یک شبکه مصنوعی الهام گرفته از نمونه زیستی

راهنمای آشنایی با شبکه‌های عصبی و نحوه پیاده‌سازی آن‌ها

شبکه‌های عمیق عصبی که برخی منابع به آن‌ها شبکه عصبی مصنوعی می‌گویند این توانایی را دارند تا از طریق برنامه‌نویسی و بر مبنای الگوریتم‌های هوشمند به‌گونه‌ای رفتار کنند که عملکردی شبیه به سلول‌های عصبی طبیعی داشته باشند. شبکه‌هایی که در عمل برای حل مشکلات و مسائل بغرنج استفاده می‌شوند. غیر ممکن است درباره هوش مصنوعی صحبت کنیم، اما اشاره‌ای به شبکه‌های عصبی نداشته باشیم، زیرا شبکه‌های عصبی از مهم‌ترین شاخه‌های هوش مصنوعی هستند. البته در این مقاله برای آن‌که صحبت ما کمتر جنبه تئوری داشته باشد نحوه طراحی یک شبکه عصبی ساده را نیز بررسی کرده‌ایم.

در حالی که می‌توانیم تشابهات میان دو واحد پردازشی طبیعی و مصنوعی را بررسی کنیم، اما تشابهات زیاد نیستند، زیرا این دو واحد پردازشی به شیوه‌ای متفاوت اطلاعات را پردازش می‌کنند. اگر با دروس مدار منطقی و سخت‌افزار آشنا باشید به خوبی می‌دانید که چیدمان ترانزیستورها در کامپیوترها بر مبنای یک الگوی ساده است، زیرا تراتزیستورها به شکل زنجیره‌های سریالی چیده می‌شوند، در حالی که سلول‌های عصبی در مغز به پیچیده‌ترین شکل و به صورت موازی قرار گرفته‌اند.

ترانزیستورها به دو یا سه ترانزیستور و به شکلی که گیت منطقی نامیده می‌شود به یکدیگر متصل می‌شوند. در حالی که یک سلول عصبی ممکن است به ده‌ها هزار سلول عصبی مجاور خود متصل باشد. تفاوت دیگری که یک سامانه کامپیوتری با مغز انسان دارد در ارتباط با نحوه ذخیره‌سازی داده‌ها است. کامپیوترها به گونه‌ای طراحی شده‌اند تا حجم بسیار زیادی از اطلاعات که در اغلب موارد مفهوم خاصی ندارند را ذخیره‌سازی کرده و این اطلاعات را دومرتبه و بر مبنای برنامه‌هایی که از آن‌ها استفاده می‌کنند سازمان‌دهی کرده و در دسترس ما قرار دهند. اما در مقابل مغز بر مبنای الگوی یادگیری آهسته و مستمر کار می‌کند. رویکردی که در آن ممکن است به ماه‌ها یا حتا سال‌ها زمان نیاز باشد تا یک مفهوم پیچیده به درستی درک شود. مغز و کامپیوتر به شیوه متفاوتی به اطلاعات ذخیره شده مراجعه می‌کنند. مغز انسان می‌تواند به شکل همزمان اطلاعات را بر مبنای روش‌های جدیدی با یکدیگر ترکیب کند (آهنگ‌های بتهوون یا موتسارت نمونه‌ای روشن از خلاقیت ذهنی هستند). الگوهای اصلی و زیربنایی را تشخیص دهد، ارتباطات جدیدی به وجود آورد و آموخته‌های خود را به شکل جدیدی به کار گیرد، در مقابل کامپیوترها بر مبنای برنامه‌هایی که برای آن‌ها نوشته شده اطلاعات را استفاده می‌کنند و قادر نیستند اطلاعات را به شیوه دیگری استفاده کنند، مگر آن‌که برنامه‌ای که از آن استفاده می‌کنند تهمیداتی در نظر گرفته باشد.

