



دانشکده مهندسی برق

تکلیف کامپیوتری درس بازشناسی آماری الگو، سری پنجم

استاد

دکتر ابریشمی مقدم

نیمسال اول ۱۴۰۵-۱۴۰۴

مهلت تحویل: ۱۴۰۴/۱۰/۱۵

با سلام و آرزوی شادی، موفقیت و سلامتی؛

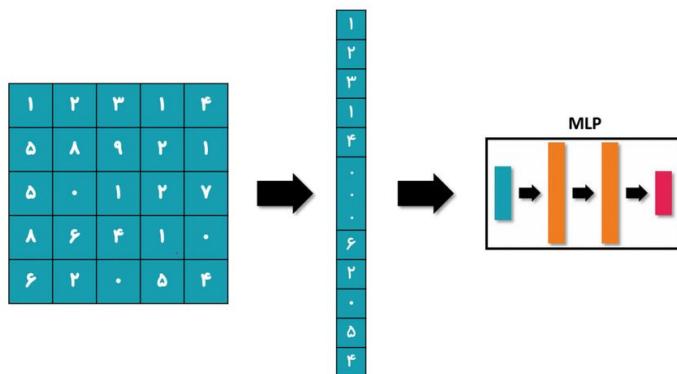
لطفا در تحویل پاسخ‌های خود موارد زیر را مدنظر داشته باشید:

- فقط برنامه‌هایی که به زبان ترجیحاً پایتون و یا متلب باشند قابل قبول خواهند بود.
- تحویل همزمان گزارش و کدها الزامی است.
- گزارش باید شامل خروجی‌های کدهای نوشته شده باشد، که موارد خواسته شده در سوالات هستند و سایر توضیحات خواسته شده دیگر در متن سوالات نیز پاسخ داده شود (از آوردن کد در گزارش خودداری کنید).
- لطفا کدهای برنامه به صورت ماجولات و همراه با توضیحات کافی باشند؛ طوری که بخش‌های مختلف برنامه کاملاً قابل تفکیک بوده و اجرا و ارزیابی هر بخش توسط کاربر به آسانی و بدون نیاز به ورود به جزئیات برنامه میسر باشد.
- فایل تحویلی پاسخ شما باید تنها یک فایل زیپ، تحت عنوان "SPR_CHW5_Student ID" محتوی دو پوشه باشد.
- گزارش خود را در پوشه اول با عنوان "Report" و کدهای خود را ترجیحاً به فرمت Jupyter Notebook در پوشه "Codes" قرار دهید.
- با این که همکاری، مشورت، و استفاده از ابزارهای کمکی در حل سوالات پیشنهاد می‌شود، حتماً به صورت مستقل به نوشتن کدها و گزارش بپردازید.
- ممکن است از دانشجویی خواسته شود در زمانی که تعیین خواهد شد جزئیات کدش را در جلسه‌ای مجازی توضیح دهد، نتایج را تحلیل کند و حتی تغییراتی در پارامترهای کد اعمال کند. در صورتی که دانشجویی تمارین را تحویل داده باشد اما نتواند کد خود را توضیح دهد و یا تغییراتی روی آن اعمال کند، و یا اینکه کد یا گزارش تحویلی به پاسخ دیگران شباهت غیرمنطقی داشته باشد، نمره تمرین صفر لاحظ شده و نمره‌ای منفی هم لاحظ خواهد شد.
- در صورت وجود هرگونه سوال یا ابهام، مشکل مربوطه را با آی دی تلگرام زیر در میان گذارید:

@omid_Emaa

1 - CNN For Classification

در این تمرین به پیاده سازی یک شبکه عصبی کانولوشنی یا همان CNN برای طبقه بندی دیتاست KMNIST می کنید. در تمارین قبلی با آن آشنا شدیم می پردازیم. در درس به تفصیل با MLP آشنا شده اید. حال فکر می کنید تفاوت آن ها در چیست؟ تفاوت در ورودی است، مثلا تصویر، تصاویر معمولا به شکل یک ماتریس دو بعدی از اعداد ۱۰۰×۱۰۰ نمایش داده می شوند. هر درایه در این ماتریس دو بعدی معادل با یک پیکسل هست. اگر یک تصویر ۱۰۰×۱۰۰ داشته باشیم، یعنی ۱۰۰۰۰ پیکسل داریم که به صورت دو بعدی کنار همدیگر نشسته اند. حال تصور کنید بخواهیم یک لایه ورودی برای این ۱۰۰۰۰ پیکسل بسازیم؛ باید ۱۰۰۰۰ نورون برای لایه ورودی شبکه MLP درنظر بگیریم. اضافه کردن نورون و لایه بیشتر به این شبکه MLP باعث می شود شبکه ما شامل حجم بزرگی از پارامترها شود، محاسباتش هزینه برد باشد و البته Overfitting اتفاق بیفتد.



