Q1)

ابتدا خلاصه ای از عملکرد شبکه CNN را مینویسم:

شبکه‌های عصبی کانولوشنی یا CNN (Convolutional Neural Networks) از نوع شبکه‌های عصبی عمیق هستند که به ویژه برای پردازش تصویر مورد استفاده قرار می‌گیرند. در زیر خلاصه‌ای از عملکرد این نوع شبکه‌ها آورده شده است:

۱. **لایه کانولوشنی :(Convolutional Layer)** در این لایه، فیلترهای کانولوشنی بر روی تصویر ورودی اعمال می‌شوند. این فیلترها ویژگی‌های مختلف تصویر را استخراج می‌کنند، مانند لبه‌ها، گوشه‌ها، یا الگوهای پیچیده‌تر.

۲. **لایه همگرایی :(Pooling Layer)** در این لایه، ابعاد فضایی تصویر کاهش می‌یابد. این باعث می‌شود که شبکه مقاوم‌تر به تغییرات کوچک در موقعیت ویژگی‌ها شود و همچنین تعداد پارامترهای شبکه کاهش یابد.

۳. **لایه کاملاً متصل :(Fully Connected Layer)** پس از مراحل کانولوشن و همگرایی، ویژگی‌های استخراج شده به یک لایه کاملاً متصل وارد می‌شوند. این لایه به تصمیم‌گیری نهایی برای تفسیر تصویر کمک می‌کند.

۴. **لایه فعال‌سازی :(Activation Layer)** این لایه از توابع فعال‌سازی مانند ReLU (Rectified Linear Unit) برای افزایش قدرت نمایشی شبکه استفاده می‌کند.

۵. **تابع هزینه :(Loss Function)** برای اندازه‌گیری تفاوت بین خروجی مدل و اطلاعات مورد انتظار، تابع هزینه مورد استفاده قرار می‌گیرد.

شبکه‌های عصبی کانولوشنی به خوبی برای مسائل دسته‌بندی تصویر، تشخیص اشیاء، ترجمه ماشینی تصویر و سایر وظایف مرتبط با تصاویر عملکرد خوبی دارند.

و همچنین خلاصه از عملکرد شبکه های مبتنی بر توجه:

شبکه‌های مبتنی بر توجه (Attention-based Networks) از نوع شبکه‌های عصبی هستند که توجه به بخش‌های خاصی از ورودی را در هر مرحله محاسباتی دارند. این نوع شبکه‌ها به ویژه در حوزه پردازش زبان طبیعی مورد استفاده قرار می‌گیرند. در زیر خلاصه‌ای از عملکرد این شبکه‌ها آورده شده است:

۱. **توجه :(Attention)** شبکه‌های مبتنی بر توجه از مفهوم توجه به ورودی استفاده می‌کنند. به ازای هر خروجی تولید شده توسط مدل، یک وزن توجه به هر بخش از ورودی اختصاص داده می‌شود. این وزن‌ها نشان‌دهنده اهمیت نقاط مختلف ورودی در تولید خروجی هستند.

۲. **لایه توجه :(Attention Layer)** در این لایه، وزن‌های توجه محاسبه می‌شوند. این محاسبات معمولاً بر اساس اشتباه مدل در تولید خروجی نسبت به ورودی صورت می‌گیرد.

۳. **توجه چندسر :(Multi-Head Attention)** برخی از مدل‌های توجه از لایه توجه چندسر استفاده می‌کنند که به مدل این امکان را می‌دهد که از چندین جهت توجه به اطلاعات مختلف ورودی داشته باشد.

۴. **کدگذار-کدگشا :(Encoder-Decoder)** در برخی از مسائل مانند ترجمه ماشینی، این شبکه‌ها به صورت کدگذار (Encoder) برای تولید نمایش متن و کدگشا (Decoder) برای تولید متن هدف استفاده می‌شوند.

