





## دانشکده مهندسی برق و کامپیوتر

داده افزایی برای هیجانات (احساسات) در گفتار

استاد راهنما: سید جلال ذهبی

متین فاضل

بهمن ۱۴۰۲

# فهرست مطالب

۱	مقدمه	۱
۱	۱.۱ داده افزایی برای هیجانات (احساسات) در گفتار	۱.۱
۱	۲.۱ شرح مسئله	۲.۱
۱	۳.۱ مشارکت ها	۳.۱
۲	۲ مفاهیم و اصطلاحات علمی	۲
۲	۱.۲ شبکه های مولد تخصصی	۱.۲
۲	۱.۱.۲ معماری	۱.۱.۲
۳	۲.۱.۲ تابع هزینه	۲.۱.۲
۳	۳.۱.۲ انواع GAN برای تشخیص هیجانات در گفتار	۳.۱.۲
۳	۴.۱.۲ شبکه های تخصصی حلقوی پایدار	۴.۱.۲
۵	۵.۱.۲ تقسیم بندی با رویکرد یادگیری ماشین یا یادگیری عمیق	۵.۱.۲
۵	۲.۲ طیف نگاری	۲.۲
۵	۳.۲ شبکه عصبی	۳.۲
۵	۴.۲ شبکه عصبی پیچشی	۴.۲

# فصل ۱

## مقدمه

### ۱.۱ داده افزایی برای هیجانات (احساسات) در گفتار

تشخیص عواطف داده های صوتی SER به تشخیص خودکار احساسات و عواطف انسانی اشاره دارد. به عنوان زمینه تحقیقاتی مهم، تشخیص عواطف دادهای صوتی به سرعت در حال رشد است، همچنین دارای پتانسیل بهبود تعامل انسان و رایانه مبتنی بر صدا، مانند سیستم داخل خودرو برای درک وضعیت عاطفی رانندگان به هنگام ایجاد تغییرات ناگهانی است.

### ۲.۱ شرح مسئله

کمبود داده از عمده چالش های پیچیده در تشخیص هیجانات در گفتار است که در این سه مورد شرح داده شده است:

- مشکل اول فقدان دیتاست اصوات صوتی طبیعی است. دیتاست های کمی برای تحقیقات در این زمینه به اشتراک گذاشته شده است. مخصوصاً، بسیاری از دیتاست های صوتی که در شرایط واقعی تولید شده اند به علت یکسری محدودیت های قانونی در دسترس عمومی قرار نگرفته اند.
- مشکل دیگر برچسب گذاری داده های صوتی است. از آنجایی که احساسات ابراز شده متفاوت هستند، دسته بندی آنها بسیار مهم است. با این حال به دلیل عدم قطعیت بالا دسته بندی و تحلیل آنها کاری به شدت زمان بر است.
- در نهایت، داده های صوتی در اکثر پایگاه های داده به صورت نامتعادلی بر روی احساسات توزیع میشوند. به طور کلی، تعداد جملات با احساسات خنثی بیشترین تعداد را در بدنه گفتاری دارد. با این حال، برای ارزیابی دقت طبقه بندی، یک پایگاه داده متعادل برای تجزیه و تحلیل نیاز است. علاوه بر این، اگر یک جمله با احساسات مختلف ضبط شود، قضاوت انسان در مورد احساس درک شده میتواند صرفاً بر اساس محتوای عاطفی جمله بدون تاثیر محتوای واژگانی آن باشد.

### ۳.۱ مشارکت ها

شبکه های مولد متخاصم (GANs)، یک روش کارآمد برای تولید دیتا هستند. با استفاده از یک بازی متخاصم بین یک تشخیص دهنده و یک مولد، شبکه های مولد متخاصم برای تولید نمونه هایی که از داده های واقعی قابل تشخیص نیستند، آموزش میبینند. علاوه بر این دارای مشخصه های زیر هستند:

- شبکه های مولد متخاصم میتوانند توزیع های احتمالی با ابعاد بالا را در مسائل پیچیده دنیای واقعی بیاموزند.
- این شبکه ها را میتوان با داده ها از دست رفته آموزش داد، که برای یادگیری نیمه نظارتی مناسب است.
- شبکه های مولد متخاصم دارای خروجی های چندوجهی هستند، به این معنی که میتوانند چندین پاسخ صحیح مختلف تولید کنند و تنوع نمونه های تولید شده را افزایش دهند.

هدف این پروژه ارزیابی عملکرد SER زمانی است که داده های آموزشی واقعی در فضای ویژگی با داده های مصنوعی تولید شده توسط شبکه های مولد متخاصم افزایش میابد. به طور خاص، نیاز به طراحی یک مدل مبتنی بر GAN است که بردارهای ویژگی مصنوعی از گفته های احساسی مختلف را تولید کند، به طوری که عملکرد یک شبکه عصبی دسته بند با دریافت بردارهای ویژگی واقعی و مصنوعی به عنوان داده های آموزشی بهبود داده شود.

