



الحمد لله رب العالمين



دانشگاه صنعتی اصفهان

دانشکده مهندسی برق و کامپیوتر

موضوع پروژه

ارائه و تحلیل روش‌های بهبود صحت و اطمینان در دسته‌بندی‌های هوش مصنوعی مبتنی بر شبکه‌های عصبی

گزارش پروژه کارشناسی مهندسی کامپیوتر

یوسف قادری

استاد راهنما

دکتر مهران صفایانی

شهریور ماه ۱۴۰۳

## تشکر و قدردانی

در ابتدا سپاس خدای را که شکیبایی و پشتکار لازم را به من در انجام پروژه و گذراندن دوره کارشناسی عطا فرمود. در ادامه قدردانی می‌کنم از زحمات جناب دکتر صفایانی که به بنده اعتماد لازم را داشتند و به صورت منظم بنده را راهنمایی می‌کردند در این مسیر بسیار از ایشان آموختم و توانستم در کنار ایشان رشد کنم.

## فهرست

۱	مقدمه	۳
۲	تعاریف اولیه	۵
۱-۲	مقدمه	۵
۲-۲	مقدمه‌ای بر یادگیری عمیق	۶
۳-۲	مقدمه‌ای بر شبکه‌های عصبی	۷
۴-۲	مقدمه‌ای بر شبکه عصبی پیچشی	۱۰
۵-۲	معیارهای ارزیابی مدل‌های یادگیری ماشین	۱۳
۳	پیاده‌سازی	۱۴
۱-۳	مقدمه	۱۴
۲-۳	تغییر در آستانه‌های مسائل مربوط به دسته بندی	۱۵
۳-۳	شانون انتروپی در طراحی یک دسته بند انتخابی	۱۹
۴-۳	روش یادگیری خود گام	۲۲
۴	جمع بندی	۳۱
۱-۴	چالش‌ها	۳۰
۲-۴	کارهای آینده	۳۱
۳-۴	نتیجه گیری	۳۲
۵	پیوست	۳۳



## چکیده

پیشرفت‌های اخیر در هوش مصنوعی، به ویژه در زمینه یادگیری عمیق<sup>۱</sup> و شبکه‌های عصبی<sup>۲</sup>، تحولات چشمگیری را به همراه داشته است. در این پروژه، تمرکز اصلی ما بر روی مسائل دسته‌بندی<sup>۳</sup> است. هدف ما بهبود صحت و دقت<sup>۴</sup> در دسته‌بندی‌های مختلف است که می‌تواند در کاربردهای متعددی که دقت و اطمینان از دسته‌بندی و پیش‌بینی اهمیت بالایی دارند، مفید باشد. یکی از کاربردهای مهم این پروژه، در شرایطی است که پیش‌بینی‌های نادرست می‌تواند هزینه‌های زیادی را به همراه داشته باشد. به عنوان مثال، در حوزه‌های پزشکی، مالی و صنعتی، دقت در پیش‌بینی و دسته‌بندی می‌تواند تأثیرات بزرگی بر نتایج نهایی داشته باشد. بنابراین، بهبود دقت و صحت در این زمینه‌ها از اهمیت بالایی برخوردار است. برای دستیابی به این هدف، از مدل‌های محاسباتی شبکه عصبی استفاده کرده‌ایم. در مراحل مختلف این پروژه، به دنبال پیاده‌سازی‌هایی بوده‌ایم که بتوانیم بهبودهای در عملکرد این مدل‌ها ایجاد کنیم. این تلاش‌ها شامل بهینه‌سازی پارامترها، استفاده از تکنیک‌های پیشرفته یادگیری و ارزیابی دقیق نتایج بوده است.

---

<sup>1</sup> Deep learning

<sup>2</sup> Neural Network

<sup>3</sup> Classification

<sup>4</sup> Accuracy

## کلیدواژه

High confidence classification, Selective classifier , Shanon Entropy, Self Paced Learning, Weak classifier, Softmax probability threshold, Easy and hard data



## فصل اول

### مقدمه

#### ۱ مقدمه

در پروژه‌های دسته‌بندی، دقت و صحت از مهم‌ترین معیارهایی هستند که موفقیت یک مدل را تعیین می‌کنند. این معیارها، به‌ویژه در کاربردهای حساس مانند تشخیص بیماری‌ها، نقش حیاتی دارند. هدف اصلی این پروژه، یافتن روش‌هایی است که بتوانند این دقت و صحت را بهبود بخشند، به گونه‌ای که نتایج دسته‌بندی نه تنها قابل اعتماد، بلکه با اطمینان و قاطعیت بالا ارائه شوند. یکی از چالش‌های اساسی در مسائل دسته‌بندی، تعادل میان دقت و پوشش داده‌های آزمون است. در بسیاری از موارد، افزایش دقت ممکن است به کاهش درجه پوشش داده‌ها منجر شود و بالعکس، این موضوع ما را در موقعیتی قرار می‌دهد که باید تصمیم بگیریم تا چه حدی از دقت و اطمینان را در مقابل پوشش داده‌ها نیاز داریم. این تصمیم‌گیری، یک سبک و سنگین کردن<sup>۱</sup> حیاتی است که بر اساس شرایط خاص هر مسئله و اهداف نهایی آن تعیین می‌شود.

در این پروژه، به دنبال استفاده از روش‌هایی هستیم که در آن‌ها شاید نتوانیم همه داده‌های تست را دسته‌بندی کنیم، اما هنگامی که دسته‌بندی انجام می‌شود، با اطمینان و دقت بسیار بالا صورت گیرد. این رویکرد به ما امکان می‌دهد که حتی در شرایطی که ریسک‌های زیادی وجود دارد، نتایج قابل اعتماد و مطمئنی ارائه کنیم.

این موضوع به‌ویژه در سناریوهایی مانند تشخیص زودهنگام بیماری‌ها یا پیش‌بینی وقایع بحرانی اهمیت پیدا می‌کند، جایی که حتی یک اشتباه کوچک می‌تواند پیامدهای بزرگی به دنبال داشته باشد.

---

<sup>۱</sup> Trade-off

این مطالعه با انجام آزمایش های دقیق و عملی بر روی مجموعه داده های مختلف در پی یافتن این روش ها و به آزمایش گذاشتن اینهاست نتایج این مطالعه می تواند در حوزه کاربرد هوش مصنوعی در زمینه های مالی و صنعتی و پزشکی مفید باشد در این فصل به مقدمه ای از اهمیت مسائل دسته بندی در هوش مصنوعی پرداخته شد .

در فصل دوم بدنبال تعاریف اولیه و معرفی ابزارهای مورد استفاده هستیم و هم چنین کارکرد هر کدام را توضیح خواهیم داد .

در فصل سوم به پیاده سازی ها و مراحل عملی که در انجام این پژوهش صورت گرفته شده و نتایجی که بدست آمده خواهیم پرداخت و در فصل چهارم به جمع بندی مباحث این مطالعه و چالش های دیگر موجود اشاره خواهد شد.

در نهایت، این پژوهش امیدوار است که بتواند ایده هایی را که در نظر گرفته است به ثمر برساند تا از زیان هایی که در صنایع مختلف ممکن است به وجود بیاید و جبران ناپذیر باشد جلوگیری کند و به عنوان منبعی برای بهبود تصمیم گیری های مهندسی در محیط هایی که از این موضوعات بهره میبرند ، عمل کند

## فصل دوم

### تعاریف اولیه

#### ۲ تعاریف اولیه

##### ۱-۲ مقدمه

در این فصل به صورت کلی مفاهیم پایه‌ای که در این پروژه نیاز است توضیح داده می‌شود ابتدا با مقدمه‌ای درباره یادگیری عمیق آشنا خواهیم شد و سپس شبکه عصبی و بخش‌های مختلف آن توضیح داده می‌شود، سپس درباره نوع خاصی از شبکه عصبی به نام شبکه عصبی پیچشی<sup>۱</sup> صحبت خواهیم کرد و در نهایت به معرفی معیارهای ارزیابی مدل‌های یادگیری ماشین خواهیم پرداخت و تلاش می‌شود تا حد ممکن شما را با اجزای مختلف آن آشنا کنیم.

---

<sup>۱</sup> CNN(Convolutional neural network)

## ۲-۲ مقدمه ای بر یادگیری عمیق

یادگیری عمیق زیر شاخه ای از علم یادگیری ماشین هست که در آن با تکیه بر شبکه های عصبی به کامپیوتر این قابلیت را میدهد که بدون استفاده از هیچگونه برنامه نویسی آشکاری بتواند یاد بگیرد و بدون اینکه لازم باشد در هنگام برنامه نویسی تمامی شرایط در غالب جملات شرطی نیاز به بیان صریح داشته باشد آنها را از داده های موجود استخراج کرده و بتواند تصمیم گیری های لازم را انجام بدهد.