۵. **استفاده در پردازش زبان طبیعی :(NLP)** شبکه‌های مبتنی بر توجه به خوبی برای مسائل پردازش زبان طبیعی مناسب هستند، از جمله ترجمه ماشینی، خلاصه‌سازی متون، تولید متن خودکار و غیره.

استفاده از توجه در شبکه‌ها به مدل این امکان را می‌دهد که به نقاط مهم تر ورودی توجه کرده و از آنها بهترین اطلاعات را جذب کند، که این امر به بهبود عملکرد و دقت مدل‌ها منجر می‌شود.

آ)

برای تصویر گربه، یک تصویر گربه اصلی است و تصویر دیگر، همان گربه ولی با پوست فیل آفریقایی است.

مدل CNN پیشبینی های خود را با توجه به بافت های تصویر انجام میدهد. بنابراین احتمالا تصویر گربه عادی را درست پیشبینی میکند اما تصویر گربه با پوست فیل آفریقایی را به اشتباه فیل پیشبینی میکند.

مدل مبتنی بر توجه اما به جای توجه به بافت، نتیجه را بر اساس اجزا و عناصر موجود در تصویر تولید میکند. برای همین تصویر گربه اصلی را درست پیشبینی میکند، تصویر گربه با پوست فیل آفریقایی را هم با توجه به این که اجزا تغییر نکرده، همان گربه پیشبینی میکند.

ب)

برای تصویر چهره انسان، یک تصویر نقاشی چهره انسان است و تصویر دیگر، تصویر چهره ای است که اجزای صورت آن به هم ریخته است.

مدل CNN به خاطر استفاده از لایه های max pooling، تنها به وجود عناصر شناسایی شده حساس میشود و نه به مکان کلی آنها به همین علت این مدل هم تصویر صورت اصلی، و هم تصویر چهره به هم ریخته را چهره انسان پیشبینی میکند.

مدل مبتنی بر توجه، با توجه به مکانیسم های توجهی که دارد میداند به کدام از قسمت از تصویر باید توجه کند. یعنی میداند که چشم و بینی و دهان در کدام قسمت از صورت باید قائل شوند برای همین تصویر اصلی را درست پیشبینی میکند اما صورت به هم ریخته را انسان پیشبینی نمیکند چون عناصر در مکان های مناسب قرار نگرفته اند.

Q2)

الف)

در زبان پزشکی و علوم پزشکی، اصطلاحات(FN (False Negative ، FP (False Positive)، TP (True Positive) وTN (True Negative)به عنوان اجزای ماتریس ارزیابی کارآیی مدل‌ها و تست‌های تشخیصی (مثل تست‌های پزشکی) استفاده می‌شوند. این اصطلاحات در ارتباط با نتایج مثبت یا منفی واقعی و نتایج مدل یا تست موردنظر قرار دارند. در ادامه توضیحات این اصطلاحات آمده است:

۱. **FN (False Negative)**

- معنی: تعداد مواردی که واقعاً مثبت هستند (حقیقی) اما مدل یا تست آنها را منفی تشخیص داده است.
- مثال: اگر یک بیمار واقعاً بیمار باشد اما تست آن را منفی تشخیص دهد، این یک FN است.

۲. **FP (False Positive)**

- معنی: تعداد مواردی که واقعاً منفی هستند (حقیقی) اما مدل یا تست آنها را مثبت تشخیص داده است.
- مثال: اگر یک تست به اشتباه به شخص سالم مثبت تشخیص دهد، این یک FP است.

۳. **TP (True Positive)**

- معنی: تعداد مواردی که واقعاً مثبت هستند (حقیقی) و مدل یا تست نیز آنها را به درستی مثبت تشخیص داده است.
- مثال: اگر یک تست به درستی یک بیمار را مثبت تشخیص دهد، این یک TP است.

۴. **TN (True Negative)**

- معنی: تعداد مواردی که واقعاً منفی هستند (حقیقی) و مدل یا تست نیز آنها را به درستی منفی تشخیص داده است.
- مثال: اگر تست به درستی یک شخص سالم را منفی تشخیص دهد، این یک TN است.