مطلب پیشنهادی

به کجا چنین شتابان
پرونده ویژه «مغزهای ماشینی؛ انسان یا برده؟» منتشر شد

الگوبرداری شبکه عصبی از مغز

شبکه‌های عصبی بر پایه این نظریه که ضرورتی ندارد برای یک شبکه عصبی یادگیری صریح را برنامه‌ریزی کرد طراحی می‌شوند، زیرا قرار است شبکه‌های عصبی شبیه به نمونه زیستی همه چیز را یاد بگیرند. این فرآیند یادگیری در لایه‌های مختلف به ترتیب کامل می‌شوند. طراحی یک شبکه عصبی به معنای آن نیست که ما در حال طراحی یک مغز هستیم. به عبارت دقیق‌تر زمانی که درباره شبکه‌های عصبی صحبت می‌کنیم در حقیقت درباره شبیه‌سازهای نرم‌افزاری صحبت می‌کنیم که از طریق الگوریتم‌ها و برنامه‌نویسی به سامانه‌های کامپیوتری که تا پیش از این غیرهوشمند و فرمان‌بردار بودند اجازه می‌دهند بر مبنای ترانزیستورها و گیت‌های منطقی رفتاری شبیه به مغز انسان‌ها داشته و مسائل را به شکل ادراکی تحلیل کرده و حل کنند. اگر به تلاش‌های دانشمندان در طول سال‌های مختلف نگاه کنید مشاهده می‌کنید تا به‌امروز هیچ شرکتی موفق نشده یک سامانه کامپیوتری طراحی کند که از طریق ترازیستورهایی با ساختار موازی رفتاری شبیه به مغز انسان‌ها داشته باشد. در شبکه‌های عصبی شما با مجموعه‌ای از فرمول‌ها، معادلات ریاضی و المان‌های جبری سروکار دارید که فرآیند انجام محاسبات را مدیریت می‌کنند. به عبارت دقیق‌تر تنها برنامه‌نویسان و دانشمندان خالق شبکه‌ها هستند که به خوبی می‌دانند شبکه‌های عصبی چه هستند. به عبارت دقیق‌تر، یک شبکه عصبی برای یک کامپیوتر هیچ معنای خاصی ندارد. شبکه‌های عصبی که بر مبنای شبیه‌سازی مغز و با اتکا بر برنامه‌نویسی الگوریتم‌های هوشمند طراحی می‌شوند به نام شبکه‌های عصبی مصنوعی ANN سرنام Artificial Neural Networks معروف هستند. شبکه‌های عصبی واقعی متشکل از مجموعه سلول‌های عصبی هستند که درون مغز ما قرار دارند. این **شبکه‌های عصبی زیستی** بوده و قابلیت رشد خودکار دارند.