در مدل های یادگیری عمیق، یادگیری می تواند به سه روش نظارت شده<sup>۱</sup>، نیمه نظارت شده<sup>۲</sup> و بدون نظارت باشد. در مدل های نظارت شده، کامپیوتر سعی میکند که یک مدل ریاضی را بر اساس داده های ورودی و خروجی ایجاد کند و در این راستا خروجی ها از قبل تفکیک شده و برچسب زده شده<sup>۳</sup> هستند در این فرایند که به فرایند یادگیری<sup>۴</sup> شناخته می شود کامپیوتر سعی میکند که با استفاده از داده های ورودی که به آن داده شده است، مدلی ریاضی ایجاد کند تا با استفاده از مدل به دست آمده و ورودی ها به خروجی مورد نظر که آن هم به کامپیوتر داده می شود برسد در این حین نیز تابعی به عنوان تابع هدف و یا تابع خطا<sup>۵</sup> در نظر گرفته می شود و دقت عملکرد مدل بدست آمده بر اساس تابع هدف یا تابع خطا محاسبه میشود (در این روش همواره بدنبال این هستیم که میزان خطا در این تابع را کاهش بدهیم) از همین منظر امید هست که کامپیوتر در حین فرایند آموزش این خطا را کاهش دهد تا در هنگام اعتبارسنجی<sup>۶</sup> و آزمودن<sup>۷</sup> مدل بر روی داده های جدید بتواند به درستی و با دقت بالا عمل کند [۱].

---

<sup>1</sup> Supervised

<sup>2</sup> Semi-supervised

<sup>3</sup> Labeled

<sup>4</sup> Training

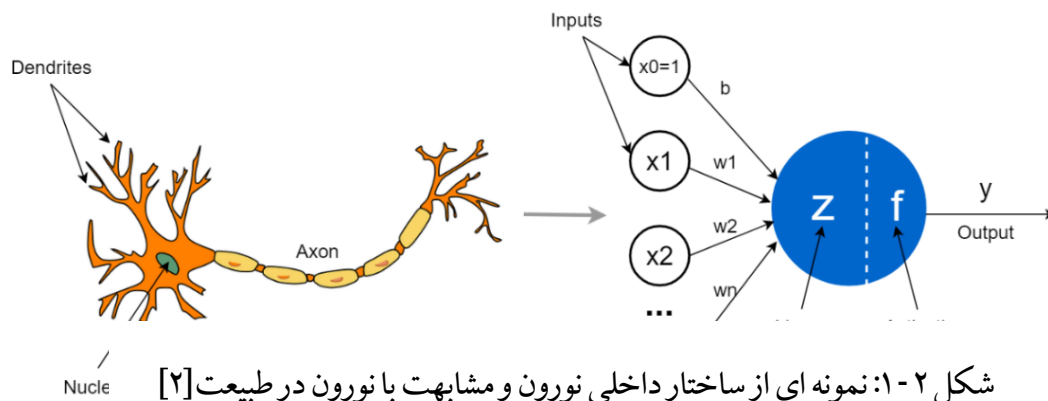
<sup>5</sup> Loss function

<sup>6</sup> Validation phase

<sup>7</sup> Test

## ۳-۲ مقدمه‌ای بر شبکه‌های عصبی

ابتدا به صورت مختصر شبکه‌های عصبی را معرفی کنیم، شبکه عصبی مصنوعی نوعی مدل محاسباتی الهام گرفته از ساختار و عملکرد مغز انسان است. شبکه‌های عصبی از مجموعه‌ای از واحدهای محاسباتی به نام نورون‌ها (یا واحدهای پردازشی) تشکیل شده‌اند که به یکدیگر متصل شده و لایه‌های مختلفی را تشکیل می‌دهند. هدف اصلی شبکه‌های عصبی این است که از طریق یادگیری الگوها از داده‌ها، قادر به انجام وظایفی مانند دسته‌بندی، پیش‌بینی و تشخیص الگوها شوند.



شکل ۲-۱: نمونه‌ای از ساختار داخلی نورون و مشابهت با نورون در طبیعت [۲]

برخی از اجزاء ساختار شبکه عصبی به صورت خلاصه توضیح داده شده است:

**نورون:** واحد بنیادی پردازش است که به طور کلی وظیفه دریافت ورودی‌ها، پردازش آن‌ها و تولید خروجی را بر عهده دارد. نورون‌ها به طور مستقیم از ساختار نورون‌های زیستی الهام گرفته‌اند و به عنوان واحدهای محاسباتی عمل می‌کنند که داده‌ها را پردازش کرده و به لایه‌های بعدی منتقل می‌کنند.

**وزن‌ها:** هر ورودی با یک مقدار وزن مرتبط است. وزن‌ها به شکل مقادیر عددی تعیین می‌شوند و نشان‌دهنده اهمیت هر ورودی برای آن نورون هستند. وزن‌ها با  $w_1, w_2, w_3, \dots, w_n$  نشان داده می‌شوند.

وزن‌ها نقش بسیار مهمی در یادگیری شبکه عصبی دارند. در طی فرآیند آموزش، این وزن‌ها تغییر می‌کنند تا نورون‌ها بتوانند به ورودی‌ها پاسخ بهتری بدهند. وزن‌ها تعیین می‌کنند که چه مقدار از یک ورودی در تصمیم‌گیری نهایی تأثیر دارد. نورون، ورودی‌های خود را با وزن‌های مربوط به آن‌ها ترکیب می‌کند.

بایاس: علاوه بر ورودی‌های وزن‌دار، نورون معمولاً یک مقدار ثابت به نام بایاس نیز دارد. بایاس به نورون کمک می‌کند تا قدرت کلی نورون را تنظیم کند و همچنین برای بهبود دقت مدل و تنظیم منحنی تصمیم استفاده می‌شود. بایاس به هر نورون اجازه می‌دهد تا حتی زمانی که ورودی‌ها صفر هستند، همچنان مقداری خروجی تولید کند.

**محاسبه مجموع وزن‌دار:** نورون مجموع وزن‌دار ورودی‌ها را به دست می‌آورد. این کار با ضرب هر ورودی در وزن مربوط به آن و سپس جمع کردن آن‌ها با هم انجام می‌شود. برای نورونی با  $n$  ورودی، مجموع وزن‌دار به شکل زیر است:

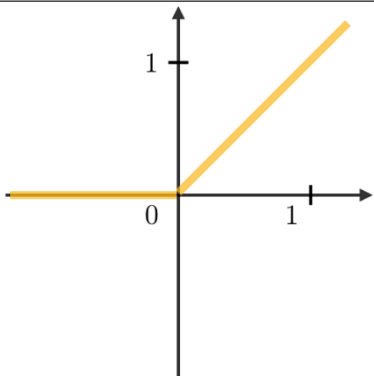
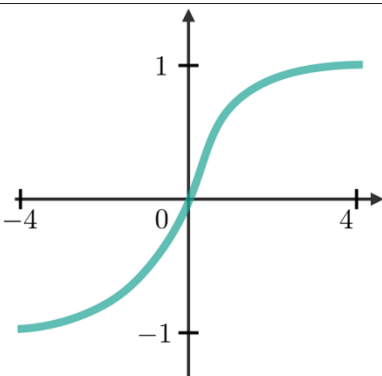
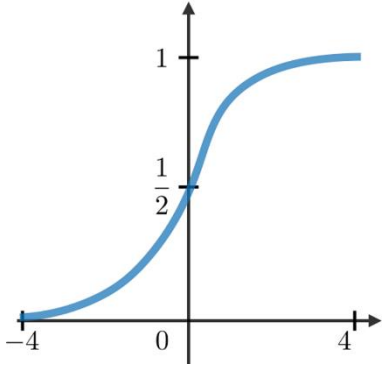
$$Z = b + w_1x_1 + w_2x_2 + \dots + w_nx_n$$

این مقدار  $Z$  نشان‌دهنده مقدار ترکیب شده ورودی‌ها به همراه بایاس است.

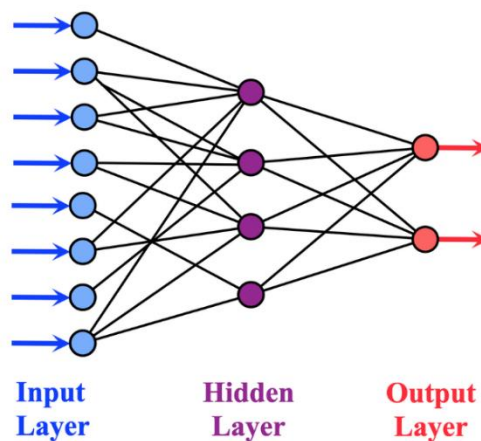
**تابع فعال‌سازی:** پس از محاسبه مجموع وزن‌دار، نورون از یک تابع فعال‌سازی استفاده می‌کند تا یک خروجی تولید کند. تابع فعال‌سازی یک تابع غیرخطی است که به نورون‌ها کمک می‌کند تا الگوهای پیچیده‌تر را مدل‌سازی کنند. بدون این تابع، شبکه عصبی فقط می‌تواند روابط خطی بین ورودی‌ها و خروجی‌ها را یاد بگیرد. در جدول زیر نمونه‌هایی از معروف‌ترین توابع فعال‌ساز معرفی شده است.

---

<sup>1</sup> Activation Function

Relu	Tanh	Sigmoid
$g(z) = \begin{cases} z & z \geq 0 \\ 0 & z < 0 \end{cases}$	$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	$g(z) = \frac{1}{1 + e^z}$
		

شکل ۲-۱: انواع توابع فعال‌سازی [۳]



شکل ۲-۲: نمونه‌ای از ساختار شبکه‌های عصبی [۴]

**لایه ورودی:** شامل نورون‌هایی است که داده‌های ورودی از طریق آنها به شبکه عصبی تزریق میشوند.

**لایه‌های مخفی:** نورون‌های موجود در این لایه‌ها پردازش اصلی داده‌ها را انجام می‌دهند و مسئولیت استخراج ویژگی‌ها و یادگیری الگوهای پیچیده را دارند. شبکه‌های عصبی می‌توانند چندین لایه مخفی داشته باشند، که در این صورت به آن‌ها شبکه‌های عصبی عمیق گفته می‌شود.