این اصطلاحات به عنوان اجزای ماتریس ارزیابی (Confusion Matrix) مورد استفاده قرار می‌گیرند تا دقت و عملکرد مدل یا تست را به طور جامعتر ارزیابی کرد.

ب)

می توانیم از معیارهای precision و recall استفاده کنیم اما هر کدام مزایا و معایب خود را دارند که با توجه به اهمیت بیشتر شناسایی نشدن افراد بی گناه به عنوان مجرم یا اهمیت بیشتر شناسایی همه مجرمین یکی را انتخاب کنیم

Presicion: این معیار به این معناست که از بین تمام مجرمینی که تشخیص میدهیم چند تا از آن ها واقعا مجرم بوده اند. در صورتی که از این معیار استفاده کنیم و بر اساس آن مدل را انتخاب کنیم، افراد بی گناه کم تری به عنوان مجرم شناسایی میشوند. اما ممکن است که همه مجرمین شناسایی نشوند.

Recall: این معیار یعنی از همه مجرمین، چند تا از آنها را شناسایی کردیم. در صورتی که از این معیار استفاده کنیم و بر اساس آن مدل را انتخاب کنیم، مجرمان بیشتری را شناسایی میکنیم اما ممکن است که تعدادی هم از افراد بی گناه را به عنوان مجرم تشخیص دهیم.

به طور کلی این دو معیار با هم یک trade off دارند. در صورتی که هیچ اولویتی بین شناسایی شدن همه مجرم ها و گناهکار شناخته نشدن افراد بیگناه نداریم، میتوانیم از معیار F1 score استفاده کنیم

F1 score: این معیار، یک برایند از دو معیار قبلی است و میتوانیم به صورت کلی از این معیار استفاده کنیم.

Q3)

الف)

تخمین چرخش یا ارزیابی تغییرات فضایی در تصاویر می‌تواند برای وظیفه طبقه‌بندی تصاویر مفید باشد. در زیر تعدادی از مواردی که نحوه تخمین چرخش می‌تواند به بهبود وظیفه طبقه‌بندی کمک کند، آورده شده است:

۱. **استفاده از ویژگی‌های چرخش:**

- تغییرات در جهت چرخش تصویر می‌تواند ویژگی‌های مهمی را در داده‌ها نشان دهد. اگر این تغییرات با ویژگی‌های مهم کلاس‌ها همخوانی داشته باشند، مدل می‌تواند این ویژگی‌ها را برای تشخیص دقیق‌تر از داده‌ها بهره‌مند سازد.

۲. **تقویت نظارت ویژگی‌ها:**

- با افزودن اطلاعات چرخش به داده‌ها، مدل می‌تواند نظارت بیشتری بر روی ویژگی‌های مرتبط با چرخش داشته باشد. این ممکن است به افزایش دقت در تفکیک بین دسته‌ها کمک کند.

۳. **افزایش تنوع داده‌ها:**

- اگر داده‌ها از زوایا و جهات مختلف گرفته شوند (به عنوان مثال، با چرخش‌های مختلف)، مدل می‌تواند بهتر از اطلاعات متنوع استفاده کند. این تنوع ممکن است به جلوگیری از برازش بیش از حد (overfitting) کمک کند.

۴. **تشخیص الگوهای چرخش مخصوص به کلاس‌ها:**

- برخی از کلاس‌ها ممکن است وابستگی به جهت و چرخش داشته باشند. با توجه به این ویژگی، مدل می‌تواند الگوهای خاص چرخش برای هر کلاس را یاد بگیرد و این اطلاعات را در تصمیم‌گیری طبقه‌بندی استفاده کند.

۵. **کاهش تأثیر چرخش بر کارایی مدل:**

- در برخی از موارد، تصویرهای چرخیده ممکن است به تصمیم‌گیری مدل نقص ایجاد کنند. با تخمین چرخش، ممکن است بتوان تأثیر این تغییرات را کاهش داد و کارایی مدل را بهبود بخشید.

