

# به نام خدا



دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده برق و کامپیوتر



درس هوش مصنوعی قابل اعتماد

مدرس: دکتر مصطفی توسلی‌پور

تمرین شماره ۱

۱۴۰۲ اسفند ماه

# فهرست

3 .....	مقدمه
4 .....	<b>سوال اول : generalization</b>
4 .....	مقدمه
4 .....	معرفی مجموعه داده و مدل
5 .....	آموزش مدل اولیه
5 .....	بهبود تعمیم‌پذیری مدل
5 .....	model architecture
6 .....	Loss function
6 .....	data augmentation
6 .....	Input features (Feature Extraction)
7 .....	optimizer
7 .....	آموزش معکوس مدل
7 .....	unsupervised
7 .....	supervised
9 .....	<b>سوال دوم : robustness</b>
11 .....	مراجع
12 .....	نکات تحویل

## مقدمه

به اولین تمرین درس "هوش مصنوعی قابل اعتماد" خوش اومدین! در این تمرین، که مربوط به بخش اول درس است، قصد داریم تا به مفاهیم اساسی تعمیم‌پذیری<sup>۱</sup> و مقاومت<sup>۲</sup> بپردازیم. در این تمرین، ما جنبه‌های ضروری ساخت مدل‌های هوش مصنوعی را بررسی می‌کنیم که می‌توانند به خوبی به داده‌های جدید و نادیده تعمیم دهند و در عین حال در مواجهه با عدم قطعیت‌ها و ورودی‌های متخاصم استحکام خود را حفظ کنند. در نهایت، هدف این سری تمرین این است که یاد بگیریم چگونه مدل خود را generalize و robust تر کنیم و مدل‌هایی را آموزش دهیم که عملکرد خوبی از این دو جنبه داشته باشند.

---

generalization <sup>۱</sup>  
robustness <sup>۲</sup>

# سؤال اول : GENERALIZATION

## مقدمه

دستیابی به توانایی قدرت تعمیم‌پذیری<sup>۱</sup> خوب در شبکه‌های عصبی عمیق برای ایجاد سیستم‌های هوش مصنوعی قابل اعتماد بسیار حائز اهمیت است. تعمیم‌پذیری در یادگیری ماشین به توانایی یک مدل برای پیش‌بینی دقیق داده‌های جدید و دیده نشده بر اساس الگوهای آموخته شده از داده‌های آموزشی اشاره دارد. در کلاس درس، با انواع و تعاریف مختلف تعمیم‌پذیری آشنا شدید و همچنین تکنیک‌های مختلفی که به کمک آن‌ها می‌توان توانایی تعمیم‌پذیری یک مدل را بالا برد، آشنا شدید. در این تمرین، قصد داریم این تکنیک‌ها را به کار گرفته و به کمک آن‌ها میزان تعمیم‌پذیری یک مدل شبکه عصبی را برای کاربرد طبقه‌بندی بهبود ببخشیم.

## معرفی مجموعه داده و مدل

در این تمرین قصد داریم تا یک مدل [Resnet18 \(لینک پایتورج\)](#) را با استفاده از مجموعه داده SVHN آموزش داده و سپس قدرت تعمیم‌پذیری آن را بر روی مجموعه داده Mnist آزمایش کنیم. سپس سعی می‌کنیم عملکرد مدل را با استفاده از تکنیک‌های مختلف موجود، بهبود دهیم.

پایگاه داده MNIST مجموعه بزرگی از ارقام دستنویس است که دارای ۱۰ کلاس مختلف است. این مجموعه داده حاوی ۶۰ هزار نمونه در مجموعه آموزشی و ۱۰ هزار نمونه در مجموعه تست است و تمامی تصاویر آن تک کanalه هستند.