**لایه خروجی:** شامل نورون‌هایی است که خروجی نهایی شبکه (پیش‌بینی یا دسته‌بندی) را ارائه می‌دهند. برای درک بهتر می‌توانید به شکل ۲-۲ مراجعه کنید. [۵]

## ۴-۲ مقدمه‌ای بر شبکه عصبی پیچشی<sup>۱</sup>

حال که با شبکه‌های عصبی بطور کلی آشنا شدیم می‌خواهیم نوع خاصی از شبکه‌ها را به نام شبکه‌های عصبی پیچشی به طور اختصاصی معرفی کنیم. بخش‌هایی در شبکه عصبی پیچشی به طور خاص وجود دارد که در سایر شبکه‌های عصبی وجود ندارد. در این شبکه‌های پیچشی سه نوع لایه خاص وجود دارد:

**لایه پیچشی<sup>۲</sup>:** این لایه اصلی‌ترین جزء شبکه عصبی پیچشی است و عمل پیچش را انجام می‌دهد. در این لایه یک فیلتر یا هسته<sup>۳</sup> کوچک روی تصویر ورودی حرکت می‌کند و مقادیر پیکسل‌های آن را با هم ترکیب می‌کند. هدف فیلتر این است که ویژگی‌های محلی (مانند لبه‌ها، بافت‌ها، الگوهای کوچک) را در تصویر شناسایی کند. هر فیلتر می‌تواند یک ویژگی خاص از تصویر را یاد بگیرد. برای مثال، یک فیلتر ممکن است

<sup>۱</sup> CNN

<sup>۲</sup> Convolutional layer

<sup>۳</sup> Kernel

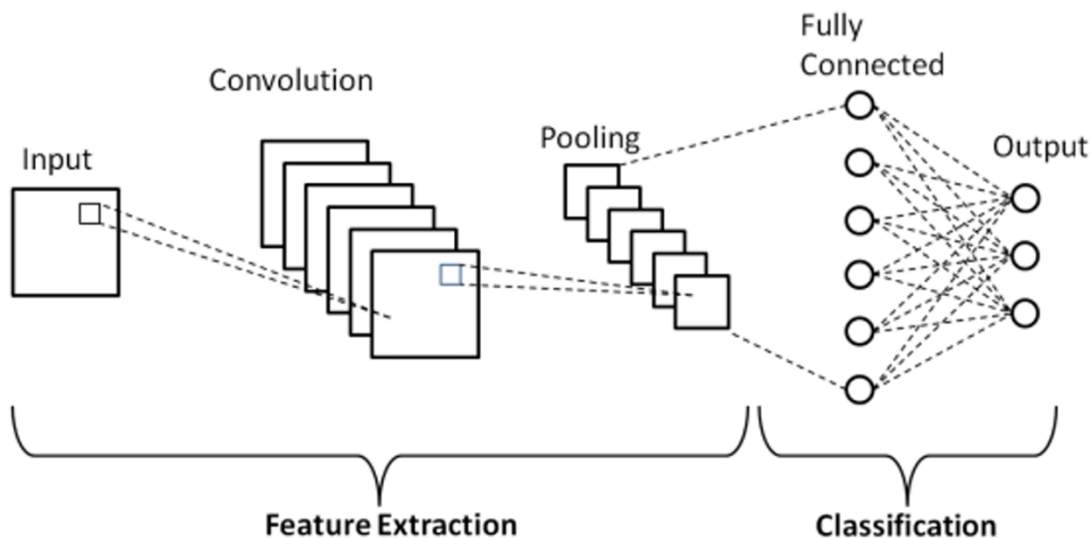


لبه‌های افقی را شناسایی کند، در حالی که دیگری ممکن است به دنبال گوشه‌ها یا منحنی‌ها باشد. نتیجه عملیات پیش‌پیش به عنوان یک نقشه ویژگی<sup>۱</sup> شناخته می‌شود که نشان‌دهنده ویژگی‌های استخراج شده است.

**لایه تجمیع<sup>۲</sup>:** یک لایه تجمیع برای کاهش ابعاد نقشه‌های ویژگی استفاده می‌شود. این لایه وظیفه کاهش ابعاد و کاهش پیچیدگی محاسباتی را بر عهده دارد. پرکاربردترین نوع تجمیع، بیشترین تجمیع<sup>۳</sup> است، که در آن بزرگ‌ترین مقدار در یک ناحیه کوچک از نقشه ویژگی انتخاب می‌شود. این لایه باعث افزایش مقاومت شبکه در برابر جابجایی‌های کوچک در داده‌های ورودی (مانند تغییرات کوچک در مکان اشیاء در تصویر) می‌شود.

**لایه کاملاً متصل<sup>۴</sup>:** در انتهای شبکه، معمولاً یک یا چند لایه کاملاً متصل قرار دارد. این لایه‌ها مانند لایه‌های سنتی شبکه عصبی عمل می‌کنند و خروجی‌های نهایی را با توجه به ویژگی‌های استخراج شده از لایه‌های پیش‌پیش و تجمیع ارائه می‌دهند.

در شکل زیر نمای شماتیک شبکه‌های عصبی پیش‌پیش ترسیم شده است. [۶]



شکل ۲-۳: شماتیک پایه از معماری شبکه‌های CNN [۷]

<sup>۱</sup> Feature Map

<sup>۲</sup> Pooling layer

<sup>۳</sup> Max Pooling

<sup>۴</sup> Fully connected Layer

## ۵-۲ معیارهای ارزیابی مدل‌های یادگیری ماشین

دقت<sup>۱</sup>: نشان می‌دهد که مدل چه میزان از پیش‌بینی‌هایش درست بوده است. این معیار تعداد پیش‌بینی‌های صحیح را به نسبت کل نمونه‌ها اندازه‌گیری می‌کند.

$$Accuracy = \frac{\text{correct prediction}}{\text{all prediction}}$$

صحت<sup>۲</sup>: معیاری است که کیفیت پیش‌بینی‌های مثبت مدل را می‌سنجد. این معیار به ما می‌گوید از تمام نمونه‌هایی که به عنوان مثبت پیش‌بینی شده‌اند، چند نمونه واقعاً مثبت بوده‌اند.

$$Precision = \frac{\text{True positives}}{\text{True Positives} + \text{False Positives}}$$

فراخوانی<sup>۳</sup>: درصد نمونه‌های مثبت واقعی را نشان می‌دهد که مدل آن‌ها را به درستی شناسایی کرده است. به عبارتی، فراخوانی نشان می‌دهد که مدل چه میزان از موارد مثبت را پوشش داده است

$$Recall = \frac{\text{True positives}}{\text{True positives} + \text{False Negatives}}$$

امتیاز<sup>۴</sup>F1: میانگین هماهنگ صحت و فراخوانی است. این معیار زمانی مفید است که به تعادل بین صحت و فراخوانی نیاز باشد. اگر یکی از این دو معیار پایین باشد، امتیاز F1 نیز کاهش می‌یابد. این معیار مناسب کاربردهایی است که نیاز به توازن میان صحت و فراخوانی داریم. [۸]

$$F1 - score = \frac{Precision \times Recall}{Precision + Recall} \times 2$$

---

<sup>1</sup> Accuracy

<sup>2</sup> Precision

<sup>3</sup> Recall

<sup>4</sup> F1-score

## فصل سوم

### پیاده‌سازی

#### ۳ پیاده‌سازی

##### ۱-۳ مقدمه

در این پروژه برای فایق آمدن بر مسائل و مشکلات مطرح شده، از روش‌های زیر استفاده شد و هر کدام از این روش‌ها بر روی یک مجموعه داده با یک شبکه عصبی مشخص به ورطه آزمایش گذاشته شده است. لازم به ذکر است تمامی پیاده‌سازی‌ها با استفاده از کتابخانه پایتورچ<sup>۱</sup> صورت گرفته است، پایتورچ یک بستر و مجموعه‌ای از کتابخانه‌ها و ابزار رایگان متن باز هست که به افراد اجازه پیاده‌سازی موثر و بهینه شبکه‌های عصبی را میدهد. در ابتدا مباحث تئوری مربوط به هر روشی گفته خواهد شد و سپس درباره جزئیات پیاده‌سازی‌هایی که انجام شده است خواهیم گفت.

