به نام خدا

دانشگاه صنعتی شریف
دانشکده‌ی مهندسی برق

یادگیری عمیق

پاییز ۱۴۰۱

استاد: دکتر فاطمی‌زاده

گردآورندگان: -

تمرین دوم — شبکه‌های پرسپترون، رگولاسیون، بهینه‌سازها — مهلت ارسال: پنج‌شنبه ۳ آذر

- مهلت ارسال پاسخ تا ساعت ۲۳:۵۹ روز مشخص‌شده است.
- در طول ترم امکان ارسال با تاخیر پاسخ همه‌ی تمارین تا سقف ۶ روز و در مجموع ۲۰ روز، وجود دارد. پس از گذشت این مدت، پاسخ‌های ارسال‌شده پذیرفته نخواهند بود. همچنین، به ازای هر روز تأخیر غیر مجاز ۱۰ درصد از نمره تمرین به صورت ساعتی کسر خواهد شد.
- هم‌کاری و هم‌فکری شما در انجام تمرین مانعی ندارد اما پاسخ ارسالی هر کس حتما باید توسط خود او نوشته شده باشد. (دقت کنید در صورت تشخیص مشابهت غیرعادی برخورد جدی صورت خواهد گرفت.)
- در صورت هم‌فکری و یا استفاده از هر منابع خارج درسی، نام هم‌فکران و آدرس منابع مورد استفاده برای حل سوال مورد نظر را ذکر کنید.
- لطفا تصویری واضح از پاسخ سوالات نظری بارگذاری کنید. در غیر این صورت پاسخ شما تصحیح نخواهد شد.
- نتایج و پاسخ های خود را در یک فایل با فرمت zip به نام HW۲-Name-StudentNumber در سایت Quera قرار دهید. برای بخش عملی تمرین نیز لینگ گیت‌هاب که تمرین و نتایج را در آن آپلود کرده‌اید قرار بدهید. دقت کنید هر سه فایل نوتبوک تکمیل شده بخش عملی را در گیت‌هاب قرار دهید .
- لطفا تمامی سوالات خود را از طریق کوئرای درس مطرح بکنید (برای اینکه تمامی دانشجویان به پاسخ‌های مطرح شده به سوالات دسترسی داشته باشند و جلوی سوالات تکراری گرفته شود، به سوالات در بسترهای دیگر پاسخ داده نخواهد شد).
- دقت کنید کدهای شما باید قابلیت اجرای دوباره داشته باشند، در صورت دادن خطا هنگام اجرای کدتان، حتی اگه خطا بدلیل اشتباه تایپی باشد، نمره صفر به آن بخش تعلق خواهد گرفت.

سوالات نظری (۳۰۰ نمره)

۱. (۵۰ نمره) همان‌طور که می‌دانید استفاده از شبکه‌های عمیق می‌تواند بسیار زمان‌بر باشد. یکی از راه‌کارهایی که برای حل این مشکل مورد بررسی قرار گرفته است فشرده‌سازی مدل پس از اتمام فرآیند یادگیری است. تکنیک‌های متفاوتی در این جهت قابل استفاده است و تنک‌سازی یکی از مهم‌ترین آن‌هاست. در این تمرین قصد داریم تا این روش را بررسی کنیم:

(آ) فرض کنید مدل ما با استفاده از گرادیان کاهشی همگرا شده است و حال می‌خواهیم یکی از وزن‌ها را حذف کنیم به طوری که با کم‌ترین افزایش خطا روبه‌رو شویم، به کمک بسط تیلور تا مرتبه ۲ مشخص کنید که کدام یک از وزن‌ها را باید برابر صفر قرار دهیم.

(ب) حال فرض کنید H، ماتریس هسیان، همانی است. عبارتی که در قسمت قبل به دست آوردید را ساده کنید و الگوریتم به دست آمده را توضیح دهید.

۲. (۵۰ نمره) فرض کنید X یک ماتریس داده رتبه کامل $N \times d$ باشد که $(N > d)$ و برچسب‌ها به صورت $y_i = b \cdot x_i + \epsilon_i$ باشد که $\epsilon_i \sim \mathcal{N}(۰, \sigma^۲)$ نویز است.

(آ) با داشتن X و y، $\hat{b}$ را طوری بیابید که خطای تجربی کمینه شود.