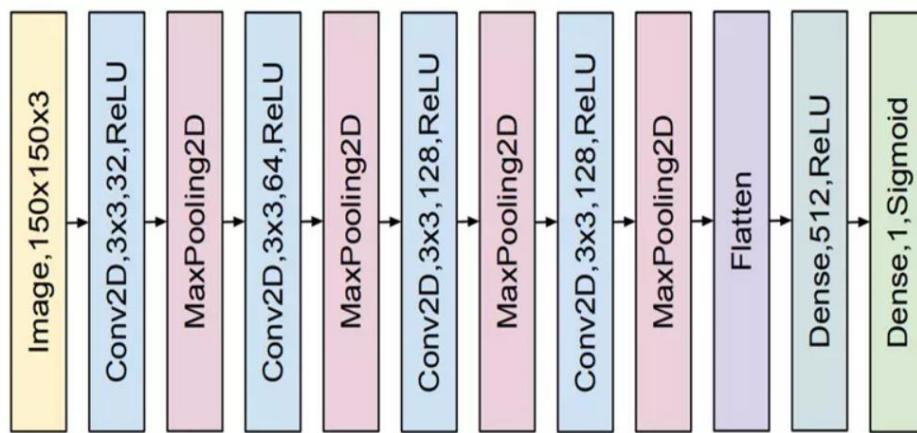
حال نقطه قوت شبکه CNN چیست؟ این شبکه ها به گونه ای طراحی شده اند که برای ورودی های با ساختار ماتریسی (دو بعدی و سه بعدی) به خوبی کار می کنند. شبکه MLP، ساختار داده های ورودی را عوض می کند و یک ماتریس دو بعدی ۱۰۰×۱۰۰ را تبدیل به یک بردار به ابعاد ۱۰۰۰۰ می کند. اما شبکه CNN ساختار ورودی را عوض نمی کند و به ارتباط بین پیکسل های همسایه اهمیت می دهد.

چرا باید به ارتباط به پیکسل های همسایه اهمیت دهیم؟ مگر غیر از این است که از کنار هم قرار گرفتن تعدادی پیکسل در راستای سطر و ستون یک تصویر مشخص تشکیل شده است؟ بنابراین، مهم است که ساختار تصویر یا ورودی را عوض نکنیم (تبدیل به بردار نکنیم) و همان ورودی اصلی را به شبکه بدهیم. دقیقا همان کاری که شبکه CNN انجام می دهد.

در شکل زیر یک مثال ساده از معماری شبکه عصبی کانولوشن نشان داده شده است. شبکه کانولوشن معمولا از بلوک های مختلفی تشکیل شده است. لایه ها یا بلوک های مختلف در شبکه CNN عبارتند از:

- لایه ورودی (Input layer)

- لایه کانولوشن (Convolutional layer)
- لایه غیرخطی (Non-linear activation function)
 - یک جا نشان می‌دهند
- لایه پولینگ (Pooling layer)
- لایه تمام متصل (Fully connected layer)



در صورت علاقه برای توضیحات بیشتر میتوانید به این [لینک](#) مراجعه کنید. همانطور که پیشتر گفته شد در این تمرین قصد داریم CNN را بر روی مجموعه داده KMNIST پیاده‌سازی کنیم.

