باسمه تعالی

دانشگاه صنعتی اصفهان

دانشکده برق و کامپیوتر

گزارش کارآموزی تابستان ۱۴۰۲

نام و نام خانوادگی کارآموز: حوری دهش

شماره دانشجویی: ۹۸۲۱۴۱۳

استاد کارآموزی: دکتر مهران صفایانی

سرپرست کارآموزی: خانم زهره جعفری

محل کارآموزی:

آدرس: شهرک علمی تحقیقاتی اصفهان

تلفن: ۰۹۱۲۰۹۴۹۸۶۵

تاریخ شروع : ۱۴۰۲/۴/۱۷ تاریخ پایان: ۱۴۰۲/۶/۱۴

فهرست مطالب

چکیده

یادگیری عمیق زیرمجموعه‌ای از یادگیری ماشین است که بر شبکه‌های عصبی عمیق تاکید دارد. این شبکه‌ها ساختارهایی چندلایه دارند که به طور مشابه به ساختار مغز انسان ترتیب داده شده‌اند همچنین یادگیری عمیق به ماشین‌ها امکان می‌دهد که از داده‌های ورودی، الگوها و ویژگی‌های پیچیده را به طور خودکار و بدون نیاز به برنامه‌ریزی دستی استخراج کنند. این شبکه‌ها می‌توانند با تعداد بسیار زیادی از ورودی‌ها و خروجی‌ها کار کنند. این تکنولوژی به طور مداوم در حال توسعه است و برای حل مسائل پیچیده هوش مصنوعی مانند تشخیص تصاویر پزشکی، پیش‌بینی رفتار مصرف‌کنندگان، و بهبود خودران‌شدن ربات‌ها از اهمیت بالایی برخوردار است.

یادگیری عمیق در پزشکی به عنوان یکی از زمینه‌های مهم و نوظهور هوش مصنوعی به‌شمار می‌آید و توانمندی‌های بسیاری را برای بهبود تشخیص بیماری‌ها، پیش‌بینی بیماری‌ها و افزایش دقت در درمان فراهم کرده است. همینطور یکی از کاربردهای بزرگ یادگیری عمیق در پزشکی، تشخیص تصاویر پزشکی مانند تصاویر رادیولوژیک (از جمله اشعه ایکس، CT Scan و MRI) و تصاویر میکروسکوپی است. مدل‌های یادگیری عمیق می‌توانند کمک کنند تا بیماری‌هایی مانند سرطان را در مراحل اولیه تشخیص دهند. در نهایت یادگیری عمیق در زمینه سرطان، به ویژه سرطان‌های مغزی، به دلایل متعددی به تحقیقات و بهبود تشخیص و درمان این بیماری‌ها کمک کرده است که این دلایل عبارتند از: تشخیص دقیق‌تر سرطان مغزی، تصویربرداری مولکولی، پیش‌بینی نتایج درمان، کاهش خطاها و آسیب‌های جانبی، تحقیقات و پیشرفت‌های دارویی و کمک به پزشکان در تصمیم‌گیری‌های بالینی.

۱- فصل اول: معرفی محل کارآموزی

شرکت فرزانگان هوش محور در سال ۱۴۰۱ تاسیس شده است، این شرکت در شهرک علمی و تحقیقاتی اصفهان قرار دارد و حداکثر ۵ نفر در این شرکت کار می‌کنند. مدیر عامل این شرکت سرکار خانم فاطمه مصطفائی هست.

زمینه‌های فعالیت این شرکت شامل فناوری اطلاعات و ارتباطات، هوش مصنوعی و پردازش تصویر است. همینطور ایده محوری این شرکت در مورد طراحی و ساخت دستگاه تفکیک جنسیت جوجه یک روزه مبتنی بر الگوریتم‌های هوش مصنوعی است. در نهایت با hooshmehvar@gmail.com می توان با این شرکت ارتباط برقرار کرد.

۲- فصل دوم: کارهای انجام شده در دوره کارآموزی

به طول کلی فعالیت‌هایی انجام شده در طول دوره کارآموزی به موارد زیر تقسیم‌بندی می‌شود:

- آموزش دوره deep learning
- استفاده پزشکان از پلتفرم‌هایی که در زمینه هوش مصنوعی هستند و تجاری‌سازی شده‌اند
- ساختن یک مدل برای دیتاست Br۳۵H
- مطالعه در مورد دیتاست BraTS ، segmentation و معماری U-Net

۲-۱- آموزش دوره deep learning

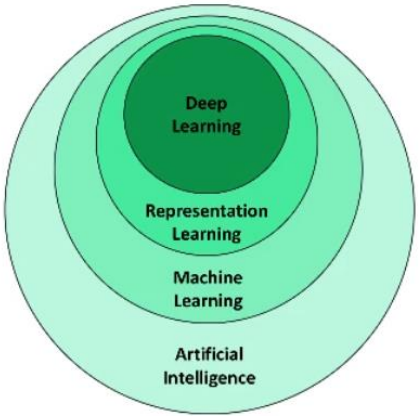
آموزش دوره deep learning توسط حسین امیرخانی توی آپارات انجام شد.

مفاهیمی که از این آموزش یاد گرفته شد عبارتند از:

- آشنایی با مفهوم یادگیری عمیق
- چگونگی کار انجام دادن یادگیری عمیق
- آشنایی با دیتاست‌های MNIST، IMDB و Reuters
- k-fold cross validation
- شاخه های یادگیری ماشین
- نشت اطلاعات
- Underfitting و Overfitting
- Max-pooling و Convolution
- Pretrained network

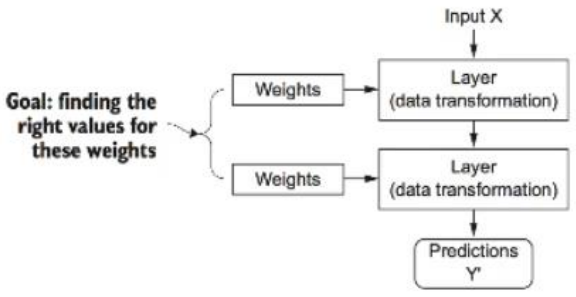
تعریف یادگیری عمیق

یادگیری عمیق یک زیرمجموعه از یادگیری ماشین است و یادگیری ماشین یک زیرمجموعه از هوش مصنوعی است. اگر بخواهیم دقیق‌تر به این موضوع اشاره کنیم به صورت شکل زیر بیان می‌شود:



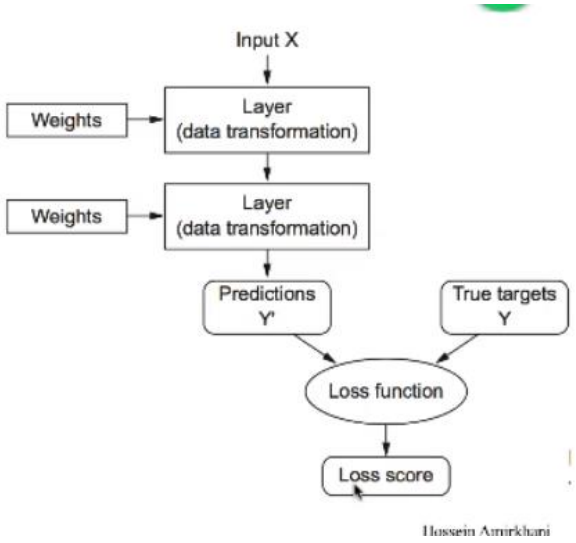
به‌طورکلی deep یک لایه نیست و چند لایه بازنمایی را دارد یاد میگیرد و همینطور هر چی عمیق‌تر می‌شود لایه‌های آخر یا عمیق‌تر به تسک مورد نظر نزدیک‌تر می‌شود مثلا توی مثال تشخیص چهره لایه اول خط‌های چهره را یاد می‌گیره و لایه دوم مثلا کناره‌های تصویر رو یاد می‌گیرد و لایه آخر مثلا نشون میده این حسن هست یا نیست پس هر چقدر توی عمق پیش برود انگار به تسک مورد نظر نزدیکتر می‌شود.

به تعداد لایه‌ها هم عمق مدل گفته می‌شود مثلا اگر سه تا لایه وجود دارد گفته می‌شود عمق این مدل سه است. یادگیری عمیق چطوری کار انجام می‌دهد؟ داده وارد یک لایه شده و بازنمایی‌اش عوض می‌شود یعنی یک بازنمایی جدید پیدا می‌کند. بازنمایی هر لایه به یک مجموعه پارامتر بستگی دارد یعنی به هر کدوم از لایه‌های ما یک weights اختصاص داده می‌شود یعنی لایه اول weights خاص خودش رو داره و لایه دوم هم همینطور و تا آخر به همین صورت پیش می‌رود. برای درک بهتر، شکل زیر این موضوع را نشان می‌دهد:



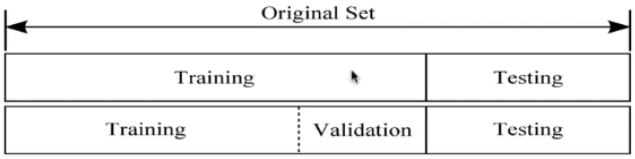
Weights یعنی یک مجموعه‌ای از اعداد که این اعداد کنترل می‌کنند که بازنمایی مورد نظر چطوری انجام بشود پس عدد ها اگر عوض بشوند عملکرد آن لایه هم عوض می‌شود یعنی اگر بخواهد لایه مورد نظر یک بازنمایی دیگه‌ای یاد بگیرد باید weights اش را عوض بکند پس در حالت کلی با عوض کردن weights بازنمایی هم عوض می‌شود.

مدل مورد نظر باید این weights ها را یاد بگیرد پس چطوری این weights ها یاد گرفته میشوند؟ برای هر ورودی سیستم یک خروجی به ما داده می‌شود و همینطور یک true targets هم وجود دارد و طبق این true targets می‌توان به سیستم فیدبک داد و طبق این فیدبک مدل یاد می‌گیرد که چطوری وزن‌هایش را تغییر بدهد پس کاری که سیستم می‌کند این است که با یک فانکشنی به اسم loss function یک مقایسه‌ای بین predictions و true targets انجام می‌دهد و هدف این است که تفاوت بین predictions و true targets کم بشود در نهایت باید این وزن‌ها طوری تغییر بکنند که loss مورد نظر تا حد ممکن کم بشود پس در مرحله بعد loss score را میگیرد و سعی می‌کند با تغییر دادن همه این وزن‌ها loss را کمینه کند در نهایت می‌توان گفت loss score به عنوان یک سیگنال فیدبکی استفاده می‌شود که وزن‌های مدل را در جهتی که این loss score کم بشود، تنظیم بکند و این فرایند چندین بار روی هزاران نمونه تکرار می‌شود برای اینکه آن ماشین یاد بگیرد. در شکل زیر این موضوع نشان داده شده است:



نکته‌ای که وجود دارد این است که اینطوری نیست که کل دیتاست یک‌جا برای آموزش به شبکه داده شود پس در حالت کلی نمی‌توان دیتاست را یکجا یا یکی یکی به شبکه داد بلکه باید یه حالت بینابین در نظر گرفته شود مثلا batch های ۱۲۸تایی یا ۲۵۶تایی را می‌توان در نظر گرفت. همینطور اگر کل دیتا یکجا به شبکه داده شود آپدیت بهتری توی یه قدم اتفاق می‌افتد اما خیلی زمان می‌برد تا همون یه قدم انجام شود پس به صورت batch دیتا به شبکه داده می‌شود.

