



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده ریاضی و علوم کامپیوتر

گزارش درس هوش مصنوعی و کارگاه

الگوریتم های پردازش زبان طبیعی، بینایی ماشین و رباتیک

دانشجو
ایلیا خلفی

استاد درس
دکتر مهدی قطعی

تدریس یار درس
روح الله احمدیان

مهر ۱۴۰۱

چکیده

هوش مصنوعی زیرمجموعه‌ای از علوم کامپیوتر است که از اواسط قرن ۲۰ میلادی مورد بحث قرار گرفت. گرچه بحث فلاسفه و ریاضیدان درباره سیستم هوشمندی که بتواند مانند انسان فکر کند به زمان فلاسفه‌ی یونان باستان بر می‌گردد، اما تنها پس از اختراع کامپیوتر و پردازنده‌ها بود که این حوزه به صورت جدی دنبال شد. همچنین با گره خوردن هوش مصنوعی به دنیای علوم کامپیوتر، دانشمندان توانستند جواب‌های دقیق‌تری به سوال اینکه "آیا یک سیستم مورد نظر هوشمند است؟" بدهند. در این گزارش، الگوریتم‌های توسعه یافته حوزه هوش مصنوعی در زمینه‌های پردازش زبان طبیعی، بینایی ماشین و رباتیک را بررسی خواهیم کرد.

واژه‌های کلیدی:

هوش مصنوعی، پردازش زبان طبیعی، بینایی ماشین، رباتیک

Contents

ا	چکیده.....
۳	فصل اول مقدمه.....
۴	فصل دوم الگوریتم های پردازش زبان طبیعی.....
۵	۱-۲- ریشه یابی.....
۵	۲-۲- گراف کلمات و موضوعات.....
۶	۳-۲- نشانه گذاری کلمات.....
۶	۴-۲- شبکه های عصبی بازگشتی.....
۹	۱-۴-۲- لایه های LSTM و GRU.....
۱۰	۲-۴-۲- لایه توجه و تغییر شکل دهنده ها.....
۱۱	فصل سوم الگوریتم های بینایی ماشین.....
۱۲	۱-۳- چهره های ویژه.....
۱۴	۲-۳- شبکه های عصبی همگشتی.....
۱۷	فصل چهارم الگوریتم های رباتیک.....
۱۸	۱-۴- الگوریتم های مبتنی بر ارزش.....
۱۹	۲-۴- الگوریتم های مبتنی بر سیاست.....
۱۹	۳-۴- الگوریتم های بازیگر و منتقد.....
۲۰	فصل پنجم جمع بندی و نتیجه گیری.....
۲۱	منابع و مراجع.....

فصل اول

مقدمه

یکی از چالش‌های جذاب برای بشر که مدت‌ها با آن مواجه بوده، طراحی موجودی است که همانند انسان توانایی شنیداری، دیداری، قوه تفکر و دیگر خصوصیات انسانی را داشته باشد.

امکان خلق همچین موجودی به دست انسان، مدت‌ها توسط فلاسفه مورد بحث و بررسی قرار می‌گرفت تا اینکه در قرن ۲۰ میلادی با اختراع پردازنده‌های کامپیوتری، این ایده بیش از هر زمان دیگری امکان تحقق پیدا کرد، در حقیقت مبحث هوش مصنوعی وابسته به علوم کامپیوتر است و به همین دلیل شالکوف (Schalkoff) در سال ۱۹۹۰ آن را اینگونه تعریف کرد:

"حیطه‌ای از مطالعه که رفتار هوشمند را تحت عنوان فرآیندهای کامپیوتری شرح داده و مورد رقابت قرار دهد"

همچنین با گره خوردن هوش مصنوعی به دنیای علوم کامپیوتر، دانشمندان توانستند جواب‌های دقیق‌تری به سوال اینکه "آیا یک سیستم مورد نظر هوشمند است؟" بدهند که معروف‌ترین آنها آزمون تورینگ است که به طور خلاصه شرط می‌کند که سیستم برای هوشمند بودن باید بتواند یک انسان را که از طریق تله تایپ با آن در ارتباط است را متقاعد کند که او نیز یک انسان است.

با وجود آنکه هوش مصنوعی با هدف طراحی موجودی متفکر مانند انسان شکل گرفت، اما امروزه سیستم‌های هوش مصنوعی تنها به دنبال تقلید توانایی‌های شنیداری و دیداری و دیگر توانایی‌های انسان نیستند، در حقیقت سیستم‌های هوش مصنوعی امروزه توانایی پیش‌بینی ارزش‌ها و طبقه‌بندی گروه‌ها از روی اطلاعات ساختار یافته را دارند که دقت آنها بسیار فراتر از انسان‌ها رفته است و هوش مصنوعی تنها محدود به شنیدن نیست.

هوش مصنوعی امروزه به قدری در تمامی زمینه‌های علمی و صنعتی نفوذ پیدا کرده است که پروفیسور اندرو انگ (Prof. Andrew Ng) که از پیشگامان این زمینه است، می‌گوید:

"هوش مصنوعی، الکترونیسته‌ی جدید است."

فصل دوم

الگوریتم های پردازش زبان طبیعی

درک زبان طبیعی یکی از چالش های اساسی هوش مصنوعی است. منظور از زبان طبیعی، زبانی است که در طی سالیان متعدد در میان روابط انسان ها ساخته شده و تکامل یافته است.

پردازش زبان طبیعی شامل زمینه های متعددی مانند تشخیص گفتار از صوت، درک قواعد گفتاری زبان، درک ویژگی کلمات و ترجمه ی زبان ها به یکدیگر است. معمولاً این زمینه در یادگیری ماشین به عنوان زیرمجموعه ی پردازش مدل های دنباله ای عنوان می شود، پردازش مدل های دنباله ای شامل هر مدل یادگیری ماشین می شود که ورودی یا خروجی آن یک دنباله است.

از جذابیت ها و خواص منحصر بفرد این زمینه آن است که هر بخش متن ورودی یا خروجی می تواند وابسته به بخش های دیگر باشد. از آنجا که تعداد واژگان در گفتار و متن ورودی در اکثر موارد متفاوت است، در نتیجه الگوریتم های پردازش زبان طبیعی کلمات متن یا گفتار را تک به تک و به صورت دنباله ورودی می گیرند و در نتیجه الگوریتم طراحی شده در هر مرحله تنها یک کلمه یا عبارت از متن را در اختیار دارد. به همین دلیل اکثر الگوریتم های طراحی شده در این زمینه دارای یک خصوصیت حافظه مانند هستند که اطلاعاتی را از بخش های دیگر متن که قبلاً وارد الگوریتم شده اند را نگهداری می کند.

