بخش اول.

۱.

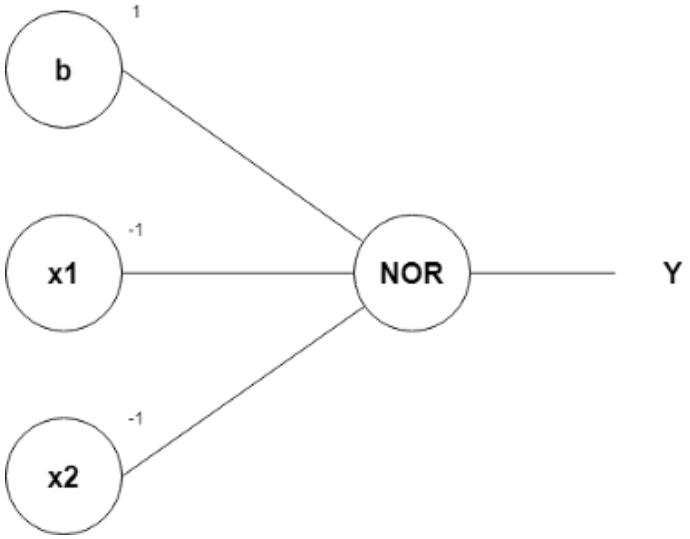
- نقش دندریت‌ها دریافت اطلاعات از نورون‌های دیگر و انتقال آن‌ها به هسته‌ی سلول عصبی برای پردازش است.
نقش هسته‌ی سلول انجام پردازش روی داده‌ی وارد شده به سلول و گاها ایجاد تغییر و تحول در آن است. در بعضی موارد نورون تنها نقش انتقال‌دهنده‌ی اطلاعات را دارد. هسته این کار را با تغییر دادن بالانس الکتریکی میان یون‌های داخل و خارج هسته‌ی سلول انجام می‌دهد که این یون‌ها از طریق کانال‌های پروتئینی و با خوانش ژن‌های مختلف که منجر به اکتیو/دی‌اکتیو شدن کانال‌ها می‌شوند به درون یا بیرون سلول منتقل می‌شوند.
نقش آکسون انتقال داده‌ی ورودی به نورون به نورون(های) بعدی است. آکسون این کار را به کمک غشای میلینی (که همیشه موجود نیست) انجام می‌دهد که این غشا منجر به افزایش سرعت انتقال اطلاعات در نورون و سریع‌تر شدن فعالیت آن می‌شود. در انتهای آکسون که ترمینال آکسون نام دارد، سلول با آزادسازی انتقال دهنده‌های عصبی به سیناپس که منجر به تحریک‌شدن بیش‌تر سلول‌های بعدی می‌شود، سیگنال را به آن‌ها انتقال می‌دهد.
سیناپس فضای بین آکسون یک نورون و دندریت یک نورون دیگر است که در آن با استفاده از تبادلات شیمیایی (با استفاده از neurotransmitterها) و الکتریکی (action potential با استفاده از جریان الکتریکی)، سیگنال‌ها بین نورون‌ها انتقال می‌یابند. (در واقع خارج از خود نورون واقع شده و غشای سلولی نورون بین این دو مرز ایجاد می‌کند.)
- سیگنال‌های بین نورون‌ها از دو جنس الکتریکی و شیمیایی هستند که به طور خلاصه با جابه‌جایی یون‌های محیط نورون از غشا و ترشح neurotransmitterها منتقل می‌شوند.