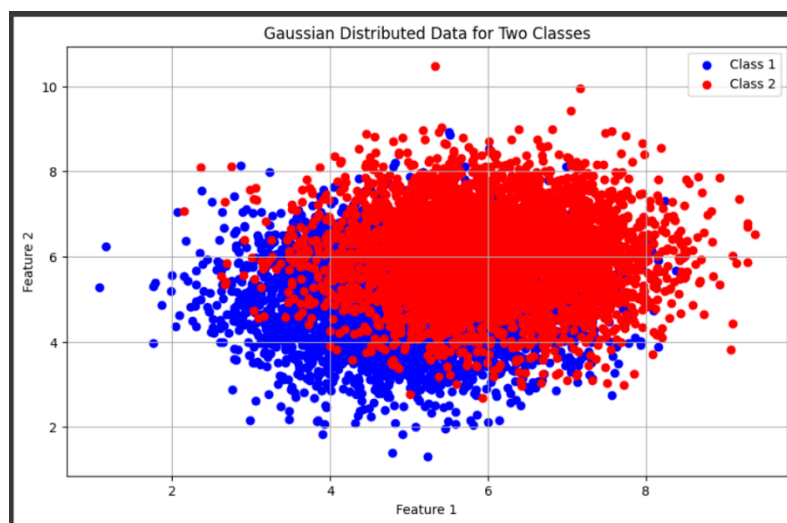
---

<sup>۱</sup> Pytorch

### ۲-۳ تغییر در آستانه های مسائل مربوط به دسته بندی

در یکی از روش های بهبود دقت (Precision) در دسته بندی شبکه های عصبی، از مفهوم حداکثر احتمال تعلق به یک دسته بندی استفاده می شود. به طور معمول، در فرآیند آموزش شبکه عصبی، داده های آزمایشی بدون در نظر گرفتن هیچ گونه آستانه یا معیار مشخصی وارد شبکه شده و به صورت خود کار طبقه بندی می شوند. این رویه می تواند منجر به کاهش دقت (Precision) در برخی از دسته ها شود که از دیدگاه ما نتیجه ای مطلوب نیست. در این ارائه، به جای آنکه تمامی داده های آزمودنی را بدون در نظر گرفتن آستانه ای مشخص طبقه بندی کنیم، از آستانه ای تعریف شده استفاده می کنیم تا کیفیت دسته بندی بهبود یابد. این موضوع در حقیقت باعث میشود درجه اطمینان<sup>۱</sup> ما از پیش بینی هایی که در هنگام آزمون انجام می شود بالاتر برود. [۹] این رویکرد را ابتدا بر روی داده های ساده مورد آزمایش قرار می دهیم و سپس آن را برای روی داده های واقعی، با استفاده از دیتاست CIFAR10، که به عنوان دیتاست اصلی دسته بندی انتخاب شده است و در نهایت نتایج این رویکرد را بررسی و ارزیابی خواهیم نمود. [۱۰]

این رویکرد، ضمن حفظ دقت در طبقه بندی، به ما اجازه می دهد تا نتایج با کیفیت تر و قابل اعتمادتری از داده های آزمودنی به دست آوریم.



شکل ۳-۱: داده های تصنعی دو ویژگی و دو کلاسه

<sup>1</sup> Confidence grade

بعد از مرحله آموزش به سراغ ارزیابی مدل خواهیم رفت و برای آستانه‌های متفاوت مقدار پارامترهای صحت، فراخوانی، دقت و امتیاز F1 را محاسبه می‌کنیم:

آستانه	دقت	صحت	فراخوانی	امتیاز F1
۰/۳	۹۲٪	۹۸٪	۸۴٪	۹۱٪
۰/۴	۹۳٪	۹۸٪	۸۶٪	۹۲٪
۰/۵	۹۳٪	۹۶٪	۸۸٪	۹۳٪
۰/۶	۹۴/۵٪	۹۵٪	۹۲٪	۹۳٪
۰/۷	۹۴٪	۹۲٪	۹۴٪	۹۲٪
۰/۸	۹۲/۵٪	۸۸٪	۹۶٪	۹۱٪
۰/۹	۹۱٪	۸۴٪	۹۸٪	۹۰٪

جدول ۱-۳: پارامترهای ارزیابی شبکه برای دسته اول

آستانه	دقت	صحت	فراخوانی	امتیاز F1
۰/۳	۹۲٪	۸۷٪	۹۹٪	۹۲٪
۰/۴	۹۳٪	۸۸٪	۹۹٪	۹۳٪
۰/۵	۹۳٪	۹۰٪	۹۷٪	۹۳٪
۰/۶	۹۴/۵٪	۹۳٪	۹۶٪	۹۴٪
۰/۷	۹۴٪	۹۵٪	۹۳٪	۹۴٪
۰/۸	۹۲/۵٪	۹۶٪	۸۸٪	۹۲٪
۰/۹	۹۱٪	۹۸٪	۸۴٪	۹۰٪

جدول ۲-۳: پارامترهای ارزیابی شبکه برای دسته دوم

## تحلیل و تفسیر نتایج

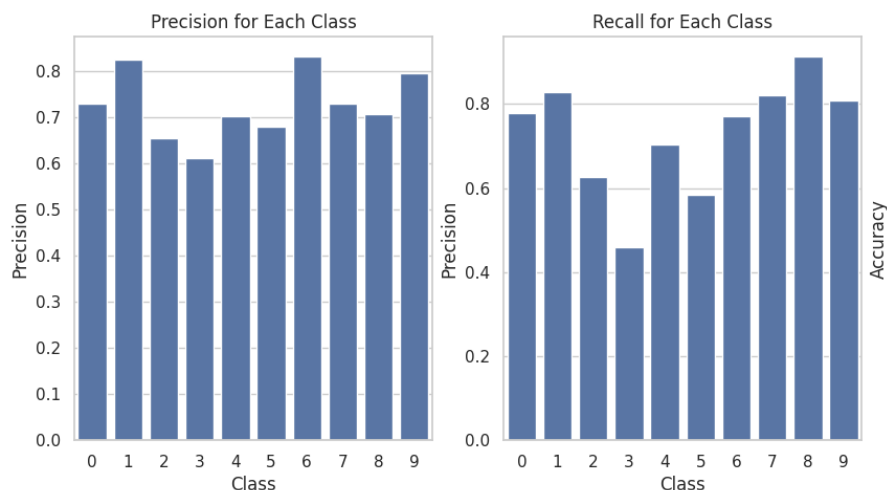
با دقت در این نتایج متوجه می شویم که ستون دقت برای هر دو جدول یکسان هست اما با افزایش میزان آستانه در دو ستون میزان صحت و فراخوانی دو روند کاملاً متفاوت را دارند در جدول ۱-۳ با افزایش آستانه مقدار صحت در حال کاهش هست این به گونه ای هست که ما داده های مربوط به دسته اول را با صحت کمتری دسته بندی می کنیم و به مرور میزان اشتباهات ما در دسته بندی کردن داده های این کلاس در حال افزایش است همینطور مقدار پارامتر فراخوانی در این جدول در حال افزایش هست به این معنی که تعداد داده های بیشتری از کلاس اول را دسته بندی می کنیم که البته این مقدار این پارامتر فارغ از اینکه نتیجه درست یا غلط هست محاسبه می شود در جدول ۲-۳ اما روند کاملاً متفاوت هست افزایش آستانه سبب افزایش میزان صحت در دسته بندی داده های مربوط به کلاس دوم شده و هم چنین کاهش میزان فراخوانی داده های کلاس دوم شده است این نتایج تفاسیر تئوری و پیش فرض هایی که در ابتدا در نظر گرفته ایم را تایید میکند

## پیاده سازی بروی دیتاست های واقعی

در ادامه پیاده سازی و بررسی تئوری روش مطرح شده، این بار به جای داده های مصنوعی، از دیتاست واقعی CIFAR10 استفاده می کنیم تا کارایی این روش را بروی داده های واقعی ارزیابی نماییم. این دیتاست شامل ۶۰,۰۰۰ تصویر است که از این تعداد، ۵۰,۰۰۰ تصویر به عنوان داده های آموزشی و ۱۰,۰۰۰ تصویر برای تست مورد استفاده قرار می گیرند. هر کلاس در این مجموعه شامل ۵,۰۰۰ تصویر آموزشی و ۱,۰۰۰ تصویر تست است. تفاوت اصلی این مرحله در مقایسه با پیاده سازی های قبلی، جستجوی آستانه هایی مرتبط با توزیع پذیری داده های اعتبارسنجی در حین فعالیت شبکه است. برای این منظور از ماژول Numpy و تابع np.percentile() استفاده می کنیم. [۱۱] این تابع دو ورودی دریافت می کند: ورودی اول مجموعه ای از بالاترین احتمالاتی است که شبکه ما برای داده های اعتبارسنجی پیش بینی کرده و ورودی دوم عددی بین صفر تا صد (به استثنای خود صد) است. خروجی تابع، عددی خواهد بود که بزرگتر از درصدی از داده ها است که به عنوان ورودی به تابع داده شده است. برای مثال، اگر پارامتر دوم برابر با ۵۰ تنظیم شود، خروجی تابع np.percentile() از ۵۰ درصد داده ها بزرگتر خواهد بود.

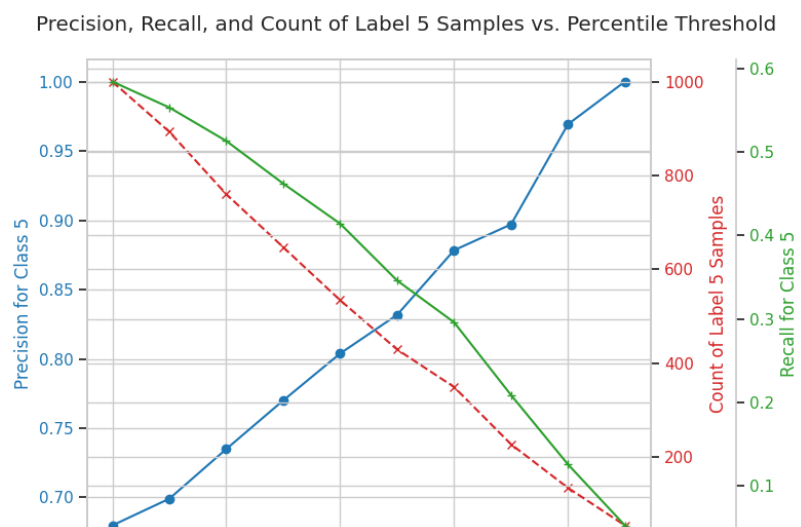
در این پیاده سازی، پس از مشاهده تمامی داده های اعتبارسنجی، درصد های مختلف از صفر تا ۱۰۰ درصد در لیستی ذخیره می شوند. در مرحله تست و ارزیابی، آستانه ها از این لیست انتخاب می شوند. اگر بالاترین احتمال پیش بینی شده برای داده های تست بزرگتر یا مساوی آستانه مربوطه باشد، آن داده به عنوان داده آسان شناخته شده و شبکه به دسته بندی آن می پردازد. در غیر این صورت، از دسته بندی آن صرف نظر می شود.