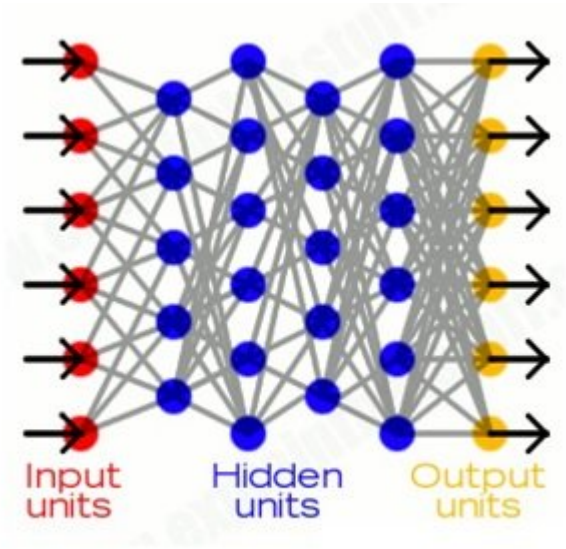
مطلب پیشنهادی

تراشه عصبی آکیدا - نگاهی به تراشه هوش مصنوعی مبتنی بر شبکه‌های عصبی SNN

شبکه‌های عصبی از چه مولفه‌ها و لایه‌هایی ساخته شده‌اند؟

در یک شبکه عصبی، سلول‌ها در مجموعه‌ای از لایه‌های مختلف در کنار یکدیگر قرار گرفته‌اند. این لایه‌ها از جهات مختلف با لایه‌های دیگر در ارتباط بوده و به یکدیگر متصل هستند. در حالی که تعدادی از این لایه‌ها نقش یک واحد ورودی را دارند و برای دریافت انواع مختلف ورودی‌ها از جهان خارج استفاده می‌شوند و به شبکه عصبی مصنوعی اجازه می‌دهند ضمن یادگیری، فرآیند تشخیص و پردازش را انجام دهد، در مقابل لایه‌های دیگر نقش خروجی را دارند. این لایه‌ها با هدف بررسی واکنش شبکه به اطلاعاتی که دریافت کرده و آن‌ها را تشخیص داده استفاده می‌شوند. در میان این لایه‌ها که برخی منابع به آن‌ها واحد (unit) می‌گویند، لایه‌های پنهان دیگری نیز قرار دارد. تجمیع این لایه‌ها در کنار یکدیگر باعث به وجود آمدن یک شبکه عصبی مصنوعی می‌شود. در حالت کلی شبکه‌های عصبی به گونه‌ای طراحی می‌شوند که همه لایه‌های ورودی، خروجی و پنهانی با یکدیگر در ارتباط باشند. این ارتباط و اتصال میان لایه‌ها با عددی به‌نام وزن (weight) نشان داده شده و ارزیابی می‌شوند. هر چه این میزان وزن بالاتر رود به معنای آن است که یک لایه تاثیر بیشتری روی لایه دیگر داشته است. در مقابل هر چه این وزن کمتر (منفی) باشد به معنای آن است که لایه‌ای بر لایه دیگر چیره شده است. این تکنیک مشابه با الگویی است که سلول‌های عصبی مغز انسان در قالب حفره‌های کوچکی که به‌نام سیناپس از آن‌ها نام برده می‌شود یکدیگر را فعال می‌کنند. دانشمندان برای آن‌که بتوانند **شبکه‌های عصبی** را به‌درستی پیاده‌سازی کنند، برای هر یک از واحدها رنگ‌های مشخصی را تعریف کرده‌اند. به‌طور مثال رنگ قرمز بیان‌گر واحدهای ورودی، رنگ آبی به معنای لایه‌های پنهان و رنگ زرد بیان‌گر واحدهای خروجی است. نکته‌ای که باید به آن دقت کنید این است که همه این لایه‌ها با لایه‌هایی که پیرامون آن‌ها

قرار دارد مرتبط هستند، درست مشابه با الگویی که در مغز وجود دارد. همان‌گونه که شکل زیر نشان می‌دهد، واحدهای ورودی قرمز رنگ از سمت چپ وارد می‌شوند، واحدهای پنهان که لایه‌های میانی را فعال می‌کنند آبی رنگ هستند و لایه‌های زردرنگ سمت چپ لایه‌های خروجی هستند.



شبکه‌های عصبی چگونه یاد می‌گیرند؟

اطلاعات درون یک شبکه عصبی به دو شکل در جریان است. شبکه عصبی فرآیند یادگیری را آغاز کرده یا فرآیند یادگیری را کامل کرده است. در این شرایط الگوهای یادگیری از طریق لایه‌های ورودی به شبکه و لایه‌های مرتبط با واحدهای پنهان اطلاعات را به سمت لایه‌های مرتبط با واحدهای خروجی ارسال می‌کنند. یک شبکه عصبی تنها زمانی می‌تواند فرآیند یادگیری را با موفقیت پشت سر بگذارد که بازخوردهایی در اختیارش قرار گیرد. درست مشابه با فرآیند آموزش کودکان که به آن‌ها اعلام می‌شود چه چیزی اشتباه و چه چیزی درست است. این یادگیری توام با اشتباه و دریافت بازخوردها باعث می‌شود میزان دقت شبکه عصبی افزایش پیدا کند. بازخوردهایی که به این شکل در اختیار شبکه عصبی قرار می‌گیرند به خوبی قادر هستند تفاوت‌های موجود را نشان داده و به دانشمندان اجازه دهند تغییراتی در دستورات اعمال کنند. در **شبکه‌های عصبی** دانشمندان همواره خروجی ارائه شده از سوی یک شبکه عصبی را با خروجی که مدنظر قرار دارند ارزیابی می‌کنند. با مقایسه تفاوت‌ها دانشمندان آگاه می‌شوند که باید چه تغییراتی اعمال کرده و چگونه وزن میان اتصالات و لایه‌ها را دستکاری کنند تا نتیجه مطلوب به دست آید.