داده مورد نظر همیشه به دو بخش اصلی تقسیم می‌شود به نام‌های: training , testing و معمولا داده training هم به دو بخش تقسیم می‌شود به نام‌های: training , validation پس در حالت کلی سه تا مجموعه وجود دارد که به صورت زیر است:



نکته‌ای که وجود دارد این است که در حین training دقت سیستم را روی داده validation نگاه می‌کنند چون نیاز هست که متوجه شوند مدل روی یک داده ندیده چطوری عمل میکند پس به جای اینکه ارزیابی مدل روی داده تست انجام بشود روی داده validation صورت می‌گیرد. پس روی داده train آموزش انجام می‌شود و روی داده validation ارزیابی انجام میشود برای اینکه متوجه بشوند مدل خوب است یا خیر در نهایت روی داده تست فقط یک بار ارزیابی نهایی انجام می‌شود.

دیتاست MNIST:

یک دیتاستی از ۶۰ هزار تصویر آموزشی از رقم‌های دست‌نویسی است. با این دیتاست قصد داریم به سیستم، این عددها را یاد بدهیم یعنی به سیستم ۶۰ هزار نمونه آموزش داده می‌شود و بعد روی ۱۰ هزار نمونه دیگر که بهش داده تست می‌گویند ارزیابی مدل انجام می‌شود.

دیتاست IMDB:

در این دیتاست می‌خواهیم کامنت‌هایی که کاربران زیر یک سری فیلم نوشته‌اند را classify کنیم به یکی از دو کلاس positive یا negative یعنی می‌خواهیم تحلیل احساسات روی کامنت‌ها انجام دهیم پس اینجا مسئله ما binary classification است. توی این مسئله از ۵۰۰۰۰ داده استفاده شده که ۲۵۰۰۰تای آن‌ها داده آموزش و ۲۵۰۰۰تای دیگر هم داده تست هستند و هر کدوم از این ۲۵۰۰۰تای داده تست و آموزش نصفشون positive است و نصف دیگر هم negative.

توی این دیتاست به هر کلمه‌ای یک عدد اختصاص داده می‌شود مثلا برای کلمه book ۱ در نظر گرفته شده و برای بقیه کلمات هم به همین صورت در نهایت کلمات را با اعداد جایگزین می‌کنند.

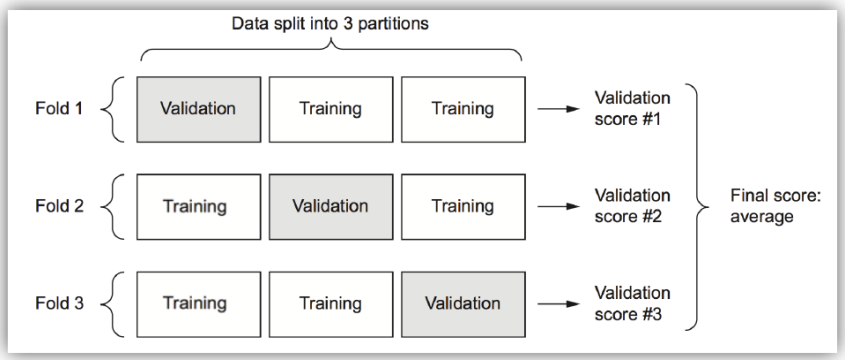
دیتاست Reuters:

در این دیتاست یک سری تیکه‌های خبری وجود دارد و می‌خواهیم ببینیم آن خبر راجع‌به چه موضوعی است پس اینجا یک مسئله multiclass classification وجود دارد. توی این دیتا ست ۴۶ تا topic هست پس یک مسئله ۴۶ کلاسه است و هر خبری به یکی از این ۴۶ تا topic تعلق دارد. در این دیتاست کلاس‌ها توزیع یکنواخت ندارند یعنی دیتامون imbalanced است و کلاس‌ها هم اندازه نیستند یعنی بعضی از topicها بیشتر از بقیه توی دیتا وجود دارند ولی دیتا اینجا طوری تهیه شده است که از هر کلاسی حداقل ۱۰ تا نمونه توی داده آموزشی وجود دارد.

نکته‌ای که وجود دارد این است که اگر دقت محاسبه شده نزدیک ۵۰ درصد بود این یک دقت بدی است چون عملکرد مدل نزدیک به رندوم می‌شود که در این حالت مدل چیزی یاد نگرفته است . از آنجایی که توی این مسئله ۴۶ تا کلاس وجود دارد و مدل باید از بین ۴۶ تا کلاس انتخاب بکند پس اگر رندوم انتخاب بکنه ۱/۴۶ احتمال داره که درست انتخاب بکند که توی این حالت دقت ۲ درصد می‌شود.

k-fold cross validation

قبل از اینکه مدل train بشود روی داده‌ها باید یک داده validation ساخته شود. یکی از راه‌ها برای ساختن داده validation زمانی که تعداد داده‌های آموزشی بسیار کم است، استفاده از روش k-fold cross validation است. منطق این روش این است داده را به تعداد k قسمت تقریبا مساوی تقسیم می‌کند مثلا توی شکل زیر k-fold سه بوده است. بعد از آن که تقسیم‌بندی انجام شد آزمایش k بار انجام می‌شود که در شکل زیر سه بار می‌شود. در آزمایش اول بخش اول validation و دو قسمت بعدی train در نظر گرفته می‌شود در نهایت validation این قسمت به ما یک امتیازی می‌دهد دفعه بعدی قسمت دوم را validation در نظر می‌گیرند و این آزمایش هم به ما یک امتیازی می‌دهد و سری بعدی هم قسمت آخر را validation می‌گیرند و این قسمت هم باز به ما یک امتیازی میدهد در نهایت روی این امتیازات میانگین می‌گیرند.



شاخه‌های یادگیری ماشین

۱- یادگیری بانظارت یا supervised learning

۲- یادگیری بدون‌نظارت یا unsupervised learning

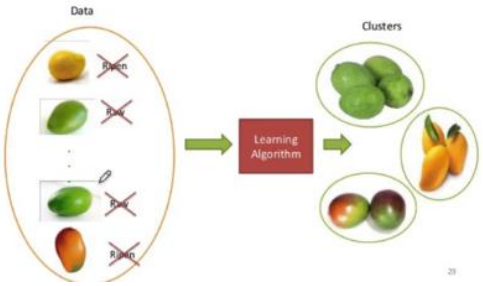
۳- یادگیری تقویتی یا reinforcement learning

یادگیری بانظارت

به سیستمی که می‌خواهد یاد بگیرد ورودی که داده می‌شود هم شامل features است و هم شامل targets مثلا توی مسئله طبقه‌بندی ارقام که با داده MNIST کار انجام می‌شد به سیستم گفته می‌شد این شکل عدد مثلا یک است و... یعنی به سیستم گفته می‌شد این ورودی باید این خروجی را بدهد پس به عنوان یک ناظر داریم به سیستم می‌گوییم که تو برای این ورودی باید این خروجی را به ما بدهی بخاطر همین بهش یادگیری بانظارت می‌گویند و وظیفه سیستم این است که یاد بگیرد.

یادگیری بدون‌نظارت

در این حالت یک مجموعه داده به سیستم داده می‌شود ولی targets داده نمی‌شود و هدف این است که ساختارهای جذاب را توی این مجموعه داده، سیستم تشخیص بدهد. خوشه‌بندی یک مثال معروف از مسائل یادگیری بدون‌نظارت است و کاری که داخلش انجام می‌شود این است که داده‌های مورد نظر را خوشه خوشه می‌کند مثلا توی شکل زیر سه تا خوشه پیدا شده است.



یادگیری خود نظارتی یا self-supervised learning

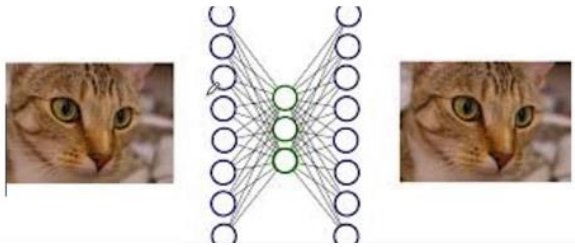
این یک یادگیری بانظارت است با این تفاوت که سیستم خودش به نمونه‌هاش label می‌زند یعنی خود سیستم یادگیر میاد و به نمونه‌ها label می‌زند. یکی از نمونه‌های موفقش Autoencoderها است.

Autoencoder

در Autoencoderها یک شبکه عصبی ساخته می‌شود که ورودی شبکه یک تصویر است و خروجی شبکه هم همان تصویر است یعنی به شبکه گفته می‌شود زمانی که این تصویر به عنوان ورودی بهت داده شد همین تصویر را به صورت خروجی بهمون تحویل بده. مزیت این شبکه این است که یک لایه ای این وسط قرار داده می‌شود که در شکل زیر بخش سبز رنگ است و کاری که آن لایه انجام می‌دهد این است که آن تصویر را از فضای پیکسلی به فضای میانی می‌آورد یعنی تصویر را بازنمایی می‌کند و بعد از این بازنمایی تصویر اصلی را می‌سازد پس این لایه در واقع کاری که انجام می‌دهد این است که بازنمایی مفیدی را از تصویر یاد می‌گیرد. می‌توان به این موضوع از دو دید نگاه کرد:

از نگاه بانظارت : چون label دارد و سیستم یاد میگیرد که برای این ورودی این خروجی را بدهد.

از نگاه بدون‌نظارت : چون label انسانی ندارد و یک مجموعه تصاویر بدون label به سیستم داده شده است.



یادگیری تقویتی

یک عامل یا agent وجود دارد که آن عامل میاد و یک اطلاعاتی راجع‌به محیط دریافت میکند و به مرور یاد می‌گیرد که چطوری عمل بکند که بیشترین پاداش را به دست بیاورد پس عامل از محیط دوتا چیز دریافت میکند:

۱- استیت محیط یعنی الان محیط توی چه استیتی است.

۲- پاداش یا جریمه از محیط دریافت میکند و کم کم یاد میگیرد که چجوری رفتار بکند که بیشترین پاداش را بدست بیارد.

