



دانشگاه صنعتی اصفهان  
دانشکده مهندسی برق و کامپیوتر

عنوان: تکلیف سوم درس یادگیری عمیق

نام و نام خانوادگی: علیرضا ابره فروش

شماره دانشجویی: ۹۸۱۶۶۰۳

نیم سال تحصیلی: پاییز ۱۴۰۲

مدرس: دکتر سمانه حسینی سمنانی

دستیاران آموزشی: مریم محمدی-علی بزرگ زادارباب

استفاده از تصاویر در شبکه‌های عصبی معمولی (MLP) دارای مشکلاتی بود که با ظهور شبکه‌های عصبی کانولوشنی (CNN) حل شد:

#### ۱. تعداد پارامترها و اتصالات:

در MLP، هر نورون لایه ورودی با تمام نورون‌های لایه خروجی متصل بود. این اتصالات کامل منجر به افزایش سریع تعداد پارامترها می‌شد، که با تعداد بزرگ تصاویر (از نظر تعداد پیکسل) به سرعت غیرقابل مدیریت می‌شد. در CNN، از لایه‌های کانولوشنی برای به اشتراک‌گذاری ویژگی‌ها در تصاویر استفاده می‌شود. این لایه‌ها با اعمال فیلترها (کرنل‌ها) به تصویر، ویژگی‌های مختلف را استخراج می‌کنند و این باعث کاهش تعداد پارامترها و اتصالات شبکه می‌شود. به عبارت دیگر، CNN با استفاده از به اشتراک‌گذاری ویژگی‌ها، تعداد پارامترها را به شدت کاهش می‌دهد.

#### ۲. مقاومت به تغییرات مکانی:

MLP به طور کامل حساس به تغییرات مکانی در تصویر بود. به عبارت دیگر، اگر یک الگو یا ویژگی در یک مکان خاص در تصویر وجود داشت، MLP قادر به تشخیص آن در سایر نقاط تصویر نبود. CNN با استفاده از لایه‌های کانولوشنی و استفاده از فیلترها، از این قابلیت برای شناسایی ویژگی‌ها در تمام تصویر به خوبی استفاده می‌کند. به این ترتیب، شبکه CNN مقاوم‌تر به تغییرات مکانی در تصاویر می‌شود و توانایی خود را در تشخیص الگوها و ویژگی‌ها در مکان‌های مختلف تصویر افزایش می‌دهد.

۳. عدم حفظ ساختار مکانی: MLP، ساختار مکانی تصویر حذف می‌شد و هر پیکسل به عنوان یک ویژگی مستقل در نظر گرفته می‌شد. این عدم حفظ ساختار مکانی اطلاعات مهمی را از دست می‌داد، به خصوص برای تصاویری که الگوها و اطلاعات مکانی مهمی دارند. CNN با استفاده از لایه‌های کانولوشنی و اعمال فیلترها، قابلیت حفظ ساختار مکانی را به شبکه اضافه کرده و اجازه می‌دهد تا ویژگی‌ها به صورت محلی در تصویر استخراج شوند. این امکان باعث می‌شود که شبکه بتواند اطلاعات مکانی را مورد توجه قرار داده و الگوهای مکانی پیچیده‌تری را در تصاویر تشخیص دهد.

#### ۴. تفسیرپذیری پایین:

MLPها به دلیل تعداد بسیار زیاد پارامترها، به صورت کلی دارای تفسیرپذیری پایین بودند. یعنی مشخص کردن دقیق اینکه شبکه چگونه تصمیمات خود را اتخاذ کرده است، معمولاً مشکل بود. CNN با تعداد کمتر پارامترها و استفاده از فیلترها، قابلیت تفسیرپذیری بیشتری دارد. این به این معناست که می‌توان بهتر فهمید که شبکه در تصمیم‌گیری‌های خود چگونه از ویژگی‌ها استفاده می‌کند.