## فصل ۲

# مفاهیم و اصطلاحات علمی

## ۱.۲ شبکه های مولد تخصصی

مدل های مولد به هر مدلی اطلاق می شود که مجموعه ای از نمونه های آموزشی گرفته شده از یک توزیع را یاد می گیرد که تخمینی از آن توزیع را نشان دهد. این شبکه ها بر اساس تولید داده به دو دسته صریح<sup>۱</sup> و ضمنی<sup>۲</sup> تقسیم می شوند. مدل های صریح تابع چگالی توزیع را مستقیماً محاسبه میکنند در حالی که مدل های ضمنی بر تولید نمونه هایی از توزیع ارائه شده توسط مدل تمرکز میکنند.

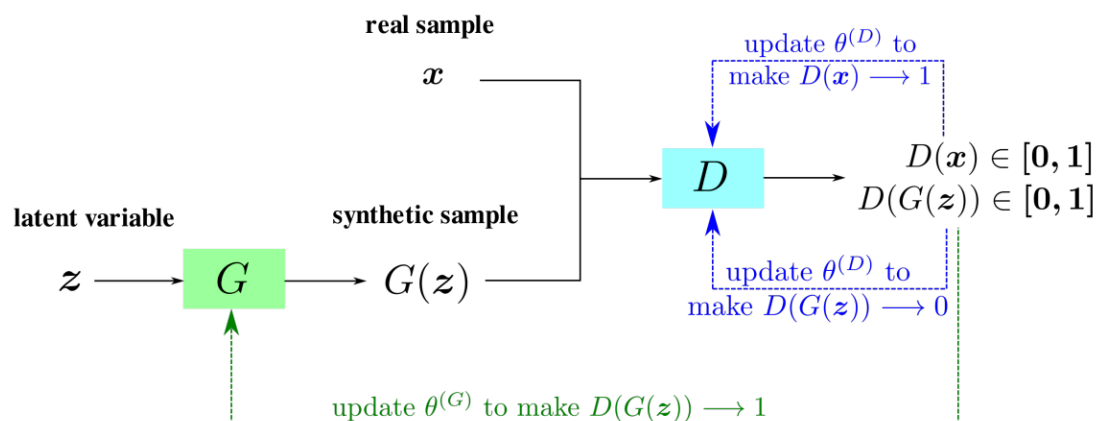
### ۱.۱.۲ معماری

یک شبکه مولد متخصصی، یک مدل مولد بر بر اساس نظریه بازی است که دو عامل را روبه روی هم قرار میدهد:

- یک تشخیص دهنده  $D$ <sup>۳</sup>

- یک مولد  $G$ <sup>۴</sup>

در شکل ۱.۱ طرح کلی یک شبکه مولد متخصصی رسم شده است، که با نام Vanilla GAN شناخته می شود. مولد به عنوان ورودی متغیر پنهان  $z$  را دریافت میکند که از یک مجموعه توزیع نویز  $p_z(z)$  تهیه میشود و داده ساخته شده  $G(z; \theta^G)$  را به عنوان خروجی تولید می کند. هدف مولد این است که یکسری داده تولید کند که از داده های واقعی غیر قابل تشخیص باشند. تشخیص دهنده از یک طرف داده های واقعی و از طرفی دیگر نمونه های مصنوعی  $G(z)$  را به عنوان ورودی دریافت میکند. خروجی این مدل  $D(x)$  یا  $D(G(z))$  که احتمال واقعی بودن نمونه دریافتی را نشان می دهد. مولد تلاش می کند تا مقدار  $D(G(z))$  را به عدد یک نزدیک کند تا تشخیص کننده را متقاعد کند که نمونه تولید شده شبیه به نمونه واقعی است، در حالی که تشخیص دهنده در تلاش است که مقدار  $D(G(z))$  را صفر نزدیک کند و مقدار  $D(x)$  را به یک نزدیک کند.



شکل ۱.۲: ساختار یک شبکه مولد متخصصی

<sup>1</sup>Implicit

<sup>2</sup>Explicit

<sup>3</sup>Discriminator

<sup>4</sup>Generator

## ۲.۱.۲ تابع هزینه

در نظر داریم که به عنوان ورودی شبکه داده‌های  $(s_i, y_i)_{i=1}^N$  را دریافت میکنیم که نیمی از آن داده‌های واقعی  $x$  و نیمی دیگر از آن داده‌های تولید شده  $G(z)$  است. هر نمونه آموزشی  $s_i$  متناظر با یک برجسب  $y_i$  است. همه داده‌های واقعی دارای برجسب یک و تمامی داد‌های واقعی حاوی برجسب صفر می‌باشند. با توجه به اینکه هدف تشخیص دهنده یک دسته‌بند دودویی است، تابع هزینه آن به صورت یک binary cross-entropy تعیین می‌شود.

$$J^{(D)}(D, G) = H((s_i, y_i)_{i=1}^N, D) = -\frac{1}{N} \sum_{i=1}^N (y_i \log(D(s_i)) + (1 - y_i) \log(1 - D(s_i))) \quad (۱.۲)$$

اگر  $y_i$  را برابر با یک برای  $s_i = x$  و برابر با صفر برای وقتی که  $s_i = G(z)$  در نظر بگیریم، همچنین با جایگذاری میانگین‌ها به رابطه تابع هزینه تشخیص دهنده به عبارت زیر میرسیم:

$$J^{(D)}(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [1 - \log D(G(z))] \quad (۲.۲)$$