۲-۱- ریشه یابی

ریشه یابی یکی از راهکاری های ابتدایی است که برای پردازش زبان های طبیعی مورد استفاده قرار میگیرد و امروزه نیز کاربرد دارد. در این روش سعی می شود که ضمائر و حروف اضافه از کلمه حذف شوند و به عنوان مثال کلماتی مانند "خوانش، خواندم، خواندی و ..." با کلمه ی ریشه آنها یعنی خواندن جایگزین می شوند. این روش نیاز به یک لغت نامه واژگان دارد که کلمات را گروه بندی کرده و کلمه ی ریشه ای هر گروه را به آن گروه متناظر کرده باشد، به اینصورت کافیسست که ابتدا کلمه موردنظرمان را در لغت نامه پیدا کرده و سپس آن با کلمه ریشه ای آن گروه جایگزین کنیم.

از راهکارهای بهینه برای ساخت لغت نامه، استفاده از درخت ترای یا هاش کردن در ساختار لغت نامه است، اما در هر صورت این راهکار نیاز به حافظه ی زیادی برای نگهداری تعداد زیادی از واژگان در لغت نامه دارد که مشکل ساز است.

توجه داشته باشید که این روش صرفاً برای کاهش گوناگونی کلمات موجود در متن کاربرد دارد و صرفاً ابزاری کمکی برای بقیه الگوریتم ها است.

۲-۲- گراف کلمات و موضوعات

در این روش سعی می شود که کلمات متعددی که ممکن است در رابطه با یک موضوع بکار گرفته شوند را شناسایی کرده و در گرافی میان کلمات و گروه موضوعات، توصیفی برای ارتباط میان کلمات و موضوعات ساخته شود، سپس در هنگام بررسی کلمات موجود در متن، به ازای هر بار ظهور کلمه ی مرتبط با موضوع، به آن موضوع امتیاز داده می شود تا در نهایت موضوعی که بیشترین امتیاز را کسب کرده برنده شود. البته این الگوریتم می تواند پیچیده تر شود و به عنوان مثال از مثبت یا منفی افعال نیز برای ضریب دادن به امتیاز موضوعات نیز استفاده کرد.

این روش از راهکارهای ابتدایی پردازش زبان طبیعی بوده است که به دلیل ضعف های بزرگ آن و روی کار آمدن الگوریتم های بهینه تر و دقیق تر، کنار گذاشته شد. از نواقص جدی این الگوریتم حافظه ی زیاد

مورد نیاز آن، وابسته بودن گراف کلمات و موضوعات به متنی که بر اساس آن ساخته شده و همچنین کارا نبود در درک روابط پیچیده‌ی زبانی است.

۲-۳- نشانه گذاری کلمات

هر الگوریتمی که سعی در پردازش زبان طبیعی داشته باشد، ابتدا باید کلمات موجود در متن مورد پردازش را به اعداد، ماتریس، تنسور یا دیگر ساختمان دادگان نشانه گذاری کرده و سپس بر روی آن نشانه‌ها فرآیند خود را اعمال کند.

یک راهکار برای نشانه گذاری کلمات آن است که کلمات موجود در زبان مورد پردازش رو در آرایه ای قرار دهیم و شماره خانه‌ی هر کلمه را به عنوان نشانه‌ی آن کلمه در نظر بگیریم. گرچه این روش باعث نشانه گذاری هر کلمه به یک عدد شده و حافظه‌ی کمی را اشغال می‌کند، اما از آنجا که کلمات متغیری کمی هستند و نه کیفی، اکثر الگوریتم‌هایی که برا مبنای روابط ریاضی کار می‌کنند با این روش به مشکل می‌خورند.

راهکار متداول‌تر آن است که مجموعه‌ای از کلمات که قرار است الگوریتم ما آن‌ها را متوجه شود مشخص کنیم، سپس بردار یا آرایه‌ای به اندازه‌ی تعداد اعضای این مجموعه بسازیم که در خانه‌ی متناظر با کلمه موردنظرمان عدد ۱ و در بقیه خانه‌ها عدد ۰ قرار دهیم. این روش حافظه‌ی زیادی برای نشانه گذاری هر کلمه نیاز دارد، به عنوان مثال اگر مجموعه لغات قابل فهم الگوریتم ما شامل ۱۰۰.۰۰۰ واژه باشد، آنگاه آرایه‌ای به اندازه ۱۰۰.۰۰۰ برای نمایش تنها یک عدد نیاز داریم.

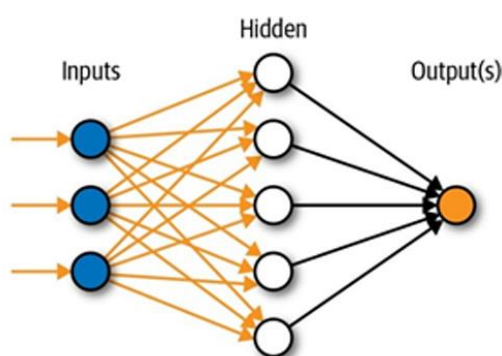
۲-۴- شبکه های عصبی بازگشتی

شاید بتوان گفت ابداع شبکه های عصبی مهم ترین رویداد در زمینه هوش مصنوعی و حتی علوم کامپیوتر است، شبکه های عصبی امروزه به قدری معروف شده اند که حتی افرادی که در این زمینه کار نمی‌کنند نیز به احتمال زیادی نام آن شنیده اند.

شبکه های عصبی به طور خلاصه سعی در ایجاد شبکه ای از پارامترها را دارند که با ورودی گرفتن مقداری اطلاعات و انجام پیش انتشار، یک خروجی را مطابق الگوی داده های تست، پیش بینی کنند.

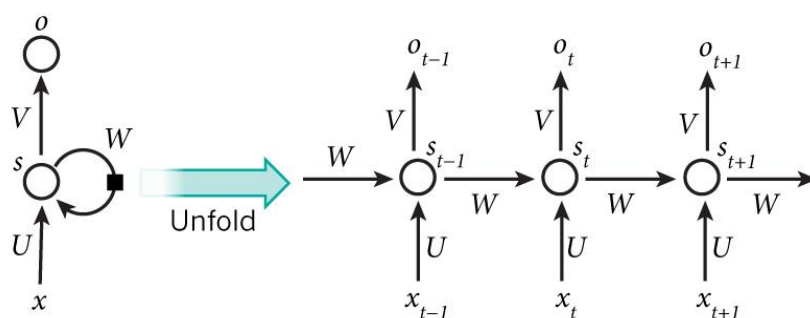
کارآمدی شبکه های عصبی به دلیل ابداع پس انتشار است که به طور خلاصه چیزی بیشتر از محاسبه ی مشتقات جزئی خطای خروجی بر حسب ورودی ها نیست، اما همین مفهوم ساده باعث خلق شبکه های عصبی شده که با انجام پیش انتشار و پس انتشار های متوالی، الگوهای موجود در داده های تست را یاد می گیرند.

در شکل زیر یک نمونه ی ساده از شبکه های عصبی کاملاً متصل نمایش شده است. کاملاً متصل به آن معناست که مقادیر هر لایه بر تمام مقادیر لایه ی بعدی اثر می گذارند.