SVHN یک مجموعه داده تصویری در دنیای واقعی برای توسعه الگوریتم‌های یادگیری ماشین و تشخیص اشیا با حداقل نیاز به پیش‌پردازش و قالب بندی داده است. این دیتابست بسیار شبیه به دیتابست Mnist است (تصاویر از ارقام برش خورده کوچک هستند)، اما دارای تعداد بیشتری از داده‌های برچسب گذاری شده است. این دیتابست شامل حدود ۷۶ هزار تصویر در مجموعه آموزش<sup>۲</sup> و حدود ۲۶ هزار داده به عنوان مجموعه آزمایش<sup>۳</sup> است. تمامی تصاویر سه کanalه و با ابعاد  $32 \times 32$  هستند.

هردو این مجموعه داده‌ها به راحتی از طریق پکیج `torchvision.datasets` قابل دسترسی هستند.

Generalization<sup>۱</sup>

Train set<sup>۲</sup>

Test set<sup>۳</sup>

## آموزش مدل اولیه

یک مدل Resnet18 را به صورت دستی (و نه با استفاده از `torchvision.models`) پیاده سازی کرده و با استفاده از `train set` تعریف شده برای دیتاست SVHN آموزش دهید. سپس نتیجه طبقه بندی را برروی `test set` همین دیتاست و همچنین برروی `test set` دیتاست Mnist گزارش کنید. (توجه داشته باشید که دادگان دیتاست Mnist بر خلاف SVHN تک کاناله هستند و هنگام تست، باید این موضوع را در نظر داشته باشید. حتما راه حل خود برای این کار را در گزارش ذکر کنید)

برای آموزش می‌توانید از روش SGD with momentum استفاده کرده و مدل را به اندازه حداقل ۱۰ ایپاک آموزش دهید. نمودار loss در هنگام آموزش را نیز حتما در گزارش خود قرار داده و تحلیل خود را از نتایج به دست آمده ذکر کنید. (۱۵ نمره)

## بهبود تعمیم‌پذیری مدل

در این بخش می‌خواهیم از تکنیک‌های موجود برای بهبود generalization (که در اسلاید های درس هم به آنها اشاره شده) استفاده کرده و تاثیر هر یک را در نتایج به دست آمده مشاهده کنیم. توجه کنید که هر یک از تکنیک‌های زیر را باید برروی مدل اولیه آموزش دیده شده (و با تنظیمات مشابه (در شرایط یکسان)) و بدون توجه به سایر تکنیک‌ها (به صورت مستقل از سایر زیربخش‌ها) اعمال کنید.

## MODEL ARCHITECTURE

یکی از راههایی که می‌توان به کمک آن میزان تعمیم‌پذیری را تحت تاثیر قرار داد، دستکاری معماری مدل است. دو مورد از عوامل موثر، استفاده از batch normalization و dropout در ساختار مدل است که به ترتیب در مقاله [1] و مقاله [2] مورد بررسی قرار گرفته‌اند.

1. ابتدا این مقالات را بررسی کرده و برای هر کدام مختصر (در حداکثر یک پاراگراف) توضیح دهید که چرا استفاده از این موارد در بهبود generalization موثر است. (۲ نمره)
2. همانطور که می‌دانید، در معماری Resnet18 از batch normalization استفاده شده است. در شبکه‌ای که در بخش قبل پیاده سازی کردید، تمام آنها را حذف کرده و مجددا شبکه را آموزش داده و نتایج (دقت و نمودار خطای) را گزارش کنید. آیا نتیجه‌ای که به دست آورده‌ید منطبق با حدس شما بود؟ تحلیل شما چیست؟ (۴ نمره)

---

## LOSS FUNCTION

یکی از تکنیک های مطرح برای بهبود تعمیم‌پذیری، طراحی یک تابع خطای مناسب است. چندین تابع خطای در ادبیات ارائه شده است که برای بهبود تعمیم‌پذیری مدل‌ها در task های مختلف طراحی شده‌اند، و بسیاری از اینها Model agnostic هستند، به این معنی که می‌توان آنها را بدون توجه به معماری خاص برای انواع مختلف شبکه‌های عصبی اعمال کرد. یکی از مثال‌های قابل توجه، تکنیک Label Smoothing برای Regularization است. ابتدا به صورت مختصر توضیح دهید که این تکنیک چگونه عمل می‌کند؟ اگر بخواهیم آن را ببروی تابع خطای Cross Entropy پیاده کنیم، عملکرد این تابع خطای چگونه خواهد شد؟

