|  |
| --- |
| اندیس‌ها |
| i | نشان‌دهنده اندیس ورودی نمونه i ام |
| j | نشان‌دهنده ویژگی‌های نمونه  |
| t | نشان‌دهنده شماره لایه  |
| n |  ماکسیمم تعداد نورون‌های هر لایه |
| c | کلاس c ام |

|  |
| --- |
| پارامترها |
|  | مقدار نمونه i ام ویژگی j ام |
|  | تابع فعال سازی گره پنهان |
|  | تابع فعال سازی گره لایه خروجی(از نوع سافت مکس) |
|  |  |
|  | مقدار پیش بینی شده شبکه |
|  | مقدار واقعی طبقه نمونه(پارامتر)C نشان دهنده تعداد کلاس خروجی نمونه i ام و کلاسc ام (ماتریس کلاسهای نمونه خروجی از قبل می‌دانیم) |

|  |
| --- |
| متغیرها |
|  | متغیر صفر و یک نشان دهنده حضور وعدم حضور نورون n ام لایه t ام |
|  | مقدار وارد شده به t امین لایه ورودی نورون n ام قبل از فعال شدن |
|  | وزن بین گره n1از لایه قبلی وn2 از لایه بعدی |
|  | ماتریس وزن بین لایه ورودی به پنهان |
|  | ماتریس وزن بین لایه پنهان و خروجی |
|  | مجموع مقدار خروجی بعد از فعال سازی با مقدار قبل از فعال سازی لایه قبلی |
|  | حاصل مقدار لایه پنهان به خروجی درمقدار لایه فعال شده  |
|  | مقدار نمونه i ام لایه t ام نورون n ام(فعال شده) |

|  |  |  |
| --- | --- | --- |
| تابع هدف: |  |  |
| 3-8 |  |   |
| محدودیت‌ها: |  |  |
| 3-9 |   |  |
| 3-10 |   |  |
| 3-11 |  |  |
| 3-12 |    |  |
| 3-13 |   |  |
| 3-14 |   |  |
|  | \*\*تضمین میکند به ازای هر لایه حداقل یه نورون وجود داشته باشد | ($\sum\_{1}^{n}d \left(t.n\right)\geq 1$ |

ماتریس ورودی با توجه متن به صورت زیر می باشد که مدل تصمیم دارد مقدار ماتریس ورودی را با توجه به لیبل های از پیش تعیین شده y(I,c) پیش بینی کند و مقدار تابع هدف را تخمین بزند

X (I,j)

$$\begin{matrix}-&j1&j2&j3&j4&j5&j6&j7&j8\\i1&0&1&1&1&0&0&0&0\\i2&0&0&1&0&1&1&1&0\\i3&1&0&1&0&0&0&0&1\end{matrix}$$

Y(I,c)

$$\begin{matrix}-&c1&c2&c3\\i1&1&0&0\\i2&0&1&0\\i3&0&0&1\end{matrix}$$

داده ها دریافت میشوند در لایه ها با توجه به وزن ها تحلیل میشوند مقدار خروجی به ما میدهد. مثلا 1000 نمونه متن را میخواند و خودش به انها برچسب مثبت منفی و خنثی میدهد سپس داد های اولیه را به الگوریتم یادگیری ماشین میدهد الگوریتم از روی ان یاد میگیرد که برای بقیه داده ها چه برچسبی بزند

با احتمال 1 نمونه اول به کلاس دوم تعلق دارد و مقداری که برای کلاس اول پیش بینی کرده مقدار بسیار کوچک است

خب... توضیح تک تک اندیس ها پارامتر ها و.

 Iاندیس ورودی نمونه آی ام..یعنی اینکه من یه متن دارم وارد میکنم که تعداد 100 جمله داره در این حالت مقدار ورودی من عدد صد هست.

J نشان دهنده ویژکی های نمونه

یعنی اینگه این صد جمله من از 500 کلمه تشکیل شده اند. و مقدار ویژکی من رو 500 تا معرفی میکنه

Nتعداد نورون های ما هست این مقدار خودمون میتونیم عدد براش تعیین کنیم

|  |
| --- |
| پارامترها |
|  | مقدار نمونه i ام ویژگی j امیعنی اینگه من ده جمله ورودی دارم اولین جمله من میشه نمونه اول و این ده جمله اگز 100 تا کلمه دارن هر کلمه رو یک حرف در نظر داریم میشه نمونه اول به ویزگی اول دوم سوم ...................صدم جمله اول توی هر ویزگی بود انتخاب میشه مقدار یک براش لحاظ میشه نبود صفر |
|  | تابع فعال سازی گره پنهانبرای این تابع سیگموید استفاد کردیم مقدار عدد بین صفر و یک رو میده |
|  | تابع فعال سازی گره لایه خروجی(از نوع سافت مکس) |
|  | مقدار پیش بینی شده شبکه |
|  | مقدار واقعی طبقه نمونه(پارامتر)C نشان دهنده تعداد کلاس خروجی نمونه i ام و کلاسc ام (ماتریس کلاسهای نمونه خروجی از قبل می‌دانیم) |