نشت اطلاعات

چرا در مجموعه داده، فقط train , test وجود ندارد؟ چون اتفاقی که می‌افتد این است که هر بار که ارزیابی روی داده تست انجام میشود، متوجه می‌شویم که مثلا تعداد نورون‌ها را باید زیاد بکنیم یا تعداد لایه‌ها را کم بکنیم در نهایت طبق این شریط مدلی درست می‌شود و هر بار طبق این داده تست، مدل بهبود پیدا می‌کند و همین موضوع باعث می‌شود که هر بار کمی از اطلاعات از داده تست نشت پیدا کند یعنی اطلاعات داده تست در داده آموزش دارد نشت پیدا می‌کند

در نهایت ممکن است سیستمی به وجود بیاید که روی داده تست خیلی خوب جواب بدهد ولی وقتی در عمل از آن استفاده می‌شود ممکن جواب خوبی ندهد.

نکته‌ای که وجود داد این است که یکی از راه‌هایی که باعث overfitting می‌شود استفاده بیش از حد از داده validation است.

برای رفع این مشکل کاری که باید انجام شود این است که روی داده train آموزش انجام بشود و بعد روی داده validation ارزیابی صورت بگیرد و براساس این موضوع مدل ساخته شود ولی نباید زیاد از داده validation استفاده کرد در نهایت بعد از اینکه آموزش تموم شد فقط یک بار روی داده تست مدل ارزیابی می‌شود.

برای اینکه کار evaluation انجام شود سه تا روش کلی وجود دارد:

۱- simple hold-out : در این روش داده شافل می‌شود و یک بخشی را برای train، یک بخشی را برای validation و یک بخش دیگری هم برای test گذاشته می‌شود.

۲- k-fold validation : وقتی داده ها کم است از این روش می‌روند که بالاتر گفته شد.

۳- iterated k-fold validation : همان k-fold است ولی چند بار انجام می‌شود یعنی مثلا ۵ بار k-fold زده می‌شود و بعد میانگین را می‌گیرند.

در آخر باید مدل نهایی روی همه داده‌های غیر تست ساخته شود یعنی داده train , validation.

Underfitting و Overfitting

underfitting

امکان بهبود هنوز وجود دارد و شبکه هنوز همه patternهای مرتبط را یاد نگرفته است به عبارت دیگر کمتر از حد به داده fit شده است.

overfitting

در این حالت شبکه شروع کرده patternهایی را یاد گرفته که خاص شبکه آموزشی هستند ولی برای داده‌های جدید به درستی کار نمی‌کند به عبارت دیگر بیش از حد به داده fit شده است و همینطور overfitting از همه lossاش کمتر است.

یک ریشه overfitting این است که داده‌های ما کم است همینطور مدل اگر پیچیدگی‌اش از یک حدی بیشتر شود به سمت overfitting می‌رود و اگر از یک حدی کمتر شود به سمت underfitting می‌رود.

توی deep learning مسئله اصلی که باهاش روبرو هستیم overfitting است چون معمولا سیستم‌های عمیق پیچیدگی خیلی بالایی دارند. بهترین کار برای اینکه این مشکل حل شود این است که داده‌ها را زیاد کنیم ولی بعضی وقت‌ها خود این زیاد کردن داده مشکل اصلی می‌شود پس راه حل بعدی این است که مدل را regularization کنیم و همینطور می‌توان به صورت دستی پیچیدگی مدل را کم کنیم.

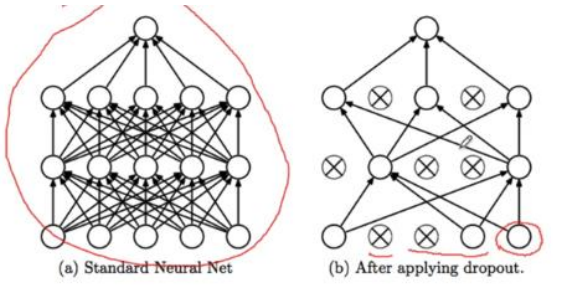
regularization

میزان اطلاعاتی که شبکه اجازه دارد یاد بگیرد را محدود کنیم. سه تا تکنیک برای regularization وجود دارد:

۱- reducing the network's size: اندازه شبکه را کوچیک کنیم (یعنی تعداد لایه‌ها یا نورون‌ها را کاهش بدهیم) وقتی شبکه کوچیک می‌شود توانایی یادگیری‌اش و پیچیدگی‌اش هم کم می‌شود.

۲- weight regularization: اجازه ندهیم وزن‌ها خیلی آزادانه تغییر بکنند (یعنی پارامترهای شبکه‌های را محدود کنیم)

۳- dropout: این یک تکنیک جدیدی است. همینطور می دانیم overfitting نتیجه هماهنگ شدن خاص یک سری از featureها است پس برای اینکه جلوی این کار گرفته شود از dropout استفاده می‌شود. ایده این روش این است که توی فرایند یادگیری توی هر لحظه‌ای یک تعدادی از نورون‌ها حذف بشود پس برای هر داده‌ای که می‌خواهد train انجام شود به طور تصادفی یک تعدادی از نورون‌ها را dropout حذف می کنیم و این باعث میشه جلوی overfitting گرفته شود. شکل زیر این موضوع را نشان می‌دهد:



این که چه درصدی dropout می‌شود با یک نرخی به اسم dropout rate مشخص می‌گردد و معمولا یک عددی بین ۲ درصد و ۵ درصد است. اگر نرخ dropout خیلی زیاد باشد ساختار شبکه خیلی ساده می‌شود و به سمت underfitting حرکت می کند و اگر خیلی کم باشد ساختار شبکه به سمت overfitting پیش می‌رود.

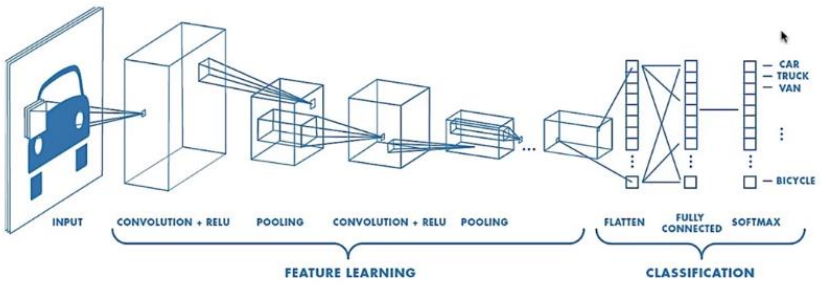
Max-pooling و Convolution

تا اینجای کار با سه تا لایه آشنایی پیدا کردیم: لایه‌های Dense و Max-pooling و Convolution

یکی از تفاوت‌هایی که لایه‌های max-pooling و convolution با لایه dense دارند این است که ورودی لایه dense از جنس وکتور بوده ولی ورودی لایه‌های max-pooling و convolution، tensorهایی هستند با ابعاد $(height, width, channels)$

به طور کلی شبکه ما دو بخش دارد: در بخش اول که داخلش لایه‌های max-pooling و convolution پشت سر هم قرار دارد تصویر وارد می‌شود و چیزی که در نهایت از بخش اول خارج میشود به بخش دوم داده می‌شود که شامل لایه dense است سپس خروجی شبکه از بخش دوم دریافت می‌شود.

کاری که بخش اول شبکه انجام میدهد کار feature learning است یعنی چیزی که از بخش اول خارج میشه featureهای خوبی (یعنی featureهای سطح بالا که می‌تواند کلاس‌ها را از هم جدا بکند) از این تصویر هستند و کاری که بخش دوم شبکه انجام می‌دهد مپ کردن featureها به کلاس‌ها است. در شکل زیر این موضوع نشان داده شده است:



نکته‌ای که وجود دارد این است که زمانی که از بخش اول شبکه می‌خواهیم به بخش دوم شبکه برویم باید از یک لایه flatten استفاده کنیم بخاطر اینکه خروجی که بخش اول شبکه به ما داده یک tensor است که متشکل از عرض و طول و channel است ولی لایه dense به وکتور (یعنی یک tensor یک بعدی) نیاز دارد که لایه flatten این تبدیل را برای ما انجام می‌دهد.

تفاوت اصلی که بین لایه dense و convolution وجود دارد این است که لایه dense گلوبال patterns را یاد می‌گیرد در حالی که لایه convolution لوکال patterns را یاد می‌گیرد.

از max-pooling زمانی استفاده می‌شود که می خواهیم ابعاد را کم کنیم.

تفاوتی که max-pooling با convolution دارد این است که max-pooling پارامتری ندارد یعنی از اولش مشخص است که توی max-pooling قراره چه اتفاقی بیوفته ولی توی convolution مشخص نیست برای همین یک w داره و این w باید یاد گرفته بشود.

برای اینکه مشکل overfitting حل بشود می‌توان بی نهایت داده به شبکه اضافه کرد که همان تکنیک data augmentation است یعنی از هر تصویری بی نهایت تصویر ایجاد بشود ولی بعضی وقت‌ها با این روش هم باز مشکل حل نمی‌شود پس به‌جای اینکه داده را زیاد بکنند از شبکه‌های پیش آموزش داده شده (Pretrained network) استفاده می کنند.

Pretrained network

یک شبکه ذخیره شده‌ای است که قبلا روی یک مجموعه بزرگ آموزش داده شده است و ایده‌اش این است اگر دیتاست بزرگی که قبلا روی آن آموزش دیده شده است کلاس‌هایش با دیتاستی که اکنون داریم متفاوت باشد باز هم مشکلی نیست چون اگر آن دیتاست به اندازه کافی بزرگ و عمومی باشد در این صورت آن سلسله مراتب featureهایی که توسط شبکه Pretrained یاد گرفته شده، در مسئله‌های دیگر قابل استفاده است. مثلا یک مجموعه داده بزرگی داریم به اسم ImageNet و روی این دیتاست آموزش دیده‌ایم و یک مدل ساخته شده است که به این مدلPretrained network گفته می‌شود بعد یک مجموعه کوچیک‌تری داریم مثل همین دیتاست سگ و گربه که الان می‌خواهیم برایش یک مدل بسازیم که مفهوم کلاس سگ را از کلاس گربه جدا بکند یک راه این است که روی همین دیتاست سگ و گربه بیایم یک مدل بسازیم و یک راه دیگر هم این است که بیایم از مدلی که روی مجموعه داده بزرگ آموزش داده شده و ممکن است با دیتاست ما متفاوت باشد به یک نحوی از این مدل استفاده بکنیم یعنی دانشی که توی یک مسئله دیگر کسب شده است این دانش را به نحوی transfer بکنیم توی یک مسئله دیگر که به این کار transfer learning یا یادگیری انتقالی گفته می‌شود.