که  $p_{data}$  یک توزیع داده بر روی نمونه داده‌های واقعی  $x$  است. در یک بازی minimax که به نام بازی zero-sum نیز شناخته می‌شود، مجموع هزینه‌های تمامی بازیکنان همواره صفر است. که نشان می‌دهد تابع هزینه مولد مخالف  $J^{(D)}$  است. در حالی که برای محاسبه تابع هزینه مولد در نزول گرادینتی عبارت دوم در معادله تابع هزینه اهمیت دارد. بنابراین تابع هزینه مولد در یک بازی minimax به صورت زیر تعریف شده است:

$$J^{(G)}(G) = \mathbb{E}_{z \sim p_z(z)} [1 - \log D(G(z))] \quad (۳.۲)$$

در حالت کلی این بازی در یک تابع ارزش خلاصه می‌شود که به صورت زیر تعریف می‌شود:

$$\min_G \max_D \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [1 - \log D(G(z))] \quad (۴.۲)$$

که مولد سعی در کمینه کردن آن و تشخیص دهنده سعی در بیشینه کردن آن دارد. در صورتی که تابع هزینه مولد در یک بازی minimax واقعیت به خوبی عمل نمی‌کند، به دلیل اینکه وقتی مولد سعی در بیشینه کردن مقداری که تشخیص دهنده سعی در کمینه کردن آن را دارد، باعث می‌شود که تشخیص دهنده به آسانی تمامی داده‌های تولید شده توسط مولد را مصنوعی تشخیص دهد و آنها را رد کند. در نتیجه، نزول گرادینتی در مولد دچار اختلال می‌شود. مولد به جای آنکه مقدار تشخیص درست تشخیص دهنده را کمینه کند، مولد در یک بازی non-saturating heuristic سعی در بیشینه کردن اشتباه تشخیص دهنده دارد. و تابع هزینه آن در این بازی به صورت زیر تعریف می‌شود:

$$-\mathbb{E}_{x \sim p_{data}(x)} [\log D(G(x))] \quad (۵.۲)$$

در شکل ۱.۲ اختلاف مقدار تابع هزینه در حالت minimax و non-saturating heuristic قابل مشاهده است. محور افقی احتمال قبول شدن یک نمونه داده مصنوعی به عنوان یک داده واقعی را نشان می‌دهد. هر چه مقدار این عدد بیشتر باشد، مولد میزان هزینه کمتری را می‌گیرد. قسمت سمت چپ تابع که میزان  $D(G(z))$  نزدیک به صفر است، در ابتدای مرحله آموزش مدل اتفاق می‌افتد. در این زمان، تشخیص دهنده به راحتی می‌تواند تشخیص بدهد که داده نمونه متعلق به کدامین کلاس است. زیرا مولد در ابتدا شروع به ساخت داده‌های مصنوعی با توجه به توزیع تصادفی  $p_z(z)$  با یکسری پارامتر رندوم می‌کند. واضح است که منحنی بازی minimax یک خط مستقیم است، که نشان می‌دهد مولد دارای گرادینت بسیار کمی است. با استفاده از نزول گرادینتی، مولد روند بهبود مدل را در مراحل ابتدایی متوقف کرده است. در نقطه مقابل، منحنی بازی non-saturating heuristic مقدار گرادینت خودش را در قسمت راست از دست می‌دهد، و در این نقطه بهینه نمونه داده‌های تولید شده قادر به گمراه کردن تشخیص دهنده هستند. بنابراین، بازی non-saturating معمولاً در دنیای واقعی کاربرد بیشتری دارد و نسبتاً بازی minimax جنبه نظری دارد.

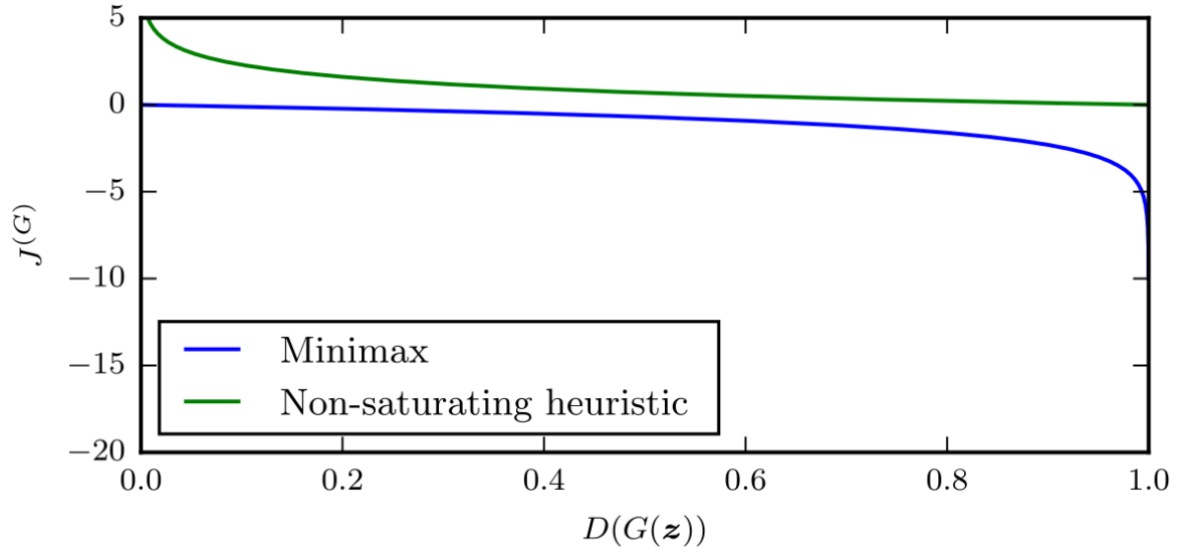
## ۳.۱.۲ انواع GAN برای تشخیص هیجانات در گفتار