#### ۵. حساسیت به اندازه تصویر:

MLPها به صورت ثابت و بدون توجه به ابعاد تصویر (مثلاً ابعاد ورودی ثابت دارای تعداد ثابت نورون‌ها) عمل می‌کردند. این باعث می‌شد که اگر تصویر ورودی ابعاد متفاوتی داشته باشد، شبکه به طور مستقیم با آن کار نکند. CNN با استفاده از لایه‌های کانولوشنی می‌تواند به تصاویر با ابعاد مختلف و با استفاده از فیلترهای متفاوت به خوبی پاسخ دهد و حساسیت کمتری نسبت به ابعاد تصویر نشان دهد.

#### ۶. برخورد با تعداد زیاد پارامترها:

با افزایش اندازه تصاویر، تعداد پارامترهای مورد نیاز برای یک MLP به صورت نمایی افزایش می‌یابد. این موضوع باعث ایجاد مدل‌های بسیار پیچیده و سنگین می‌شود که دشواری در آموزش، نگهداری و استفاده از آن‌ها را افزایش می‌دهد.

۷. عدم توانایی در مدل سازی ویژگی های سلسله مراتبی:

MLP به طور مستقیم قادر به مدل سازی ویژگی های سلسله مراتبی و پیچیده تصاویر نیستند. به عبارت دیگر، آن ها قادر به استخراج ویژگی های موقعیت مکانی، الگوها و ساختارهای سلسله مراتبی نیستند که در بینایی ماشین بسیار مهم است.

۸. حساسیت به تغییرات شدت نور و ظروف نوری:

MLP در مقابل تغییرات شدت نور و ظروف نوری حساس هستند. این به این معناست که تصاویر با نور متفاوت یا ظروف نوری متفاوت ممکن است تأثیر زیادی بر عملکرد MLP داشته باشند.

۹. اشتباه زهای غیرقابل کنترل:

به دلیل تعداد بالای پارامترها و عدم استفاده از الگوهای مکانی، MLP ها ممکن است به آموزش بیش از حد به داده ها وابسته شوند و اشتباه زهای غیرقابل کنترل در عملکرد آن ها ایجاد شود.

۱۰. کارایی ضعیف در مسائل تشخیص الگو:

در مسائل تشخیص الگو و ویژگی های پیچیده در تصاویر، MLP ها عملکرد ضعیفی دارند. زیرا این مدل ها نمی توانند ویژگی های سلسله مراتبی و پیچیده را به صورت کامل مدل کنند.

شبکه های عصبی کانولوشنی (CNN) با توجه به مزایایی که در پردازش تصاویر دارند، این مشکلات را به حداقل می رسانند و بهبودهای مهمی را در زمینه بینایی ماشین و پردازش تصویر به ارمغان آورده اند.