(۲ نمره)

سپس تابع خطای Label Smoothing Cross Entropy را پیاده سازی کرده و بدون تغییر سایر موارد، شبکه را یکبار دیگر با این تابع خطای جدید آموزش دهید. نتایج خود را به طور کامل گزارش کرده و تحلیل خود را ارائه دهید. (مقدار  $\text{smoothing}=0.25$  در نظر بگیرید) (۸ نمره)

---

## DATA AUGMENTATION

یکی دیگر از رایج ترین راه حل‌های موجود برای بهبود تعمیم‌پذیری، استفاده از data augmentation است. با استفاده از این روش، که به شکل‌های گوناگونی هم می‌توان آن را انجام داد، تنوع داده را بالا برد و مدل را با حجم داده بیشتری آشنا می‌کنیم که باعث بهبود تعمیم‌پذیری می‌شود. با توجه به این نکته که با توجه به جنس داده‌های ما در این سوال، برخی از انواع augmentation قبل استفاده نیستند(چرا؟)، مجموعه‌ای از augmentation های مناسب را (با آزمون و خطای) به گونه‌ای پیدا کنید که منجر به افزایش دقت مدل بر روی داده‌ی دیده نشده از دیتابست دیگر(قدرت تعمیم‌پذیری) شود. مدل برای augmentation های خود به همراه نتایج آموزش در گزارش ذکر کرده و تحلیل خود را ارائه دهید.

(۸ نمره)

---

## INPUT FEATURES (FEATURE EXTRACTION)

همانطور که می‌دانید، ویژگی‌های ورودی و استخراج ویژگی مناسب، یکی دیگر از راه‌هایی است که به کمک آن می‌توان عملکرد مدل در زمینه تعمیم پذیری را بهبود داد. مدل‌های از پیش آموزش دیده<sup>۱</sup>، به ویژه آن‌هایی که بر روی مجموعه داده‌های بزرگ مانند ImageNet آموزش دیده‌اند، بازنمایی ویژگی‌های غنی را از حجم وسیعی از داده‌های متتنوع آموخته‌اند.

---

Pre-trained models<sup>1</sup>

استفاده از این نمایش‌های از پیش آموزش دیده می‌تواند چندین مزیت را برای بهبود عملکرد تعمیم ارائه دهد از جمله آنها می‌توان به transfer learning و به خصوص استخراج ویژگی اشاره کرد؛ چرا که حتی بدون finetuning، مدل‌های از پیش آموزش دیده می‌توانند به عنوان استخراج‌کننده ویژگی‌های قدرتمند عمل کنند.

به همین منظور خوب است که عملیات آموزش را یکبار هم با استفاده از مدل Resnet18 ای که بروی دیتاست ImageNet از قبل آموزش دیده نیز انجام دهید. سپس نتیجه را گزارش کرده و تحلیل خود را ارائه دهید. (در این بخش می‌توانید از `torchvision.models` استفاده کرده و در هنگام لود کردن مدل `Resnet18`، آرگومان `pretrained=True` قرار دهید و یا اینکه وزن‌های مربوطه را بروی شبکه ای که از پیش ساخته‌اید لود کنید). نتایج و تحلیل خود را در گزارش ارائه دهید. (۴ نمره)

## OPTIMIZER

در این بخش با استفاده از یک بهینه‌ساز دیگر (و مجدداً با ثابت در نظر گرفتن سایر تنظیمات مدل اولیه)، آموزش مدل را تکرار کنید (می‌توانید از Adam استفاده کنید). مطابق معمول نتایج را به طور کامل گزارش کرده و تحلیل خود را ارائه دهید. (۴ نمره)