در سال‌های گذشته شبکه‌های مولد تخصصی برای تشخیص هیجانات در گفتار استفاده شده است. برای مثال در یک آزمایش از یک DCGAN برای تحلیل گفتار احساسی به روش نیمه نظارت شده استفاده کردند. در این قسمت ما تمرکز بر استفاده از شبکه‌های مولد تخصصی برای تولید داده داریم، که در نهایت به ما کمک می‌کند که داده‌هایی تولید کنیم که شبیه به توزیع داده‌های واقعی در مسئله داده شده باشد. در این قسمت ما به تحلیل و بررسی سه نوع رایج شبکه‌های مولد تخصصی می‌پردازیم که شامل: adversarial autoencoder, conditional GAN, and CycleGAN است.

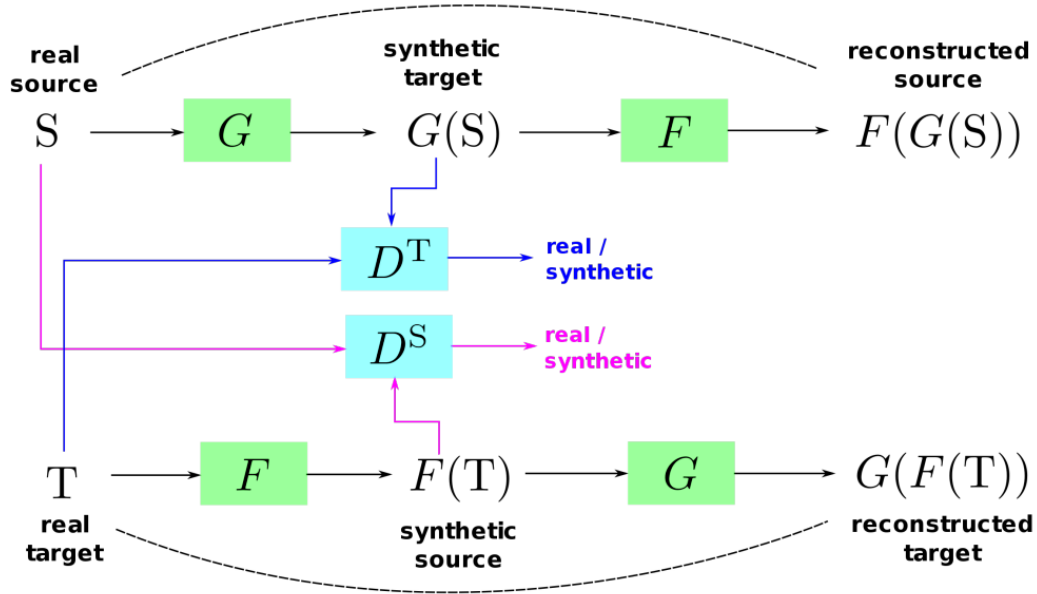
## ۴.۱.۲ شبکه‌های تخصصی حلقوی پایدار

<sup>۱</sup>شبکه‌های تخصصی حلقوی پایدار که معروف به CycleGAN است یک روش بسیار موفق برای حل مسائل ترجمه متن به متن با به کارگیری مجموعه داده‌های جفت نشده است. نگاشت دو طرفه‌ای که توسط شبکه CycleGAN آموخته می‌شود، می‌تواند ویژگی‌های خاص یک مجموعه تصویر را ثبت کند و بفهمد که چگونه می‌توان این ویژگی‌ها را به مجموعه تصویر دیگر ترجمه کند. موفقیت چشم‌گیر این نوع شبکه در انگیزه‌ای برای استفاده

<sup>1</sup>Cycle-consistent adversarial networks



شکل ۲.۲: مقایسه مقدار تابع هزینه مولد در بازی های minimax و non-saturating heuristic