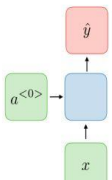
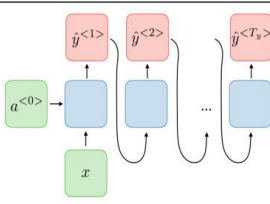
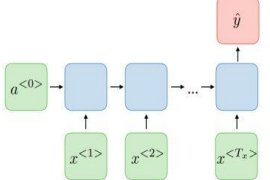
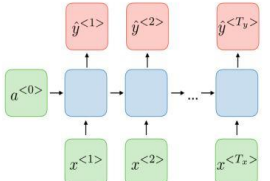
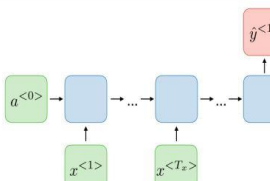


اما شبکه های عصبی کاملاً متصل برای پردازش زبان طبیعی کارایی ندارند، زیرا همانطور که پیشتر گفتیم، در پردازش متن یا گفتار، اندازه ورودی می تواند وابسته به تعداد کلمات متن یا گفتار متفاوت باشد در حالیکه اندازه ی ورودی شبکه های کاملاً متصل باید ثابت باشد.

به دلیل ثابت نبودن اندازه ی ورودی و همچنین متکی بودن بخش های مختلف متن ورودی به یکدیگر، ایده ی شبکه های عصبی بازگشتی مطرح شد. در این نوع از شبکه ها از لایه های بازگشتی استفاده می شود که در ساختار آن مقادیر اشتراکی در شبکه وجود خواهد داشت که به ازای ورودی کلمات پشت سر هم، این مقادیر اشتراکی تغییر می کنند و بدین صورت هر کلمه ورودی می تواند بر خروجی شبکه برای کلمات بعدی تاثیر گذار باشد. شکل زیر ساختار این لایه را نشان می دهد.



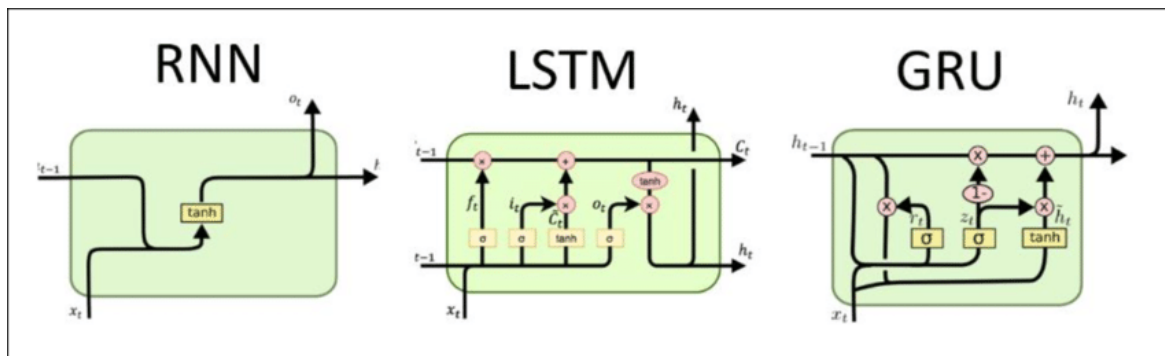
در نظر داشته باشید که در شبکه های عصبی بازگشتی، لزوما نیاز نیست ورودی و خروجی هردو دنباله باشند و این شبکه ها می تواند برای یادگیری الگوهای دنباله به دنباله (مانند ترجمه متن و تبدیل گفتار به متن) یا متغیر به دنباله (مانند توصیف تصویر با متن) یا دنباله به متغیر (مانند تشخیص مثبت یا منفی بودن نظرات) بکار رود. در تصویر زیر مثالی از ساختارهای متفاوت شبکه های عصبی همگشتی را آورده ایم.

نوع RNN	نگاره	مثال
یک به یک $T_x = T_y = 1$		شبکه ی عصبی سنتی
یک به چند $T_x = 1, T_y > 1$		تولید موسیقی
چند به یک $T_x > 1, T_y = 1$		دسته بندی حالت احساسی
چند به چند $T_x = T_y$		بازشناسایی موجودیت اسمی
چند به چند $T_x \neq T_y$		ترجمه ماشینی

۲-۴-۱- لایه های LSTM و GRU

مشکل بسیار بزرگ استفاده از لایه های بازگشتی تنها، در شبکه های عصبی آن است که به دلیل تکرار لایه ی بازگشتی به تعداد اندازه ی ورودی، در اصل یک شبکه عصبی به عمق ورودی پدید می آید در نتیجه با مشکل گرادیان محو شونده مواجه خواهیم شد که باعث می شود فرآیند پس انتشار به مشکل بخورد و در طی یادگیری، پارامترهای لایه ی بازگشتی با توجه به بخش های اول ورودی تغییر نکند و به طور خلاصه بخش های انتهایی متن بسیار بیشتر از بخش های ابتدایی در آموزش شبکه تاثیر خواهند گذاشت و الگوهای ابتدای متن نادیده گرفته می شود.

لایه های LSTM و GRU نیز لایه هایی بازگشتی به حساب می آیند که ولی در آنها پارامترهایی برای به خاطر داشتن یا از یاد بردن ورودی های قبلی اضافه شده است که بدین ترتیب در هنگام پس انتشار آن بخش از متن که توسط LSTM به عنوان بخش مهم به خاطر سپرده شده است، سیگنال گرادیان بیشتری دریافت خواهد کرد و اگر این بخش در ابتدای متن باشد مشکل گرادیان محو شونده را بسیار کم خواهد کرد.



لایه LSTM پیش از GRU ابداع شده است و تعداد پارامترهای بیشتری دارد که در نتیجه استفاده از آن حافظه و منابع محاسباتی بیشتری را نیاز دارد، به همین دلیل و با آزمایش تعداد مختلف پارامترهای حافظه در LSTM، دانشمندان دریافتند که می توان تا حد معقولی تعداد پارامترهای حافظه را بدون از دست دادن دقت، کم کرد تا حافظه و منابع محاسباتی کمتری نیاز باشد که در نتیجه ی آن لایه ی GRU ابداع شد. این لایه نسبت به LSTM پارامترهای کمتری دارد.

۲-۴-۲- لایه توجه و تغییرشکل دهنده‌ها

همانطور که از ابتدا گفتیم، الگوریتم‌های هوش مصنوعی سعی در تقلید توانایی انسانی دارند و در الگوریتم‌های پردازش زبان طبیعی هم این قاعده برقرار است، اما سوالی که با روی کار آمدن شبکه‌های عصبی بازگشتی مطرح شد آن بود که "چرا باید شبکه در موقع بررسی انتهای یک متن طولانی، ابتدای آن را نیز به خاطر داشته باشد؟" به طور دقیق‌تر دانشمندان مشاهده کردند که با افزایش طول متن ورودی، دقت شبکه‌های عصبی بازگشتی کاهش می‌یابد زیرا در انتهای متن سعی در استخراج محتوا از ابتدای متن را نیز دارد که این مشکل ساز است. همچنین با توجه به رفتار انسانی، موقع بررسی یک متن، تنها چند کلمه قبل و بعد از آن در ذهن انسان باقی می‌ماند و نه کل متن.

