

سوال ۱: شبکه عصبی (تئوری)

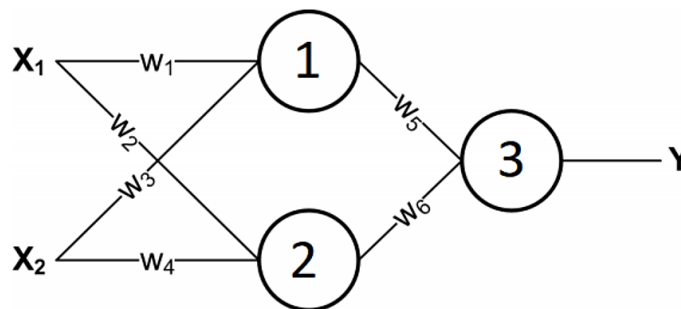
مقدمه: در این بخش با مقدمات شبکه عصبی به صورت تئوری آشنا می‌شوید.

شبکه عصبی تصویر ۱-۱ را در نظر بگیرید. در نورون‌هایی که با شماره‌های ۱ تا ۳ مشخص شده‌اند باید تابع فعالیت (Activation Function) قرار گیرد. می‌خواهیم از دو تابع فعالیت زیر استفاده کنیم:

$$S: \text{signed sigmoid function } S(a) = \text{sign}(\sigma(a) - 0.5) = \text{sign}\left(\frac{1}{1 + e^{-a}} - 0.5\right)$$

$$L: \text{Linear function } L(a) = c a$$

که در روابط بالا $a = \sum_i w_i X_i$ است.



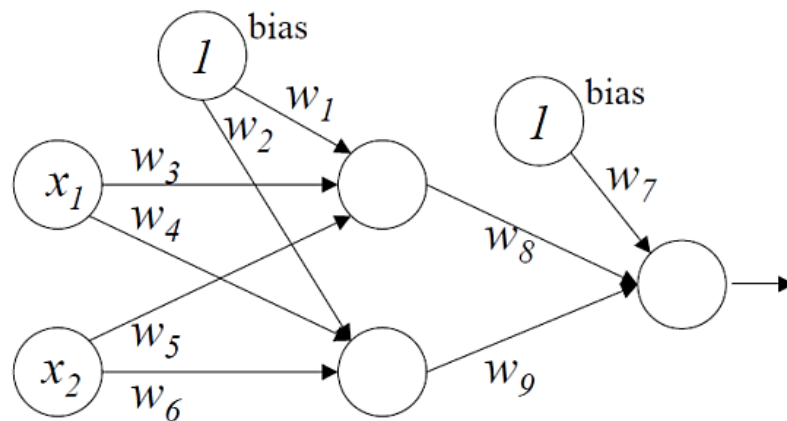
تصویر ۱-۱: شبکه عصبی بخش اول سوال ۱

قسمت الف: رگرسیون خطی

اگر بخواهیم شبکه‌ی بالا معادل یک رگرسیون خطی عمل کند، یعنی $Y = \beta_1 X_1 + \beta_2 X_2$ ، تابع فعالیت مناسب (S و L) برای هر یک از نورونهای ۱ تا ۳ چه خواهد بود؟

قسمت ب: طبقه بندی دو کلاسه

شبکه عصبی تصویر ۲-۱ را برای طبقه بندی دو کلاسه در نظر بگیرید. فرض کنید که لایه های میانی از تابع فعال سازی S و L در لایه خروجی استفاده می کند. خروجی شبکه عصبی را بر حسب پارامترهای شبکه (X, W) و ثابت c نوشته و مرز تصمیم نهایی را به دست آورید.



تصویر ۱-۱: شبکه عصبی بخش دوم سوال ۱

آیا می‌توان شبکه عصبی بدون لایه مخفی به دست آورد که معادل شبکه عصبی فوق باشد؟ در صورت وجود شبکه پیشنهادی را رسم کنید.

سوال ۲: شبکه عصبی (تئوری)

مقدمه: در این بخش به صورت تحلیلی به بررسی برخی مطالب مقدماتی شبکه‌های عصبی می‌پردازیم.