شکل ۳.۲: ساختار CycleGAN

در انتقال احساسات است. شکل ۱.۳ ساختار کلی یک CycleGAN را تصویر کشیده است. این شبکه حاوی دو تابع نگاشت  $G$  و  $F$  است. تابع  $G$  یاد می‌گیرد که داده‌های نمونه را از منبع  $S$  به دامنه هدف  $T$  تبدیل کند. همچنین، تابع نگاشت  $F$  نیز یک نگاشت معکوس نسبت به  $G$  است. هر دوی این توابع نگاشت  $G$  و  $F$  را می‌توان یکسری مولد برای هدف و منبع تولید داده در نظر گرفت. به علاوه، این شبکه دارای دو تشخیص دهنده  $D^T$  و  $D^S$  است. تشخیص دهنده  $D^T$  در مقابل مولد  $\Pi$  وظیفه تشخیص واقعی بودن داده‌های تولید شده توسط  $G$  را برای داده‌های  $T$  دارد. همچنین،  $D^S$  تشخیص هم وظیفه تشخیص داده واقعی  $S$  را از مجموعه داده تولید شده  $F(T)$  دارد. همچنین در ادامه برای اطمینان حاصل کردن از اینکه تصویر ساخته شده قابلیت بازگردانی به داده نمونه اصلی را دارد، این شبکه سعی در ساخت نمونه داده‌های هدف و منبع را دارد به طوری که  $F(G(S))$  باید شبیه به  $S$  و  $G(F(T))$  باید شبیه به  $T$  باشد، که مربوط به قسمت پایداری این شبکه تخصصی است. میزان خطای CycleGAN تشکیل شده از یک خطای تخصصی و خطای حلقوی پایدار است. خطای تخصصی را می‌توانیم برای دو قسمت  $S$  و  $T$  در نظر بگیریم. خطای تخصصی به صورت زیر محاسبه می‌شود:

$$\mathcal{L}_{GAN}(G, D^T, S, T) = \mathbb{E}_{t \sim p_t(t)}[1 - \log D^T(t)] + \mathbb{E}_{s \sim p_s(s)}[1 - \log 1 - D^T(G(s))] \quad (6.2)$$

$$\mathcal{L}_{GAN}(F, D^S, T, S) = \mathbb{E}_{s \sim p_s(s)}[1 - \log D^S(s)] + \mathbb{E}_{t \sim p_t(t)}[1 - \log 1 - D^S(G(t))] \quad (7.2)$$

قابل ذکر است که میزان خطای تخصصی در قالب یک تابع ارزش بیان شده است. بنابراین حذف ما از این خطا  $\min_G \max_{D^T} \mathcal{L}_{GAN}(G, D^T, S, T)$  و  $\min_F \max_{D^S} \mathcal{L}_{GAN}(F, D^S, T, S)$  است. با داشتن ظرفیت به اندازه بزرگ، شبکه می تواند همگی تصاویر ورودی را به هر ترتیب تصادفی از تصاویر در دامنه مقصد نگاشت کند. به گونه ای که هر یک از نگاشت های یاد گرفته شده می تواند یک توزیع خروجی را ایجاد کند که با توزیع مقصد همخوانی داشته باشد. بنابراین این شبکه به یک تابع خطای دیگر هم نیاز دارد، که در پایین ذکر شده است:

$$\mathcal{L}_{cyc}(G, F) = \mathbb{E}_{t \sim p_t(t)}[||G(F(t)) - t||_1] + \mathbb{E}_{s \sim p_s(s)}[1 - \log 1 - D^T(G(s))] \quad (8.2)$$

در نتیجه به صورت کلی تابع هزینه به این صورت نمایش داده میشود:

$$\mathcal{L}(G, F, D^T, D^S) = \mathcal{L}_{GAN}(G, D^T, S, T) + \mathcal{L}_{GAN}(F, D^S, T, S) + \lambda \mathcal{L}_{cyc}(G, F) \quad (9.2)$$

## ۵.۱.۲ تقسیم بندی با رویکرد یادگیری ماشین یا یادگیری عمیق

یادگیری ماشینی برای تقسیم خودکار قسمت های مختلف یک تصویر استفاده می شود. معماری های U-Net در حل مسائل تقسیم بندی کارآمد هستند. در ادامه با توجه به اینکه روش استفاده شده مبتنی بر یادگیری عمیق است ابتدا اصطلاحات و تئوری مربوط به آن را بررسی می نمایم.

## ۲.۲ طیف نگاری

یک طرح طیف نگاره<sup>۱</sup> روشی بصری برای نمایش قدرت سیگنال یا بلندی سیگنال در طول زمان در فرکانس های مختلف موجود در یک شکل موج خاص است. این به ما کمک میکند که در طول زمان در فرکانس های مختلف مشاهده کنیم چگونه سطوح انرژی تغییر می کند. طیف نگاره ها نمودارهای دو بعدی هستند که بعد سوم با رنگ ها نشان داده می شود. زمان از چپ (قدیمی ترین) به راست (جوان ترین) در امتداد محور افقی می گذرد. محور عمودی نشان دهنده فرکانس است که می توان آن را به صورت زیر و بم یا تن نیز در نظر گرفت، با کمترین فرکانس در قسمت پایین و بیشترین فرکانس در قسمت بالایی قرار گرفته است. دامنه (یا انرژی یا "بلندی") یک فرکانس خاص در یک زمان خاص با بعد سوم، رنگ، با آبی تیره مربوط به دامنه های کم و رنگ های روشن تر تا قرمز مربوط به دامنه های به تدریج قوی تر (یا بلندتر) نشان داده می شود.