علاوه بر همه این مزایا، از وزن های آموخته شده در این شبکه میتوان برای ساخت مدل های دیگر استفاده کرد. تمام این مزایای گفته شده همزمان با انتقال وزن ها به شبکه جدید و fine tune کردن آن، به آن منتقل میشوند.

از این روش که یک روش self supervised است زمانی استفاده میشود که تصویرو لیبل به اندازه کافی موجود نیست. بنابراین تصاویر و شبه لیبل ها به صورت خودکار توسط این شبکه تولید میشوند و وزن ها آموزش داده شده و به مدل اصلی منتقل میشوند. سپس مدل اصلی با همان تعداد نمونه های کم که در اختیار داریم، fine tune میشود.

ب)

بردارهای "One-Hot" (یا بردارهای یک‌گانه) یک روش نمایش داده‌ها در ماشین لرنینگ هستند که به طور خاص برای نمایش متغیرهای دسته‌ای (متغیرهایی که مقادیر گسسته و متمایز دارند، مانند دسته‌ها یا برچسب‌ها) استفاده می‌شوند. در این نوع نمایش، هر مقدار از متغیر به یک بردار با اندازه تعداد دسته‌ها تبدیل می‌شود و تنها یکی از عناصر آن بردار مقدار ۱ (وضعیت فعال) دارد و بقیه صفر هستند.

برای مثال، اگر یک متغیر دسته‌ای سه دسته داشته باشد، نمایش One-Hot برای این متغیر به صورت زیر خواهد بود:

- دسته ۱: [1, 0, 0]
- دسته ۲: [0, 1, 0]

- دسته ۳[0, 0, 1] :

حالت انتخابی مقدار ۱ به این معناست که داده مربوطه در آن دسته قرار دارد.

مشکلات استفاده از بردارهای:One-Hot

۱. **بزرگی ابعاد بردار:**

- اگر تعداد دسته‌ها زیاد باشد، ابعاد بردارهای One-Hot نیز زیاد خواهد بود. این می‌تواند منجر به افزایش حجم داده و افزایش پیچیدگی مدل شود.

۲. **عدم در نظر گرفتن ارتباطات بین دسته‌ها:**

- بردارهای One-Hot تمام دسته‌ها را به صورت مستقل در نظر می‌گیرند و هیچ اطلاعاتی درباره ارتباطات بین دسته‌ها را نگه نمی‌دارند. این می‌تواند مشکلاتی را در مواردی که ارتباطات میان دسته‌ها مهم هستند، ایجاد کند.

۳. **مصرف حافظه:**

- زمانی که داده‌ها به شکل بردارهای One-Hot نمایش داده می‌شوند، این نمایش ممکن است حجم زیادی حافظه را مصرف کند، به ویژه زمانی که تعداد دسته‌ها بسیار زیاد باشد.

۴. **عدم مقیاس‌پذیری:**

- در مواقعی که تعداد دسته‌ها متغیر باشد، نیاز به تغییر اندازه بردارها و دوباره آموزش مدل ممکن است مشکلاتی را ایجاد کند.

در کل، بردارهای One-Hot برای مسائلی که متغیرهای دسته‌ای دارند، استفاده می‌شوند، اما در مواردی که مشکلات فوق برای ماهیت مسئله مهم هستند، روش‌های دیگری ممکن است بهتر باشند.

پ)

در یادگیری خودنظارتی، مدل برای یادگیری به تنهایی از داده‌های ورودی استفاده می‌کند. اینجا برچسب‌های خارجی در داده‌ها وجود ندارند. به جای آن، مدل با استفاده از ویژگی‌های موجود در داده‌ها خود برچسب‌ها را ایجاد یا پیش‌بینی می‌کند.

در روش word2vec هم، هیچ نمونه و لیبل از پیش آماده ای نداریم بلکه شبه لیبل ها توسط خود مدل از روی متن تولید میشوند.

