به نام خداوند بخشنده مهربان

عنوان تمرین

رژیم پارامتری

عنوان درس

یادگیری ماشین

استاد

دکتر الهام قصرالدشتی

دستیاران آموزشی

مهرداد قصابی

مریم صفوی

گردآورنده

سید حسین حسینی

تابستان ۱۴۰۴

دانشکده مهندسی کامپیوتر

دانشگاه اصفهان

فهرست مطالب

۱ .چکیده مدیریتی

این مستند، فرآیند طراحی، پیاده‌سازی و ارزیابی یک شبکه عصبی کانولوشنی (CNN) بسیار سبک با نام **LightCNN**را برای طبقه‌بندی ارقام دست‌نویس MNIST تشریح می‌کند. چالش اصلی پروژه، دستیابی به حداکثر دقت طبقه‌بندی با استفاده از حداقل تعداد پارامترهای قابل آموزش بود. معماری پیشنهادی با بهره‌گیری هوشمندانه از لایه‌های کانولوشنی و جایگزینی لایه‌های تماماً متصل سنگین با یک لایه **Global Average Pooling (GAP)**، به این هدف دست یافت. نتایج نهایی نشان می‌دهد که مدل LightCNN با تنها **۶,۲۱۸ پارامتر** به دقت **٪۹۸.۶۱** بر روی مجموعه داده آزمون رسیده است. این عملکرد، در مقایسه با معماری‌های کلاسیک که بیش از ۸۰,۰۰۰ پارامتر برای رسیدن به دقت مشابه نیاز دارند، یک بهبود چشمگیر در بهره‌وری محاسباتی محسوب می‌شود. این مستند به تحلیل دقیق این معماری، مقایسه آن با مدل‌های دیگر و بررسی نقاط قوت و ضعف آن می‌پردازد.

۲ .مقدمه

۲.۱ .شرح مسئله

مجموعه داده MNIST یک استاندارد کلاسیک در زمینه بینایی ماشین برای ارزیابی الگوریتم‌های طبقه‌بندی است. هدف، طبقه‌بندی صحیح تصاویر ۲۸x۲۸ پیکسلی از ارقام دست‌نویس (۰ تا ۹) است. با وجود سادگی نسبی این مجموعه داده، طراحی مدلی که همزمان دقیق و بهینه از نظر محاسباتی باشد، یک چالش مهندسی ارزشمند است.

۲.۲ .اهداف پروژه

1. **طراحی یک معماری CNN سفارشی** بدون استفاده از مدل‌های از پیش آموزش‌دیده.
2. **به حداقل رساندن تعداد پارامترهای قابل آموزش** مدل تا حد امکان.
3. **دستیابی به دقت طبقه‌بندی بالا** (حدود ٪۹۸ یا بیشتر) بر روی داده‌های آزمون.
4. **ارائه تحلیل مقایسه‌ای** بین مدل پیشنهادی و سایر معماری‌های استاندارد.

۲.۳ .اهمیت و کاربردها

مدل‌های سبک و کارآمد در دنیای امروز اهمیت فزاینده‌ای دارند. کاربردهای اصلی آن‌ها عبارتند از:

- **پردازش لبه(Edge Computing) :** اجرا بر روی دستگاه‌هایی با منابع محدود مانند تلفن‌های هوشمند، دوربین‌های مداربسته هوشمند و دستگاه‌های اینترنت اشیاء.(IoT)
- **کاهش هزینه‌های سرور :**مدل‌های کوچک‌تر به توان پردازشی و حافظه کمتری برای استنتاج (Inference)نیاز دارند.
- **افزایش سرعت پاسخ‌دهی :**مدل‌های سبک‌تر زمان تاخیر (Latency) کمتری دارند.