## ۳.۲ شبکه عصبی

شبکه های عصبی<sup>۲</sup> از مغز ما الگو گرفته اند. نودهایی<sup>۳</sup> وجود دارند که لایه ها<sup>۴</sup> را در شبکه تشکیل می دهند و دقیقاً مانند نورون های مغز ما، نواحی مختلف را به هم متصل می کنند. به ورودی های نودها در یک لایه، وزنی اختصاص می یابد که تأثیری را که پارامتر بر نتیجه پیش بینی کلی دارد، تغییر می دهد. از آنجا که وزن ها به پیوندهای بین نودها اختصاص داده می شوند، ممکن است هر نود تحت تأثیر وزن های مختلف قرار گیرد. شبکه عصبی تمام داده های آموزش را در لایه ورودی می گیرد. سپس داده ها را از میان لایه های پنهان عبور داده، مقادیر را براساس وزن هر نود تغییر می دهد و در نهایت مقداری را در لایه خروجی برمی گرداند. تنظیم درست یک شبکه عصبی برای رسیدن به نتایج سازگار و قابل اعتماد ممکن است کمی زمان بر باشد. آزمایش و آموزش شبکه عصبی، یک فرآیند متعادل سازی برای تعیین مهم ترین ویژگی های مدل است.

## ۴.۲ شبکه عصبی پیچشی

شبکه عصبی پیچشی<sup>۵</sup> نوع خاصی از شبکه عصبی با چندین لایه است که داده هایی را که آرایش شبکه ای دارند، پردازش کرده و سپس ویژگی های مهم آن ها را استخراج می کند. یک مزیت بزرگ استفاده از CNN ها این است که نیازی به انجام پیش پردازش زیادی روی تصاویر نیست. در بیشتر الگوریتم هایی که پردازش تصویر را انجام می دهند، فیلترها معمولاً توسط یک مهندس بر اساس روش های اکتشافی (heuristic) ایجاد می شوند. CNN ها می توانند مهم ترین ویژگی فیلترها را بیاموزند و چون به پارامترهای زیادی احتیاج نیست، صرفه جویی زیادی در وقت و عملیات آزمون و خطا صورت می گیرد.

<sup>1</sup>Spectrogram

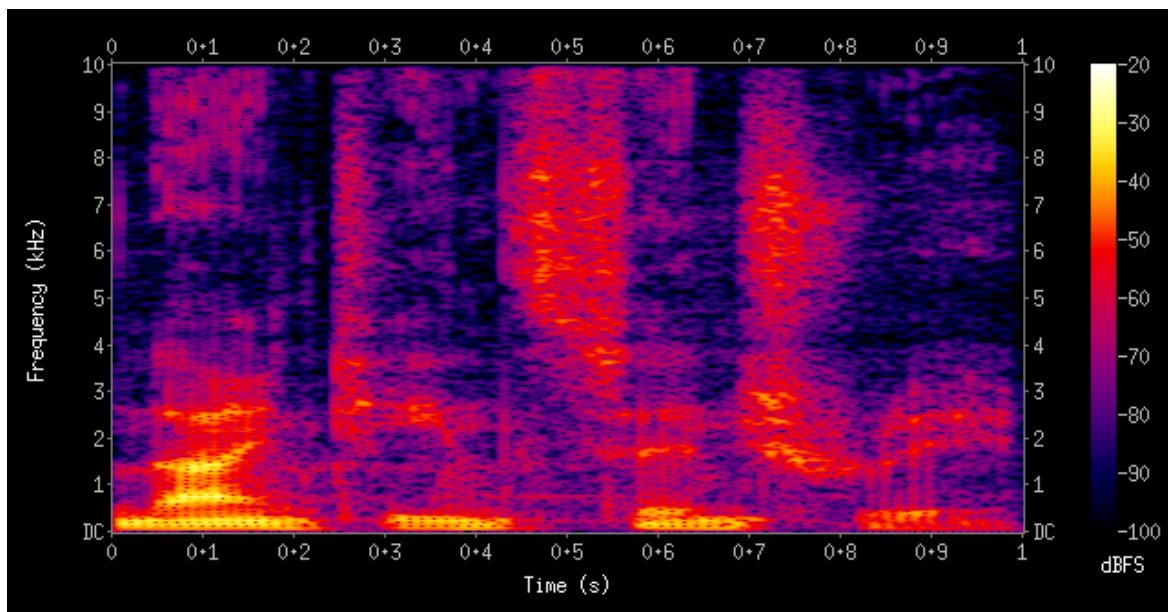
<sup>2</sup>Neural Networks - NNs

<sup>3</sup>Nodes

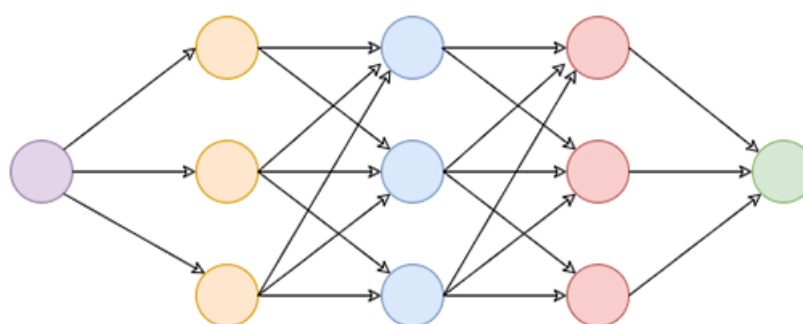
<sup>4</sup>Layers

<sup>5</sup>Convolutional Neural Network - CNN





شکل ۴.۲: طرحی از یک طرح نگاره



شکل ۵.۲: شبکه عصبی با چندین لایه پنهان. هر لایه چندین گره دارد.