همچنین نبود امکان موازی سازی مشکل دیگر شبکه‌های عصبی بازگشتی است، زیرا برای خروجی گرفتن از این شبکه‌های توالی واژگان ورودی مهم است و به همین دلیل نمی‌توان موازی سازی در آن انجام داد.

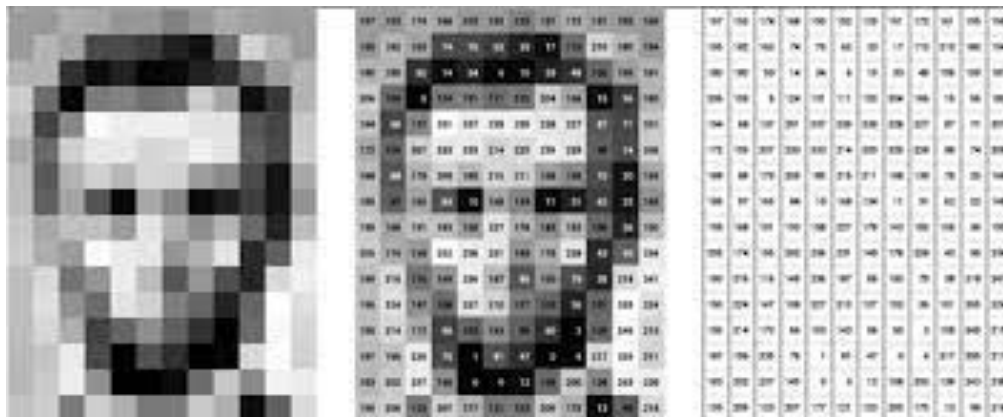
تغییرشکل دهنده‌ها هر دوی این مشکلات را حل کرده‌اند. در این مدل‌ها از لایه‌های توجه چندسرها استفاده می‌شود که در اصل چند لایه توجه هستند که خروجی‌های آن‌ها در انتهایشان در کنار یکدیگر قرار گرفته است، موازی سازی در همین بخش رخ می‌دهد زیرا می‌توانیم خروجی‌های هر سر که یک لایه توجه است را بر روی یک پردازنده محاسبه کرده و در نهایت خروجی‌ها را در کنار یکدیگر قرار دهیم.

به طور خلاصه در لایه توجه، زمانی که شبکه می‌خواهد بخشی از دنباله را پردازش کند، از چند کلمه قبل تا چند کلمه بعد از آن کلمه را به عنوان ورودی گرفته و سعی می‌کند مفهوم کلمه را بر اساس کلمه‌های اطرافش درک کند زیرا یک کلمه ممکن است در ترکیب‌های مختلف معانی متفاوتی داشته باشد. شباهت عملکرد تغییرشکل دهنده‌ها به عملکرد انسان و همچنین قابلیت موازی سازی آنها، به قدری این مدل‌ها را محبوب کرده که ابداع آنها جزو انقلاب‌های حوزه‌ی یادگیری ماشین تلقی می‌شود.

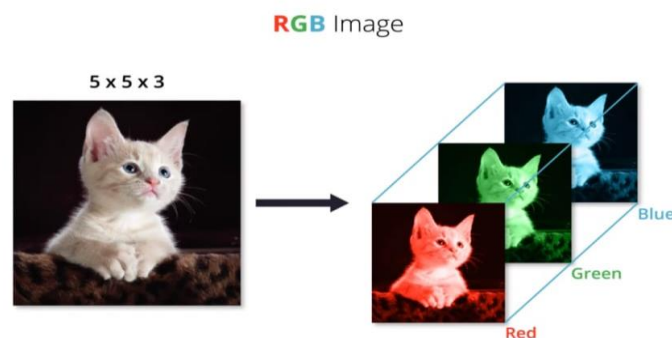
فصل سوم

الگوریتم های بینایی ماشین

تصاویر در ماشین به صورت مجموعه پیکسل های در کنار یکدیگر نمایش داده می شوند به طوری که هر پیکسل کوچکترین واحد رنگی یک تصویر است. همچنین برای ذخیره و نمایش مجدد تصاویر، آنها به صورت یک ماتریس یا تنسور از اعداد در ماشین ذخیره می شوند که هر عدد مربوط به یک پیکسل است. مانند شکل زیر هر تصویر را می توان با یک ماتریس نمایش داد.



همچنین برای نمایش تصاویر رنگی نیاز به کانال های رنگی نیز داریم که در نتیجه با توجه به فرمت تصویر (مثلا RGB یا HSV) ممکن است تصویرمان چند کانال رنگی داشته باشد که در اینصورت از تنسورها برای ذخیره ی تصویرمان استفاده می کنیم. در شکل زیر کانال های رنگی در فرمت RGB را نمایش داده ایم.

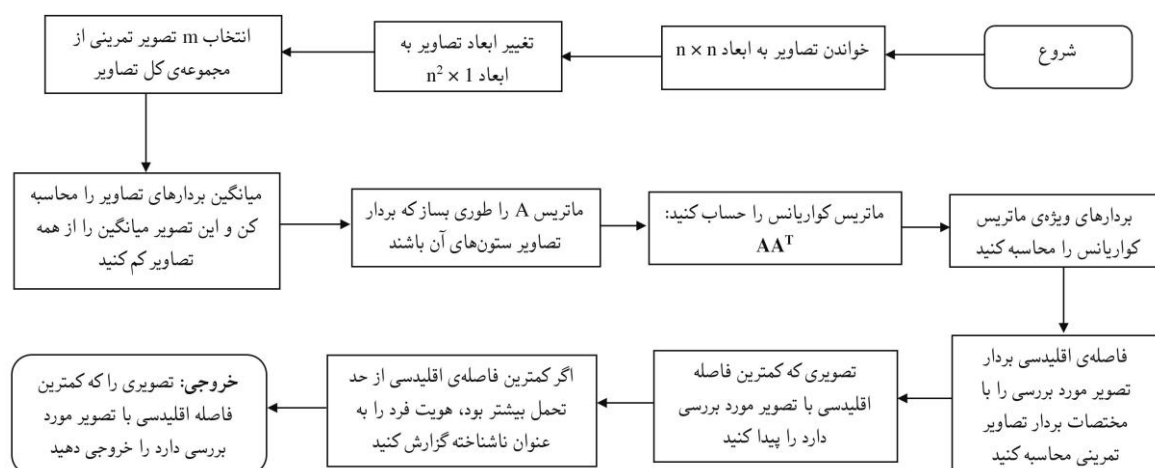


در نتیجه با توجه ساختار تصاویر در ماشین، می توان گفت که در حوزه ی بینایی ماشین به دنبال یافتن الگو در ماتریس ها و تنسورها هستیم به همین دلیل نفوذ جبرخطی عددی در این زمینه بسیار بالاست و حتی قبل از ابداع شبکه های عصبی، این حوزه الگوریتم های غنی ای داشت که از الگوریتم های جبرخطی عددی الهام گرفته شده بود.