۳ .مجموعه داده و پیش‌پردازش

۳.۱ .معرفی مجموعه دادهMNIST

- **تعداد تصاویر آموزشی :**۶۰,۰۰۰
- **تعداد تصاویر آزمون :**۱۰,۰۰۰
- **ابعاد تصاویر :**۲۸x۲۸ پیکسل، تک کاناله (سیاه و سفید)
- **تعداد کلاس‌ها :**۱۰ (ارقام ۰ تا ۹)

۳.۲ .فرآیند آماده‌سازی داده‌ها

پیش‌پردازش داده‌ها به سادگی و تنها با یک مرحله انجام شد:

- transforms.ToTensor(): این تبدیل، تصاویر PIL یا آرایه‌های NumPy را به تنسورهای PyTorch تبدیل کرده و مقادیر پیکسل‌ها را از بازه [0, 255]به بازه [0.0, 1.0]نرمال‌سازی می‌کند.

برای بارگذاری داده‌ها از torch.utils.data.DataLoaderاستفاده شد که امکان بارگذاری دسته‌ای(mini-batch)و بُر زدن داده‌ها را فراهم می‌کند.

۴ .معماری پیشنهادیLightCNN :

۴.۱ .فلسفه طراحی

فلسفه اصلی در طراحیLightCNN ، **استخراج سلسله‌مراتبی ویژگی‌ها با لایه‌های کانولوشنی و سپس تجمیع هوشمندانه آن‌ها بدون نیاز به لایه‌های تماماً متصل (Fully Connected) بزرگ** بود. این رویکرد از هدررفت پارامترها جلوگیری کرده و مدل را به یادگیری ویژگی‌های کلی‌تر و مقاوم‌تر تشویق می‌کند.

۴.۲ .تشریح دقیق لایه‌ها

ورودی[Batch_Size, 1, 28, 28] :

مرحله	لایه	مشخصات	شکل خروجی
۱	conv1 + ReLU	۱ ورودی، ۸ خروجی، کرنل 3*3، پدینگ ۱	[Batch_Size, 8, 28, 28]
۲	pool	Max Poolingبا کرنل 2*2	[Batch_Size, 8, 14, 14]
۳	conv2 + ReLU	۸ ورودی، ۱۶ خروجی، کرنل 3*3، پدینگ ۱	[Batch_Size, 16, 14, 14]
۴	pool	Max Poolingبا کرنل 2*2	[Batch_Size, 16, 7, 7]
۵	conv3 + ReLU	۱۶ ورودی، ۳۲ خروجی، کرنل 3*3، پدینگ ۱	[Batch_Size, 32, 7, 7]
۶	global_pool	Adaptive Average Poolingبا خروجی 1*1	[Batch_Size, 32, 1, 1]
۷	view (Flatten)	تغییر شکل تنسور	[Batch_Size, 32]
۸	fc	لایه خطی با ۳۲ ورودی و ۱۰ خروجی	[Batch_Size, 10]

۴.۳ .تحلیل نقش کلیدی لایه Global Average Pooling (GAP)

این لایه مهم‌ترین عنصر در بهینه‌سازی معماری است. در روش‌های سنتی (مانندLeNet-5)، خروجی آخرین لایه کانولوشنی (32x7x7) مسطح (Flatten) می‌شود که نتیجه آن یک بردار با 1568 = 7 * 7 * 32عنصر است. این بردار بزرگ سپس به یک یا چند لایه FC حجیم متصل می‌شود که صدها هزار پارامتر تولید می‌کنند.

GAPاین فرآیند را متحول می‌کند:

- به جای مسطح‌سازی، میانگین مقادیر هر یک از ۳۲ نقشه ویژگی (7x7) را محاسبه می‌کند.
- خروجی آن یک بردار با تنها ۳۲ عنصر است که هر عنصر نماینده چکیده اطلاعات یک نقشه ویژگی است.
- **مزایا:**

1. **کاهش شدید پارامترها :**لایه FC نهایی تنها 330 = 10 + 10 * 32پارامتر دارد.
2. **جلوگیری ازOverfitting :** با حذف لایه‌های FC بزرگ، مدل کمتر مستعد حفظ کردن نویز داده‌های آموزشی است.
3. **حفظ ارتباط بین ویژگی و کلاس :**هر نورون خروجی در لایه GAP مستقیماً با یک نقشه ویژگی مرتبط است و مدل را به یادگیری ویژگی‌های معنادارتر تشویق می‌کند.