۲

۳

۴

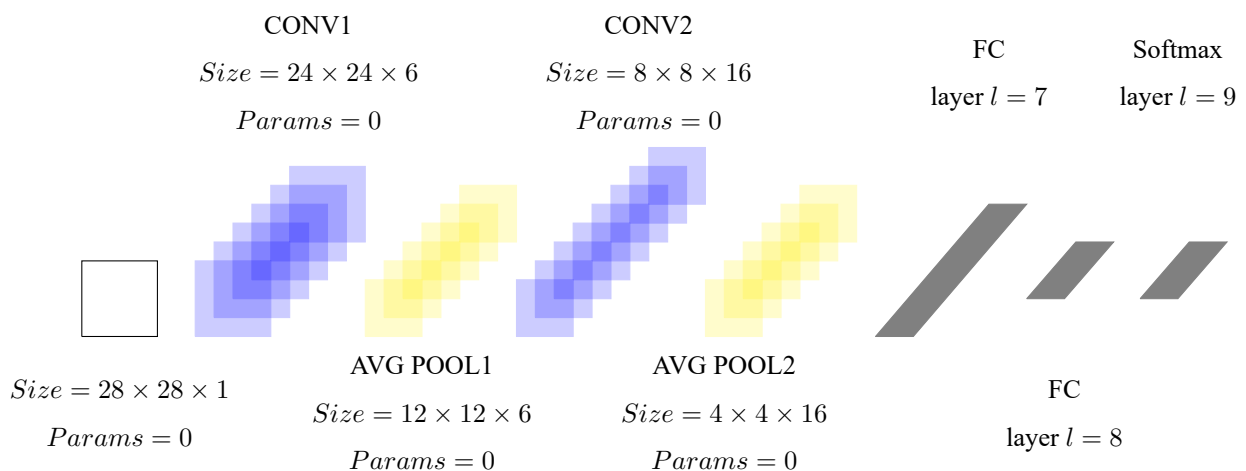


Figure 1

$$\text{size after applying convolution layer} = \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor \times k$$

### Input Layer:

- Size:  $28 \times 28 \times 1$
- Params: 0
- Number of Channels: 1

### CONV1 Layer:

- Filter size:  $5 \times 5$
- Stride: 1
- Number of filters ( $k$ ): 6
- Padding: 0
- Size:  $\left\lfloor \frac{28+2 \times 0-5}{1} + 1 \right\rfloor \times \left\lfloor \frac{28+2 \times 0-5}{1} + 1 \right\rfloor \times 6 = 24 \times 24 \times 6$
- Params:  $(5 \times 5 \times 1 + 1) \times 6 = 156$  (weights + bias for each filter)

### Average Pooling Layer 1:

- Filter size:  $2 \times 2$
- Stride: 2
- Size:  $\left\lfloor \frac{24+2 \times 0-2}{2} + 1 \right\rfloor \times \left\lfloor \frac{24+2 \times 0-2}{2} + 1 \right\rfloor \times 6 = 12 \times 12 \times 6$
- Params: 0

### CONV2 Layer:

- Filter size:  $5 \times 5$
- Stride: 1
- Number of filters ( $k$ ): 16
- Padding: 0
- Size:  $\left\lfloor \frac{12+2 \times 0-5}{1} + 1 \right\rfloor \times \left\lfloor \frac{12+2 \times 0-5}{1} + 1 \right\rfloor \times 16 = 8 \times 8 \times 16$
- Params:  $(5 \times 5 \times 6 + 1) \times 16 = 2416$  (weights + bias for each filter)

**Average Pooling Layer 2:**

- Filter size:  $2 \times 2$
- Stride: 2
- Size:  $\left\lfloor \frac{8+2 \times 0-2}{2} + 1 \right\rfloor \times \left\lfloor \frac{8+2 \times 0-2}{2} + 1 \right\rfloor \times 16 = 4 \times 4 \times 16$
- Params: 0

**FC1 Layer (Fully Connected):**

- Number of neurons: 128
- Params:  $(4 \times 4 \times 16 + 1) \times 128 = 32896$  (weights + bias for each neuron)

**FC2 Layer (Fully Connected):**

- Number of neurons: 64
- Params:  $(128 + 1) \times 64 = 8256$  (weights + bias for each neuron)

**Softmax Layer:**

- Number of neurons: 10 (assuming classification)
- Params:  $(64 + 1) \times 10 = 650$  (weights + bias for each class)

**Sum Parameters** =  $156 + 2416 + 32896 + 8256 + 650 = 44374$

۵

۶

۷

منابع

[1] <https://www.shiksha.com/online-courses/articles/relu-and-sigmoid-activation-function/>

[2] <https://medium.com/@amanatulla1606/vanishing-gradient-problem-in-deep-learning-understanding-intuition-and-solutions-da90ef4ecb54>

[3] [https://en.wikipedia.org/wiki/Rectifier\\_\(neural\\_networks\)](https://en.wikipedia.org/wiki/Rectifier_(neural_networks))

- [4] <https://wandb.ai/ayush-thakur/dl-question-bank/reports/ReLU-vs-Sigmoid-Function-in-Deep-Neural-Networks-VmlldzoyMDk0MzI>
- [5] <https://medium.com/swlh/why-are-neural-nets-non-linear-a46756c2d67f>