قسمت الف: Multi-Layer Networks

(۱) با استفاده از Perceptron عملگرهای منطقی NAND و NOR را پیاده‌سازی کنید.

(۲) سه عدد حقیقی z و y و x را خوب می‌نامیم اگر داشته باشیم:

$$(x > y \text{ and } y < z) \text{ or } (x < y \text{ and } y > z)$$

با استفاده از MLP شبکه‌ای طراحی کنید که سه عدد z و y و x را به عنوان ورودی دریافت کرده و اگر خوب بودند خروجی یک و در غیر این صورت خروجی منفی یک تولید کند.

قسمت ب: Regularization

در درس با L2 Regularization که تابع هدف را به شکل زیر تغییر می‌دهد آشنا شدید:

$$E = \left(y - \sum_i w_i x_i \right)^2 + \lambda \sum_i w_i^2$$

(۱) از عبارت بالا مشتق گرفته و با اعمال روش SGD توضیح دهید که چرا نام دیگر این روش Weight Decay است.

(۲) حال تابع هدف را به شکل زیر در نظر بگیرید:

$$E = \left(y - \sum_i w_i x_i \right)^2 + \lambda \sum_i |w_i|$$

با تکرار روند بخش ۱ برای تابع بالا توضیح دهید که چه تفاوتی در این روش با روش بخش ۱ وجود دارد؟ اگر وزنی کوچک باشد، چه اتفاقی برای آن می‌افتد؟ آیا این روش مزیتی هم دارد؟

(۳) برای سه بردار X_1, X_2, X_3 مقدار نرم‌های ۱ و ۲ را حساب کنید. حال با توجه به این مقادارها به صورت شهودی توضیح دهید که استفاده از کدام یک از Regularization های L1 یا L2 برای وزن های مدل، تعداد وزن های با مقدار صفر را بیشتر میکند؟

$$X_1 = \begin{bmatrix} 2 \\ 2 \\ 3 \end{bmatrix}, X_2 = \begin{bmatrix} 4 \\ 3 \\ 0 \end{bmatrix}, X_3 = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix}$$

قسمت ج: تحلیلی

(۱) فرض کنید می‌خواهیم تابع هدف f را تخمین بزنیم. از قبل می‌دانیم که پیچیدگی تابع هدف بالاست و دارای Stochastic Noise بسیار کمی است. در چه شرایطی بهتر است از مدلی نسبتاً ساده استفاده کنیم (مثلاً چند جمله‌ای های درجه ۲) و در چه شرایطی از مدلی نسبتاً پیچیده (مثلاً چند جمله‌ای های درجه ۲۰)

۲) فرض کنید شبکه‌ای داریم که ورودی‌های منطقی منفی یک یا یک دریافت کرده و تابع فعال ساز آن به ازای ورودی مثبت، عدد یک و به ازای ورودی منفی، عدد منفی یک را خروجی می‌دهد. با توجه به این که هر تابع منطقی را می‌توان به شکل Sum of Products نشان داد، بیشینه تعداد لایه‌های مورد نیاز برای نشان دادن یک تابع منطقی دلخواه چندتاست؟

۳) توضیح دهید تفاوت داده‌های test و validation چیست؟ چرا از داده‌های test برای validation استفاده نمی‌کنیم؟

سوال ۳: شبکه‌های عصبی پیچشی^۱ (پیاده‌سازی)

مقدمه: در این سوال می‌خواهیم با پیاده‌سازی مثال‌هایی، هم آموخته‌هایمان در مورد شبکه‌های عصبی را به کاربردهای عملی و واقعی نزدیکتر کنیم و هم کار با چارچوب‌ها و کتابخانه‌های مهم حوزه یادگیری ماشین؛ یعنی PyTorch و Tensorflow در زبان پایتون را تمرین کنیم.

بخش اول: طبقه‌بندی ساده با CNN

یکی از انواع بسیار مهم شبکه‌های عصبی که کاربردهای بسیار زیادی مخصوصاً روی دادگان تصویری دارد شبکه عصبی پیچشی است. با این نوع از شبکه‌ها و طرز کار آنها در درس آشنا شدید. در این قسمت از تمرین می‌خواهیم یک طبقه‌بند ساده با استفاده از کتابخانه‌های Keras و Tensorflow بسازیم. دادگان مورد استفاده در این سوال دادگان بسیار معروف CIFAR10 است که می‌تواند آن را با استفاده از کتابخانه keras دانلود کنید.