۳-۱- چهره های ویژه

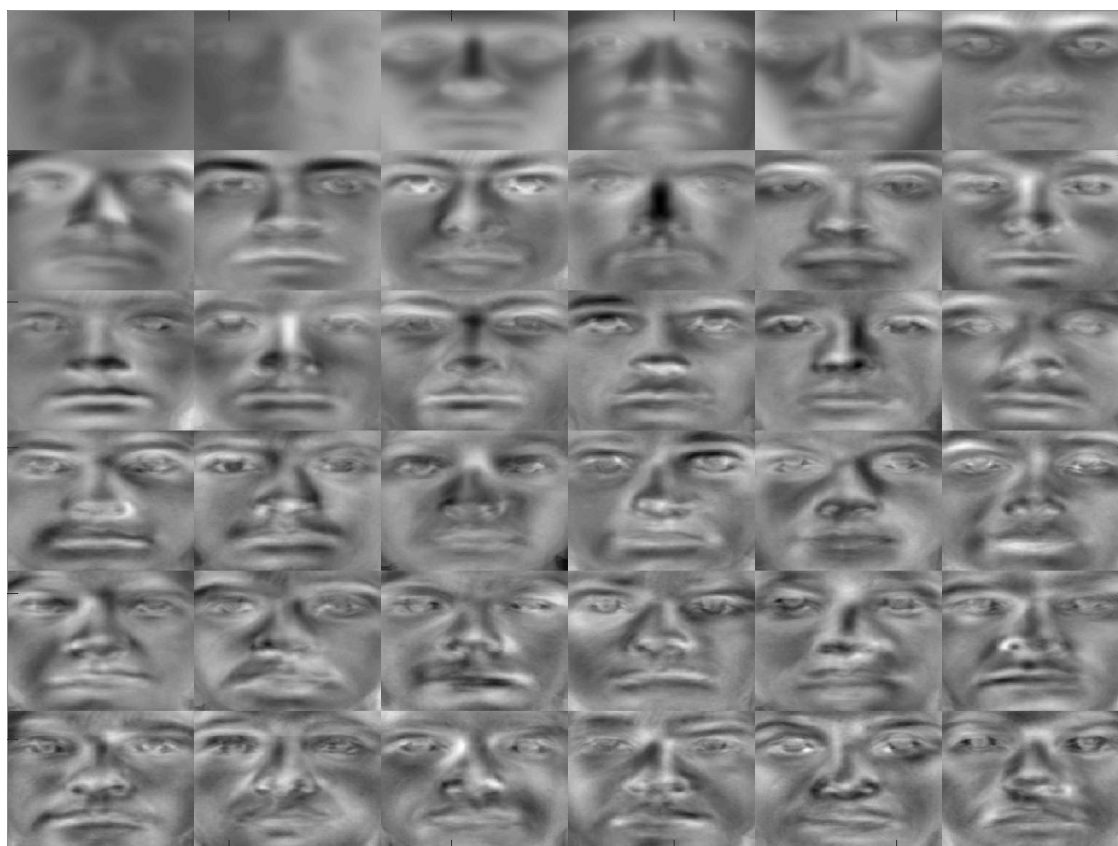
فرض کنید مجموعه ای تمرینی طبقه بندی شده از تصاویر چهره ها را در اختیار داشته باشیم و بخواهیم که الگوریتم ما هویت تصاویری از چهره که در مجموعه تمرینی نیست را مشخص کند.

در این روش برای تشخیص هویت در تصاویر، از روش تحلیل مولفه های اصلی بهره می گیرد، بدین صورت که با اعمال PCA بر روی تصاویر بردار شده، بردارهای پایه ای برای فضای برداری حاصل از تصاویر پیدا می کند، سپس مختصات تصویر چهره ی جدیدی که می خواهیم هویت آن را مشخص کنیم، بر اساس این پایه ها محاسبه می کند و هرکدام از تصاویر مجموعه تمرینی که مختصاتش به مختصات چهره مورد پیش بینی نزدیک تر بود را به عنوان هویت شخص اعلام می کند. برای افزایش سرعت و پایداری محاسباتی این الگوریتم از تجزیه SVD در پیاده سازی آن استفاده می شود. شکل زیر مراحل این الگوریتم را نشان می دهد.



لازم به ذکر است که پایه های فضای تصاویر چهره ها که در اصل همان ستون های ماتریس U در تجزیه ی SVD هستند را چهره های ویژه می نامند و دلیل نامگذاری این الگوریتم نیز همین است.

از آنجا که طبق تجزیه خواص تجزیه SVD ستون های ماتریس U متعامد هستند، در نتیجه طیف وسیعی از چهره ها را می توان با ترکیب بردارهای چهره های ویژه تولید کرد. همچنین از آنجا که مقادیر منفرد متناظر با ستون های اول ماتریس U بزرگتر هستند، در نتیجه اولین (چپ ترین) چهره های ویژه خصوصیات کلی تری از چهره را در بر می گیرند و چهره های ویژه آخر اطلاعات بسیار ریزتری را در بر می گیرند، به طوریکه بیشتر به حالت نویز تصاویر هستند تا چهره. تصویر زیر ۳۶ چهره ویژه اول را در یک بار انجام این الگوریتم بر روی یک پایگاه داده از تصاویر نشان می دهد:



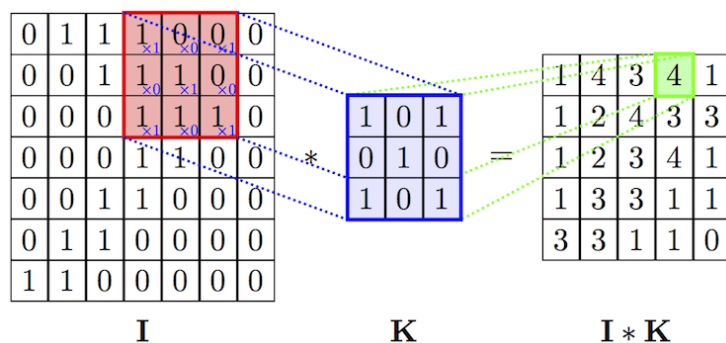
جذابیت این الگوریتم آن است که برخلاف روش شبکه های عصبی یک راهکار تکراری نیست و زمان انجام آن قابل پیش بینی است. همچنین با وجود آنکه این الگوریتم اولین برای تشخیص چهره طراحی شده است، اما در عمل هر وظیفه ی طبقه بندی تصاویر رو می توان تنها با تغییر مجموعه تمرینی تصاویر انجام داد. به عنوان کافیسست به جای تصاویر افراد و شماره ی فرد، تصاویر غده های سرطانی خوش خیم و بدخیم را به آن بدهیم تا بدون تغییر الگوریتم تصاویر غده ها را طبقه بندی کند. البته این خصوصیت باعث می شود که این الگوریتم برخلاف شبکه های عصبی، قابل توسعه و انعطاف پذیر نباشد.