(ب) نشان دهید که امید ریاضی خطای تمرین تنها برحسب N, d, σ قابل بیان است. به طور خاص نشان دهید:

$$E\left[\frac{1}{N}\left\|\left(X\left(X^TX\right)^{-1}X^T-I\right)\epsilon\right\|_2^2\right]=\frac{(N-d)}{N}\sigma^2$$

راهنمایی:

- اگر تعریف کنیم $A=X\left(X^TX\right)^{-1}X^T$، آیا A متقارن است؟ A^TA چیست؟ A^2 چطور؟
- مقادیر ویژه‌ی ماتریس متقارنی که در تساوی $A^2=A$ صدق می‌کند چگونه خواهد بود؟
- اگر X رتبه کامل باشد، آنگاه رتبه A چه خواهد بود؟

(ج) با توجه به عبارتی که در بخش قبل به دست آوردید، سعی کنید توضیح دهید چرا زمانی که d به N نزدیک است (زمانی که $overfit$ رخ می‌دهد)، خطای تمرین بسیار کم است.

۳. (۵۰ نمره) مسئله رگرسیون خطی چندگانه را در نظر بگیرید که در آن $Y=X\beta+\epsilon$ که X ماتریس داده، Y بردار پاسخ باشد، و ϵ یک بردار نویز گوسی است ($\epsilon\sim\mathcal{N}(\cdot,\Sigma)$). همچنین تابع خطا MSE است.

(آ) نشان دهید $\hat{\beta}=\left(X^TX\right)^{-1}X^TY$.

(ب) اگر به تابع خطا MSE، یک عبارت رگولاسیون L_2 اضافه کنیم، فرم بسته $\hat{\beta}$ چگونه خواهد شد؟

(ج) تخمین‌گر $\beta^*=\left(X^T\Sigma^{-1}X\right)^{-1}X^T\Sigma^{-1}Y$ را در نظر بگیرید. نشان دهید این تخمین‌گر با $\hat{\beta}$ برابر است، اگر و تنها اگر ماتریس غیر تکینی همچون F باشد که $\Sigma X=XF$.

(د) تابع خطای $L\left(\beta,\lambda_1,\lambda_2\right)=|y-X\beta|^2+\lambda_1\|\beta\|_2^2+\lambda_2\|\beta\|_1$ را در نظر بگیرید. نشان دهید این تابع معادل حالتی است که تعدادی داده به مسئله اصلی اضافه کرده‌ایم و به تابع خطا رگولاسیون L_1 اضافه شده است.

۴. (۵۰ نمره) تابع خطا در یک شبکه با اعمال $Dropout$ گوسی‌ـضربی به شکل $J=(y_d-\sum_{k=1}^{n}\delta_k w_k x_k)^2$ است که در آن $\delta_k\sim\text{Normal}\left(1,\sigma^2\right)$ می‌باشد.

(آ) امید ریاضی گرادیان تابع هدف نسبت به متغیر w_i را محاسبه و تا حد امکان ساده کنید.

(ب) آیا می‌توانید تعبیری از رگولاسیون با استفاده از این نوع $Dropout$ ارائه دهید؟

۵. (۵۰ نمره) تصور کنید که تابع هدف یک مدل یادگیری ماشین به صورت w^THw باشد که اگر از تجزیه مقادیر ویژه استفاده کنیم $H=Q\Lambda Q^T$

(آ) اگر از روش گرادیان کاهشی با طول گام ϵ استفاده کنیم، فرمول یادگیری ضرایب به چه صورت است؟

(ب) با شروع از حالت اولیه w_0 ضرایب در گام t به چه صورت است؟

(ج) تحت چه شرایطی این الگوریتم همگرا می‌شود؟

(د) حال بررسی کنید اگر از روش نیوتن استفاده کنیم، یادگیری به چه صورت خواهد بود؟ چند گام طول می‌کشد تا همگرا شویم؟

(ه) چرا با وجود اینکه روش مرتبه ۲ نیوتن از روش مرتبه ۱ گرادیان کاهشی بسیار سریع‌تر همگرا می‌شود، در آموزش شبکه‌های عمیق از آن استفاده نمی‌شود؟