برای این که از شبکه های Pretrained استفاده بشود دو تا روش کلی وجود دارد :

۱- روش اول این است که از آن شبکه برای feature extraction استفاده بشود یعنی آن شبکه پیش آموزش داده شده به عنوان یک ابزاری استفاده میشه برای اینکه برای ما یک feature بیاد extract بکنه که خود این روش توی دوتا رویکرد انجام میشود:

۱- با data augmentation

۲- بدون data augmentation

۲- روش دوم fine-tuning است.

۲-۲- استفاده پزشکان از پلتفرم‌هایی که در زمینه هوش مصنوعی هستند و تجاری‌سازی شده‌اند

پلتفرم‌هایی که مورد استفاده قرار گرفته‌اند عبارتند از:

1. Aidoc
2. Mindstrong Health
3. Cogniciti
4. BrainScope
5. Epihunter

Aidoc

یک نرم‌افزار مجهز به هوش مصنوعی است که برای تجزیه و تحلیل تصاویر رادیولوژی و MRI برای کمک به تشخیص سریع بیماری‌های مغزی استفاده می‌شود. این نرم‌افزار در سال ۲۰۱۶ تاسیس شد. Aidoc برای الگوریتم‌های سکته مغزی، آمبولی ریوی، شکستگی دهانه رحم، خونریزی داخل جمجمه، گاز آزاد داخل شکمی و آمبولی ریوی اتفاقی، تاییدیه FDA و CE را دریافت کرده است و الگوریتم‌های آن در بیش از ۹۰۰ بیمارستان و مرکز تصویربرداری، مرکز پزشکی دانشگاه روچستر و مرکز پزشکی شبا استفاده می‌شود.

توضیح Aidoc روی خونریزی داخل جمجمه

خونریزی داخل جمجمه (ICH) زمانی رخ می‌دهد که یک رگ خونی بیمار در مغز می‌ترکد و اجازه می‌دهد خون به داخل مغز نشت کند. مدلی که اینجا مطرح می‌شود مبتنی بر تکنیک‌های یادگیری عمیق است که می‌تواند برای طبقه‌بندی ICH در بیماران از برش‌های تصویر توموگرافی کامپیوتری (CT) استفاده کند.

روش نقشه برداری فعال‌سازی کلاس (Grad-CAM)، که در آن توضیحات بصری را از طریق محلی سازی مبتنی بر گرادیان ارائه می‌دهند. این رویکرد یک تصویر را به عنوان ورودی دریافت می‌کند و از گرادیان‌های هر نورون خروجی در یک شبکه طبقه‌بندی که به لایه کانولوشنی نهایی جریان دارد، استفاده می‌کند تا یک نقشه حرارتی ایجاد کند که مناطق

مربوطه را در تصویر برجسته می‌کند تا پیش‌بینی مربوطه را انجام دهد. Grad-CAM شکلی از توجه پس از آن است یعنی روشی است که برای یک شبکه عصبی قبلا آموزش دیده اعمال می‌شود. ایده اصلی پشت این روش استفاده از اطلاعات مکانی است که از طریق شبکه عصبی حفظ می‌شود تا پیدا شود که کدام قسمت‌های یک تصویر ورودی برای تصمیم‌گیری طبقه‌بندی مهم هستند.

Grad-CAM یک نقشه حرارتی ارائه می‌دهد که بر روی پیکسل‌هایی تمرکز می‌کند که بیشترین تاثیر را در تصمیم‌گیری دارند.

معماری که اینجا به کار می‌رود ResNet است که از سلول‌های هرمی در قشر مغز الهام گرفته شده است. ResNet سعی می‌کند با معرفی بلوک‌های باقیمانده این مشکلات را کاهش دهد. این بلوک‌ها یک اتصال مستقیم را معرفی می‌کنند، که برخی از لایه‌ها را در میان می‌گذراند. این اتصال، اتصال پرش نامیده می‌شود و هسته بلوک‌های باقیمانده است. استفاده از ResNet در این مدل با EfficientDet یکپارچه شده است که شبکه هرمی ویژگی دو جهته وزنی (BiFPN) را با یک روش مقیاس‌بندی ترکیبی، ترکیب می‌کند پس اولین معماری اساسی مورد استفاده در این مدل ResNet است و دومین EfficientDet. انگیزه اصلی این معماری بهینه‌سازی منابع مدل یادگیری عمیق برای تشخیص اشیاء تصویر دیجیتال است.

مجموعه داده ما شامل ۷۵۲۷۹۹ برش اسکن در قالب تصویربرداری دیجیتال و ارتباطات در پزشکی، از ۱۸۹۳۸ بیمار است و مجموعه بیماران به سه گروه به طور تصادفی تقسیم شده‌اند، گروه های: train (٪۹۰، ۱۷۰۴۴ بیمار)، validation (٪۵، ۹۴۷ بیمار) و test (٪۵، ۹۴۷ بیمار). با این حال، توجه به این نکته مهم است که عدم تعادل قابل توجهی وجود دارد زیرا گروه‌ها، شامل تعداد برش‌های زیر هستند:

۱. train: ۹۷۵۲۵ با ICH، ۵۸۰۹۳۴ بدون ICH
۲. validation: ۵۴۰۱ با ICH، ۳۱۱۷۴ بدون ICH
۳. test: ۵۰۰۷ با ICH، ۳۲۷۵۸ بدون ICH

برای کاهش این عدم تعادل، به طور تصادفی همان تعداد برش غیر ICH انتخاب می‌شود که برش‌های ICH به ۱۹۵۰۵۰ برش از ۱۷۰۴۴ بیمار برای آموزش تبدیل می‌شود. ۱۰۸۰۲ برش از ۹۴۷ بیمار برای validation و ۱۰۰۱۴ برش از ۹۴۷ بیمار برای training است. علاوه بر این، از آنجایی که شبکه عصبی ما سه کانال را به عنوان ورودی می‌گیرد، اسکن‌ها

با ماتریس‌های HU باید از قبل پردازش شوند تا سه ماتریس با اندازه ۵۱۲ × ۵۱۲ پیکسل برای هر کانال به دست آید. برای به دست آوردن این سه ماتریس، سه پنجره مختلف برای هر برش اعمال می‌شود.

اعمال پنجره‌ای با X به عنوان کران پایین و Y به عنوان کران بالا به این معنی است که تمام مقادیر HU پایین تر از X به X تبدیل می‌شوند، تمام مقادیر HU بزرگتر از Y به Y تبدیل می‌شوند و بقیه ثابت می‌مانند.

در نهایت برای پیاده سازی Grad-CAM، از کتابخانه pytorch-grad-cam استفاده می‌شود.

Mindstrong Health

Mindstrong در سال ۲۰۱۴ با تمرکز بر تشخیص و درمان بیماری‌های روانی از طریق قدرت هوش مصنوعی و گوشی‌های هوشمند راه‌اندازی شد. این شرکت از هوش مصنوعی و حسگرهای غیرفعال برای ردیابی علائم سلامت روان استفاده کرد سپس Mindstrong به ارائه خدمات سلامت روان مبتنی بر اپلیکیشن روی آورد.

این نرم‌افزار هوش مصنوعی برای نظارت بر رفتار مغز از طریق استفاده از گوشی هوشمند برای کمک به تشخیص و پیش‌بینی بیماری‌های مغزی مانند افسردگی و اضطراب استفاده می‌شود همینطور Mindstrong Health با استفاده از هوش مصنوعی و تحلیل داده، به پزشکان و تیم‌های درمانی کمک می‌کند تا رفتارها و الگوهای مغزی بیماران را مانیتور کنند و اطلاعات مفیدی درباره وضعیت روانی و شناختی آنها کسب کنند. این اطلاعات می‌تواند در تشخیص و مدیریت بیماری‌های مغزی، بهبود روند درمانی، و پیشگیری از تشدید علائم بیماری‌ها کمک کند.

عملکرد دقیق نرم‌افزار Mindstrong Health به صورت زیر است:

۱. مانیتورینگ روانشناختی: این نرم‌افزار از طریق تحلیل رفتارها و الگوهای مغزی کاربران از طریق تلفن‌همراه، اطلاعاتی درباره حالت روانی و شناختی آن‌ها کسب می‌کند. به کمک این اطلاعات، پزشکان می‌توانند تغییرات در روند روانشناختی بیماران را مشاهده کرده و به موقع واکنش نشان دهند.
۲. پیش‌بینی بیماری‌های مغزی: نرم‌افزار Mindstrong Health با استفاده از الگوریتم‌های هوش مصنوعی و یادگیری ماشینی، قادر است به صورت زودهنگام علائم بیماری‌های مغزی مانند افسردگی و اضطراب را پیش‌بینی کند. این امر می‌تواند به پزشکان کمک کند تا درمان‌ها و مداخلات مناسب‌تری انجام دهند و از تشدید بیماری جلوگیری کنند.
۳. شناخت پزشکان: اطلاعات جمع‌آوری شده توسط این نرم‌افزار به پزشکان کمک می‌کند تا با دقت بیشتر مشکلات بیماران را شناخته و برنامه‌های درمانی بهتری را برای هر فرد تنظیم کنند.

۴. بهبود مداخلات: نرم‌افزار Mindstrong Health به پزشکان اجازه می‌دهد تا تاثیر مداخلات درمانی را ارزیابی کنند و در صورت لزوم برنامه‌ها را تغییر دهند تا بهترین نتایج را برای بیماران به‌دست آورند.

نحوه استفاده از Mindstrong

Health توسط Mindstrong، در پس‌زمینه تلفن اجرا می‌شود و نحوه لمس تلفن هنگام استفاده از آن را ردیابی می‌کند. برای روشن بودن، آنچه را که در تلفن خود انجام می‌دهید ردیابی نمی‌کند، فقط نحوه انجام آن را ردیابی می‌کند. برای مثال، برنامه می‌تواند نحوه تایپ شما را ردیابی کند، اما نمی‌تواند آنچه را که تایپ می‌کنید ببیند. این بدان معنی است که می‌تواند الگوهای استفاده از تلفن را تشخیص دهد.

چه اتفاقی برای داده‌های سلامت می‌افتد؟

هنگامی که شروع به استفاده از برنامه Health می‌کنید، برنامه تعاملات صفحه نمایش شما را ضبط می‌کند و آنها را به یکی از سرورهای Mindstrong می‌فرستد و همینطور تمام داده‌ها رمزگذاری می‌شود. هنگامی که در سرور قرار گرفت، داده‌های جمع‌آوری‌شده توسط برنامه، توسط الگوریتم‌های یادگیری ماشینی که به پنج نشانگر زیستی برای درک سلامت مغز شما نگاه می‌کنند، تجزیه و تحلیل می‌شوند. این نشانگرهای زیستی عبارتند از: عملکرد اجرایی، کنترل شناختی، حافظه فعال، سرعت پردازش و ظرفیت عاطفی

Cogniciti

Cogniciti یک شرکت فناوری بهداشتی (HealthTech) است که در زمینه تشخیص زودهنگام بیماری آلزایمر فعالیت می‌کند. این شرکت در کانادا در سال ۲۰۱۱ توسط Baycrest Health Sciences تاسیس شد. Baycrest Health Sciences یک موسسه پژوهشی و درمانی معتبر در زمینه آسایش و بهبود کیفیت زندگی کارکنان و ساکنان سالمند است. هدف اصلی Cogniciti توسعه نرم‌افزارهای هوش مصنوعی برای تشخیص زودهنگام بیماری آلزایمر و کمک به تحسین زندگی افراد مبتلا به این بیماری می‌باشد. این اپلیکیشن از تست‌های شناختی برای کمک به کاربران برای شناسایی علائم اولیه بیماری و دریافت درمان‌های مناسب استفاده می‌کند.