قسمت الف: پیش‌پردازش داده‌ها

پس از اینکه دادگاه CIFAR10 را دانلود کردید مشاهده می‌کنید که این دادگان به دو قسمت train و test تقسیم شده است. کل داده‌ها را با هم ترکیب کنید و ۸۰٪ آنها را به آموزش، ۱۰٪ به آزمون و ۱۰٪ را هم به اعتبارسنجی اختصاص دهید. تعداد کلاس‌های این دادگان و ابعاد و محدوده مقادیر هر تصویر را مشخص کنید. از هر کلاس به طور نمونه یک تصویر را نمایش دهید.

برای رسیدن به نتایج بهتر نیاز است تا داده‌ها نرمال شوند یعنی محدوده مقادیر هر تصویر به $[0, 1]$ نگاشت شوند. این نگاشت را روی تصاویر اعمال کنید.

برچسب داده‌ها را که خروجی مدل است به شکل one-hot کدگذاری کنید.

قسمت ب: پیاده‌سازی مدل اولیه و تعیین هایپرپارامترها

یک CNN مطابق فایل پیوست BaseModel.png، برای طبقه‌بندی دادگان CIFAR10 با استفاده از کتابخانه‌های Keras و Tensorflow بسازید. همانطور که مشاهده می‌کنید این مدل از ۳ بلوک تشکیل شده است که هر بلوک قسمت استخراج ویژگی^۲ این مدل شامل دو لایه Convolution و یک لایه Max Pooling است. از تابع فعالساز relu استفاده کنید.

چه تابع فعالساز را برای لایه آخر طبقه‌بند پیشنهاد می‌کنید؟

چه تابع خطایی را با توجه به قسمت‌های قبل برای این مسئله پیشنهاد می‌کنید؟

از تابع بهینه‌ساز Adam برای حل مسئله استفاده کنید.

¹ Convolutional Neural Networks (CNN)

² Feature extraction

تذکر: در تعریف مدل و توابع مورد نیاز اگر نیاز به تعیین مقدار پارامتری باشد، خودتان آن را تعیین کرده و با توضیحات کافی در گزارش بیاورید.

قسمت ج: آموزش و ارزیابی

با بسته‌بندی^۳ مناسب داده‌ها و با تعداد دور^۴ کافی، مدلی را که ساخته‌اید آموزش دهید. نمودارهای دقت^۵ و خطا^۶ را بر حسب تعداد دور رسم کنید. منحنی مربوط به داده‌های آموزش و اعتبارسنجی در یک نمودار رسم شود.

مدل آموزش دیده خود را با داده‌های آزمون محک بزنید. دقت، خطا و ماتریس آشفتگی^۷ را گزارش کنید.

قسمت د: بهبود مدل

در مدل قبلی از ۲ بلوک در قسمت استخراج ویژگی استفاده کرده بودیم. مدل‌های جدیدی با ۱ و ۳ بلوک تعریف کرده و قسمت ج را برای آنها تکرار کنید. بر حسب نتایج بدست آمده، بهترین مدل را در ادامه استفاده کنید.

بین لایه‌های مدل خود از لایه‌های *Dropout* و *Batch Normalization* استفاده و قسمت ج را تکرار کنید.

بخش ب: یادگیری انتقالی^۸

در بسیار از کاربردهای عملی نگاهی ابزاری به شبکه‌های عصبی داریم یا تخصص کافی برای طراحی شبکه‌های عصبی قدرتمند را نداریم حتی شاید دسترسی به دادگان غنی یا منابع سخت‌افزاری کافی برای آموزش مدل‌های بزرگ نداشته باشیم. در چنین شرایطی یکی از روش‌هایی که می‌تواند به ما در استفاده مفید از شبکه‌های عصبی پیچیده و قدرتمند کمک کند یادگیری انتقالی است.