۳-۲- شبکه های عصبی همگشتی

همانطور در بخش پردازش زبان طبیعی توضیح دادیم، ابداع شبکه های عصبی انقلابی در حوزه هوش مصنوعی بود، اما شبکه های عصبی کاملاً متصل به چند دلیل کارایی کافی برای پردازش تصویر را نداشتند:

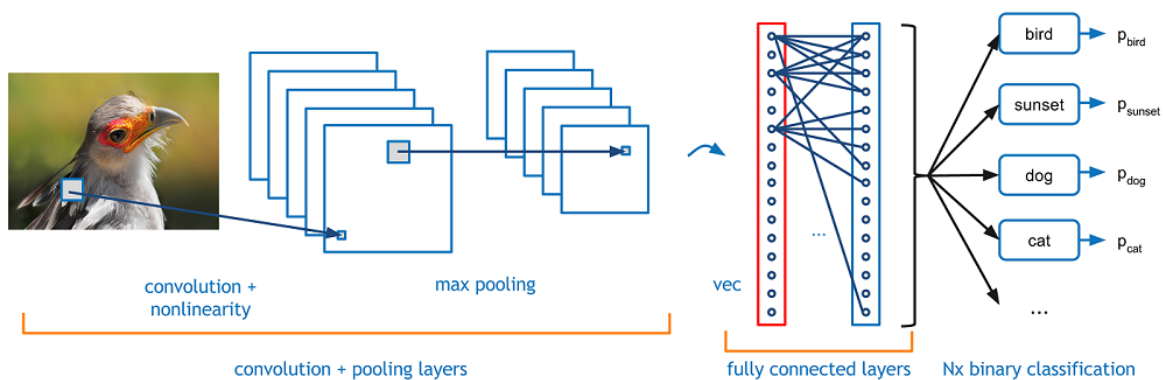
- برای آنکه شبکه ای کاملاً متصل بسازیم که تصاویر را به عنوان ورودی دریافت کند، نیاز به تعداد بسیار زیادی پارامتر داریم، به عنوان مثال برای یک تصویر 64×64 در فرمت RGB باید شبکه ما دارای 13872 مقدار ورودی بپذیرد که اگر لایه بعدی حتی ۱۲۸ واحد هم داشته باشد، برای اتصال این دو لایه نیاز به ۱۷۷۵۶۱۶ پارامتر خواهیم داشت که برای تنها یک لایه بسیار زیاد است و نه تنها زمان آموزش مدل، بلکه زمان خروجی گرفتن از آن نیز غیرمعقول خواهد بود.
- با بردار کردن تصویر می تواند آن را به عنوان ورودی به یک شبکه عصبی داد، اما آنگاه پارامترهایی که میان هر پیکسل از تصویر تا لایه بعدی هستند، باید به صورت مستقل الگوهای تکراری را یاد بگیرند. به صورت خلاصه در شبکه های عصبی کاملاً متصل، اشتراک پارامتر میان پیکسل مختلف تصویر وجود ندارد تا الگوهای تکراری در نقاط مختلف تصویر با یک پارامتر شناسایی شوند

در شبکه های عصبی همگشتی که از عملگر همگشت استفاده می کند، هر دو مشکل بالا رفع می شود. در عمل همگشت یک فیلتر داریم که بر روی تصویر حرکت کرده و ضرب عنصر به عنصر میان فیلتر و هر بخش تصویر صورت می گیرد و در نهایت مجموع مقادیر محاسبه می شود. شکل زیر یک مرحله از عمل همگشت را نشان می دهد که در آن K فیلتر و I تصویر ورودی است.



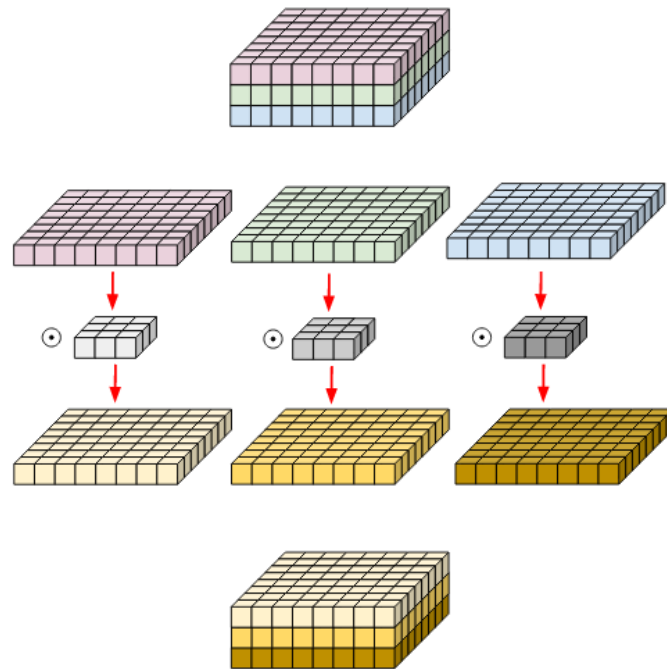
در یک لایه همگشتی از شبکه، یک یا چند فیلتر بر روی تصویر اعمال شده و نتایج آنها در کنار یکدیگر قرار می‌گیرد تا به لایه بعدی وارد شود. در مرحله پس انتشار نیز مقادیر هر فیلتر به صورت یک پارامتر مستقل، با گرادیان خود جمع می‌شود.

در نتیجه در ساختار یک شبکه عصبی همگشتی، فیلترها بر روی تصویر اعمال شده و در انتهای آن زمانی که تعداد خروجی ها کمتر بود، یک لایه کاملاً متصل قرار می‌گیرد. در حقیقت از عمل همگشت برای استخراج اطلاعات از تصویر و از لایه کاملاً متصل برای تصمیم گیری و طبقه بندی استفاده می‌شود.