عملکرد شبکه‌های عصبی چگونه است؟

برای آن‌که به توان یک شبکه عصبی را آموزش داد به صدها یا شاید هزاران نمونه نیاز است تا فرآیند یادگیری تکمیل شود. پس از کامل شدن این فرآیند، در ادامه ورودی‌های جدیدی در اختیار شبکه عصبی قرار می‌گیرد که تاکنون دریافت نکرده است. دانشمندان واکنش شبکه عصبی به این ورودی‌ها را ارزیابی می‌کنند. فرض کنید، شبکه عصبی باید یاد بگیرد که میزها و صندلی‌ها را تشخیص دهد. در این حالت طیف گسترده‌ای از تصاویر میز و صندلی به شبکه نشان داده می‌شود. پس از تکمیل این فرآیند یک مدل خاص از صندلی که پیش از این شبکه عصبی آن‌را مشاهده نکرده به عنوان ورودی در اختیارش قرار می‌گیرد. در این مرحله شبکه عصبی بر مبنای آموزش‌ها باید سعی کند ورودی جدید را طبقه‌بندی کرده و اعلام کند ورودی یک میز یا صندلی است. اینکار درست مشابه با فرآیندی است که

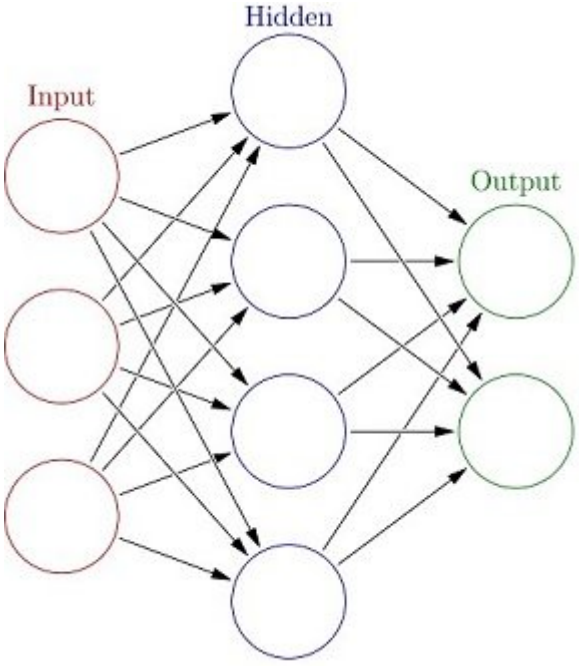
ما در طول زندگی خود بارها و بارها آن‌را تکرار می‌کنیم. البته فراموش نکنید دقت و سرعت عمل شبکه عصبی همانند انسان‌ها نبوده و پایین‌تر است. تفاوت زیربنایی در همین نقطه خود را نشان می‌دهد. یک شبکه عصبی هیچ‌گاه با نگاه کردن به یک تصویر نمی‌تواند تشخیص دهد یک صندلی یا میز در تصویر قرار دارد. بلکه بر مبنای صفرها و یک‌هایی که به عنوان ورودی دریافت می‌کند قادر است مشخصات و تفاوت‌های صندلی‌های مختلفی در شکلی که مشاهده می‌کند را تشخیص دهد.