این روش به ما امکان می دهد تا داده هایی که تشخیص آنها با اطمینان کافی صورت نگرفته، به درستی از جریان دسته بندی حذف شوند و دقت کلی سیستم افزایش یابد. ابتدا صحت و فراخوانی را برای داده های هر کلاس از طریق روشی که بطور عمومی استفاده میشود (دسته بندی همه داده های آزمون) در نمودارهای زیر مشاهده میکنید:



شکل ۳-۲: الف) میزان صحت ب) میزان فراخوانی برای هر کلاس

اما هنگامی که از روشی که در بالا ذکر شد استفاده می کنیم نتایج ما متفاوت خواهد شد که در نمودارهای شکل زیر سه پارامتر مهم بررسی شده است یکی تعداد داده هایی هست که در صورت استفاده از آستانه های مختلف به عنوان داده های آسان شناخته می شود دیگری میزان صحت و آخری فراخوانی مربوط به یک دسته خاص هست که در اینجا کلاس پنجم به عنوان نمونه نمایش داده شده است همانطور که در شکل بالا مشاهده میکنید صحت برای کلاس پنجم برای با ۶۵٪ بوده است:



شکل ۳-۳: خط آبی: نمایش صحت، خط قرمز: تعداد نمونه های آسان و خط سبز: فراخوانی در روش تغییر آستانه

## تفسیر و تحلیل نتایج

به طور کلی بعد از آنکه خروجی مدل را به ازای ورودی‌های داده‌آزمون بدست آوردیم از یک لایه سافت مکس استفاده می‌کنیم که بعد از اعمال آن به ازای هر داده احتمال تعلق آن داده به هر کلاس را خروجی می‌دهد در نهایت بیشینه احتمال را انتخاب خواهیم کرد و اگر این بیشینه احتمال از امتیاز اطمینان ما بیشتر باشد به منزله این هست که داده ما آسان هست و ما آن را دسته بندی خواهیم کرد این روش باعث افزایش میزان صحت در دسته بندی‌های ما خواهد شد پس این روش در پیاده سازی بر روی داده‌های واقعی نیز کارساز هست و نتایج مطابق پیش بینی‌های ما هستند.

## ۳-۳ آنتروپی شانون در طراحی یک دسته بند انتخابی

تا به اینجا به خوبی روشن شده است که در فرآیند تست یک شبکه عصبی، رویکردی مبتنی بر انتخاب گزینشی اتخاذ کرده‌ایم. طراحی و پیاده‌سازی یک دسته‌بند گزینشی مناسب، به عنوان یکی از اصلی‌ترین چالش‌های این پروژه مطرح بوده است. اکنون در ادامه به بررسی روش جدیدی می‌پردازیم که بر اساس مفهوم آنتروپی شانون طراحی شده است. این روش با دقت و جزئیات بیشتری مورد بحث قرار خواهد گرفت و نقش موثری در بهبود عملکرد دسته‌بندی گزینشی ایفا می‌کند.

### معرفی آنتروپی شانون

آنتروپی شانون، که توسط کلود شانون در سال ۱۹۴۸ معرفی شد، یکی از مفاهیم اساسی در تئوری اطلاعات است. این مفهوم معیاری برای سنجش عدم قطعیت یا تصادفی بودن اطلاعات است. اگر سیستم یا پدیده‌ای دارای چندین خروجی ممکن با احتمالات مشخص باشد، آنتروپی شانون میزان اطلاعاتی را که با مشاهده یک خروجی به دست می‌آید، مشخص می‌کند. در حقیقت این فرمول ابهام موجود در توزیع یک متغیر تصادفی را تخمین می‌زند فرمول محاسبه آن به صورت زیر است: [۱۲]

$$H(x) = - \sum_{i=1}^n p(x_i) \log p(x)_i$$

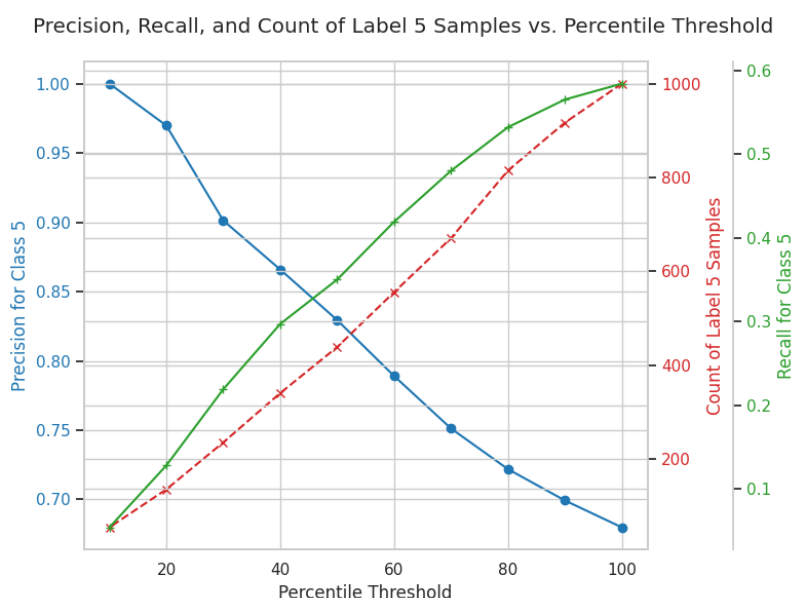
که در آن  $p(x_i)$  برابر احتمال رخ دادن  $x_i$  خواهد بود هرچه احتمالات به طور یکنواخت‌ترین حالات مختلف توزیع شده باشد، آنتروپی بیشتر است و عدم قطعیت افزایش می‌یابد پس با استفاده از این فرمول می‌توانیم برای داده‌های آزمودنی که بدست می‌آیند میزان عدم قطعیت آنها را بدست آوریم و در صورتی که نمره اطمینان آنها به حد نصاب برسد آنها را دسته بندی کنیم.



## پیاده سازی بروی دیتاست CIFAR10

اکنون که با مفاهیم نظری این فرمول آشنا شدیم، قصد داریم آن را به مرحله پیاده سازی برسانیم و بررسی کنیم که آیا می توان با استفاده از این روش یک دسته بند انتخابی مناسب طراحی کرد یا خیر. برای این منظور، پس از آموزش شبکه عصبی با استفاده از تمامی داده های آموزشی، در مرحله اعتبارسنجی، به محاسبه ی آنتروپی برای هریک از داده های اعتبارسنجی خواهیم پرداخت. سپس، لیستی از این مقادیر آنتروپی ایجاد می کنیم. در مرحله ارزیابی مدل، برای صدک های مختلف این لیست، آستانه هایی تعیین خواهیم کرد و نتایج حاصل از این فرآیند را ترسیم و تحلیل خواهیم نمود.

بعد از اعمال این موارد در حین انجام تست اکنون نتایج بدست آمده در مرحله آزمون را برای دسته خاص (دسته شماره ۵) مشاهده می کنید:



شکل ۳-۴: خط آبی: نمایش صحت، خط قرمز: تعداد نمونه های آسان و خط سبز: فراخوانی در روش آنتروپی شانون

### تفسیر و تحلیل نتایج

در روش آنتروپی شانون، هر چه درصد داده ها افزایش یابد، به این معناست که مقدار خروجی تابع `np.percentile()` بزرگ تر خواهد شد. این افزایش مقدار خروجی به معنای افزایش عدم قطعیت است که در نتیجه، شبکه را وادار می کند داده های بیشتری را دسته بندی کند. در پی این امر، صحت کاهش یافته و فراخوانی افزایش می یابد.

در هر دوروش تعیین آستانه و استفاده از آنتروپی شانون، از مکانیزمی تحت عنوان درجه اطمینان استفاده شده است. این مکانیزم به داده‌های آزمون اعمال می‌شود؛ بدین معنا که برای آنکه این داده‌ها دسته‌بندی شوند، باید یک حداقل امتیازی را کسب کنند.