۶. (۵۰ نمره) شبکه‌ی یادگیری‌ای را در نظر بگیرید که دو ورودی $x_1, x_2\in R^n$ را می‌گیرد و به عنوان خروجی، $h_1=\mathbf{tanh}(Wx_1+b), h_2=\mathbf{tanh}(Wx_2+b)$ که $W\in R^{m\times n}, b\in R^m$ ضرایب شبکه هستند. فرض کنید تابع خطا به صورت $J=|h_1-h_2|_2^2+\|W\|_F^2$ باشد که $\|W\|_F^2=\sum_{i,j}|A_{i,j}|^2$

(آ) این شبکه چه چیزی یاد می‌گیرد؟

(ب) اگر از SGD با سایز $batch$ برابر b استفاده کنیم، معادله‌ی آموزش وزن‌ها و بایاس را به دست آورید.

سوالات عملی (۳۰۰ نمره)

۱. (۱۲۰ نمره) **پیاده‌سازی یک شبکه عصبی از پایه** m.eshtehardian@yahoo.com

در این سوال قصد داریم یک شبکه عصبی را با استفاده از (Pytorch) از پایه پیاده‌سازی نماییم. دیتاستی که در این سوال از آن استفاده می‌شود، دیتاست (Fashion-MNIST) می‌باشد که شامل ۶۰۰۰۰ داده آموزش و ۱۰۰۰۰ داده تست می‌باشد. برای اطلاعات بیشتر در رابطه با این دیتاست می‌توانید به این لینک مراجعه نمایید.

(آ) **اضافه کردن دیتاست و نمایش آن**

در ابتدا باید فایل‌های مربوط به این دیتاست شامل تصاویر و لیبل‌های آموزش و تست را دانلود نمایید و از روی آن یک نمونه از کلاس Dataset ایجاد کنید. برای اینکار می‌توانید به جای دانلود فایل‌های اصلی و پیاده‌سازی نمونه دیتاست گفته شده، از کلاس torchvision.datasets.FashionMNIST استفاده نمایید. برای اطلاعات بیشتر رابطه با نحوه کار با این کلاس می‌توانید به این لینک مراجعه نمایید. پس از اضافه کردن دیتاست، از هر ۱۰ کلاس دیتاست، یک نمونه تصادفی را به همراه لیبل مربوط به آن نمایش دهید.

(ب) **پیاده سازی شبکه**

در ادامه، به پیاده‌سازی شبکه می‌پردازیم. شبکه مورد نظر برای این سوال، یک شبکه fully-connected با تعداد لایه دلخواه خودتان می‌باشد(حداکثر ۵ لایه با احتساب لایه ورودی و خروجی) اما دقت کنید که استفاده از torch.nn ، توابع فعالسازی آماده و همچنین توابع هزینه آماده **مجاز نمی‌باشد** . استفاده از هریک از موارد ذکر شده، منجر به صفر شدن نمره کل سوال خواهد شد.

برای مثال فرض کنید که می‌خواهیم یک شبکه یک لایه با تعداد نورون n در لایه ورودی و تعداد نورون m در لایه خروجی پیاده‌سازی نماییم. همچنین در هر بار ورودی دادن به شبکه، یک batch شامل k داده به آن ورودی داده می‌شود که آن را با X_{Batch} نمایش می‌دهیم و یک تانسور با ابعاد $k * n$ می‌باشد. برای اینکار باید دو tensor با نام‌های weights و bias به ترتیب با ابعاد $n * m$ و $1 * m$ ایجاد نماییم. همچنین هر یک از توابع فعالسازی نظیر ReLU و Softmax که قصد استفاده از آن را داریم را در یک تابع پیاده‌سازی می‌نماییم. در این صورت خروجی این مدل یک لایه برابر با $Y_{Batch} = Activition(X_{Batch}.weigths + bias)$ خواهد بود.

برای پیاده‌سازی شبکه با تعداد لایه بالاتر بایستی برای هر لایه مانند مثال قبل عمل کرده و متغیرهایی برای ذخیره کردن وزن و بایاس هر لایه تعریف نمایید. در نهایت برای این بخش باید موارد زیر پیاده‌سازی شوند:

i. ایجاد تانسورهای وزن و بایاس برای هر لایه
ii. پیاده‌سازی توابع فعالسازی مورد استفاده در مدل
iii. پیاده‌سازی تابعی به نام model که ورودی و خروجی آن متغیرهای xb و yb از نوع tensor می‌باشند.