مطلب پیشنهادی

فقط مکان و زمان مبارزه را به من نشان دهید...
علی‌بابا و غول عصبی؛ قلب و مغز دنیای فناوری در تسخیر هوش مصنوعی

مفاهیم اولیه شبکه‌های عصبی

برای پیاده‌سازی یک شبکه عصبی در اولین گام باید با مفاهیم اولیه و مقدماتی و همچنین مباحث ریاضی و آمار آشنا باشید. سعی کردیم تا حد امکان توضیحاتی ارائه کنیم، اما به مطالعه دقیق‌تر نیاز دارید. همچنین، اگر تصمیم دارید مدل‌ها را روی سامانه خود اجرا کنید، باید کتابخانه‌های موردنیاز را روی سامانه خود نصب کرده و با زبان برنامه‌نویسی پایتون آشنا باشید. برای اجرای نمونه مثال‌های این مقاله باید کتابخانه‌های Pandas، Numpy، Matplotlib، Sckiit-Learn، Tensorflow و Keras را روی سامانه خود نصب کنید. در این مقاله قصد داریم شبکه‌ای مشابه با آن چیزی که در شکل زیر مشاهده می‌کنید را ایجاد کنیم.



مدل نشان داده شده در شکل بالا، از سه لایه ورودی، پنهان منفرد و خروجی ساخته شده است. اندازه این لایه‌ها تابع استاندارد و قواعد مشخصی است و قرار نیست اندازه دلخواهی داشته باشند. در زمان ساخت شبکه عصبی مصنوعی و زمانی که باید درباره تعداد سلول‌های عصبی (نورون‌ها) که قرار است در هر لایه قرار بگیرند تصمیم‌گیری کنیم، مجبور هستیم قواعد را رعایت کنید. قصد داریم بر مبنای نیاز کاری خودمان در مدل شکل بالا تغییراتی اعمال کنیم. ما به جای 3 سلول عصبی ورودی از 4 سلول عصبی، به جای 4 سلول عصبی در لایه نهان از 8 سلول عصبی و به جای 1 سلول عصبی در بخش سلول‌های عصبی خارجی از 3 سلول عصبی استفاده می‌کنیم. انتخاب این مقادیر دلایل خاص خود را دارند که در ادامه به ذکر این دلایل خواهیم پرداخت. اندازه لایه ورودی شبکه ما بر مبنای تعداد قابلیت‌هایی که در مجموعه داده‌های ما قرار دارد مشخص می‌شود. به‌طور مثال، اگر در نظر داشته باشیم مجموعه داده‌های سنتی Iris (مجموعه داده گل زنبق) را طبقه‌بندی کنیم، تنها آن تعداد سلول‌های عصبی ورودی را درون مدل خود قرار می‌دهیم که ویژگی‌های مدنظر ما را تامین کنند. در مجموعه داده Iris این حالت بیان‌گر به‌کارگیری تمامی ویژگی‌های رایج مجموعه داده و وارد کردن طول کاسبرگ (sepal length)، عرض کاسبرگ (sepal width)، طول گلبرگ (petal length) و عرض گلبرگ (sepal width) است. شکل زیر مجموعه داده که ویژگی‌ها و خروجی‌ها را شناسایی می‌کنند نشان می‌دهد.

	sepal length	sepal width	petal length	petal width	target
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa

در انتها، لایه‌های پنهان هستند که در وضعیت آزمون و خطا قرار می‌گیرند. لایه‌های پنهان از جمله مباحث مهم **شبکه‌های عصبی** هستند که پژوهش‌های بسیاری در ارتباط با آن‌ها انجام شده است. در حالت کلی، یک لایه پنهان تکی کافی است. صفر لایه پنهان تنها در ارتباط با موقعیت‌هایی که جداسازی به صورت خطی است استفاده می‌شود. یک لایه پنهان امکان نگاشت از یک فضای محدود به فضای دیگر را فراهم می‌کند. 2 لایه پنهان برای موقعیت‌هایی مناسب است که مرزها یا کرانه‌های تصمیم‌گیری دلخواه هستند و محدودیتی برای آن‌ها در نظر نگرفته شده است.