۵ .فرآیند آموزش و بهینه‌سازی

۵.۱ .تابع هزینه(Loss Function)

از nn.CrossEntropyLossاستفاده شد. این تابع برای مسائل طبقه‌بندی چندکلاسه ایده‌آل است، زیرا محاسبات LogSoftmaxو NLLLossرا در یک مرحله به صورت بهینه انجام می‌دهد.

۵.۲ .الگوریتم بهینه‌سازی(Optimizer)

الگوریتم optim.Adamانتخاب شد Adam .یک بهینه‌ساز تطبیقی است که نرخ یادگیری را برای هر پارامتر به صورت جداگانه تنظیم می‌کند و به دلیل همگرایی سریع و عملکرد قوی، یک انتخاب استاندارد در یادگیری عمیق است.

۵.۳ .ابرپارامترهای آموزش

- **نرخ یادگیری(Learning Rate)** :0.001
- **اندازه دسته(Batch Size)** :64
- **تعداد دوره‌ها(Epochs)** :50

۶ .نتایج و ارزیابی عملکرد

۶.۱ .معیارهای ارزیابی

- **دقت(Accuracy)** : درصد نمونه‌هایی که به درستی طبقه‌بندی شده‌اند.
- **تعداد پارامترهای قابل آموزش** :کل پارامترهایی که در طول فرآیند آموزش به‌روزرسانی می‌شوند.

۶.۲ .نتایج نهایی

- **دقت نهایی روی داده‌های آزمون :٪۹۸.۶۱**
- **تعداد کل پارامترهای قابل آموزش :۶,۲۱۸**

این نتایج نشان‌دهنده موفقیت کامل پروژه در دستیابی به اهداف تعیین‌شده است.

۷ .تحلیل مقایسه‌ای با معماری‌های مبنا(Baseline)

برای ارزیابی عمق عملکردLightCNN ، نتایج آن با چندین معماری مرجع مقایسه می‌شود.

۷.۱ .معرفی معماری‌های مبنا

ما دو دسته معماری را به عنوان مبنا در نظر می‌گیریم: مدل‌های سنگین که بر دقت تمرکز دارند و مدل‌های بسیار سبک که بر حداقل‌سازی پارامترها متمرکز هستند.

۷.۲ .معماری‌های سنگین(Heavyweight)

1. **شبکه تماماً متصل (MLP) کلاسیک:**
   - **توصیف معماری :**یک شبکه چندلایه پرسپترون بدون لایه‌های کانولوشنی. یک ساختار رایج 10 <- 256 <- 512 <- 784است. این مدل ساختار فضایی تصویر را نادیده می‌گیرد.
   - **تعداد پارامترها:** ≈ (784*512+512) + (512*256+256) + (256*10+10) 550,000(بسیار سنگین).
   - **دقت :**حدود ۹۸٪.
2. **CNNکلاسیک (مشابه LeNet-5 با لایه‌های FC بزرگ):**
   - **توصیف معماری :**دو یا سه لایه کانولوشنی و سپس دو لایه FC بزرگ (مثلاً با ۵۱۲ و ۱۲۸ نورون).
   - **تعداد پارامترها :**حدود ۸۰,۰۰۰ تا ۱۲۰,۰۰۰.
   - **دقت :**حدود ۹۹٪.

۷.۳ .معماری‌های بسیار سبک(Ultra-Lightweight)

1. **مدل:Ultra-Light**
   - **توصیف معماری :**یک لایه کانولوشن با ۴ فیلتر، و یک لایه FC کوچک.
   - **تعداد پارامترها :**حدود ۲۱۰.

- **دقت :**حدود ٪۷۴ .این مدل به شدت دچار کم‌برازش (Underfitting) می‌شود.

