

بسم الله الرحمن الرحيم

دانشگاه علم و صنعت ایران

بهار ۱۴۰۰

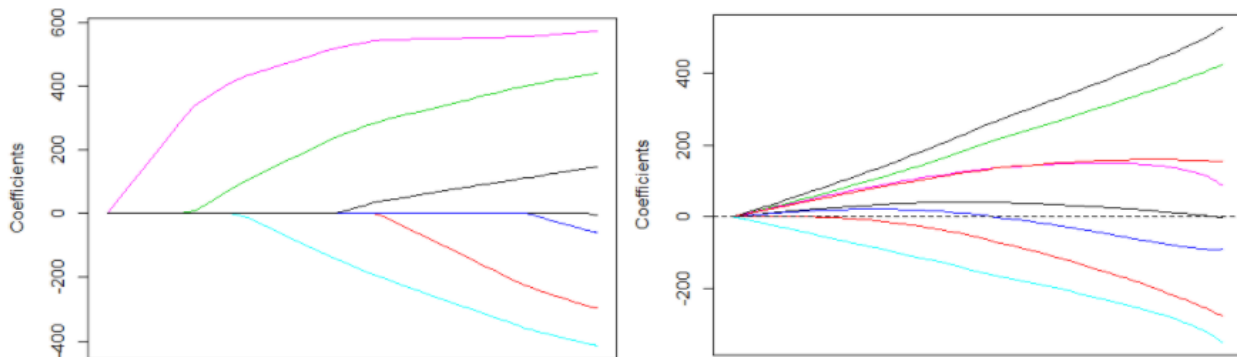
یادگیری عمیق

تمرین سری چهارم

تحویل: شنبه ۲۱ فروردین

۱. مشکل بیش‌برازش (overfitting) را توضیح دهید و حداقل ۵ راه‌حل برای کاهش آن پیشنهاد دهید (مراجع مورد استفاده را ذکر کنید).

۲. در شکل زیر محور افقی نرم (۱ یا ۲) ضرایب یک شبکه را نشان می‌دهد و محور عمودی مقادیر مربوط به ضرایب این شبکه را مشخص می‌کند. به نظر شما کدام نمودار مربوط به استفاده از L1-Regularization و کدام مربوط به L2-Regularization می‌باشد و پاسخ خود را از منظر ضرایب تحلیل کنید.

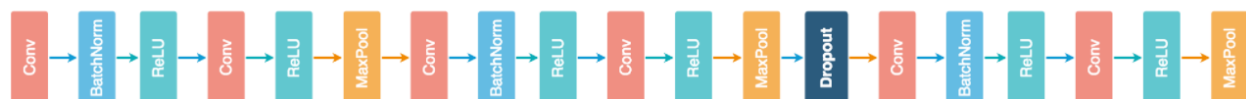


۳. شبکه‌های عصبی عمیق در کنار توانمندیشان اما همیشه قابل اطمینان نیستند و بحث نشتی اطلاعات و attack در این شبکه‌ها مطرح است. با مطالعه مقاله زیر، adversarial attack را توضیح دهید و بفرمایید به چه صورت کارکرد درست شبکه را از بین می‌برند و راه‌های مقابله با آن چیست.

https://openaccess.thecvf.com/content_cvpr_2016/papers/Moosavi-Dezfooli_DeepFool_A_Simple_CVPR_2016_paper.pdf

۴. یک شبکه عمیق کانولوشنی (CNN) با معماری زیر بر روی مجموعه داده CIFAR10 با استفاده از PyTorch در این [نوت بوک](#) پیاده سازی شده است (لطفا تغییری در معماری شبکه ندهید و تنها موارد خواسته شده را انجام دهید).

Convolutional Layers



Fully Connected Layers



در قسمت‌های مرتبط آن موارد زیر را پیاده سازی کنید. (تغییرات لازم را اعمال کنید یا بخش جدید اضافه نمایید)

- (A) data augmentation
- (B) train with ADAM
- (C) dropout with different parameters
- (D) L2-regularization with different parameters

سپس دقت های بدست آمده را در جدول زیر با استفاده از روش های مختلف (حالت های مختلف موارد فوق و بدون آنها) یادداشت نمایید و نتایج را مقایسه و تحلیل نمایید.
(تعداد و نوع لایه ها و پارامترهایی که تنظیم کردید)

| model | Train accuracy | Test accuracy |
|--------------------------------------|----------------|---------------|
| without regularization | | |
| with data augmentation | | |
| with dropout=0.5 | | |
| with dropout= your setting | | |
| with dropout= your setting | | |
| with L2-regularization= 0.0001 | | |
| with L2-regularization= your setting | | |
| with L2-regularization= your setting | | |

در شکل زیر یک نمونه کد برای augmentation قرار داده شده است.

```
transform_train = transforms.Compose([
    transforms.RandomCrop(32, padding=4),
    transforms.RandomHorizontalFlip(),
    transforms.ToTensor(),
    transforms.Normalize((0.4914, 0.4822, 0.4465), (0.2023, 0.1994, 0.2010)),
])

# Normalize the test set same as training set without augmentation
transform_test = transforms.Compose([
    transforms.ToTensor(),
    transforms.Normalize((0.4914, 0.4822, 0.4465), (0.2023, 0.1994, 0.2010)),
])
```

این لینک ها برای یادگیری و پیاده سازی روش های مختلف augmentation مفید می باشند:

1. <https://pytorch.org/vision/stable/transforms.html>
2. <https://modelzoo.co/model/data-augmentation-and-sampling-for-pytorch>

برای یادگیری کتابخانه PyTorch نیز مطالعه لینک های زیر توصیه می شود.

1. [An Introduction to PyTorch – A Simple yet Powerful Deep Learning Library](#)
2. [PyTorch Examples](#)

نکات تکمیلی

- (۱) لطفاً پاسخ سوالات (تئوری و توضیحات پیاده سازی) را به طور گویا و به زبان فارسی و در صورت امکان تایپ همراه با سورس کدهای نوشته شده، در یک فایل فشرده شده به شکل HW4_YourStudentID.zip قرار داده و بارگذاری نمایید.
- (۲) منابع استفاده شده را به طور دقیق ذکر کنید.
- (۳) برای سهولت در پیاده سازی ها و منابع بیشتر، زبان پایتون پیشنهاد می شود. لطفا کدهای مربوطه را به طور جداگانه در فرمت py یا ipynb ارسال نمایید.
- (۴) ارزیابی تمرین ها براساس صحیح بودن راه حل ها، گزارش مناسب، بهینه بودن کدها و کپی نبودن می باشد.
- (۵) در مجموع تمام تمرین ها، تنها ۷۲ ساعت تاخیر در ارسال پاسخ ها مجاز است اما پس از آن به صورت خطی از نمره کسر خواهد شد (معادل با روزی ۵۰ درصد).
- (۶) در رابطه با پرسش و پاسخ در رابطه با تمرین ها می توانید در گروه مربوطه مطرح کنید.

sh_ezatzadeh@comp.iust.ac.ir

موفق باشید.