تعداد سلول‌های عصبی در هر لایه

تعداد سلول‌های عصبی درون لایه‌های پنهان نیز در وضعیت آزمون و خطا قرار می‌گیرند، اما دستورالعمل‌ها و فرمول‌هایی وجود دارند که اجازه می‌دهند به درستی تعداد سلول‌های عصبی در لایه پنهان را انتخاب کنید. فرمول زیر فرمولی که برای انتخاب تعداد سلول‌های عصبی در لایه‌های پنهان از آن استفاده می‌شود را نشان می‌دهد.

$$N_h = \frac{N_s}{(\alpha * (N_i + N_o))}$$

- Nh: تعداد سلول‌های عصبی در لایه پنهان
- Ns: تعداد نمونه‌های موجود در داده‌های آموزشی
- @: آلفا عامل گسترش‌پذیری
- Ni: تعداد سلول‌های ورودی
- No: تعداد سلول‌های خروجی

فرمول فوق دید روشنی از تعداد لایه‌ها و سلول‌های عصبی که باید در هر لایه قرار گیرند در اختیار ما قرار می‌دهد. برنامه‌نویسان مجرب از راهکارهای دقیق‌تری برای انتخاب تعداد سلول‌های عصبی که باید در لایه پنهان قرار گیرند استفاده می‌کنند. تعداد سلول‌های عصبی بسته به نتیجه‌ای که قرار است به دست آورید متفاوت هستند. اگر از یک رگرسور (regressor) که ترجمه تحت‌الفظی آن بازگشت می‌شود استفاده کنیم، تعداد سلول‌های عصبی برابر با یک خواهند بود. اگر از برچسب‌های گره برای کلاس استفاده کنید (به‌طور مثال softmax) باید بیش از یک سلول عصبی را استفاده کنید. اگر تمایلی ندارید از softmax استفاده کنید، بدون مشکل می‌توانید از یک سلول عصبی برای طبقه‌بندی داده‌ها استفاده کنید. در این آموزش تصمیم داریم از softmax استفاده کنیم تا ببنیم مدل ما چگونه مجموعه داده‌های Iris را جست‌وجو خواهد کرد. در مجموعه داده Iris به دنبال بررسی این موضوع هستیم که یک مشاهده منفرد (منظور همان جست‌وجویی است که مدل انجام می‌دهد) برای پیدا کردن گونه‌های Virginica، Versicolor یا Setosa انجام دهیم. در این‌جا ما 3 سلول عصبی برای 3 کلاس خواهیم داشت.

مطلب پیشنهادی

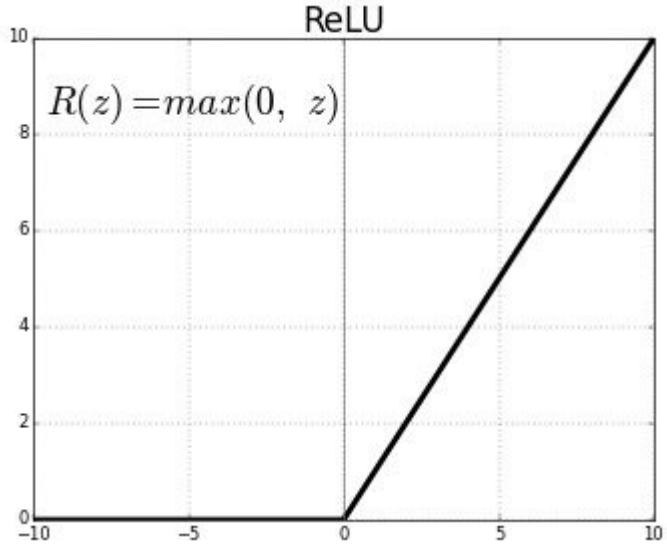
فناوری‌های عصبی و لزوم احترام به حریم شخصی
چهار اولویت اخلاقی در رابطه با فناوری‌های عصبی و هوش مصنوعی