Cogniciti یک تست رایگان و دارای اعتبار علمی است که توسط دانشمندان موسسه تحقیقاتی روتمن ساخته شده است تا به بزرگسالان بالای ۴۰ سال کمک کند که آیا تغییرات حافظه آنها باید توسط پزشک ارزیابی شود یا خیر. این آزمون ۲۰ دقیقه‌ای کاملا خصوصی است و شامل سوالات و کارهای ساده برای ارزیابی عملکرد مغز و حافظه است.

شرکت‌کنندگان گزارشی دریافت می‌کنند که نتایج آن‌ها را با سایر آزمایش‌کنندگان مقایسه می‌کند و مهم‌تر از همه، به آنها توصیه می‌کند که آیا باید به دنبال ارزیابی بیشتر از پزشک خود باشند یا نباشند.

ابزار ارزیابی آنلاین Cogniciti اکنون توسط بیش از ۶۵۰۰۰ بزرگسال در ۵۰ کشور استفاده شده است.

توضیح Cogniciti

به طور معمول، Cogniciti از دو بخش اصلی تشکیل شده است:

۱. ارزیابی شناختی: Cogniciti ممکن است ابزارها و تست‌هایی را ارائه دهد که به ارزیابی شناخت فرد کمک می‌کنند. این ارزیابی‌ها می‌توانند معیارهایی را در مورد حافظه، تمرکز، توجه و دیگر قابلیت‌های شناختی فرد ارائه دهند. با استفاده از این ارزیابی‌ها، افراد می‌توانند اطلاعاتی در مورد وضعیت شناختی خود دریافت کنند.
۲. تمرینات شناختی: Cogniciti ممکن است برنامه‌ها و تمرینات خاصی ارائه دهد که به بهبود و تقویت حافظه و شناخت کمک می‌کنند. این تمرینات ممکن است شامل بازی‌ها و فعالیت‌های ذهنی مختلفی باشند که به تقویت تمرکز، حافظه و دیگر عملکردهای شناختی کمک می‌کنند.

استفاده از Cogniciti ممکن است به صورت آنلاین انجام شود و افراد می‌توانند از طریق وب سایت یا برنامه‌های تلفن همراه به این سرویس دسترسی پیدا کنند.

BrainScope

BrainScope یک شرکت فناوری بهداشتی است که در زمینه تشخیص آسیب‌های مغزی و اختلالات عصبی فعالیت می‌کند. این شرکت به توسعه فناوری‌های هوش مصنوعی و تحلیل داده متمرکز است که برای ارزیابی سریع و غیرتهاجمی وضعیت مغز بیماران با آسیب‌های مغزی استفاده می‌شود همینطور BrainScope در سال ۲۰۰۶ تاسیس شد و دفتر مرکزی آن در Bethesda، مریلند واقع شده است.

BrainScope به‌ویژه در حوزه تشخیص آسیب‌های مغزی مرتبط با ضربه‌های سر و اختلالات عصبی، به ویژه در مواقع اورژانسی، فعالیت دارد. از جمله استفاده‌های این فناوری می‌توان به تشخیص موارد ضربه‌های سری خفیف، متوسط و شدید، تشخیص شوک و آسیب‌های عصبی اشاره کرد.

این فناوری ابتدا از طریق اسکن‌های مغزی و تصاویر MRI، اطلاعات جمع‌آوری می‌کند و سپس با استفاده از الگوریتم‌های هوش مصنوعی و تحلیل داده، اطلاعات مغز بیمار را تحلیل و بررسی می‌کند. با توجه به نتایج به‌دست آمده،

این فناوری به پزشکان و تیم‌های درمانی اطلاعاتی درباره وضعیت مغز بیمار ارائه می‌دهد و در تصمیم‌گیری‌های درمانی کمک می‌کند.

BrainScope با ارائه روش‌های مقیاس‌پذیر و قابل استفاده در محیط‌های اورژانسی، به تشخیص و مدیریت سریع‌تر و دقیق‌تر آسیب‌های مغزی کمک می‌کند و می‌تواند نقش مهمی در بهبود نتایج بالینی و افزایش ایمنی بیماران با مشکلات مغزی داشته باشد.

BrainScope یک دستگاه پزشکی چندوجهی است که توسط FDA پاکسازی شده است که به پزشکان توانایی شناسایی آسیب‌های مغزی خفیف (mTBI) ساختاری (خونریزی‌های مغزی) و عملکردی (تکان‌های مغزی) را می‌دهد. این دستگاه از داده های الکتروانسفالوگرافی (EEG)، هوش مصنوعی، فناوری یادگیری ماشین و عملکرد شناختی و ارزیابی‌های بالینی برای شناسایی نشانگرهای عینی mTBI استفاده می‌کند. در زیر نمایی از شکل این دستگاه نشان داده شده است:

توضیح BrainScope

t-SNE (تعبیه همسایگی تصادفی توزیع شده) یک روش کاهش ابعاد غیر خطی است. قدرت t-SNE از این واقعیت ناشی می‌شود که سعی می‌کند به طور دقیق همسایه‌های محلی نقاط را نشان دهد بنابراین همسایگان در نمودار با داده‌های با ابعاد بالا اصلی مطابقت دارند. در عوض فواصل بین نقاط غیرمشابه، به خوبی حفظ می‌شوند. t-SNE برای تولید نقشه‌های رونویسی از مناطق مغز در اطلس مغز آلن (ABA) استفاده شده است و در تجزیه و تحلیل داده‌های مولکولی تک سلولی محبوب است. در اینجا از BrainScope استفاده شده که پورتالی است که از نقشه‌های t-SNE از نمونه‌ها و ژن‌ها در تجسم تعاملی چشم‌انداز رونویسی مغز، استفاده می‌کند. بر اساس داده‌های مغز انسان موسسه آلن یک نمای وسیع مغزی و رونویسی از بیان ژن و شباهت رونویسی نواحی مغز ارائه می‌دهد. این امکان را برای تجزیه و تحلیل تعاملی بیان ژن در مغز انسان به روش بصری فراهم می‌کند. برای اتصال نماهای ژنومرکز و نمونه محور، از نقشه‌های پیوندی استفاده می‌شود. اولین نمونه از این موضوع کاوشگر دوگانه‌ای است که دارای نقشه asingletrunscriptome-widegenemapanda-

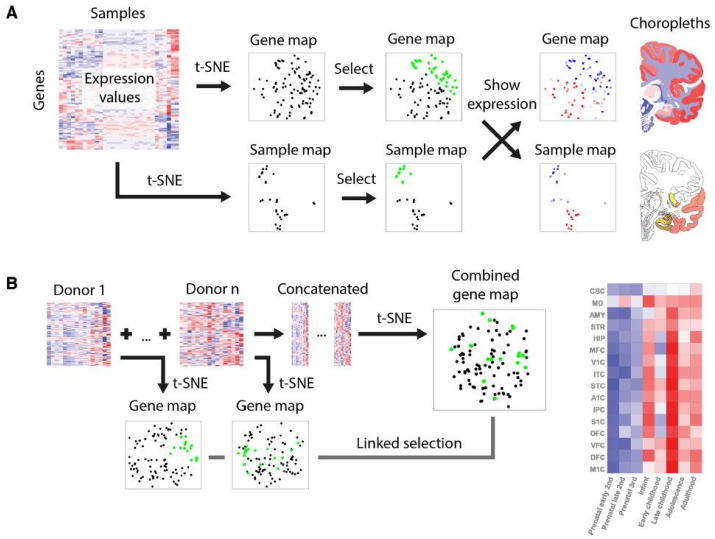
wadesample مغز است. کاربران می توانند ژن‌ها یا نمونه‌ها را انتخاب کنند و الگوهای میانگین خود را در نقشه دیگر نشان دهند. علاوه بر این، این بخش از پورتال حاوی choroplethهای مغز است (برش‌های انتخاب شده توسط کاربر از مغز انسان که برای بومی سازی نمونه‌ها و نشان دادن الگوهای بیان فضایی استفاده می‌شود.) علاوه بر کاوشگر دوگانه، پورتال حاوی کاوشگر مقایسه‌ای است که بر مقایسه چندین نقشه ژنی تمرکز دارد که نشان دهنده مغزهای اهداکننده متمایز است.

داده‌های ژن از موسسه آلن برای علوم مغز به دست آمده است. اطلس مغز انسان بالغ، شامل اندازه گیری ژن شش اهداکننده بالغ سالم است. نمونه‌ها با استفاده از کالبد شکافی ماکرو و ریز از مناطق مشروح آناتومیکی گرفته شده است. تعداد نمونه‌ها به ازای هر اهداکننده متفاوت است، از ۳۶۳ تا ۹۴۶، با مجموع ۳۷۰۲ نمونه. مقادیر بیان در هر نمونه با یک تراشه ریز آرایه سفارشی با اندازه گیری ۵۸۶۹۲ پروب تعیین شده است. پردازش اولیه داده‌ها توسط موسسه آلن انجام شد و داده‌ها در وب سایت آن‌ها در دسترس قرار گرفت.

در مجموع از ۴۲ مغز نمونه‌برداری شد که در محدوده سنی ۸ هفته پس از لقاح تا ۴۰ سالگی قرار داشتند. تعداد نمونه‌ها در هر مغز از ۱ تا ۱۶ با مجموع ۵۲۴ نمونه مشروح آناتومیک متغیر است. بیان ژن با استفاده از توالی یابی RNA تعیین شد و مقادیر RPKM به صورت آنلاین برای ۵۲۳۷۶ ژن در دسترس است.