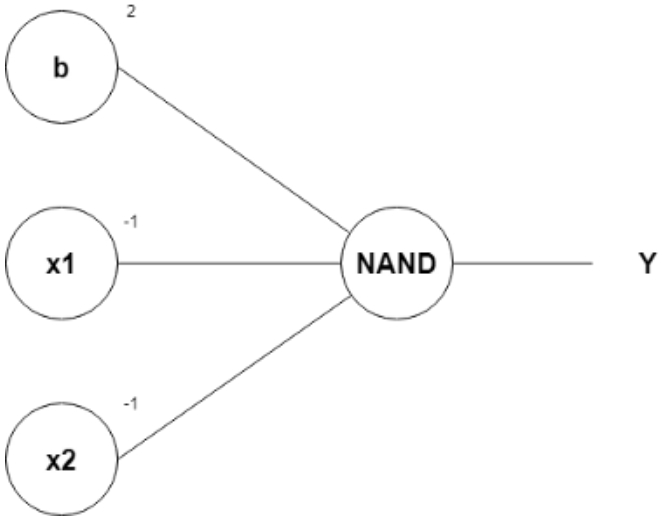
..۲

- مقدار نرخ یادگیری برای مینیمم کردن میزان خطای سیستم (با تلاش برای رسیدن به مشتق صفر) است که بزرگی یا کوچکی آن روی سرعت رسیدن به نقطه‌ی مشتق صفر تاثیر می‌گذارد. (با کم کردن ضرب نرخ یادگیری در مشتق خطای step فعلی) این پارامتر باید به اندازه‌ای بزرگ باشد که تعداد stepهای رسیدن به مشتق صفر خیلی زیاد نشود و به اندازه‌ای کوچک باشد که مداوما بین دو طرف نمودار سهمی جابه‌جا نشویم.
- افزایش عمق شبکه‌ی عصبی با استفاده از تعداد ضریب‌های بیش‌تر و فرمول‌های پیچیده‌تر، می‌تواند به مدل‌کردن مساله‌های پیچیده‌تر کمک کند.
- تعداد بیش‌تر نورون‌های یک لایه‌ی میانی منجر به ایجاد ترکیب‌های بیش‌تری از خروجی‌های نورون‌های قبلی می‌شود و می‌تواند تا حد خوبی به regulation لایه‌ی خروجی کمک کند.
- وقتی خطای مدل روی دیتای test خیلی بیش‌از دیتای training باشد متوجه می‌شویم که مدل نسبت به دیتای training بیش‌براز شده است.
برای جلوگیری از بیش برازش می‌توان چندین کار انجام داد. از جمله حجم افزایش دیتای training، کم‌کردن تعداد نورون‌های لایه‌های میانی، بایاس‌کردن مدل بر علیه وزن‌های بزرگ، اضافه کردن نویز در دیتای training و نادیده گرفتن خروجی بعضی نورون‌ها به صورت رند.وم.

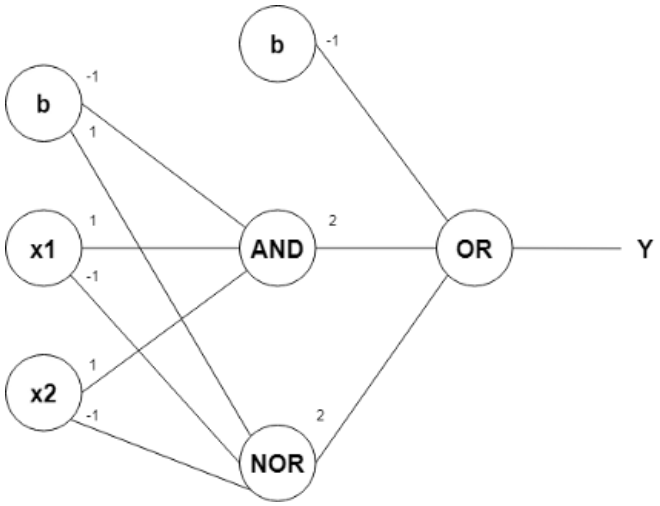
۳.

With Heaviside step function H(n) as the activation function for all models,





XNOR:



-۴

یادگیری نظارت‌شده، یادگیری‌ای است که در آن فرمول یک تابع با استفاده از tupleهای ورودی-خروجی‌هایی که از قبل مشخص‌شده یاد گرفته می‌شود. (یک لیسست ورودی و یک خروجی). الگوریتم‌های نظارت‌شده، مدل درونی خودشان را با استفاده از هر tuple جدید آپدیت می‌کنند با هدف این‌که برای ورودی‌های دیده‌نشده نیز بتوانند تقریب خوبی داشته‌باشند.