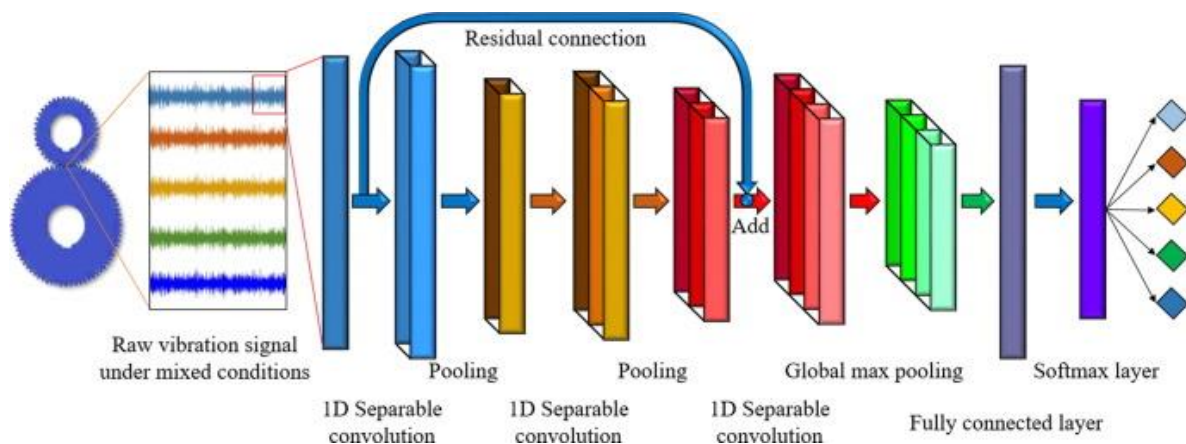


در تصویر بالای نمونه‌ای از یک شبکه عصبی همگشتی آورده شده است. در شکل به لایه های جمع‌آوری بیشینه نیز اشاره شده است که در حقیقت مانند لایه‌های همگشتی هستند که به جای عمل همگشت، مقدار بیشینه ای را که در داخل قاب فیلتر قرار می‌گیرد را خروجی می‌دهند. این لایه پارامتری برای آموزش ندارد و صرفاً برای دور ریختن خانه های با اطلاعات کمتر و کوچک کردن ورودی لایه های بعد هستند.

در شبکه های عصبی همگشتی، برای افزایش سرعت عمل همگشت، از فیلترهای تک بعدی به جای فیلترهای چند بعدی نیز استفاده می‌شود که به آن لایه همگشتی متکی بر عمق می‌گویند. در مقاله‌ای که این نوع لایه همگشتی پیشنهاد شده، گفته شده است که سرعت آن تقریباً ۱۰ برابر سرعت لایه همگشتی عادی است. در این نوع از لایه های همگشتی، بر هر کانال از تصویر به صورت جدا یک فیلتر متفاوت اعمال می‌شود. این لایه در ساختار شبکه عصبی MobileNet مورد استفاده قرار گرفته است و هدف طراحی این شبکه قابل استفاده بودن برای تلفن های هوشمند است. یک مرحله از لایه همگشتی متکی بر عمق در شکل زیر آورده شده است.



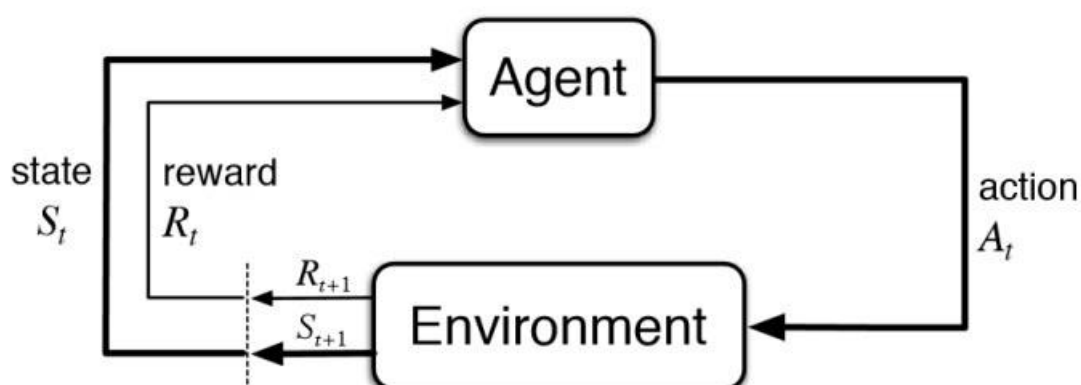
ساختار خاص دیگری که در طراحی شبکه های همگشتی کاربرد دارد، استفاده از اتصالات رو به جلو است که برای جلوگیری از مشکل گرادیان محو شونده در شبکه های عصبی همگشتی به کار می رود. اتصال رو به جلو به این صورت است که برای کوتاه کردن مسیر حرکت گرادیان در شبکه، خروجی یک لایه همگشتی را به خروجی چند لایه جلوتر از آن که اندازه خروجی یکسانی دارد، اضافه می کنیم. نمونه ای از این اتصال در شبکه زیر با رنگ آبی نشان داده شده است.



فصل چهارم

الگوریتم های رباتیک

در زمینه رباتیک اکثر الگوریتم‌های پرکاربرد مربوط به حوزه یادگیری تقویتی هستند. در یادگیری تقویتی یک عامل داریم که قرار است با محیط اطرافش تعامل کند تا هدفی را دنبال کند. برای پیگیری آنکه عامل هدف را خوب دنبال می‌کند یا نه، به ازای هر تصمیمی که عامل اتخاذ می‌کند یک پاداش به آن می‌دهیم، در اینصورت به طور خلاصه می‌توان گفت که عامل ما سعی در بیشینه کردن پاداش را خواهد داشت. همچنین در مرحله برای آنکه عامل ما تصمیم‌گیری کند باید بتواند محیط اطرافش را مشاهده کند و بدین ترتیب می‌تواند گفت عامل ما تابعی است که با دریافت مشاهدات و پاداش یک عمل را خروجی می‌دهد. شکل زیر نمودار کلی آنچه گفتیم را نشان می‌دهد.



آنچه که یادگیری تقویتی را جذاب می‌کند آن است که تصمیم‌عامل بر انتخاب یک عمل باعث تغییر ورودی‌های بعدی آن خواهد شد در حالیکه این خاصیت در بقیه حوزه‌های یادگیری ماشین وجود ندارد. الگوریتم‌های یادگیری تقویتی با توجه به اینکه نحوه ساز و کار محیط را بدانیم یا نه، به ۲ دسته‌ی زیر تقسیم می‌شوند:

۱. مستقل از محیط (Model-Free)

۲. متکی بر محیط (Model-Base)

الگوریتم های متکی بر محیط زمانی قابل استفاده هستند که از نحوه کار و قواعد محیط اطلاع داشته باشیم. در نتیجه الگوریتم های مستقل از محیط، الگوریتم های جامع تری هستند که حتی در شرایطی که از نحوه کار محیط اطلاع داریم هم می توانیم با نادیده گرفتن نحوه کار محیط، از این الگوریتم ها استفاده کنیم. الگوریتم های برنامه نویسی پویا جزو این بخش محسوب می شوند.