آشنایی بیشتر با تابع softmax

اجازه دهید توضیح مختصری درباره تابع softmax ارائه کنیم تا ببنیم چرا تابع فوق در ارتباط با لایه خروجی ایده‌آل است. تابع softmax زمانی استفاده می‌شود که قصد داریم از روش‌های طبقه‌بندی چند کلاسی مختلف استفاده کنیم. فرمول زیر تابع softmax را نشان می‌دهد. Softmax در هسته اصلی خود اجازه می‌دهد مقادیر کمتر از شاخص خود را حذف کرده (یا به عبارت دقیق‌تر سرکوب) و مقادیر بالاتر از میانگین خود را برجسته‌تر کنیم.

$$P(y = j \mid \mathbf{x}) = \frac{e^{\mathbf{x}^\mathsf{T}\mathbf{w}_j}}{\sum_{k=1}^{K} e^{\mathbf{x}^\mathsf{T}\mathbf{w}_k}}$$

دلیل استفاده از تابع softmax این است که خروجی تابع فوق به ما اجازه می‌دهد یک توزیع احتمال را روی نتایج مختلف و متنوع انجام دهیم. تابع فوق پویایی زیادی در اختیار ما قرار می‌دهد، زیرا اجازه می‌دهد برخی از مسائل پیچیده یادگیری ماشین و یادگیری عمیق را با اتکا بر طبقه‌بندی چند کلاسی حل کنیم.
در نتیجه تابع softmax گزینه ایده‌آلی برای به‌کارگیری روی مجموعه داده Iris است. برای کاربردهایی همچون طبقه‌بندی‌های باینری، باید از توابعی همچون Sigmoid استفاده کنیم، زیرا تابع فوق نسخه خاصی از تابع softmax است که تعداد کلاس‌ها را به دو مورد تقلیل می‌دهد. ما در لایه پنهان مدل خودمان reLU را تابع فعال‌ساز انتخاب کرده‌ایم. ReLU پر کاربردترین تابع در لایه پنهان است. تابع فوق به دلیل ماهیت طبیعی خود نسبت به سایر گزینه‌های یادگیری متداول سرعت یادگیری بیشتری دارد. در ReLU مولفه‌های منفی با 0 تنظیم می‌شوند. شکل زیر تابع ReLU را نشان می‌دهد.



بهینه‌سازی مدل

اکنون زمان آن رسیده تا بهینه‌ساز خود را انتخاب کنید. در این مقاله ما از بهینه‌ساز Adam استفاده می‌کنیم، زیرا روی سامانه‌هایی با حافظه کم عملکرد مطلوبی دارد. برعکس شیب کاهشی تصادفی (stochastic gradient descent)، بهینه‌ساز Adam از پارامتر نرخ یادگیری با استفاده از میانگین حرکت/متحرک نمایی (exponential moving average) استفاده می‌کند. Adam در مدت زمان یادگیری از میانگین گشتاور دوم برای انطباق نرخ یادگیری استفاده می‌کند.

مطلب پیشنهادی

ماشین‌هایی با توانایی فکر کردن
در آینده بهترین قابلیت‌های مغز ما به ماشین‌ها خواهد رسید

جمع‌بندی کارهایی که انجام دادیم

1 لایه ورودی، 1 لایه پنهان، 1 لایه خروجی و چهار سلول عصبی ورودی برای 4 ویژگی، 8 سلول عصبی در لایه نهان که هماهنگ با تعداد سلول‌های عصبی ورودی، سلول‌های عصبی خروجی و اندازه مجموعه داده‌ها هستند و 3 سلول عصبی در لایه خروجی برای 3 کلاسی که در اختیار داریم، حاصل کار ما است. ما توابع فعال‌ساز خود را به این شکل انتخاب کردیم که از ReLU برای لایه پنهان و از softmax برای لایه خروجی استفاده می‌کنیم. در بحث تابع بهینه‌ساز نیز از الگوریتم بهینه‌ساز Adam به شرحی که در فهرست 1 مشاهده می‌کنید استفاده می‌کنیم.

```
def baseline_model():
    model = Sequential()
    model.add(Dense(8,input_dim=4,activation='relu'))
    model.add(Dense(3, activation='softmax'))
    model.compile(loss='categorical_crossentropy',optimizer='adam', metrics=['accuracy'])
    return model
```

