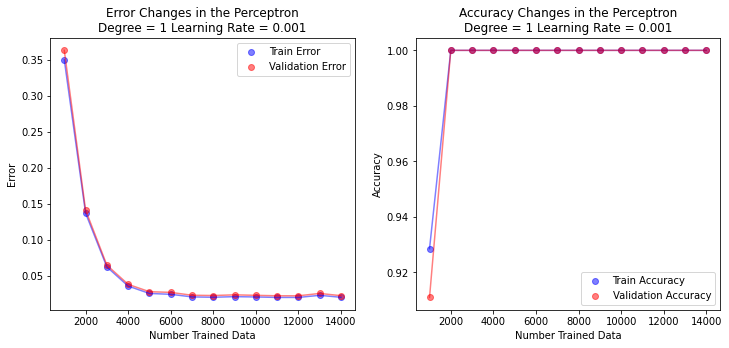
حال نوبت به بررسی نورون آدالاین می‌رسد. برای این نورون استفاده از نرخ‌های یادگیری مشابه با نورون پرسپتورون منجر به خطای زمان اجرا می‌شد و لذا به ناچار از رنج مقادیر کمتر برای نرخ یادگیری استفاده کردم. اعداد ،، ،، به ترتیب مورد بررسی قرار گرفت که نتایج آن به همراه دقت نهایی در ادامه آورده شده است. برای این حالت دو گام آموزش درنظر گرفته شده است. دو نرخ یادگیری و در زمان دو گام آموزش به همگرایی نرسیدند و مشخصا تعداد گام بیشتری برای آموزش نیاز داشتند و دقت نهایی آن‌ها هم به ترتیب 0.44 و 0.51 بوده است. از طرف دیگر برای نرخ یادگیری دقت مدل دائما نوسان داشته است و دقت 0.3 و 1.0 را در زمان کمی همزمان دیده است. این نشان می‌دهد که این نرخ یادگیری برای مدل زیاد است. دو مقدار یادگیری دیگر یعنی و از نظر دقت به سرعت به مقدار کامل همگرا شده‌اند. بین این دو بهترین گزینه به نظر می‌رسد چراکه گزینه دیگر از نظر همگرایی تابع خطا کمی کند بوده است و عملا در دو گام به همگرایی نرسیده است.

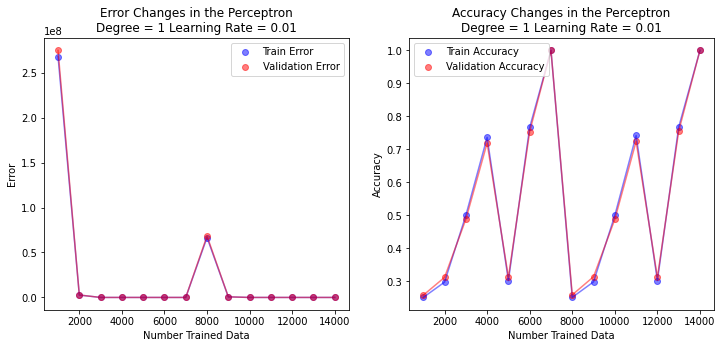
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *نرخ یادگیری* |  |  |  |  |  |
| *صحت مجموعه آموزشی* | *0.436* | *0.505* | *1* | *1* | *0.999* |
| *صحت مجموعه اعتبارسنجی* | *0.441* | *0.513* | *1* | *1* | *1* |