یادگیری نظارت‌نشده، یادگیری‌ای است که در آن هدف پیدا کردن پترن‌های از قبل دیده‌نشده در یک dataset بدون هیچ‌گونه اطلاعات جانبی و با حداقل فعالیت مستقیم انسانی است، برخلاف یادگیری نظارت‌شده که از داده‌های لیبل‌خورده توسط انسان‌ها استفاده می‌کند، داده‌های نظارت‌نشده هیچ‌گونه اطلاعات جانبی‌ای درباره‌ی خودشان ندارند. از متدهای اصلی این شیوه، می‌توان به پیداکردن عنصر اصلی و آنالیز کلاستر اشاره کرد که در اولی سعی می‌شود یک المان مشترک از عناصر استخراج شود تا بتوان با دیدن همان المان در دیتاهای جدید، آن را به سرعت پیدا کرد. در دومی سعی می‌شود داده‌ها را به چندین دسته‌ی مختلف بخش‌بندی کرد و خروجی مدل نشان‌گر دسته‌بندی خاص دیتای ورودی آن است.

در یادگیری تقویتی، یک agent نرم‌افزاری را در یک محیط شبیه‌سازی شده قرار می‌دهیم که هدف آن agent، حداکثر کردن میزان rewardای است که می‌تواند از محیط دریافت کند. در این یادگیری، تمرکز روی یافتن یک بالانس میان کشف اطلاعات دیده‌نشده و تحلیل اطلاعات در اختیار است.

بخش دوم.

.۲

با فرض این‌که log(y) = ln(y)، داریم:

$$\frac{dy}{dW} = \frac{d\,S(W.X+b)}{dW} = X * S(W.X+b) * (1 - S(W.X+b)$$

$$\frac{dCost}{dW} = \frac{dCost}{dy} * \frac{dy}{dW} = -\sum_{\forall x} y_t * \frac{1}{y} * \frac{dy}{dW} + (1-y_t) * \frac{-1}{1-y} * \frac{dy}{dW}$$

$$=> \frac{dCost}{dW} = \sum_{\forall x} X(y_t - y)$$

In the same manner, $\frac{dCost}{db} = \sum_{\forall x}(y_t - y)$

.۵

$$\frac{dCost}{dW} = 2 * (y - y_t) * \frac{d\,S(Z.U+b_2)}{dW} = 2 * (y - y_t) * \frac{dy}{dz_0}$$

$$= 2 * (y - y_t) * u_0 * y * (1-y) * \frac{dz_0}{dW}$$

$$\frac{dCost}{dW} = 2 * (y - y_t) * y * (1-y) * u_0 * z_0 * (1 - z_{0)} * X$$

$$In\ the\ same\ manner, \frac{dCost}{db_0} = 2 * (y - y_t) * y * (1-y) * u_0 * z_0 * (1 - z_{0)} * 1$$

$$\frac{dCost}{dU} = 2 * (y - y_t) * y * (1 - y) * Z$$

$$\frac{dCost}{db_2} = 2 * (y - y_t) * y * (1 - y) * 1$$

$$\frac{dCost}{dV} = 2 * (y - y_t) * y * (1 - y) * u_1 * z_1 * (1 - z_{1)} * X$$

$$\frac{dCost}{db_1} = 2 * (y - y_t) * y * (1 - y) * u_1 * z_1 * (1 - z_{1)} * 1$$

می‌دانیم که به طور کلی، شبکه‌هایی که تعداد پرستپرون‌های بیش‌تر و لایه‌های نهفته‌ی بالاتری دارند به مدل کردن مسائل پیچیده تر و غیرخطی تر کمک بهتری می‌کنند.
در مورد مقایسه‌ی این شبکه با شبکه‌ی قبلی، شبکه‌ی اول ۶۰۰۰ *epoch* نیاز داشت تا بتواند بالاخره مدل را به درستی تشخیص دهد و نتیجه‌ی مطلوب حاصل کند، اما در شبکه‌ی دوم اگر تعداد *epoch*ها را زیاد کنیم شبکه نسبت به داده‌های تمرین *overfit* می‌شود و دیگر نتیجه‌ی مطلوب نمی‌دهد.

در مورد شبکه‌ی اول تعداد ۶۰۰۰ *epoch* و شبکه‌ی دوم ۵۰۰ *epoch* برای درست تمرین شدن نیاز دارند و به نظر می‌رسد *learning rate = 0.15* برای هردوی آن‌ها مناسب است.

.۷

به نظر می‌رسد که بهتر است به تعداد دسته‌های *classification* مان در سطح اول پرسپترون داشته‌باشیم که هر کدام مسوول تشخیص یکی از آن‌ها شوند (*grandmother neuron*) و در نتیجه یک نورون نهایی بتواند نتیجه ی نورون‌های قبلی را تفسیر کند.