پیش پردازش داده‌ها

در داده‌های مغز انسان بالغ، ۵۸۶۹۲ کاوشگر با استفاده از شناسه‌های Entrez به ۱۹۹۹۲ ژن نگاشت شدند. برای ژن‌هایی که دو پروب دارند، پروبی با بیشترین واریانس انتخاب شد. برای ژن‌هایی با بیش از دو پروب، ما کاوشگری را انتخاب کردیم که بالاترین اتصال را به همه پروب‌های دیگر داشت (به عنوان مجموع همبستگی‌های پیرسون تعریف می‌شود). تعداد نمونه‌ها در هر مغز اهداکننده متفاوت است. برای فعال کردن ترکیب داده‌ها برای t-SNE دوگانه، همه مجموعه‌های بیانی کاهش یافتند تا ۱۰۵ مقدار در هر ژن داشته باشند، که مربوط به مناطق حاشیه‌نویسی شده بود که در هر مغز نمونه‌برداری شد. در نهایت، برای به دست آوردن یک ژن و نقشه نمونه در کاوشگر دوگانه، مقادیر بیان برای هر ترکیبی از نمونه و ژن در شش اهداکننده به طور میانگین محاسبه شد. کاوشگر مقایسه‌ای مغز بزرگسالان در عوض از داده‌های پردازش شده برای تمام نمونه‌های مغز استفاده می‌کند. در شکل زیر کاری که در اصل صورت می‌گیرد نشان داده شده است:



نتایج کاهش ابعاد t-SNE به مقیاس و مکان داده‌ها بستگی دارد. برای نقشه‌های ژن، همه ژن‌ها با نمره z نرمال‌سازی شدند تا میانگین صفر و انحراف استاندارد ۱ داشته باشند. کاهش ابعاد با جاسازی همسایگی تصادفی توزیع شده t (t-SNE) است که یک تکنیک تعبیه غیرخطی است. این یک نقشه با ابعاد کم از داده‌های با ابعاد بالا ایجاد می‌کند در حالی که تا حد ممکن ساختار محلی را حفظ می‌کند.

کاوشگر دوگانه شباهت رونویسی محلی را در آناتومی عصبی انسان نشان می‌دهد همینطور کاوشگر دوگانه شامل دو نقشه است: یک نقشه نمونه و یک نقشه ژن.

Epihunter

Epihunter در سال ۲۰۱۷ تاسیس شد. این شرکت متعلق به کشور بلژیک است. ایده اصلی ایجاد Epihunter برای کمک به شناسایی و تعقیب حملات صرع بر اساس فناوری‌های هوش مصنوعی و تحلیل داده بوده است. با استفاده از سنسورها و دستگاه‌های مغزی، Epihunter فعالیت‌های مغزی کاربران را نظارت می‌کند و در صورت شناسایی حملات صرع هشدارها و پیام‌های اطلاع‌رسانی، به کاربران و تیم درمانی ارسال می‌شود. این راهکار بهبود کیفیت زندگی افراد مبتلا به صرع را هدف قرار داده است.

Epihunter از فناوری‌های هوش مصنوعی و تحلیل داده استفاده می‌کند تا فعالیت‌های مغزی کاربران را نظارت کند. این نظارت شامل ثبت و تجزیه و تحلیل الگوهای مغزی است که باعث شناسایی حملات صرع می‌شود. با استفاده از سنسورها و دستگاه‌های مغزی، Epihunter اطلاعات مربوط به فعالیت‌های مغزی را جمع‌آوری می‌کند.

هدف اصلی Epihunter از استفاده از فناوری هوش مصنوعی، کمک به بهبود کیفیت زندگی افراد مبتلا به صرع است و تلاش می‌کند تا با ارائه راهکارهای موثر، زندگی روزمره آن‌ها را بهتر و مدیریت بیماری‌شان را آسان‌تر کند.

توضیح Epihunter

هدف اصلی ما در Epihunter اندازه‌گیری دقت تشخیص تشنج خاموش با استفاده از دستگاه الکتروانسفالوگرافی پوشیدنی (EEG) است.

روش‌ها

ما یک کارآزمایی بالینی فاز ۳ را با یک طرح مطالعاتی آینده‌نگر، چند مرکزی و کور انجام دادیم. ورودی دستگاه EEG one-c Hannel است که با الکترودهای خشک تعبیه شده در یک دستگاه هدبند پوشیدنی متصل به تلفن هوشمند ضبط شده است.

یافته‌هایی که به دست آمده است این است که از ۱۰۲ بیمار (۵۷ زن، میانه سنی ۱۰ سال) که مشکوک به تشنج خاموش بودند ثبت شدند در نهایت ۳۶۴ تشنج خاموش را در ۳۹ بیمار ثبت کردیم.

معیارهای ورود به مطالعه به شرح زیر بود:

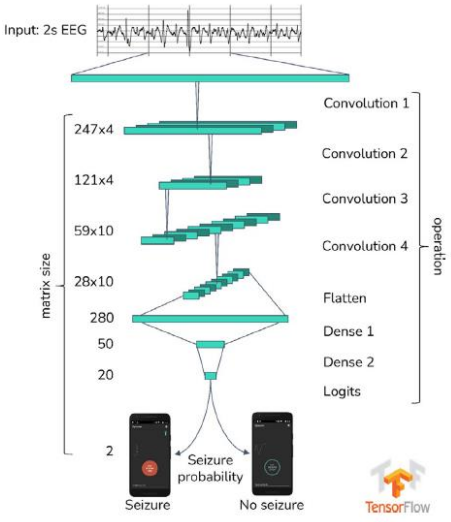
بیماران ۳ ساله یا بالاتر که مشکوک به تشنج خاموش بودند به عنوان بخشی از ارزیابی بالینی خود به نظارت ویدئویی EEG مراجعه کردند. معیارهایی که از مطالعه خارج شدند عبارت بودند از: اندازه دور سر بزرگتر از محدوده دستگاه پوشیدنی (۴۰ تا ۶۰ سانتی متر)، ناتوانی در پیروی از دستورالعمل‌ها و رفتارهایی که شامل برداشتن دستگاه قبل یا در حین ضبط بود.

EEG پوشیدنی با استفاده از دستگاه Brainlink Lite (Macrotellect)، یک هدبند EEG مصرفی مورد تایید برای استفاده در اتحادیه اروپا که می‌تواند توسط پرسنل آموزش ندیده نصب شود، ثبت شده است و از این رو برای خانه نیز مناسب است. ضبط کردن هدبند از یک ماژول NeuroSky ThinkGear ASIC (NeuroSky) استفاده می‌کند. این تراشه به سه الکترود خشک EEG (فعال، مرجع و زمین) متصل می‌شود که توسط یک نوار الاستیک روی پیشانی قرار می‌گیرد و در نتیجه یک کانال EEG دوقطبی مربوط به F۷-Fp۱.۱۱ دستگاه از طریق بلوتوث به یک واحد گوشی هوشمند برای پردازش بیشتر داده‌ها استفاده می‌شود. باتری لیتیومی ۳.۷ ولتی ۱۶۰ میلی آمپری موجود در دستگاه، ۴ تا ۵ ساعت جریان EEG مداوم را امکان پذیر می‌کند. برای نظارت بر کیفیت سیگنال‌های ثبت شده EEG، دستگاه از الگوریتمی استفاده می‌کند که ارزیابی تماس الکترود با پوست (امپدانس) و سطح نویز ناشی از عوامل محیطی یا بیولوژیکی (الکترویوگرام، الکتروکاردیوگرام، الکترواکولوگرام) را ترکیب می‌کند. کیفیت سیگنال یک بار در ثانیه در مقیاسی از ۰

(خیلی خوب) تا ۲۰۰ (بسیار بد) اندازه گیری می‌شود. اگر میانگین کیفیت سیگنال در ۵ ثانیه گذشته بیش از ۴۰ باشد، آنگاه ضعیف در نظر گرفته می‌شود. شکل دستگاه به صورت زیر است:

برای تشخیص کاملا خودکار تشنج‌های خاموش، از یک الگوریتم از پیش تعریف‌شده (که قبلا توسعه داده شده) با مقادیر قطع از پیش تعریف‌شده استفاده می‌شود. الگوریتم تشنج‌های EEG را در زمان واقعی تجزیه و تحلیل میکند و نقاط زمانی شناسایی شده را ثبت می‌کند.

شبکه عصبی از چهار لایه کانولوشن، به دنبال آن دو لایه dense و یک لایه logits نهایی تشکیل شده است. (که نمایی از آن در زیر نشان داده شده است) ما از ترکیبی از نرمال‌سازی دسته‌ای و لایه‌های حذفی برای جلوگیری از overfitting استفاده می‌کنیم. الگوریتم سیگنال‌های EEG را در پنجره‌های ۲ ثانیه، با همپوشانی ۱-s تجزیه و تحلیل می‌کند. تشنج‌های خاموش شناسایی شده به صورت خودکار به عنوان سه پنجره تشنج متوالی تعریف شده‌اند. آموزش الگوریتم قبلا بر روی ۱۴۱ ساعت داده‌های EEG بالینی از افراد مبتلا به صرع خاموش، از جمله ۲۷۱ تشنج خاموش انجام شد. برای آموزش، فقط سرنخ‌های ۷ Fp۱-F جلویی حفظ شدند، زیرا آنها نزدیک ترین مکان‌ها به الکترودهای پوشیدنی هستند.



۲-۳- ساختن یک مدل برای دیتاست Br۳۵H یا ۲۰۲۰ Brain Tumor Detection

تومور مغزی یکی از بیماری‌های تهاجمی در بین کودکان و بزرگسالان محسوب می‌شود. تومورهای مغزی ۸۵ تا ۹۰ درصد از تومورهای اولیه سیستم عصبی مرکزی (CNS) را تشکیل می‌دهند. هر سال حدود ۱۱۷۰۰ نفر به تومور مغزی تشخیص داده می‌شوند. میزان بقای ۵ ساله برای افراد مبتلا به تومور سرطانی مغز یا CNS تقریبا ۳۴ درصد برای مردان و ۳۶ درصد برای زنان است. تومورهای مغزی به چند دسته تقسیم‌بندی می‌شوند: تومور خوش خیم، تومور بدخیم، تومور هیپوفیز و... درمان مناسب، برنامه‌ریزی و تشخیص دقیق باید برای بهبود امید به زندگی بیماران انجام شود. بهترین روش برای تشخیص تومورهای مغزی، تصویربرداری تشدید مغناطیسی (MRI) است. حجم عظیمی از داده‌های تصویری از طریق اسکن‌ها تولید می‌شود. این تصاویر توسط رادیولوژیست بررسی می‌شود. یک معاینه دستی به دلیل سطح پیچیدگی‌های دخیل در تومورهای مغزی و خواص آنها می‌تواند مستعد خطا باشد.

استفاده از تکنیک‌های طبقه‌بندی خودکار با استفاده از یادگیری ماشینی (ML) و هوش مصنوعی (AI) به طور مداوم دقت بالاتری نسبت به طبقه‌بندی دستی نشان داده است. از این رو، پیشنهاد سیستمی که تشخیص و طبقه‌بندی را با استفاده از الگوریتم‌های یادگیری عمیق با استفاده از شبکه عصبی پیچیده (CNN)، شبکه عصبی مصنوعی (ANN) و آموزش انتقال (TL) انجام می‌دهد برای پزشکان در سراسر جهان مفید خواهد بود.