### ۴-۳ روش یادگیری خود گام<sup>۱</sup>

در آموزش معمول شبکه‌های عصبی، روند به گونه‌ای است که در هر مرحله، شبکه عصبی با استفاده از تمامی داده‌های آموزشی، وزن‌های خود را تنظیم می‌کند. به این صورت که ابتدا داده‌های خود را در قالب یک مجموعه داده سازماندهی کرده و سپس آن‌ها را به بخش‌هایی تقسیم می‌کنیم. در نهایت، طی یک حلقه‌ی تکرار در هر مرحله، تمام داده‌ها به شبکه عصبی داده می‌شود و خروجی‌های آن با استفاده از یک تابع خسارت ارزیابی می‌شوند. پس از محاسبه‌ی مقدار خطا، وزن‌ها با استفاده از روش‌های بهینه‌سازی به‌روزرسانی می‌گردند. این فرایند به صورت مستمر ادامه می‌یابد تا در پایان دوره‌ی آموزش، شبکه عصبی به وزن‌های بهینه‌تری دست یابد. اما در رویکرد یادگیری خود گام که الهام گرفته از نحوه‌ی یادگیری انسان در دنیای واقعی است هدف آن است که شبکه عصبی بتواند وزن‌های دقیق‌تری را برای شبکه عصبی تنظیم کند. [۱۳]

#### معرفی روش عملکرد یادگیری خود گام

این روش با الهام از فرآیند یادگیری انسان طراحی شده است. همان‌طور که در یادگیری انسان‌ها ابتدا مفاهیم پایه و ساده آموزش داده می‌شوند و سپس به مرور مطالب پیچیده‌تر ارائه می‌گردند، در فرآیند آموزش شبکه عصبی نیز از این رویکرد بهره گرفته می‌شود. در هر مرحله بعد از محاسبه خروجی مدل یک بار تمامی داده‌ها به همراه برجسب‌های آنها به تابع خسارت داده می‌شود تفاوتی که در این روش با سایر روش‌ها وجود دارد در این هست که این با به جا اینکه خروجی تابع خسارت صرفاً یک عدد باشد یک ماتریس هست که در آن به ازای هر داده و برجسب خروجی آن میزان خسارت با توجه به تابع خسارت محاسبه شده است سپس تمامی این خسارت‌ها با یک آستانه به نام لامبدا ( $\lambda$ ) مقایسه خواهند شد و در صورتی که میزان خسارت از این آستانه کمتر باشد داده این داده به عنوان داده آسان شناخته می‌شود در حقیقت داده‌های آموزشی بر اساس فرمول زیر به دو دسته آسان و سخت دسته بندی میشوند

$$v_i = \begin{cases} 1 & \text{if } l(y_i, f(x_i, w)) < \lambda \\ 0 & \text{otherwise} \end{cases}$$

که در حقیقت  $v_i$  نشان می‌دهد که داده در محاسبه خسارت و تنظیم وزن‌ها نقشی دارند یا خیر مقدار  $y_i$  برجسب داده مورد نظر هست و  $f(x_i, w)$  برابر با خروجی مدل شبکه عصبی برای داده  $x_i$  همین‌طور  $w$  به عنوان وزن‌های شبکه عصبی ما شناخته می‌شود. مقدار خسارت نیز از طریق فرمول زیر محاسبه می‌شود:

---

<sup>1</sup> Self-paced learning (SPL)

$$\min E(w, v; \lambda) = \frac{1}{m} \sum_{i=1}^n v_i l(y_i, f(x_i, w))$$

که در آن  $m$  تعداد داده‌هایی هست که  $v_i$  آنها برابر با یک هست.

در نهایت بعد از هر مرحله از یک معیار رشد استفاده می‌کنیم این معیار رشد تعداد داده‌هایی که به عنوان داده آسان در نظر می‌گیرد را افزایش می‌دهد و شبکه عصبی در مراحل بعدی تعداد داده‌های بیشتری را مشاهده می‌کند.

## پیاده سازی روش یادگیری خود گام در دیتاست واقعی

پس از آشنایی با تئوری مباحث مرتبط با یادگیری خود گام، تصمیم داریم آن را در عمل پیاده سازی کرده و نتایج حاصل را بررسی کنیم. در گام نخست باید اشاره کرد که در این آزمایش در هر مرحله مقدار آستانه جدید در ابتدا تنظیم خواهد شد سپس با توجه به این آستانه داده های سخت و آسان جدا خواهند شد و از داده های آسان برای آموزش شبکه عصبی استفاده میشود. این آزمایش را بر روی دیتاست CIFAR10 انجام می دهیم.



شکل ۳-۵: میزان خسارت در آموزش با استفاده از یادگیری خود گام و هم چنین روش عادی در CIFAR10

در جدول زیر مقایسه میزان دقت برای هر کدام از روش های آموزش را می توانید بررسی کنید:

یادگیری خود گام	آموزش به روش معمولی	
۷۰/۸۷٪	۷۱٪	دقت

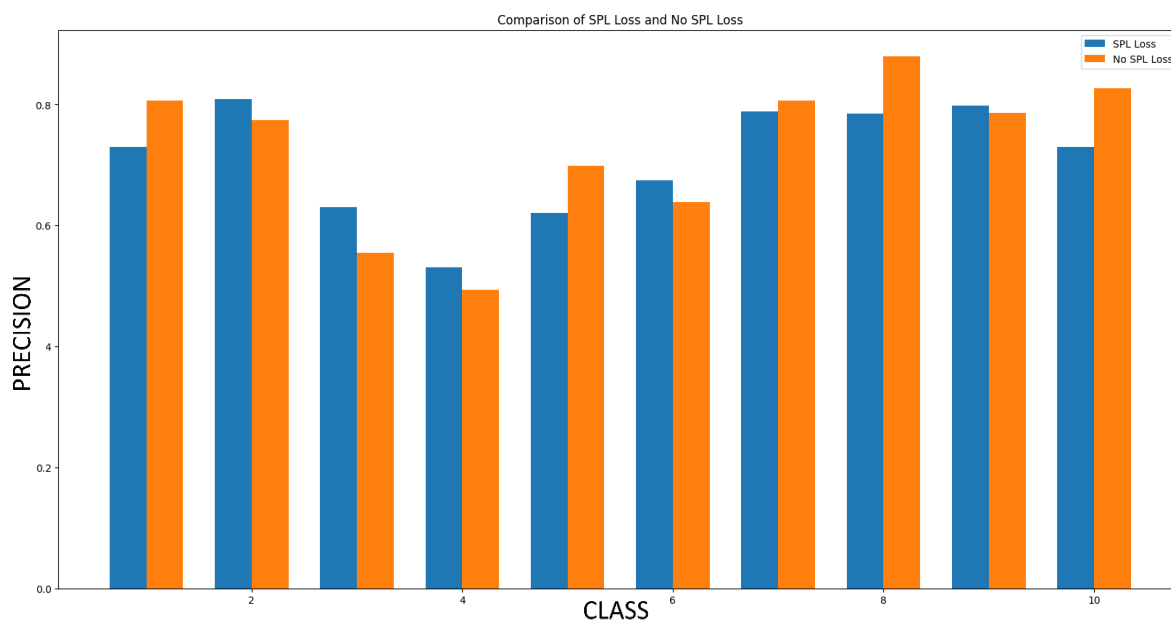
جدول ۳-۳: درصد دقت در روش های مبتنی بر یادگیری خود گام و روش معمولی

هم چنین در هر مرحله تعداد داده هایی که به عنوان داده آسان شناخته میشوند به صورت زیر بوده است:

مرحله	تعداد داده های آسان
۱	۵۴۶۹
۲	۱۰۱۵۶
۳	۱۴۸۴۴
۴	۲۰۳۱۲
۵	۲۵۰۰۰
۶	۲۹۶۸۷
۷	۳۱۱۵۶
۸	۳۹۸۴۳
۹	۴۴۵۳۱
۱۰	۴۹۸۱۲

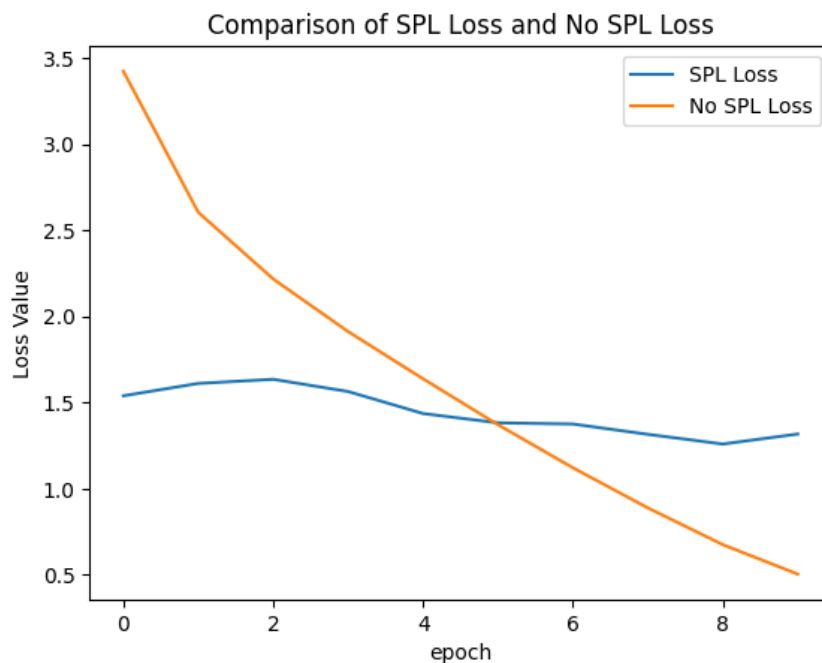
جدول ۳-۴: تعداد داده های آسان در هر مرحله از آموزش خود گام

در نمودار شکل زیر مقایسه میان نتایج بدست آمده در دوروش یادگیری خود گام و همینطور یادگیری به روش معمولی نمایش داده شده است:



شکل ۳-۶: مقایسه صحت در دسته بندی در روش های معمولی و یادگیری خود گام مجموعه داده CIFAR10

بعد از بررسی شکل بالا متوجه خواهید شد که در چهار کلاس از ده کلاس روش های مبتنی بر یادگیری خود گام بهتر از روش های معمولی عمل می کنند این خود امیدوار کننده هست و نیازمند این هست که بر روی مجموعه داده های دیگر نیز مورد آزمون قرار بگیرد. در ادامه، به بررسی دیتاست CIFAR100 می پردازیم. این دیتاست شامل ۱۰۰ کلاس مختلف است که هر کلاس شامل ۶۰۰ تصویر می باشد. از این تعداد، ۵۰۰ تصویر به منظور آموزش مدل و ۱۰۰ تصویر دیگر به عنوان داده های تست استفاده خواهند شد. تصویر زیر مربوط به میزان خسارت برای هر مرحله هست (همانطور که در بالا نحوه محاسبه خسارت گفته شد) و هم چنین در هر مرحله نشان می دهد که چه تعدادی از داده ها به عنوان داده آسان در نظر گرفته شده اند.