هدف اصلی الگوریتم CNN این است که با حفظ ویژگی‌هایی که برای فهم آنچه داده‌ها نشان می‌دهند مهم هستند، داده‌ها را به فرم‌هایی که پردازش آن‌ها آسان‌تر است، درآورد. آن‌ها همچنین گزینه خوبی برای کار با مجموعه داده‌های عظیم هستند.

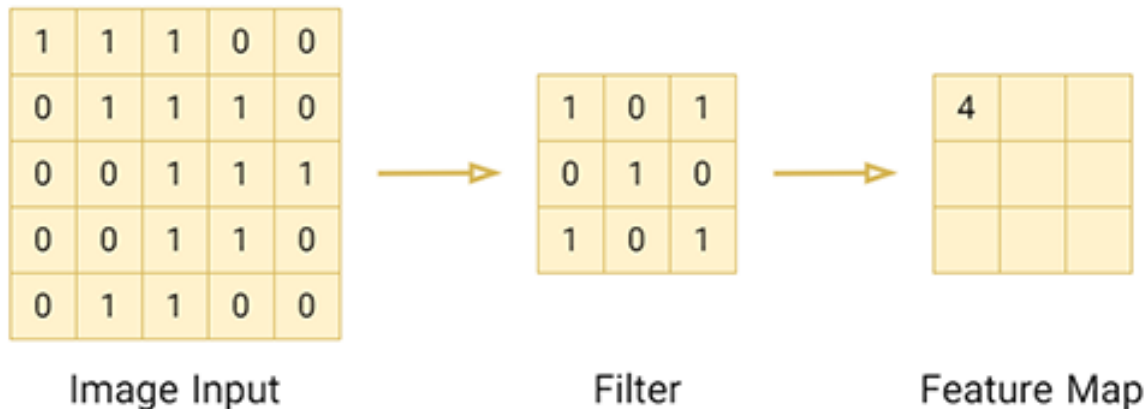
یک تفاوت بزرگ بین CNN و شبکه عصبی معمولی این است که CNN ها برای مدیریت ریاضیات پشت صحنه، از کانولوشن استفاده می‌کنند. حداقل در یک لایه از CNN، به جای ضرب ماتریس از کانولوشن استفاده می‌شود. کانولوشن‌ها تا دو تابع را می‌گیرند و یک تابع را برمی‌گردانند. CNN ها با اعمال فیلتر روی داده‌های ورودی شما کار می‌کنند. چیزی که آن‌ها را بسیار خاص می‌کند، این است که CNN ها می‌توانند فیلترها را هم‌زمان با فرایند آموزش، تنظیم کنند. به این ترتیب، حتی وقتی مجموعه داده‌های عظیمی مانند تصاویر داشته باشید، نتایج به‌خوبی و در لحظه دقیق‌تر می‌شوند.

از آنجا که می‌توان فیلترها را برای آموزش بهتر CNN تازه‌سازی کرد، نیاز به فیلترهای دستی از بین می‌رود و این انعطاف‌پذیری بیشتری در تعداد و ارتباط فیلترهایی که بر روی مجموعه داده‌ها اعمال می‌شوند، به ما می‌دهد. با استفاده از این الگوریتم، می‌توانیم روی مسائل پیچیده‌تری مانند تشخیص چهره کار کنیم.

کمبود داده یکی از مشکلاتی است که مانع استفاده از CNN می‌شود. با وجود اینکه می‌توان شبکه‌ها را با تعداد داده نسبتاً کمی، تقریباً ۱۰,۰۰۰ آموزش داد، هرچه اطلاعات بیشتری در دسترس باشد، CNN بهتر تنظیم می‌شود. داده‌ها باید بدون نقص و دارای برجسب باشند تا CNN بتواند از آن‌ها استفاده کند و این چیزی است که باعث می‌شود کار کردن با آن‌ها زمان‌بر و نیازمند منابع سنگین محاسباتی باشد.

شبکه‌های عصبی پیچشی براساس یافته‌های علوم اعصاب<sup>۱</sup> عمل می‌کنند. آن‌ها از لایه‌هایی از نورون‌های مصنوعی به نام نود<sup>۲</sup> ساخته شده‌اند. این نودها توابعی هستند که مجموع وزنی ورودی‌ها را محاسبه می‌کنند و یک نگاشت فعال‌سازی<sup>۳</sup> را برمی‌گردانند. این بخش پیچشی<sup>۴</sup> شبکه عصبی<sup>۵</sup> است.