اما الگوریتم های مستقل از محیط بسیار پیچیده تر و جذاب تر هستند، زیرا اطلاعی از محیط ندارند و صرفا به دنبال پیشنهاد کردن جایزه بر اساس مشاهدات هستند، به همین دلیل کاربرد این الگوریتم ها بسیار گسترده و جذاب است. در ادامه به انواع الگوریتم های مستقل از محیط می پردازیم.

۴-۱- الگوریتم های مبتنی بر ارزش

تقریبا در تمامی الگوریتم های مبتنی بر ارزش از معادله بلمن برای یافتن بهترین حرکت استفاده می شود. در ساده ترین الگوریتم این دسته یک Q-table ساخته می شود که هر خانه آن متناظر با یک جفت وضعیت و عمل است که مقدار Q آن عمل در وضعیت موردنظر، در جدول قرار می گیرد. البته که در ابتدا مقادیر Q برابر صفر قرار داده می شود و به کمک معادله بلمن و به صورت تکراری این مقادیر تقریب زده می شود. این الگوریتم به Q-Learning معروف است.

اما مشکل بسیار بزرگ الگوریتم Q-Learning آن است که برای محیط های با تعداد وضعیت های ممکن بسیار زیاد، حجم حافظه بسیار زیادی نیاز خواهد داشت. ایده قدرتمندتر که از شبکه های عصبی استفاده می کند آن که یک شبکه عصبی طراحی می کند که یک وضعیت از محیط را گرفته و ارزش آن را خروجی می دهد، بدین صورت همان محاسباتی که در Q-Learning انجام می شد را بر روی خروجی شبکه ی عصبی اعمال می کنیم. بدین شکل در اصل سعی داریم که Q-function را به کمک شبکه عصبی و با برازش تقریب بزنیم که ارزش هر وضعیت را تعیین کند. در این الگوریتم معمولا از شبکه های عصبی عمیق استفاده می شود و به همین دلیل به شبکه ی آن Deep Q-Network نیز می گویند.

همچنین در DQN مشکل گرادیان انفجاری بسیار مشاهده می شود و متداول است که از ۲ شبکه برای تقریب مقادیر Q استفاده شود تا واریانس مقادیر ورودی کاهش یابد.

۴-۲- الگوریتم های مبتنی بر سیاست

در الگوریتم های مبتنی بر ارزش، پس از تقریب مقادیر تابع Q نیاز داریم که در هر وضعیت به ازای هر عمل، مقدار Q وضعیت بعدی را محاسبه کنیم و به همین دلیل الگوریتم های مبتنی بر ارزش در محیط هایی که تعداد عمل های مجاز زیاد است، بسیار کند خواهند بود. اما در الگوریتم های مبتنی بر سیاست هدف ما آن است که بر اساس مشاهدات عامل یک توزیع احتمالاتی ارائه دهیم که به ازای هر عمل، احتمال مطلوب بودن آن عمل را به ما بدهد، سپس به راحتی عملی که بیشترین احتمال مطلوب بودن را دارد انتخاب می کنیم.

معروف ترین الگوریتم این دسته REINFORCE است که زیرمجموعه ی روش های تصادفی مونته کارلو قرار می گیرد. در این روش به دلیل انتخاب تصادفی عمل از توزیع احتمالاتی خروجی، دیگر با مشکل تعادل میان اکتشاف و بهره برداری مواجه نخواهیم بود، اما به دلیل تصادفی بودن این روش، مشکل گرادیان های انفجاری رخ می دهد که باعث تغییرات زیاد دقت الگوریتم شود.

۴-۳- الگوریتم های بازیگر و منتقد

همانطور که گفتیم الگوریتم های مبتنی بر سیاست اکثرا با مشکل گرادیان انفجاری مواجه هستند. در الگوریتم های هنرمند و منتقد، غیر از یک شبکه عصبی که توزیع احتمالاتی مطلوب بودن حرکات را خروجی می دهد و به آن بازیگر نیز می گوییم، یک شبکه عصبی نیز برای تقریب مقدار Q عمل انتخاب شده توسط شبکه هنرمند می سازیم که با انجام اعمال ریاضی بر روی گرادیان شبکه بازیگر به کمک خروجی شبکه دوم، سعی می کنیم که واریانس گرادیان را کم کرده تا انفجاری بودن آن کنترل شود. به شبکه عصبی دوم که مقدار Q را تقریب می زند، منتقد می گویند.

نکته جالب آن است که ممکن است تنها یک شبکه عصبی با ۲ خروجی طراحی کرده و یک خروجی را به عنوان خروجی بازیگر و خروجی دیگر را خروجی منتقد در نظر بگیریم. این ساختار به خصوص زمانی که نیاز به استفاده از لایه های همگشتی داریم بسیار کاربرد دارد.

فصل ششم

جمع‌بندی و نتیجه‌گیری

امروزه هوش مصنوعی و یادگیری ماشین روز به روز در حال تحول و پیشرفت هستند، به طوریکه کمتر کسی جرئت آن را دارد که به راحتی عملکردی از انسان را برای کامپیوترها غیرقابل دستیابی بداند. به طور مثال امروزه حتی با روی کار آمدن مدل‌هایی مانند DALL-E2 کامپیوترها در هنر هم در حال رقابت با انسان هستند.

در این گزارش پیشرفت‌های دانشمندان در زمینه طراحی الگوریتم‌هایی برای تقلید از توانایی‌های دیداری، شنیداری و حرکتی را بررسی کردیم اما چالش اصلی حوزه هوش مصنوعی طراحی الگوریتم‌هایی برای تفکر و حل مساله است. به عنوان مثال طراحی مدل Copilot توسط مایکروسافت یکی از قدم‌های در راستای تقلید از قوه تفکر انسانی است.

با وجود تمام چالش‌ها، باید منتظر ماند و دید الگوریتم‌های نوین طراحی شده چگونه قوه تفکر انسانی را تقلید خواهند کرد؟ آیا از الگوریتم‌های امروزی الهام خواهند گرفت و یا الگوریتم‌های کاملاً متفاوتی مانند تغییرشکل دهنده‌ها که در قسمت پردازش زبان طبیعی دیدیم روی کار خواهند آمد؟

منابع و مراجع

- He, K. Z. (2016). Deep residual learning for image recognition. *In Proceedings of the IEEE conference on computer vision and pattern recognition*.
- Hochreiter, S. a. (1997). Long short-term memory. *Neural computation*, 46.
- Howard, A. G. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv*.
- John Schulman, F. W. (2017). Proximal Policy Optimization Algorithms. *arXiv*.
- Özen., M. Ç. (2012). A face recognition system based on eigenfaces method. *Procedia Technology*. doi:<https://doi.org/10.1016/j.protcy.2012.02.023>
- Pentland, M. T. (1991). Eigenfaces for recognition. *Journal of cognitive neuroscienc.*
- Strang, G. (1993). *Introduction to linear algebra. Vol. 3*. Wellesley-Cambridge Press.
- Vaswani, A. S. (2017). "Attention is all you need." *Advances in neural information processing systems* 30.