|  |
| --- |
| متغیرها |
|  | متغیر صفر و یک نشان دهنده حضور وعدم حضور نورون n ام لایه t امحالا این یعنی چی...؟یعنی اینکه من مثلا 50 تا نورون فعال دارم.از این ها چند تا قرار فعال باشه که وقتی داره ضرب میشه و مقدار صفر میده یعنی اون نورون دیگه حذف میشه (غیر فعال میشه) قراره به من ده تا خروجی بده ، همه رو نمیتونه تو همه لایه ها با خودش ببره برای همون ما از این متغیر در محدودیت ها استفاده میکنیم.. حالا این نقدار ورودی که در وزن ضرب شد در این مقدار هم ضرب بشه.نورون ان ام از لایه تی ام حضور داشته باشد یا خیر |
|  | مقدار وارد شده به t امین لایه ورودی نورون n ام قبل از فعال شدن |
|  | وزن بین گره n1از لایه قبلی وn2 از لایه بعدی |
|  | ماتریس وزن بین لایه ورودی به پنهان |
|  | ماتریس وزن بین لایه پنهان و خروجی |
|  | مجموع مقدار خروجی بعد از فعال سازی با مقدار قبل از فعال سازی لایه قبلی |
|  | حاصل مقدار لایه پنهان به خروجی درمقدار لایه فعال شده  |
|  | مقدار نمونه i ام لایه t ام نورون n ام(فعال شده) |

یک شبکه عصبی یک مدل خاص است که ارتباط بین ویژگی‌ها و هدف تبدیل مجموعه داده را با توجه به یک لایه از نورون‌ها به دست می‌آورد

هر نورون مقداری ورودی می‌گیرد، آن‌ها را تبدیل می‌کند و یک خروجی باز می‌گرداند. خروجی یک نورون می‌تواند ورودی نورون‌های لایه بعدی شود و به همین ترتیب، ساختارهای پیچیده‌تر و پیچیده‌تر بسازد. لایه اول، که لایه ورودی نامیده می‌شود، توسط نورون‌هایی ساخته می‌شود که مقادیر ویژگی‌ها را به خود برمی‌گردانند. سپس هر نورون از لایه اول به تمام نورون‌های لایه پنهان متصل می‌شود که مسئول قابلیت‌های یادگیری شبکه است. لایه پنهان می‌تواند توسط چندین لایه پنهان دیگر دنبال شود

 در نهایت، خروجی آخرین لایه پنهان به یک لایه خروجی داده می‌شود که نتیجه را می‌دهد (یعنی مقدار متغیر هدف

در ساده‌ترین شکل آن، یک شبکه عصبی تنها یک لایه پنهان دارد، همانطور که ما می‌توانیم از شکل بالا ببینیم. تعداد نورون‌های لایه ورودی برابر با تعداد ویژگی‌ها است. تعداد نورون‌های لایه خروجی با توجه به متغیر هدف تعریف می‌شود. در اینجا مساله یافتن تعداد صحیح نورون‌ها برای لایه پنهان مطرح می‌شود.

یک عدد کوچک می‌تواند کم‌تر از حد لازم تولید کند، زیرا شبکه ممکن است به درستی یاد نگیرد. یک عدد بالا می‌تواند بیش از حد لازم تولید کند، زیرا شبکه از داده‌های آموزشی چیزهای زیادی یاد می‌گیرد و تعمیم نمی‌یابد. بنابراین، باید تعداد متوسطی از نورون‌ها وجود داشته باشند که آموزش خوب را تضمین کنند.

https://blog.tarjomyar.ir/%DA%86%D9%86%D8%AF-%D9%86%D9%88%D8%B1%D9%88%D9%86-%D8%A8%D8%B1%D8%A7%DB%8C-%DB%8C%DA%A9-%D8%B4%D8%A8%DA%A9%D9%87-%D8%B9%D8%B5%D8%A8%DB%8C-y4gc2wlh7yjm

**شبکه عصبی انتشار برگشتی (Backpropagation):** در این نوع شبکه، خطای خروجی با مقایسه مقدار خروجی با مقدار تجربی یا مقدار مورد نظر محاسبه می گردد. از مقدار و میزان خطا برای تصحیح شبکه و تغییر وزن ها استفاده می شود. این محاسبات از گره خروجی شروع شده و به لایه های پنهان ادامه می یابد. این عمل به ازای تمامی داده های اولیه تکرار می گردد. به هربار اجرای این الگوریتم برای تمام داده ها یک دوره (epoch) گفته می شود. دوره (epoch) آنقدر تکرار می شوند تا مقدار خطا به مقدار ثابتی رسیده و تغییر نکند

softmax را به اختصار بررسی می‌کنیم تا بفهمیم چرا چنین کاربرد رایجی در تابع فعال‌سازی لایه خروجی یافته است. softmax برای روش‌های طبقه‌بندی چند کلاسی مختلف مورد استفاده قرار می‌گیرد و شبکه‌های عصبی تنها یکی از این کاربردهای متداول هستند.

softmax در هسته مرکزی خود امکان سرکوب مقادیر پایین‌تر و برجسته‌سازی مقادیر بالاتر را می‌دهد. ما از softmax به این جهت استفاده می‌کنیم که خروجی تابع به ما امکان می‌دهد که یک توزیع احتمال را روی برآیندهای متفاوت و متنوع بررسی کنیم. بدیهی است که این وضعیت بسیار قدرتمند است، زیرا به ما امکان می‌دهد که بسیاری از مسائل پیچیده یادگیری ماشین و یادگیری عمیق را از طریق طبقه‌بندی چند کلاسی حل کنیم. این امر موجب می‌شود که softmax برای استفاده روی مجموعه داده Iris مناسب باشد. در مورد کاربردهایی مانند طبقه‌بندی دودویی می‌بایست به چیزی مانند sigmoid مراجعه کنیم، زیرا sigmoid یک حالت خاص از softmax است که در آن تعداد کلاس‌ها به دو مورد کاهش یافته است

https://blog.faradars.org/building-your-first-neural-network/