فهرست 1

تابع فوق مدل پایه ما را ایجاد می‌کند. ما می‌توانیم این تابع را هر زمان که قصد ایجاد مدل جدیدی داریم فراخوانی کنیم. تابع فوق شبکه‌ای که در این مقاله به توصیف آن پرداختیم را ایجاد می‌کند (فهرست 2).

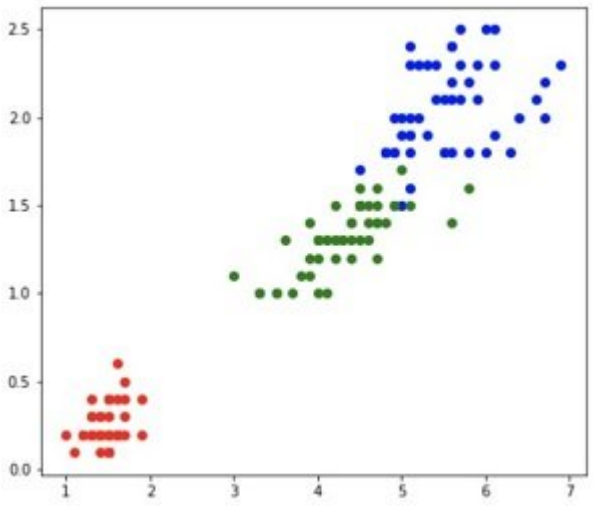
```
model = baseline_model();
model.fit(X,dummy_y,epochs=125,verbose=1)
y_pred = model.predict_classes(X)
```

فهرست 2

3 خط نشان داده شده در فهرست 2 ساده هستند. ابتدا مدل را مقداردهی اولیه می‌کنیم. ابتدا باید مقدار مدنظر خود را درون پارامتر epochs درج کنیم و در ادامه مقدار verbose را به 1 تنظیم کنیم.
مقدار 1 وضعیتی در اختیار ما قرار می‌دهد تا بررسی کنیم که آیا تعداد انتخاب‌های epoch فراتر از آستانه تعیین شده است که منجر به هدر رفت زمان آموزش می‌شود یا خیر. در هر epcho تمامی مجموعه یک‌بار به جلو و یک‌بار به عقب شبکه عصبی حرکت می‌کنند. این حرکت پاندولی باعث به‌روزرسانی وزن‌ها در هر حرکت می‌شوند. شکل زیر نمایی از وضعیت آموزش مدل را نشان می‌دهد.

training process

همان‌گونه که شکل بالا نشان می‌دهد، دقت به میزان قابل قبولی بهبود پیدا کرده و پیرامون مقدار 150 در حال انجام است. بر همین مبنا باید تعداد epochها را تنظیم کرد. اکنون می‌توانیم طبقه‌بندی را با استفاده از مشخصه‌های مختلف روی مجموعه داده‌ها اعمال کنیم.



منبع:
towardsdatascience
explainthatstuff
تاریخ انتشار:
10 آذر 1398

نشانی منبع:
https://www.shabakeh-mag.com/information-feature/artificial-intelligence/16240/%D8%B1%D8%A7%D9%87%D9%86%D9%85%D8%A7%DB%8C-%D8%A2%D8%B4%D9%86%D8%A7%DB%8C%DB%8C-%D8%A8%D8%A7-%D8%B4%D8%A8%DA%A9%D9%87%E2%80%8C%D9%87%D8%A7%DB%8C-%D8%B9%D8%B5%D8%A8%DB%8C-%D9%88-%D9%86%D8%AD%D9%88%D9%87-%D9%BE%DB%8C%D8%A7%D8%AF%D9%87%E2%80%8C%D8%B3%D8%A7%D8%B2%DB%8C