## آموزش معکوس مدل

## UNSUPERVISED

در این بخش از شما می‌خواهیم که با استفاده از نتایجی که در بخش قبلی گرفتید، بهترین setting ممکن را اعمال کرده و بار دیگر مدل را آموزش دهید. منتها این بار مدل را با استفاده از داده‌های train set تست کنید (تست باید بروی `test set` هردو دیتاست انجام آموزش داده و بروی دیتاست SVHN تست کنید (تست باید بروی `test set` هردو دیتاست انجام شود.). نتایج خود را گزارش کنید. آیا دقت مشابه با حالت قبلی است؟ تحلیل خود را ارائه دهید. (۵ نمره)

## SUPERVISED

تا به اینجا، تمامی ارزیابی‌هایی که بروی مدل‌های آموزش دیده شده داشتیم، به صورت unsupervised بودند. یعنی از مدل آموزش دیده می‌خواستیم تا داده‌ای را برای ما طبقه‌بندی کند که تا حال ندیده‌ایم می‌خواهیم این کار را به صورت supervised انجام داده و تاثیر آن را ببینیم. به همین منظور، این بار پس از آموزش مدل با استفاده از دیتاست Mnist، یکبار آنرا با استفاده از تعداد کمی (مثلاً 1000-500 نمونه) از داده‌های SVHN fine tune کرده و سپس دقت را بروی `test set`ها به دست آورید. تحلیل

خود را از نتیجه به دست آمده ارائه دهید. (توجه داشته باشید که در این بخش تنها لازم است که classifier انتهایی را fine tune کنید و لایه‌های کانولوشنی را freeze کنید) **(۸ نمره)**

## ROBUSTNESS: سوال دوم

تابع هزینه [3] Circle از وزن دهی تطبیقی<sup>۱</sup> برای انعطاف پذیری در بهینه سازی شباهت های درون کلاسی و همچنین بین کلاسی استفاده می کند که باعث بهبود عملکرد آن نسبت به روش های پیشین شده است.

در این بخش قصد داریم با مفهوم حملات متخاصمانه<sup>۲</sup> بیشتر آشنا شده و همچنین مقاومت<sup>۳</sup> دادگان CIFAR10 را با کمک این تابع هزینه و روش adversarial training بهبود دهیم. بدین منظور از بازنمایی مدل از پیش آموزش داده<sup>۴</sup> شده ResNet-18 استفاده می کنیم.

دیتابست CIFAR10 شامل ۶۰۰۰۰ عکس رنگی<sup>۵</sup> در ۱۰ کلاس می باشد که هر کلاس دارای ۶۰۰۰ تصویر است. در ابتدا مدل ResNet و همچنین دادگان Cifar10 را به صورتی که ۲۰ درصد برای داده آموزش<sup>۶</sup> و ۸۰ درصد داده ها برای برآش<sup>۷</sup> استفاده شود، لود کنید (به گونه ای که از هر کلاس به صورت برابر در فرایند آموزش استفاده شود).

در مورد حملات FGSM<sup>۸</sup> و PGD<sup>۹</sup> توضیح مختصری دهید. سپس نمونه های adversarial بر روی دادگان تست را از طریق اعمال fast gradient sign method با پارامتر اپسیلون ۰.۱ و همچنین اضافه کردن نویز (به صورت رندوم تغییر دادن تعدادی از پیکسل های تصویر) ایجاد کنید و تعدادی از این تصاویر را نمایش دهید. (۷ نمره)

• در ادامه، در تمامی حالات زیر دقت بر روی دادگان تست را گزارش کرده و نمودار loss را رسم کنید، و همچنین به منظور نمایش داده ها، دادگان ۵۱۲ بعدی را در دو بعد به کمک UMAP تصویر کنید:

الف) آموزش مدل با استفاده از داده original و تابع هزینه cross entropy

- عملکرد مدل را یکبار با استفاده از داده های تست original و بار دیگر با استفاده از داده های تست adversarial ارزیابی کنید. (۱۰ نمره)

adaptive re-weighting<sup>۱</sup>

adversarial attack<sup>۲</sup>

Robustness<sup>۳</sup>

pre-trained<sup>۴</sup>

train<sup>۵</sup>

validation<sup>۶</sup>

Fast Gradient Sign Method<sup>۷</sup>

projected gradient descent<sup>۸</sup>

ب) آموزش مدل با استفاده از دیتاست augmented (به هر داده با احتمال 50 درصد اغتشاش وارد شود) و تابع هزینه cross entropy