مجموعه داده Br۳۵H شامل سه پوشه yes، no و pred است که شامل ۳۰۶۰ تصاویر MRI مغز است. در اینجا فقط از پوشه yse و no استفاده می‌شود برای اینکه تشخیص دهد آن عکس سرطان دارد یا ندارد پس مسئله ما یک مسئله binary classification است.

پوشه yes دارای ۱۵۰۰ تصویر MRI مغز است که تومور دارند و پوشه no دارای ۱۵۰۰ تصویر MRI است که تومور ندارد.

مراحل ساختن مدل

۱. در اولین مرحله در گوگل درایو دو پوشه yes و no ایجاد می‌کنیم سپس عکس‌ها را به این دو پوشه منتقل می‌کنیم زمانی که عکس‌ها به گوگل درایو منتقل شدند در محیط colab به گوگل درایو متصل می‌شویم.
۲. در گام بعدی کتابخانه‌های مورد نیاز را فراخوانی می‌کنیم.
۳. درون گوگل درایو غیر از دو پوشه yes , no سه پوشه دیگر هم به اسم های train , validation , test ایجاد کرده و درون هر کدوم از این پوشه‌ها دو پوشه دیگر هم به اسم yes , no درست می‌کنیم و در نهایت

آدرس‌دهی مربوطه را در colab انجام می‌دهیم. (دو پوشه yes , no ایجاد شده درون سه پوشه train , validation , test بخاطر برچسب‌گذاری روی عکس‌ها است.)

۴. در مرحله بعد ابتدا باید داده‌ها را شافل کرده و در نهایت به نسبت ۷۰ درصد برای train، ۲۰ درصد برای validation و ۱۰ درصد برای test جدا می‌کنیم و درون پوشه‌های مربوطه کپی می‌کنیم در نهایت مقدارهای هر پوشه به صورت زیر می‌شود:

```
total training yes images: 1050
total training no images: 1050
total validation yes images: 300
total validation no images: 300
total test yes images: 150
total test no images: 150
```

۵. زمانی که عملیات مورد نظر که مدنظرمان بود روی داده‌ها انجام شد معماری مدل را می‌سازیم به این صورت که یک مدل sequential ایجاد کرده و بعد از آن از لایه‌های convolution , max_pooling استفاده می‌کنیم در نهایت از یک لایه flatten استفاده کرده و بعد از آن لایه‌های dense را قرار می‌دهیم.

۶. در مرحله بعد داده‌های مورد نظر را پیش‌پردازش می‌کنیم به این صورت که از کلاس ImageDataGenerator ، object می‌سازیم و درون train_datagen , test_datagen قرار می‌دهیم. در این قسمت از فانکشن flow_from_directory هم استفاده می‌شود و batch_size، train_generator را ۲۰ در نظر گرفته و برای validation_generator را هم ۵۰ می‌گیریم.

۷. در این گام مدل را fit می‌کنیم و برای آموزش epochs را ۸۰ در نظر می‌گیریم. برای این مدل یک بخش early stopping , save best model هم قرار می‌دهیم. از early stopping برای جلوگیری از overfitting استفاده می‌شود و save best model هم زمانی که مدل در بهترین حالت قرار دارد آن را ذخیره می‌کند. بخاطر اینکه ما در این بخش از early stopping استفاده کردیم آموزش مدل روی epochs=۳۴ خاتمه پیدا می‌کند و در این حالت دقت train ما ۹۹ درصد و دقت validation، ۹۷ درصد می‌شود.

۸. سپس برای مدل دو نمودار دقت و loss برای train , validation می‌کشیم.

۹. در آخر دقت test را محاسبه کرده و predict را بر روی داده تست انجام می‌دهیم و در آخر دقت تست ۹۸ درصد می‌شود که در شکل زیر می‌توان این موضوع را دید:

	precision	recall	f1-score	support
no	0.99	0.97	0.98	150
yes	0.97	0.99	0.98	150
accuracy			0.98	300
macro avg	0.98	0.98	0.98	300
weighted avg	0.98	0.98	0.98	300

ماتریس درهم‌ریختگی recall , precision در مسائل binary classification دارای ۴ خونه است:

True positive: تعداد نمونه‌هایی که به درستی به عنوان مثبت تشخیص داده شده است.

False positive: تعداد نمونه‌هایی که به عنوان مثبت تشخیص داده شده در حالی که منفی بوده است.

True negative: تعدا نمونه‌هایی که به درستی به عنوان منفی تشخیص داده شده است.

False negative: تعداد نمونه‌هایی که اشتباها به عنوان منفی تشخیص داده شده در حالی که مثبت بوده است.

در شکل زیر این موضوع نشان داده شده است:

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

فرمول‌هایی که برای محاسبه accuracy , precision , recall , f۱_score استفاده شده است شامل موارد زیر است:

$$accuracy = \frac{true\ positive + true\ negative}{true\ positive + true\ negative + false\ negative + false\ positive}$$

$$precision = \frac{true\ positive}{true\ positive + false\ positive}$$

$$recall = \frac{true\ positive}{true\ positive + false\ negative}$$

$$f1_{score} = \frac{2\ \times precision\ \times recall}{precision + recall}$$

در نهایت می‌توان کد مدل ساخته شده برای دیتاست Br۳۵H را در لینک زیر مشاهده کرد:

https://colab.research.google.com/drive/۱igG۲QH-۵SMiANFKD۵ApxjGCA۱z۴kriv۴

۲-٤- دیتاست BraTs

دیتاست BraTS یک مجموعه داده است که برای تحقیقات در زمینه تصویربرداری پزشکی و تشخیص تومورهای مغزی استفاده می‌شود. نام BraTS مخفف "BraTS Multimodal Brain Tumor Segmentation" است و به تمرکز بر روی تشخیص و تقسیم‌بندی تومورهای مغزی اشاره دارد. این دیتاست به محققان و توسعه‌دهندگان اجازه

می‌دهد تا الگوریتم‌ها و مدل‌های یادگیری عمیق را برای تشخیص و تجزیه و تحلیل تومورهای مغزی با استفاده از تصاویر پزشکی ارائه شده در آن توسعه دهند.

هدف این دیتاست تشخیص دقیق تومورهای مغزی و نشان دادن آن‌ها در تصاویر MRI است. تصاویر MRI شامل مقاطع مختلفی از مغز است که عبارتند از : T۱, T۲, T۱c, flair

T۱: این نوع تصویر به بافت‌های مختلف مغز اطلاعات می‌دهد و معمولا در تحلیل ساختارهای مغزی استفاده می‌شود.

T۲: این نوع تصویر به بافت‌های مغزی مختلف به خصوص بافت‌های غده‌ای و مایلین اطلاعات می‌دهد.

T۱c: این تصویر برای تشخیص تومورها و تحلیل تغییرات مرتبط با آن‌ها مفید است و با افزایش کانتر تصویر، این نوع تصویر به تشخیص تومورها و تغییرات مرتبط به آن ها کمک می کند.

Flair: این نوع تصویر برای تشخیص ویژه تومورها و ضایعات مغزی استفاده می‌شود.

همه این مقاطع به طور مختصر به تحلیل و تشخیص تومورها و ویژگی های مغز کمک می کند.

در این دیتاست رنگ‌های مشخصی به عنوان برچسب برای تومور و اجزای مختلف استفاده می‌شود که عبارتند از:

- رنگ آبی: معمولا نواحی تومور اصلی را نشان می‌دهد. این شامل بخش‌های اصلی تومور مغزی است.
- رنگ سبز: به عنوان ناحیه نرمال مغز و بدون تومور به کار می‌رود.
- رنگ صورتی: این رنگ ممکن است برای اجزای فرعی تومور یا نواحی که ممکن است به نوعی مرتبط با تومور باشند.
- رنگ زرد: معمولا برای نواحی مشکوک به تومور یا نواحی مرتبط با حد فاصل بین تومور و ناحیه نرمال استفاده می‌شود.

مسئله‌ای مورد نظر ما در این دیتاست از نوع segmentation است.

Segmentation -۵-۲

در پردازش تصویر و یادگیری عمیق، مفهوم Segmentation به معنای تقسیم‌بندی تصویر یا تفکیک تصویر به بخش‌های مختلف است. در واقع هدف از تقسیم‌بندی یا Segmentation، تجزیه و تحلیل تصویر به صورتی است که هر بخش از تصویر که مرتبط با یک شی و یا ویژگی خاصی است، به صورت مستقل از دیگر بخش‌ها تشخیص داده شود. برای مثال اگر ما یک تصویر پزشکی از مغز داشته باشیم و بخواهیم تومورهای مغزی را تشخیص دهیم، باید تصویر را به

گونه‌ای تقسیم کنیم که بتوانیم محل و مرز تومورها را تعیین کنیم به عبارت دیگر، در اینجا تقسیم‌بندی به معنای تفکیک تومورها از سایر بخش‌های تصویر است.

Segmentation در دسته‌های مختلفی از پردازش تصویر مورد استفاده قرار می‌گیرد، از جمله پزشکی (برای تشخیص تومورها، ساختارهای انسانی، و...)، بینایی ماشین (برای تشخیص و شناسایی اشیا و اجزای تصاویر)، تصویربرداری ماهواره‌ای (برای تفکیک مختلف مناظر زمینی)، و بسیاری دیگر از حوزه‌ها. Segmentation معمولا با استفاده از الگوریتم‌ها و مدل‌های یادگیری عمیق انجام می‌شود و در بسیاری از برنامه‌های کاربردی مهم استفاده می‌شود.

در حالت کلی دو نوع مسئله segmentation وجود دارد:

۱. Semantic segmentation: هدف طبقه‌بندی هر پیکسل در یک تصویر به یک کلاس یا شی است و تولید یک نقشه تقسیم‌بندی پیکسلی متراکم از یک تصویر است که در آن هر پیکسل به یک کلاس یا شی خاصی اختصاص دارد و یکی از مهم‌ترین ویژگی‌های پردازش تصویر و بینایی ماشین است. در این نوع تقسیم‌بندی، هدف از تحلیل یک تصویر، تخصیص یک برچسب (عموما نام یک شی یا شی‌های مشابه) به هر پیکسل موجود در تصویر است به عبارت دیگر، تصویر به بخش‌های کوچکتر تقسیم می‌شود و برای هر بخش یک برچسب مناسب انتخاب می‌شود.