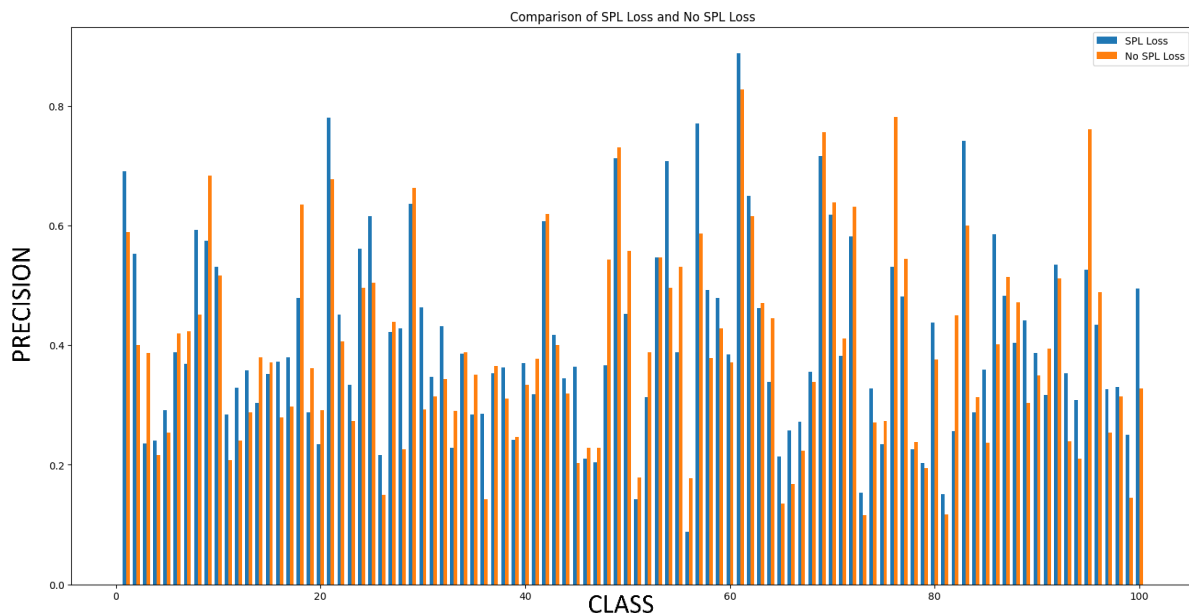
شکل ۳-۷: میزان خسارت در آموزش با استفاده از یادگیری خود گام و هم چنین روش عادی در CIFAR100

بعد از آموزش شبکه عصبی به روش یادگیری خود گام و معمولی در جدول زیر دقت را برای هر کدام از مدل های شبکه عصبی بدست آمده مشاهده می کنید:

دقت	آموزش به روش معمولی	یادگیری خود گام
	۳۸/۶٪	۳۹/۸٪

جدول ۳-۵: درصد دقت در روش های مبتنی بر یادگیری خود گام و روش معمولی مجموعه داده CIFAR100

در نمودار زیر می‌توانید مقایسه بین صحت دوروش را برای کلاس‌های مختلف بررسی کنید این مجموعه داده شامل ۱۰۰ کلاس هست که در هنگام استفاده از روش یادگیری خود گام ۵۶ کلاس صحت بزرگتر از آموزش به روش معمولی دارند که این برای ما بسیار مناسب هست و در این موضوع استفاده از روش‌های یادگیری خود گام ارجحیت پیدا می‌کند.



شکل ۳-۸: مقایسه صحت در دسته بندی در روش‌های معمولی و یادگیری خود گام مجموعه داده CIFAR100

حال به سراغ دیتاست دیگری به نام SVHN خواهیم رفت که شامل تصاویر اعداد خیابانی گرفته شده از Google Street View می‌باشد. این دیتاست به منظور تشخیص اعداد نوشته شده بر روی خانه‌ها طراحی شده و شباهت زیادی به مسئله‌ی تشخیص ارقام دست‌نویس (مثل دیتاست MNIST) دارد، با این تفاوت که پیچیدگی بیشتری دارد.

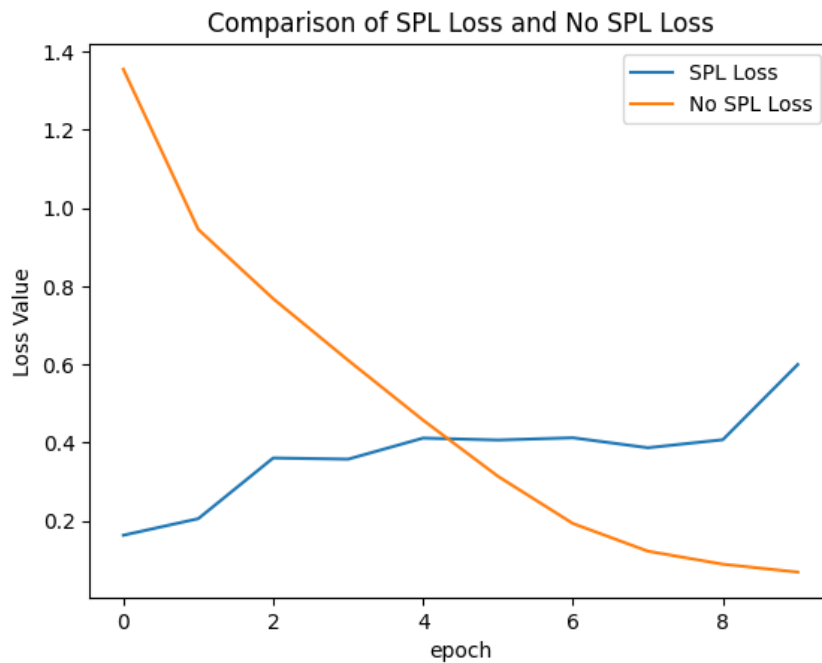
ویژگی‌های دیتاست SVHN:

۱. نوع داده‌ها: شامل تصاویر رنگی از اعداد خیابانی است.
۲. فرمت داده‌ها: تصاویر به صورت فرمت RGB با اندازه ۳۲\*۳۲ پیکسل ارائه می‌شوند.
۳. مجموعه‌های داده:

○ مجموعه آموزشی: حدود ۷۳,۲۵۷ تصویر



○ مجموعه تست: حدود ۲۶,۰۳۲ تصویر [۱۴]



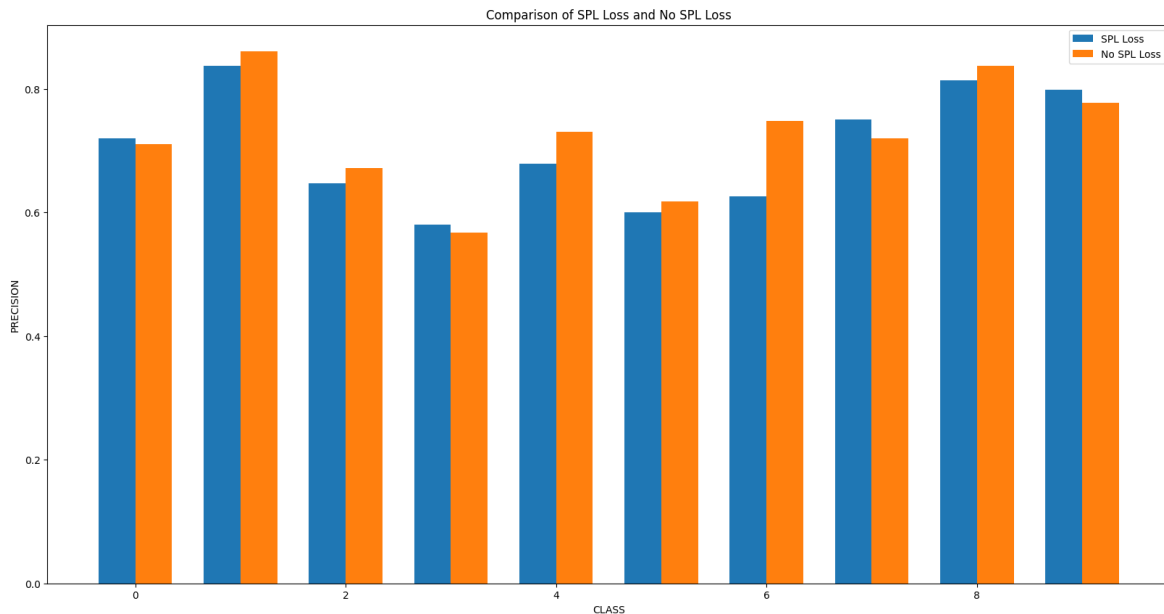
شکل ۳-۹: میزان خسارت در آموزش با استفاده از یادگیری خود گام و هم چنین روش عادی در SVHN

در جدول زیر مقایسه درصد صحت برای هر کدام از روش ها در هنگام استفاده از مجموعه داده SVHN نمایش داده شده است:

یادگیری خود گام	آموزش به روش معمولی	
۷۰/۵۵%	۷۲/۴۴%	دقت

جدول ۲-۳: درصد دقت در روش های مبتنی بر یادگیری خود گام و روش معمولی مجموعه داده SVHN

در نمودار زیر مقایسه بین دقت در کلاس های مختلف این مجموعه داده در دو روش آموزش معمولی شبکه عصبی و همینطور آموزش خود گام نشان داده شده است که در آن چهار کلاس مربوط به آموزش خود گام درصد صحت بهتری نسبت به آموزش به روش معمولی دارند.