2. **مدل Medium-Light :**

- **توصیف معماری :**دو لایه کانولوشن با تعداد فیلترهای کم و یک لایه FC کوچک.
- **تعداد پارامترها :**حدود **۳,۰۰۰**.
- **دقت :**حدود ٪**۹۲**.

۷.۴ .جدول مقایسه‌ای و تحلیل نتایج

تحلیل مقایسه‌ای	دقت در آزمون	تعداد پارامترها	مدل
دقت بالا با هزینه محاسباتی بسیار زیاد. بخش عمده پارامترها در لایه‌های FC هدر می‌رود.	~٪۹۹	~۸۰,۰۰۰	**CNN کلاسیک سنگین**
تعادل بهینه :دقتی بسیار نزدیک به مدل‌های سنگین، با حدود **۱۳ برابر پارامتر کمتر**.	**٪۹۸.۶۱**	**۶,۲۱۸**	**مدل پیشنهادی (LightCNN)**
با نصف کردن پارامترها، دقت به شدت افت کرده و نشان‌دهنده طراحی غیربهینه است.	~٪۹۲	~۳,۰۰۰	**مدل Medium-Light**
ظرفیت یادگیری بسیار محدود و برای کاربردهای عملی نامناسب است.	~٪۷۴	~۲۱۰	**مدل Ultra-Light**

تحلیل:

مدل **LightCNN**در یک "نقطه شیرین (Sweet Spot) "قرار دارد. این مدل نشان می‌دهد که افزایش بی‌رویه پارامترها لزوماً منجر به بهبود چشمگیر دقت نمی‌شود. با یک طراحی هوشمندانه، می‌توان با **کمتر از ۱۰٪ پارامترهای یک CNN کلاسیک**، به بیش از **۹۹.۵٪ از دقت آن** دست یافت. این یک معامله فوق‌العاده از نظر کارایی (Efficiency) است.

۸ .بحث و بررسی

۸.۱ .نقاط قوت و نوآوری‌ها

- **بهره‌وری پارامتر :**مزیت اصلی این مدل، نسبت بالای دقت به تعداد پارامترها است.
- **استفاده موثر از GAP:** این پروژه به خوبی قدرت Global Average Pooling را در ساخت مدل‌های سبک و کارآمد نشان می‌دهد.
- **سادگی و شفافیت :**معماری مدل ساده، قابل فهم و پیاده‌سازی آن آسان است.

۸.۲ .محدودیت‌ها و زمینه‌های بهبود

- **دقت حداکثری :**برای رسیدن به دقت‌های بالاتر از ۹۹.۵٪(State-of-the-Art) ، احتمالاً به تکنیک‌های پیشرفته‌تری مانندDropout ، Batch Normalizationیا معماری‌های پیچیده‌تر نیاز خواهد بود.
- **وابستگی به مجموعه داده :**این معماری برای MNIST بهینه شده است و ممکن است عملکرد مشابهی روی داده‌های پیچیده‌تر مانند CIFAR-10 نداشته باشد.

۹ .نتیجه‌گیری

پروژه حاضر با موفقیت یک شبکه عصبی کانولوشنی (LightCNN) را طراحی و پیاده‌سازی کرد که توانست با تعداد پارامترهای بسیار کم (۶,۲۱۸) به دقت بسیار بالا (٪۹۸.۶۱) در طبقه‌بندی مجموعه داده MNIST دست یابد. تحلیل مقایسه‌ای نشان داد که این مدل از نظر بهره‌وری محاسباتی به مراتب برتر از معماری‌های کلاسیک سنگین است و ثابت می‌کند که **طراحی هوشمندانه معماری مهم‌تر از افزایش بی‌رویه پیچیدگی و تعداد پارامترهاست** .این دستاورد، راه را برای طراحی مدل‌های یادگیری عمیق قابل اجرا بر روی دستگاه‌های با منابع محدود هموار می‌سازد.