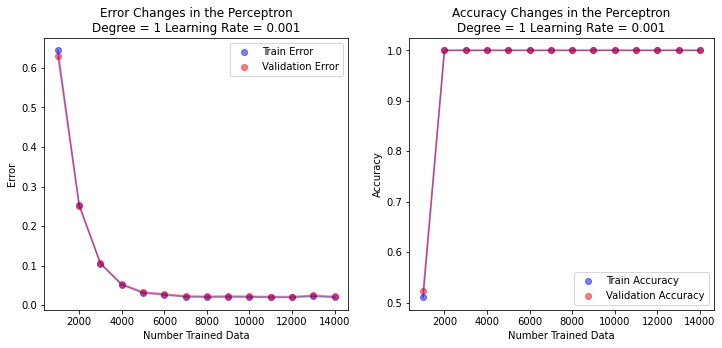


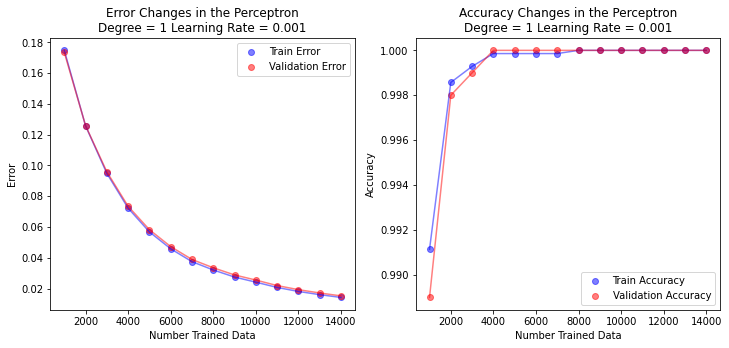


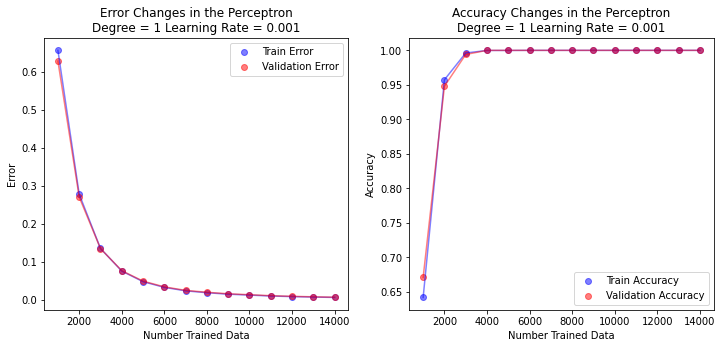


در آزمایش دیگر باید تاثیر تابع فعال‌سازی سنجیده شود. با توجه به نتایج قسمت قبل نرخ یادگیری برای این قسمت برابر با در نظر گرفته شده است. مدل آدالاین به صورت اصلی از یک تابع همانی برای فعال‌سازی استفاده می‌کند. اما می‌توان توابع دیگری را هم استفاده کرد. برای بررسی تاثیر تابع فعال‌سازی علاوه بر تابع همانی، از تابع سیگموید قطبی و تانژانت هایپربولیک استفاده کرده‌ام. لازم به ذکر است برای بررسی تاثیر نرخ یادگیری از تابع همانی کمک گرفته بودم. در ادامه به ترتیب نمودارهای مربوط به تابع همانی، سیگموید و تانژانت هایپربولیک به همراه جدول مقادیر دقت نهایی آورده شده است. با نگاه به نتایج در می‌یابیم هر سه تابع مناسب است ولی تابع سیگموید حداقل در یک بار اجرا و جمع‌آوری نتایج همگرایی کندتری نسبت به دو تابع دیگر داشته است.

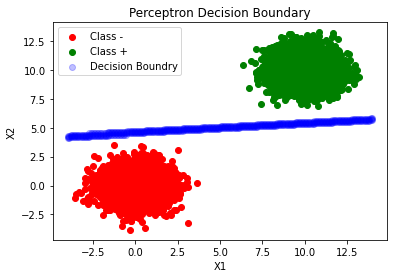
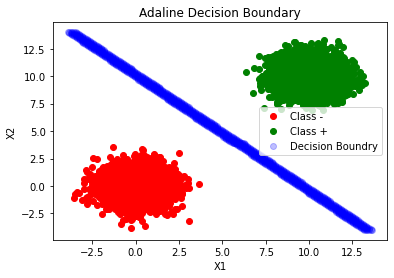
|  |  |  |  |
| --- | --- | --- | --- |
| *تابع فعال‌سازی* | همانی | سیگموید | تانژانت هایپربولیک |
| *صحت مجموعه آموزشی* | *۱* | *۱* | *۱* |
| *صحت مجموعه اعتبارسنجی* | *۱* | *۱* | *۱* |





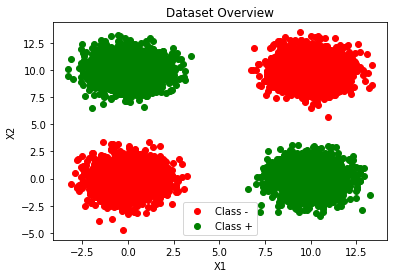


حال که نتایج هر دو نورون ارائه شده است باید بین این دو نورون و برای این مجموعه‌داده یکی از دو نورون را انتخاب کرد. اگر ملاک سرعت همگرایی باشد به نظر می‌آید نورون پرسپترون همگرایی خوبی در تمام حالات داشته است درحالی که برای آدالاین این همگرایی بعضا کمی کندتر بوده است و در برخی از آزمایش‌ها نیاز به حداقل دو گام آموزش احساس می‌شده است. اما در کنار این‌ها باید به یک عامل دیگر هم توجه داشت. در نورون پرسپترون تابع خطا صرفا به درست بودن پیشبینی اهمیت می‌دهد و یک خطای گسسته را ارائه می‌دهد درحالی که در آدالاین خطا پیوسته است و برایش مهم است که مدل با چه اطمینانی (چه فاصله‌ای از خط تصمیم) داده را به یک کلاس تعلق داده است؛ در نتیجه برای نورون پرسپترون وقتی تمام داده‌های آموزشی یک کلاس در یک طرف خط بیافتد و داده‌های دیگر در ظرف دیگر، کار خاتمه بیابد. در حالی که ممکن است این خط مماس به داده‌ها باشد ولی در آدلاین این خط تا حد ممکن از دو کلاس فاصله می‌گیرد. در تصویر زیر خط تصمیم برای هر دو نورون با بهترین تنظیمات نرخ یادگیری آورده شده است. بدیهی است که خروجی آدالاین کیفیت و اطمینان بیشتری دارد. برای همین در مجموع من ترجیح می‌دهم برای این مجموعه‌داده از یک نورون آدالاین استفاده کنم ولی پرسپترون هم بدون مشکل است.



سوال ۴

باتوجه به آنکه در صورت سوال به صورت غیرمستقیم درخواست شده است که مجموعه‌داده از نوع XOR باشد، من چهار توزیع گاوسی در چهار نقطه از فضا با کواریانس در نظر گرفتم. میانگین این چهار توزیع عبارت است از ، ،، و . توزیع اول و چهارم برای کلاس و دو توزیع دیگر به کلاس دیگر متعلق است. مشابه استدلال‌های ارائه‌شده در سوال ۱ می‌توان با اطمینان بسیار بالا مطمئن شد که داده‌ها این چهار توزیع باهم ترکیب نخواهند شد. مجموعه‌داده به شکل زیر به وجود می‌آید. بدیهی است که مجموعه خطی جداپذیر نیست ولی استفاده از نورون با درجات بالا ممکن است مشکل را حل کند.



سوال ۵

سوال ۶

1. <https://sebastianraschka.com/faq/docs/diff-perceptron-adaline-neuralnet.html> [↑](#footnote-ref-1)
2. <https://datascience.stackexchange.com/q/36368/121138> [↑](#footnote-ref-2)