شکل ۳-۱۰: مقایسه صحت در دسته بندی در روش های معمولی و یادگیری خود گام مجموعه داده SVHN

### تحلیل و تفسیر نتایج

همانطور که نتایج را مشاهده می کنید بهترین نتیجه مربوط به استفاده از یادگیری خود گام مربوط به مجموعه داده CIFAR100 هست ، در این مجموعه داده تعداد داده های آموزشی مربوط به هر کلاس نسبت به دو مجموعه داده دیگر بسیار کمتر هست همین موضوع باعث شده است که وقتی داده های آسانتر را ابتدا به شبکه عصبی وارد می کنیم وزن های شبکه بهتر تنظیم بشوند و در نهایت در اغلب کلاس های این مجموعه داده میزان دقت بالاتر از حالت معمولی هست هم چنین دقت بالاتری را برای شبکه عصبی آموزش دیده با یادگیری خود گام شاهد هستیم.

## فصل چهارم

### جمع‌بندی

#### ۴ جمع‌بندی

##### ۱-۴ چالش‌ها

یکی از چالش‌های اساسی که احتمالاً به ذهن شما خطور کرده است و در فصل پیشین با آن مواجه شدیم این هست که متدها و روش‌هایی مثل یادگیری خودگام در چه مجموعه داده‌هایی میتوانند مفید واقع شوند هنگامی که ما این متد را درباره مجموعه داده CIFAR10 اجرا کردیم نتوانستیم به نتیجه مطلوب برسیم این درحالی هست

که CIFAR100 توانست ما را به نتیجه ای که فکرش را می کردیم برساند در حقیقت یکی از کارهایی که می بایست ما بر روی آن تحقیق و پژوهش انجام بدهیم مرز تشخیص بین مشاهده داده های جدید یا صرف نظر از بررسی کردن آنهاست گاهی اوقات با اعمال یادگیری خود گام می توانیم حتی نتایج بدتر از زمانی بگیریم که به صورت کاملاً عادی شبکه خودمان را آموزش داده ایم.

سوالی که پیش می آید این هست که داده های ما باید چه ویژگی هایی داشته باشند که بتوانیم اذعان کنیم که استفاده از یادگیری خود گام در این موارد می تواند برای ما مفید باشد از سوی دیگر کارهای صورت گرفته در این مطالعه به صورت تجربی بوده اند و هنگامی که آستانه ای برای گرفتن نتیجه های مورد نظر بیان شده از طریق تست کردن و تجربه به دست آمده است چالشی که می توانیم بیان کنیم که با آن مواجه هستیم در عمل و کاربرد های عملی چگونه می توانیم در سریع ترین زمان ممکن به این آستانه ها و دقت مورد نظر در شبکه عصبی برسیم.

## ۲-۴ کارهای آینده

یکی از کارهایی که میتوان در این حوزه انجام داد ادغام روش های ذکر شده هست از آنجایی که روش های یادگیری خود گام در حین آموزش پیاده سازی می شوند و روش های طراحی دسته بند انتخابی در حین انجام تست می توان ترکیبی از دوروش را پیاده سازی کرد و نتایج را بررسی نمود این مورد یکی از کارهای آینده خواهد بود از سمت دیگری از کارهایی که میتوان انجام داد در بحث یادگیری خود گام قرار دادن ظرفیت خاص برای کلاس های مختلف هست به طوری که اعضای هر کلاسی بتواند فقط یک تعداد محدودی را به عنوان داده آسان به شبکه عصبی معرفی کند در صورتی که این مشکل مرتفع نشود ممکن هست در حین آموزش شبکه عصبی منجر به مشکل مجموعه داده نامتوازن در حین آموزش شود و این موضوع میتواند باعث سوگیری وزن های شبکه شود از سمتی دیگر می توانیم روش های دیگری را برای پیاده سازی یادگیری خود گام استفاده کنیم تا داده های سخت تر هم بتوانند در حین آموزش استفاده شوند اما به نحوی که وزن بسیار کمتری نسبت به داده های آسان داشته باشند که البته همه اینها باید پیاده سازی شده و آزمایش شوند [۱۵].

## ۳-۴ نتیجه‌گیری

در این پروژه در ابتدا ما با یکسری از فرضیات و مفاهیم تئوری شروع به نظریه پردازی و تحقیق در مورد نحوه بهبود صحت و اطمینان در شبکه های دسته بند پرداختیم و آنها را برای دیتاست های مختلف به مورد آزمایش قرار دادیم چیزی که مشخص هست در دسته بندها ما بطور کامل توانستیم به آن مقصودی که مد نظرمان بود برسیم و با قاطعیت بالا نمونه های آسان را دسته بندی کنیم خود این موضوع میتواند در عمل به کار بیاید و باعث شود در شرایط بحرانی با وجود اینکه از پیش بینی کردن یکسری از مسائل دسته بندی خودداری می کنیم اما وقتی که پیش بینی برای یک نمونه انجام می دهیم این خروجی با قاطعیت زیادی درست باشد و ما با اطمینان بالا به آن نتیجه تکیه کنیم.

در مورد مسائل مربوط به یادگیری خود گام کمی تامل بیشتری لازم بود همانطور که از نتایج بالا مشخص هست برای یک شبکه عصبی کاملاً یکسان برای مجموعه داده های مربوط به CIFAR100 , CIFAR10 و SVHN مقدار عددی پارامتر دقت تفاوت قابل ملاحظه ای داشت هم چنین برای یک تابع خسارت مشابه یعنی تابع خسارت یادگیری خود گام سه مجموعه داده مورد آزمون نتایج کاملاً متفاوتی را به ما نشان دادند در دیتاست مربوط به SVHN, CIFAR10 وقتی که همه داده ها در تمامی مرحله ها مشاهده میشوند باعث میشود شبکه عصبی وزن های بهتری بدست بیاورد و در هنگام تست پیش بینی های دقیق تری انجام بدهد اما در CIFAR100 این اتفاق نمی افتد، توجیه این موضوع به علت تفاوت در تعداد داده های آموزشی برای مجموعه داده CIFAR100 هست که اختلاف زیادی با بخش آموزشی دو مجموعه داده دیگر دارد همین موضوع باعث می شود که اضافه کردن داده های آسان به شبکه عصبی ، وزن ها را بهتر تنظیم بکند.

## ۵ پیوست

جهت مشاهده کدهای پیاده سازی شده به همراه بررسی نتایج آنها میتواند از این [لینک](#) اقدام کنید.

## ۶ منابع

- [1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, 2015.
- [2] "Biological Neuron versus McCulloch and Pitts Artificial Neuron Model." [Online]. Available: [https://www.researchgate.net/figure/Biological-neuron-versus-McCulloch-and-Pitts-artificial-neuron-model\\_fig2\\_359233566](https://www.researchgate.net/figure/Biological-neuron-versus-McCulloch-and-Pitts-artificial-neuron-model_fig2_359233566). Accessed: Sep. 2024.
- [3] "A Neuron and a MLP." [Online]. Available: [https://www.researchgate.net/figure/a-neuron-and-MLP\\_fig3\\_344954689](https://www.researchgate.net/figure/a-neuron-and-MLP_fig3_344954689). Accessed: Sep. 2024.
- [4] M. A. Nielsen, *Neural Networks and Deep Learning*. Determination Press, 2015.
- [5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [6] "Shallow vs Deep Learning Models," MRI Questions. [Online]. Available: <https://mriquestions.com/shallow-v-deep-ml.html>. Accessed: Sep. 2024.
- [7] V. H. Phung and E. J. Rhee, "A high-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets," *Applied Sciences*, vol. 9, no. 21, p. 4500, 2019.
- [8] "Confusion Matrix, Accuracy, Precision, Recall, F1 Score," Analytics Vidhya. [Online]. Available: <https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63cd>. Accessed: Sep. 2024.

- [9] Y. Geifman and R. El-Yaniv, "Selective classification for deep neural networks," 2017.
- [10] "CIFAR Dataset." [Online]. Available: <https://www.cs.toronto.edu/~kriz/cifar.html>. Accessed: Sep. 15, 2024.
- [11] "Numpy Percentile Documentation." [Online]. Available: <https://numpy.org/doc/stable/reference/generated/numpy.percentile.html>. Accessed: Sep. 15, 2024.
- [12] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, 1948.
- [13] W. Hou, N. Chen, J. Peng, W. Sun, and Q. Du, "Pyramidal dilation attention convolutional network with active and self-paced learning for hyperspectral image classification," 2023.
- [14] "Street View House Numbers Dataset." Kaggle. [Online]. Available: <https://www.kaggle.com/datasets/stanfordu/street-view-house-numbers>. Accessed: Sep. 2024.
- [15] L. Li, K. Zhao, S. Li, R. Sun, and S. Cai, "Extreme learning machine for supervised classification with self-paced learning," *Neural Processing Letters*, 2020.