بارگذاری مجموعه داده:

مجموعه داده KMNIST را با استفاده از فریم‌ورک انتخابی خود بارگذاری نمایید. به‌منظور کاهش زمان اجرا، مجاز هستید تنها از N نمونه‌ی اولیه (برای مثال 10000 یا 20000 تصویر) استفاده نمایید. مقدار N باید در گزارش ذکر شود. پیش‌پردازش‌های لازم را انجام دهید، از جمله: تبدیل داده‌ها به نوع عددی مناسب (float) و نرمال‌سازی داده‌ها و همچنین اطمینان از شکل صحیح ورودی تصاویر. برای هر کلاس، یک نمونه نمایش دهید و تعداد نمونه‌های موجود در هر کلاس را گزارش کرده و در مورد توازن یا عدم توازن کلاس‌ها توضیح دهید.

پیاده سازی CNN:

با استفاده از لایه های آماده ی چارچوب انتخابی، یک شبکه کانولوشنی طراحی نمایید که دارای ویژگی های زیر باشد: حداقل دو لایه کانولوشن - استفاده از Dropout - استفاده از Batch Normalization

کاهش overfitting - لایه خروجی متناسب با ۱۰ کلاس KMNIST

استفاده از توابع آماده کتابخانه ها مجاز است و نیازی به پیاده سازی دستی مراحل آموزش یا بهینه سازی نیست

تنظیم فرآیند آموزش:

- یک optimizer مناسب (مانند Adam یا SGD) انتخاب کنید و به طور خلاصه توضیح دهید.
- تابع هزینه مناسب (Cross Entropy) را مشخص نمایید.
- مقادیر epoch ها شامل learning rate و تعداد batch size hyperparameter را تعیین و در گزارش ذکر کنید.

ارزیابی مدل با روش Hold-out:

- مشابه با تکلیف کامپیوتری ۴، داده های train را به دو بخش train و validation تقسیم کنید.
- مجموعه داده test را کاملاً کنار گذاشته و در هیچ یک از مراحل تنظیم مدل از آن استفاده نکنید.
- مدل را آموزش داده و نمودارهای Loss و Accuracy بر حسب epoch را برای دو بخش train و validation رسم نمایید.

پس از پایان آموزش، مدل نهایی را تنها یکبار روی مجموعه داده test ارزیابی کرده و موارد زیر را گزارش دهید:

- Accuracy
- Confusion Matrix
- دقت هر کلاس یا Per-class accuracy

```
[Hold-out] Confusion Matrix:  
[[963  4  2   0  20  2   0   5   1   3]  
 [ 1 949  3  1   6  2  19  3   2  14]  
 [ 6  2 902 34  7  8  24  6   6   5]  
 [ 0   1  5 978  1  7  4   2   2   0]  
 [ 15  5  2  2 949  2  2   3  15  5]  
 [ 1   3 25  7  3 942  6  2   5  6]  
 [ 2   4  9  2  1   0 981  0   0  1]  
 [ 1   0  2  0  2   0   1 983  1  10]  
 [ 2   5  4  7  2  1   2   0 975  2]  
 [ 3   1  2  0  5  2  4   2  1 980]]  
[Hold-out] Per-class accuracy: [0.963, 0.949, 0.902, 0.978, 0.949, 0.942, 0.981, 0.983, 0.975, 0.98]  
[Hold-out] TEST accuracy: 0.9602
```

خروجی نمونه

ارزیابی مدل با روش : K-Fold Cross Validation

مانند قبل مجموعه داده `test` را کنار گذاشته و تنها روی داده های `train`، اعتبار سنجی را انجام دهید از K برابر با ۱۰ استفاده کنید. در هر فولد مدل را آموزش دهید و مقدار `accuracy` روی داده هی اعتبار سنجی را گزارش نمایید. این کار را برای چندین مجموعه های پر پارامتر انجام داده و بهترین مجموعه را انتخاب کنید.

- با استفاده از بهترین مجموعه انتخاب شده، مدل نهایی را آموزش دهید.

- مدل را تنها یکبار روی مجموعه داده `test` ارزیابی کنید.

- نتایج بدست آمده را با نتایج روش Hold-out مقایسه نمایید.

راهنمایی : اگر با `k-fold` به نتایج خوبی نمی رسید میتوانید از `stratified k-fold` استفاده نمایید.

(امتیازی +٪۱۰) سعی کنید به دقت تست بالای ۹۰ درصد دست پیدا کنید.

(امتیازی +٪۲۰) سعی کنید به دقت تست بالای ۹۵ درصد دست پیدا کنید.

"موفق باشید"