برای مثال، یک متن طولانی چند صد هزار کلمه ای انتخاب میشود، از روی ای متن چندین نمونه انتخاب و لیبل مناسب آنها تولید میشود. مثلا از روی متن، کلمات مختلف به عنوان context انتخاب شده و برای هر کانتکست، یک کلمه تصادفی به صورت خودکار به عنوان target انتخاب مشود. سپس این جفت های context و label، به عنوان نمونه های ورودی به شبکه داده میشوند تا آموزش انجام گیرد.

میبینیم که در اینجا، متن اصلی چند صد هزار کلمه ای هیچ نمونه و لیبلی از پیش ندارد و همه sample ها توسط خود مدل ایجاد میشوند، به همین علت این روش، یک روش self supervised است.

Q4)

الف)

به طور خاص، فرایند شامل آموزش یک کنترل‌گر (Controller) به صورت یک شبکه عصبی بازگشتی (RNN) است که ساختارهای مدل فرزند را پیشنهاد می‌دهد تا ارزیابی شوند. هدف نهایی این است که طراحی معماری شبکه‌های عصبی برای یک وظیفه خاص بهینه‌سازی شود.

مفاهیم اصلی و فرآیند بهینه‌سازی با استفاده از یادگیری تقویتی به شرح زیر است:

۱. **طراحی کنترل‌گر:**

- کنترل‌گر به صورت یک RNN پیاده‌سازی می‌شود و یک دنباله متغیر طولانی از توکن‌ها را خروجی می‌دهد. این توکن‌ها برای پیکربندی ساختار یک شبکه عصبی کودک استفاده می‌شوند.

۲. **فضای عمل:**

- فضای عمل برای وظیفه یادگیری تقویتی به عنوان یک لیست از توکن‌ها برای تعریف ساختار یک شبکه فرزند تعریف می‌شود. کنترل‌گر یک دنباله از اقدامات را خروجی می‌دهد و طول این دنباله توسط T، تعداد کل توکن‌ها، نشان داده می‌شود.

۳. **پاداش:**

- پاداش برای آموزش کنترل‌گر بر اساس دقت حاصل از شبکه فرزند در زمان همگرایی است. به عبارت دیگر، هرچه عملکرد بهتر شبکه فرزند باشد، پاداش برای کنترل‌گر بیشتر خواهد بود. دقت به عنوان معیار موفقیت در نظر گرفته می‌شود.

۴. **تابع هزینه:**

- کنترل‌گر با استفاده از الگوریتم REINFORCE آموزش می‌بیند. هدف بهینه‌سازی پارامترهای کنترل‌گربا استفاده از تابع هزینه REINFORCE است. ما می‌خواهیم پاداش مورد انتظار (دقت بالا) را بیشینه کنیم و از گرادیان به عنوان زیر استفاده می‌شود. نکته جالب در اینجا این است که با استفاده از گرادیان سیاست، می‌تواند حتی زمانی که پاداش قابل تفاوت نیست کار کند.

همچنین روش MetaQNN از یادگیری Q برای انتخاب توالی لایه‌های شبکه عصبی پیچیده بااستفاده از استراتژی ϵ-greedy و بازیابی تجربی عمل می‌کند. پاداش نیز دقت اعتبارسنجی مدل است.

ب)

بله، رویکردی که در متن توضیح داده شده و شامل استفاده از یادگیری تقویتی (Reinforcement Learning) برای جستجوی ساختارهای مدل شبکه عصبی (Neural Architecture Search - NAS) است، می‌تواند به منظور تنظیم اندازه ورودی و تعداد لایه‌ها در مدل زیرمجموعه (child model) مورد استفاده قرار گیرد. در این سیاق، کنترل‌گر مبتنی بر یادگیری تقویتی به عنوان هدف دارد تا دنباله‌های

اقدامات (توکن‌ها) ارائه دهد که ساختار شبکه عصبی فرزند را تعریف می‌کند. اقدامات می‌توانند شامل مشخص کردن اندازه ورودی و تعداد لایه‌ها باشند.