۱۰ .مراجع و منابع(References and Sources)

این بخش به معرفی مقالات علمی، مجموعه داده‌ها و ابزارهای نرم‌افزاری می‌پردازد که بنیان‌های نظری و عملی این پروژه را تشکیل داده‌اند.

۱۰.۱ .مقالات بنیادی و الهام‌بخش

1. **Network In Network**
   - **Citation:** Lin, M., Chen, Q., & Yan, S. (2013). Network In Network. *arXiv preprint arXiv:1312.4400*.
   - **ارتباط با پروژه :**این مقاله برای اولین بار مفهوم **Global Average Pooling (GAP)**را به عنوان جایگزینی برای لایه‌های تماماً متصل (Fully Connected) در شبکه‌های کانولوشنی معرفی کرد. ایده کلیدی معماری **LightCNN**در این پروژه، یعنی حذف لایه‌های سنگین و کاهش چشمگیر پارامترها، مستقیماً از این مقاله الهام گرفته شده است.

2. **Gradient-Based Learning Applied to Document Recognition**
   - **Citation:** LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278-2324.

- **ارتباط با پروژه :**این مقاله، معماری **LeNet-5**را معرفی کرد که به عنوان یکی از اولین و موفق‌ترین شبکه‌های عصبی کانولوشنی شناخته می‌شود. ساختار سلسله‌مراتبی لایه‌های کانولوشن و ادغام (Pooling) در مدل **LightCNN**، از اصول پایه‌ای که در این مقاله معرفی شد، پیروی می‌کند LeNet-5 .یک معیار مقایسه (Baseline) کلاسیک برای عملکرد روی MNIST است.

3. **Adam: A Method for Stochastic Optimization**

- **Citation:** Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- **ارتباط با پروژه :**الگوریتم بهینه‌سازی Adamکه در این پروژه برای آموزش مدل استفاده شده، در این مقاله معرفی شده است. انتخاب این بهینه‌ساز به دلیل همگرایی سریع و عملکرد پایدار آن در طیف وسیعی از مسائل یادگیری عمیق بوده است.

4. **SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size**

- **Citation:** Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. *arXiv preprint arXiv:1602.07360*.
- **ارتباط با پروژه :**هرچند معماری SqueezeNet در این پروژه پیاده‌سازی نشده، اما فلسفه آن (دستیابی به دقت بالا با پارامترهای بسیار کم) کاملاً با اهداف این پروژه همسو است. این مقاله به عنوان یک مرجع الهام‌بخش برای طراحی شبکه‌های سبک و بهینه در نظر گرفته می‌شود.

۱۰.۲ .مجموعه داده(Dataset)

1. **The MNIST Database of Handwritten Digits**

- **Citation:** LeCun, Y., Cortes, C., & Burges, C. J. C. (1998). The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/.

- **ارتباط با پروژه :**این مرجع، منبع اصلی مجموعه داده‌ای است که تمام فرآیندهای آموزش و ارزیابی مدل **LightCNN**بر روی آن انجام شده است.

۱۰.۳ .ابزارها و کتابخانه‌ها(Tools and Libraries)

1. **PyTorch**

- **Citation:** Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems, 32*.
- **ارتباط با پروژه :**تمام مراحل پیاده‌سازی، آموزش و ارزیابی مدل در این پروژه با استفاده از کتابخانه یادگیری عمیق PyTorch انجام شده است. قابلیت‌های دینامیک و واسط کاربری ساده آن، فرآیند توسعه را تسهیل کرده است.

2. **Torchvision**

- **Citation:** Marcel, S., & Rodriguez, Y. (2010). Torchvision: The machine vision package of torch. *Proceedings of the 18th ACM international conference on Multimedia*.
- **ارتباط با پروژه :**از این کتابخانه برای دسترسی آسان به مجموعه داده MNIST و همچنین اعمال تبدیل‌های پیش‌پردازشی (transforms) بر روی تصاویر استفاده شده است.

موفق باشید 😊😁