ویژگی‌ها و کاربردهای Semantic Segmentation عبارتند از:

1. تفکیک اجزای تصویر: این تکنیک به ما امکان می‌دهد تا اجزای مختلف تصویر را از یکدیگر متمایز کنیم. به عنوان مثال، در یک تصویر خیابان، می‌توانیم هر خودرو، پیاده‌رو، درخت و ساختمان را با برچسب‌های مختلف تشخیص دهیم.
2. بهبود بینایی ماشین: با تفکیک دقیق اجزای تصویر، برنامه‌های بینایی ماشین می‌توانند اطلاعات دقیق‌تری در مورد محتوای تصویر به دست آورند. این اطلاعات می‌توانند در برنامه‌هایی مانند خودروهای خودران، پردازش تصویر پزشکی، رباتیک، و دیگر حوزه‌ها مفید باشند.
3. تشخیص و تعیین موقعیت اشیا: Semantic Segmentation به ما اجازه می‌دهد تا نه تنها اشیا را تشخیص دهیم بلکه موقعیت آنها را نیز تعیین کنیم. این معمولا در سیستم‌های هوش مصنوعی برای هدایت ربات‌ها، تعقیب اشیا، یادگیری تقویتی و کاربردهایی مشابه مورد استفاده قرار می‌گیرد.

4. پردازش تصویر پزشکی: در پزشکی Semantic Segmentation می‌تواند به تشخیص و مرزبندی ساختارهای مهم در تصاویر پزشکی کمک کند. به عنوان مثال، تشخیص و مرزبندی تومورها در تصاویر MRI مغز از این تکنیک بهره می‌برد.
5. بینایی ماشین در خودروهای خودران: در خودروهای خودران Semantic Segmentation برای تشخیص و متمایز کردن اجزای مختلف محیط از جمله جاده، خودروهای دیگر، پیاده‌ها و چراغ‌های راهنما مورد استفاده قرار می‌گیرد.

برای انجام Semantic Segmentation، معمولا از شبکه‌های عصبی عمیق (Deep Neural Networks)، به ویژه شبکه‌های کانولوشنی (Convolutional Neural Networks یا CNNs)، استفاده می‌شود. این شبکه‌ها با یادگیری از داده‌های بزرگ و تنوع‌پذیر، می‌توانند تصاویر را به طور دقیق‌تر تقسیم‌بندی کنند.

۲. Instance segmentation: شامل شناسایی و جداسازی اشیا منفرد در یک تصویر از جمله تشخیص مرزهای هر شی و اختصاص یک برچسب منحصر به فرد هر شی است و همینطور Instance segmentation یکی از پیشرفته‌ترین و مهم‌ترین وظایف در حوزه بینایی ماشین و پردازش تصویر است. این وظیفه ترکیب دو مسئله دیگر یعنی Semantic Segmentation و Object Detection است. در واقع در Instance Segmentation هدف اصلی تشخیص و تمیز کردن هر شی موجود در تصویر به صورت جداگانه و با تخصیص یک شناسه یکتا به هر شی است.

ویژگی‌ها و کاربردهای Instance Segmentation عبارتند از:

1. تفکیک شی از پس‌زمینه: Instance Segmentation به ما این امکان را می‌دهد که هر شی در تصویر را از پس‌زمینه تشخیص داده و تمیز کنیم. این به معنای این است که ما می‌توانیم اشیا مختلف را از یکدیگر جدا کرده و به هر کدام یک شناسه یکتا اختصاص دهیم.
2. تمیزکاری دقیق شی: Instance Segmentation دقیق‌تر از Object Detection عمل می‌کند. زیرا در Object Detection، معمولا فقط اطلاعات در مورد مکان و مرزهای شی ارائه می‌شود، در حالی که در Instance Segmentation، ما دقیقا می‌دانیم کدام پیکسل‌ها به هر شی تعلق دارند.
3. تعقیب اشیا: Instance Segmentation برای تعقیب و پیگیری اشیا در تصاویر متحرک (مثلا در ویدیوها) بسیار مفید است. با تخصیص یک شناسه یکتا به هر شی می‌توانیم مسیر و حرکت آنها را در تصاویر متوالی پیگیری کنیم.

4. بینایی ماشین در خودروهای خودران: Instance Segmentation در خودروهای خودران به عنوان یکی از مهم‌ترین وظایف مورد استفاده قرار می‌گیرد. با تشخیص و تمیز کردن شی‌های مختلف مانند خودروهای دیگر، پیاده‌روها و سایر اشیا، خودروها می‌توانند به ایمنی و هوش مصنوعی بهبود بخشند.

برای انجام Instance Segmentation، معمولا از شبکه‌های عصبی عمیق (Deep Neural Networks)، به ویژه شبکه‌های کانولوشنی (Convolutional Neural Networks یا CNNs) و شبکه‌های مبتنی بر ماسک (Mask R-CNN) استفاده می‌شود. این شبکه‌ها به صورت همزمان شناسایی شی، تعیین مرزهای آن و تمیزکاری دقیق آن انجام می‌دهند و نتایج دقیقی را در مسائل تشخیص و تعقیب اشیا ارائه می‌دهند.

۲-٦- معماری U-Net

معماری U-Net یک معماری شبکه عصبی عمیق برای تقسیم‌بندی تصاویر (Image Segmentation) است که ابتدا توسط Ronneberger و همکارانش در سال ۲۰۱۵ معرفی شد. این معماری به خصوص برای مسائل Semantic Segmentation تصاویر پزشکی مانند تشخیص تومورها در تصاویر MRI طراحی شده است و از آن زمان تبدیل به یکی از معماری‌های محبوب در این زمینه شده است.

U-Net به عنوان یک معماری پایه برای تقسیم‌بندی تصاویر مورد استفاده قرار می‌گیرد و می‌تواند با تغییرات و توسعه‌های مختلف به منظور حل مسائل خاص مورد تنظیم قرار گیرد. این معماری بسیار موثر برای تشخیص و تقسیم‌بندی اشیا در تصاویر با ابعاد مختلف و در حوزه‌های مختلف از پزشکی تا بینایی ماشین استفاده می‌شود.

معماری U-Net دارای ساختاری خاص و متمرکز بر اساس Encoder و Decoder است که به شکل حرف "U" شکل‌گیری کرده است. دو بخش اصلی Encoder و Decoder به صورت زیر هستند:

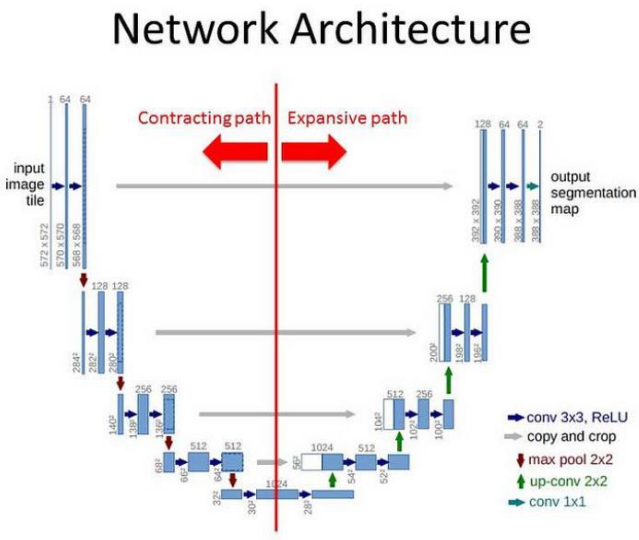
1. Encoder: شبکه encoder از چندین لایه کانولوشنی عمیق تشکیل شده است که تصاویر ورودی را به فضای ویژگی‌های انتزاعی تبدیل می‌کند. این لایه‌های کانولوشنی به تدریج اندازه تصویر را کاهش می‌دهند (از طریق استفاده از max_pooling یا اشتراک گذاری ویژگی‌ها) و ویژگی‌های مهم را استخراج می‌کنند پس توی این بخش از لایه های کانولوشنی و max_pooling و تابع فعال سازی relu استفاده می‌شود و هر بار که اندازه تصویر کوچکتر می‌شود اطلاعات بصری مهمتری استخراج می‌شود.

2. Decoder: این بخش به همراه اطلاعات ویژگی‌ها که توسط encoder استخراج شده‌اند به بازسازی تصویر اصلی پرداخته و از طریق لایه‌های transposed convolution و عملیات unpooling اندازه تصویر را افزایش می‌دهند.

یکی از ویژگی‌های بارز U-Net، استفاده از اتصالهای جانبی (Skip Connections) است. این اتصالهای جانبی از لایه‌های encoder به لایه‌های معادل در decoder می‌رسند و کمک به انتقال اطلاعات دقیق از لایه‌های پایین به لایه‌های بالاتر می‌کنند.

معماری U_Net از Autoencoder ها الهام گرفته است اما دقیقا نمی‌توان گفت که U_Net یک Autoencoder است به طور کلی U_Net برای مسائل تشخیص و تجزیه و تحلیل تصاویر به ویژه در حوزه تشخیص شی و ترسیم آن‌ها در تصاویر پزشکی مورد استفاده قرار می‌گیرد.

شکل زیر یک نمایی از معماری U_Net را نشان می‌دهد:



عملیات copy and crop در معماری U_Net:

هنگامی که از لایه‌های convolution و max_pooling برای کاهش ابعاد استفاده می‌شود ممکن است اطلاعات مکانی از دست برود و برای اینکه اطلاعات مکانی حفظ شود از copy and crop استفاده می‌کنند.

Copy: در عملیات Copy، اطلاعات و ویژگی‌های موجود در لایه‌های انتخاب شده از encoder به decoder کپی می‌شوند. این اطلاعات از لایه‌های encoder با ابعاد کوچکتر به لایه‌های معادل در decoder انتقال می‌یابند. این

اتصالات جانبی به decoder اطلاعات معنی‌داری از تصویر ورودی (به ویژه در مورد ویژگی‌هایی که در فازهای ابتدایی encoder استخراج می‌شوند) ارائه می‌دهند تا بتواند ویژگی‌های جزئی و سطوح بالاتر را بازیابی کند.

Crop: عملیات Crop به معنای تنظیم اندازه و ابعاد اطلاعات کپی شده در مرحله قبل است. از آنجا که اطلاعات encoder و decoder ممکن است ابعاد مختلفی داشته باشند (به عنوان مثال، به دلیل استفاده از لایه‌های کانولوشنی با مقیاس‌های مختلف)، باید این ابعاد تنظیم شوند تا اتصالات جانبی به درستی انجام شود. crop به این ترتیب انجام می‌شود که اطلاعات از لایه‌های encoder به اندازه و ابعاد مشابه لایه‌های معادل در decoder برش داده می‌شوند و همینطور از آنجایی که عملیات کپی ممکن است به افزایش حجم حافظه و محاسبات منجر شود به جای کپی کامل از برش استفاده می‌شود.

از اتصالات جانبی Copy and Crop در U-Net به عنوان اتصال جانبی کوتاه (Short Skip Connections) نیز یاد می‌شود زیرا این اتصالات به تقویت و ارتقا اطلاعات در سطوح پایین تر تصویر کمک می‌کنند و برای دقیق‌تر کردن تقسیم‌بندی تصویر و بازیابی ویژگی‌ها از دست رفته مفید هستند. این اتصالات جانبی از ارائه دقت و کیفیت بهتر در تقسیم‌بندی تصاویر شناخته شده‌اند.