راهنمایی برای اعمال این رویکرد به منظور تنظیم اندازه ورودی و تعداد لایه‌ها به شرح زیر است:

۱. **توسعه فضای عمل:**

- تغییر فضای عمل در وظیفه یادگیری تقویتی به منظور شامل توکن‌ها یا اقداماتی که اندازه ورودی و تعداد لایه‌ها را نمایند. این توسعه به کنترل‌گر این امکان را می‌دهد که ساختارهای گوناگونی از جمله تغییرات در اندازه ورودی و تعداد لایه‌ها ارائه دهد.

۲. **تعریف پاداش:**

- تعریف پاداش بر اساس عملکرد شبکه فرزند نسبت به اندازه ورودی و تعداد لایه‌ها مشخص شده است. به عبارت دیگر، هر چه عملکرد بهتر شبکه فرزند باشد، پاداش برای کنترل‌گر بیشتر خواهد بود. دقت به عنوان معیار موفقیت در نظر گرفته می‌شود.

۳. **اعمال اقدامات:**

- پیاده‌سازی اقدامات مرتبط با تغییر اندازه ورودی و تعداد لایه‌ها در مدل فرزند. کنترل‌گر باید دنباله‌های اقدامات را خروجی دهد که ساختار کلی را، از جمله این جنبه‌ها، مشخص کند.

۴. **تابع هزینه و بهینه‌سازی:**

- بهینه‌سازی پارامترهای کنترل‌گر با استفاده از الگوریتم REINFORCE با هدف بیشینه کردن پاداش مورد انتظار. کنترل‌گر باید یاد بگیرد تا ساختارهایی را پیشنهاد دهد که در ابعاد مختلف اندازه ورودی و تعداد لایه‌ها مؤثر باشند.

۵. **نمایش وضعیت:**

- بسته به اجرای خاص، در نظر بگیرید که نمایش وضعیت در وظیفه یادگیری تقویتی به عنوان یک تاپل شامل اطلاعاتی از جمله اندازه ورودی، تعداد لایه‌ها و پارامترهای مرتبط دیگر باشد. این اطلاعات، عامل یادگیری تقویتی را در انتخاب‌های آگاهانه هدایت می‌کند.

با گسترش فضای عمل و تعریف مناسب پاداش‌ها، می‌توانید از رویکرد یادگیری تقویتی مبتنی بر NAS برای تنظیم اندازه ورودی و تعداد لایه‌ها در مدل فرزند استفاده کنید. کنترل‌گر یاد می‌گیرد که ساختارهایی را ارائه دهد که بر اندازه ورودی و پیکربندی لایه‌ها موثر باشند.

Q5)

این سوال از نمونه سوال های امتحان Stanford است که پاسخ آن هم طبق داک خود استنفورد به این صورت است:

(1 point) You are training a standard GAN, and at the end of the first epoch you take note of the values of the generator and discriminator losses. At the end of epoch 100, the values of the loss functions are approximately the same as they were at the end of the first epoch. Why are the quality of generated images at epoch 1 and epoch 100 not necessarily similar? (1-2 sentences)

Solution: You should not necessarily expect them to be the same since the losses are with respect to different quality models over time. That is, the loss of the generator at epochs 1 and 100 are with respect to a discriminator which might have significantly improved, and the same follows for the loss of the discriminator.

به بیان دیگر دو مدل مولد و ممیز با هم دیگر در رقابت هستند، در ابتدا هر دو مدل ضعیف هستند و یک لاس مشخص تولید میکنند، هر چه جلو تر میرویم، هر دو مدل با هم پیشرفت میکنند. بنابراین رقابت میان این دو مدل حفظ میشود و هیچ کدام بر دیگری پیروز نمیشوند که لاس آن کم بشود. هر دو مدل قوی تر میشوند و لاس همان لاس قبلی میماند.