## Convolution



شکل ۶.۲: بخش پیچشی شبکه عصبی

هر نود در یک لایه توسط مقادیر وزنی آن تعریف می‌شود. وقتی به یک لایه داده‌هایی را می‌دهید، برای مثال یک تصویر، مقادیر پیکسل را می‌گیرد و برخی از ویژگی‌های بصری را جدا می‌کند. هنگامی که داده‌ها را به CNN می‌دهید، هر لایه نگاشت‌های فعال‌سازی را برمی‌گرداند. این نگاشت‌ها ویژگی‌های مهم مجموعه داده را شناسایی می‌کنند. اگر به CNN تصویری را بدهید، ویژگی‌های مبتنی بر مقادیر پیکسل مانند رنگ‌ها را شناسایی می‌کند و تابع فعال‌سازی را به شما ارائه می‌دهد. معمولاً در تصاویر، CNN در ابتدا لبه‌های تصویر را پیدا می‌کند. سپس این تعریف جزئی از تصویر به لایه بعدی منتقل می‌شود و آن لایه شروع به شناسایی مواردی مانند گوشه‌ها و گروه‌های رنگی می‌کند. سپس این تعریف جدید از تصویر به لایه بعدی منتقل می‌شود و چرخه تا پیش‌بینی ادامه پیدا می‌کند.

همان‌طور که در تصویر زیر مشخص است، با افزایش تعداد لایه‌ها حداکثر تجمع (max-pooling) باید انجام شود. حداکثر تجمع فقط مرتبط‌ترین ویژگی‌ها از لایه موجود در نقشه فعال‌سازی را برمی‌گرداند و به لایه‌های بعدی منتقل می‌کند تا زمانی که به لایه آخر برسید. آخرین لایه CNN لایه طبقه‌بندی است که مقدار پیش‌بینی شده را براساس نگاشت فعال‌سازی تعیین می‌کند. اگر یک نمونه دست‌خط را به CNN بدهید، لایه‌ی طبقه‌بندی حروف موجود در تصویر را به شما می‌گوید. این همان چیزی است که وسایل نقلیه خودران برای تعیین اینکه یک شیء اتومبیل، شخص یا یک مانع است، استفاده می‌کنند.

انواع شبکه عصبی پیچشی به شرح زیر است:

- CNN یک‌بعدی: در این حالت، کرنل CNN در یک جهت حرکت می‌کند. CNN های یک‌بعدی معمولاً روی داده‌های سری زمانی استفاده می‌شوند.

- CNN دوبعدی: در این نوع از CNN، کرنل‌ها در دو جهت حرکت می‌کنند. CNN های دوبعدی در برجسب‌گذاری و پردازش تصویر کاربرد دارند.

- CNN سه‌بعدی: این نوع CNN دارای کرنلی است که در سه جهت حرکت می‌کند. محققان از این نوع CNN در تصاویر سه‌بعدی مانند سی‌تی‌اسکن و MRI استفاده می‌کنند.

از آنجایی که بیشتر مسائل با داده‌های تصویر مرتبط هستند، اغلب از CNN های دوبعدی استفاده می‌شود. در ادامه برخی از کاربردهایی که ممکن است از CNN ها استفاده شود، آورده شده است.

- تشخیص تصاویر با پیش‌پردازش کم

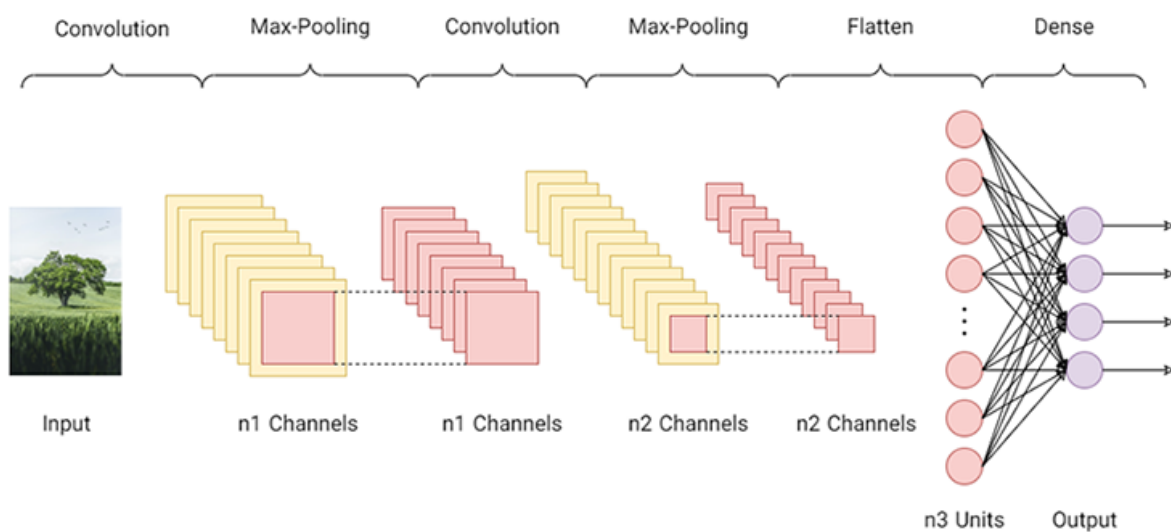
<sup>1</sup>neuroscience

<sup>2</sup>node

<sup>3</sup>activation map

<sup>4</sup>Convolutional

<sup>5</sup>Neural Network



شکل ۷.۲: معماری و لایه‌های یک شبکه عصبی پیچشی

- تشخیص دست‌خط‌های مختلف
- کاربردهای بینایی کامپیوتر (Computer Vision)
- استفاده در بانک‌داری بای خواندن ارقام در چک
- استفاده در سرویس‌های پستی برای خواندن کدپستی روی پاکت نامه