در یادگیری انتقالی به دنبال به‌سازی شبکه‌های عصبی‌ای هستیم که قبلاً روی دادگان بزرگ به نحو خوبی آموزش دیده‌اند. ما با ایجاد تغییرات جزئی در ساختار این شبکه‌ها و انجام یک فرآیند آموزش ساده، آنها را مناسب کاربرد خودمان می‌کنیم.

در نظر بگیرید که خواهیم دادگان *tf_flowers* را طبقه‌بندی کنیم. برای اینکار می‌خواهیم از یکی از شبکه‌های عصبی که روی دادگان عظیم *imagenet* آموزش دیده است استفاده کنیم. ابتدا یکی از مدل‌های *ResNet50*, *VGG16*, *VGG19*, *InceptionV3*, *Xception*, ... را به همراه وزن‌های از قبل آموزش‌دیده^۹ آن روی دادگان *imagenet* از کتابخانه *Keras* دریافت کنید. سپس معماری مدلی که انتخاب کرده‌اید را در گزارش بررسی کنید. بگویید که کدام قسمت از این مدل مربوط به استخراج ویژگی و کدام قسمت مربوط به طبقه‌بندی است؟

³ Batch

⁴ Epoch

⁵ Accuracy

⁶ Loss

⁷ Confusion matrix

⁸ Transfer learning

⁹ Pre-trained

طبقه‌بند مدل شما به تعداد کلاس‌های دادگانی که روی آن آموزش دیده است خروجی دارد. قسمت طبقه‌بند مدل انتخابی را مطابق با تعداد کلاس‌های `tf_flowers` اصلاح کنید.

حالا فقط آن وزن‌هایی را که مربوط به قسمت طبقه‌بند جدید می‌شوند، با تعریف تابع خطا و بهینه‌ساز مناسب، آموزش دهید و وزن‌های قسمت استخراج ویژگی را ثابت نگهدارید. نمودار دقت و خطای داده‌های آموزش و اعتبارسنجی را بر حسب تعداد دور آموزش رسم کنید.

در نهایت چند دور هم وزن‌های کل مدل، اعم از طبقه‌بند جدید و استخراج ویژگی را آموزش دهید و نمودار دقت و خطای داده‌های آموزش و اعتبارسنجی را بر حسب تعداد دور آموزش، در ادامه نمودار قبلی، رسم کنید.

دقت، خطا و ماتریس آشفتگی را روی داده‌های آزمون گزارش کنید.

نکته: حل هر کدام از بخش‌های این سوال با استفاده از کتابخانه PyTorch نمره امتیازی دارد. حل با کتابخانه Keras الزامی است.

نکات تحویل

- ۱- مهلت تحویل این تمرین **۵ دی‌ماه** می‌باشد.
- ۲- انجام این تمرین به صورت یک‌نفره است.
- ۳- برای انجام این تمرین تنها مجاز به استفاده از زبان برنامه نویسی پایتون هستید.
- ۴- در صورت وجود تقلب نمره تمامی افراد شرکت کننده در آن **نمره صفر** لحاظ می‌شود.
- ۵- در صورتی که از منبعی برای هر بخش استفاده می‌شود، حتماً لینک مربوط به آن در گزارش آورده شود. وجود شباهت بین منبع و پیاده‌سازی در صورت ذکر منبع بلامانع است. اما در صورت مشاهده شباهت با مطالب موجود در سایت‌های مرتبط نمره کسر می‌گردد.
- ۶- نتایج و تحلیل‌های شما در روند نمره‌دهی دستیاران آموزشی تأثیرگذار است.
- ۷- لطفاً پاسخ تمرین خود را (به همراه کد/گزارش سوال کامپیوتری) به صورت زیر در صفحه درس آپلود نمایید:
HW[HW number]_[Last_name]_[Student number].zip
- ۸- در صورت وجود هر گونه ابهام یا مشکل می‌توانید از طریق ایمیل با طراحان تمرین در تماس باشید:
 - نیلوفر فریدنی (nilu.1380@gmail.com)
 - امیررضا وفازاده (amirvafazadeh@gmail.com)
 - سید محمد متین آل محمد (sm.matin.alemohammad@gmail.com)