- عملکرد مدل را یکبار با استفاده از داده های تست original و یکبار نیز با داده های تست adversarial ارزیابی کنید. (8 نمره)

ج) در مورد تابع هزینه Circle Loss به طور مختصر توضیح دهید ، فواید این تابع هزینه چیست و توانسته چه مشکلاتی که روش های قبل داشته اند را برطرف کند؟ (6 نمره)

د) مدل را بار دیگر با استفاده از تابع هزینه circle loss آموزش دهید

- عملکرد مدل را یکبار با استفاده از داده های تست original و یکبار نیز با داده های تست adversarial ارزیابی کنید. (9 نمره)

- در نهایت نتایج بدست آمده در تمامی حالات بخش الف و ب و د را با یکدیگر مقایسه و تحلیل کنید. (10 نمره)

\* توجه: دقت بفرمایید که هر batch می بایست شامل تعداد مناسبی از هر کلاس باشد.

- [1] N. Srivastava, G. Hinton, A. Krizhevsky, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014, Available: [https://www.jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf?utm\\_content=buffer79b43&utm\\_medium=social&utm\\_source=twitter.com&utm\\_campaign=buffer](https://www.jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf?utm_content=buffer79b43&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer)
- [2] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” *proceedings.mlr.press*, Jun. 01, 2015. <https://proceedings.mlr.press/v37/ioffe15.html>
- [3] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In CVPR, pages 6398–6407, 2020.

## نکات تحویل

- مهلت ارسال این تمرین تا پایان روز "دوشنبه ۲۰ فروردین ماه" خواهد بود.
- این زمان قابل تمدید نیست و در صورت نیاز می‌توانید از grace time استفاده کنید.
- در نظر داشته باشید که حداکثر مهلت آپلود تمرین در سامانه تا ۷ روز پس مهلت تحویل است و پس از آن سامانه بسته خواهد شد.

- پیاده سازی با زبان برنامه نویسی پایتون باید باشد و کدهای شما باید قابل اجرا بوده و به همراه گزارش آپلود شوند.
- انجام این تمرین به صورت یک نفره می‌باشد.
- در صورت مشاهده هر گونه تشابه در گزارش کار یا کدهای پیاده‌سازی، این امر به منزله تقلب برای طرفین در نظر گرفته خواهد شد.
- استفاده از کدهای آماده بدون ذکر منبع و بدون تغییر به منزله تقلب خواهد بود و نمره تمرین شما صفر در نظر گرفته می‌شود

- در صورت رعایت نکردن فرمت گزارش کار نمره گزارش به شما تعلق نخواهد گرفت.
- تحویل تمرین به صورت دستنویس قابل پذیرش نیست.
- تمامی تصاویر و جداول مورد استفاده در گزارش کار باید دارای توضیح (caption) و شماره باشند.
- بخش زیادی از نمره شما مربوط به گزارش کار و روند حل مسئله است.
- لطفا گزارش ، فایل کدها و سایر ضمائم مورد نیاز را با فرمت زیر در سامانه بارگذاری نمایید.

- HW1\_[Lastname]\_[StudentNumber].zip
  - در صورت وجود سوال و یا ابهام می‌توانید از طریق رایانame زیر با موضوع TAI\_HW1 با دستیاران آموزشی در ارتباط باشید:
  - سوال اول

alirezaghafouri1@[تلگرام](https://t.me/alirezaghafouri) یا [ایمیل](mailto:alirezaghafouri@ut.ac.ir)

◦ سوال دوم

anna12hh@[تلگرام](https://t.me/ana.hashemzadeh) یا [ایمیل](mailto:ana.hashemzadeh@ut.ac.ir)

با آرزوی سلامتی و موفقیت روزافزون. عیدتونم مبارک ☺