(ج) **آموزش مدل**

پس از تعریف مدل، لازم است که این مدل آموزش ببیند. برای آموزش مدل، از روش SGD استفاده می‌کنیم به این صورت که در هر epoch ، داده‌های آموزش به batch های متفاوت تقسیم می‌شوند و برای هر batch ، خروجی مدل محاسبه و سپس با اعمال یک تابع هزینه بر روی خروجی مدل و خروجی داده، هزینه مورد نظر محاسبه می‌شود. لازم به ذکر است که پیاده‌سازی تابع هزینه نیز باید **توسط خودتان باشد** و مجاز به استفاده از توابع آماده موجود نظیر توابع موجود در torch.nn **نمی‌باشید** .پس از محاسبه هزینه هر batch در هر epoch ، با اسفاده از تابع backward، گرادیان پارامترهای مدل که همان وزن‌ها

و بایاس‌های تعریف شده در بخش ۱ می‌باشند را محاسبه می‌نمایید. در نهایت با مشخص کردن یک learning rate ثابت، برای هر batch در هر epoch ، مقادیر وزن‌ها و بایاس‌ها را برابر با مقادیر جدید (با توجه به رابطه SGD) قرار می‌دهید. بدیهی است که در این بخش نیز استفاده از torch.optim مجاز نمی‌باشد. همچنین در هر epoch لازم است که دقت مدل آموزش داده شده بر روی هر دو دیتاست آموزش و تست چاپ شود.

(د) **تست مدل و نمایش نتایج**

پس از تکمیل آموزش مدل، دقت مدل خود بر روی داده تست را به دست آورده و گزارش نمایید. دقت نهایی شما باید مقدار قابل قبولی (حدود ۸۰ درصد روی داده های تست مورد انتظار می باشد)داشته باشد، در غیر این صورت نمره‌ای به این بخش تعلق نخواهد گرفت.
همچنین برای ۹ نمونه تصادفی از دیتاست، تصویر مربوطه را به همراه لیبل واقعی و لیبل پیش‌بینی شده توسط مدل خود نمایش دهید.

۲. (۱۰۰ نمره) **پیشبینی مسابقات گروهی جام جهانی** amiroo23jf@gmail.com

هدف این مساله پیشبینی نتایج تیم ملی ایران در مسابقات دور گروهی جام جهانی قطر ۲۰۲۲ می باشد. برای این منظور از داده های موجود در فایل international_matches.csv استفاده نمایید. این فایل تمامی مسابقات ملی برگزار شده از سال ۱۹۹۳ تا ۲۰۲۲ را در برمی گیرد و شامل ستون های تاریخ، نام تیم های میزبان و میهمان، قاره تیم های میزبان و میهمان، رتبه تیم های میزبان و میهمان در فیفا، امتیاز تیم های میزبان و میهمان در فیفا، تورنمنتی که مسابقه در آن صورت گرفته است، شهر و کشور محل برگزاری مسابقه، نتیجه مسابقه و ... است. همچنین برای پیاده سازی های این سوال باید از توابع موجود در کتابخانه pytorch استفاده کنید.

(آ) **بارگذاری داده ها در پایتون**

داده های موجود در فایل csv را با استفاده از کتابخانه pandas بارگذاری کنید و ۱۰ داده آخر آن را نمایش دهید.

(ب) **رسم نقشه پراکندگی داده ها**

با استفاده از کتابخانه matplotlib نمودار پراکندگی (scatter) داده ها را با استفاده از تنها دو ویژگی home_team_fifa_rank و away_team_fifa_rank رسم کنید و نتیجه بازی (برد، باخت یا مساوی تیم میزبان) را با سه رنگ متفاوت روی شکل نشان دهید.

(ج) **مرتب سازی داده ها**

با استفاده از ویژگی های home_team_total_fifa_points، home_team_fifa_rank، away_team_fifa_rank و away_team_total_fifa_points میخواهیم نتیجه بازی (ستون home_team_result) را پیشبینی کنیم. برای اینکار ابتدا داده هایی که در آنها امتیازهای تیم ها صفر لحاظ شده است را حذف نمایید. از بقیه بازی ها ۷۵ درصد را برای train باقی داده ها را برای test قرار دهید.

(د) **نمایش همبستگی ویژگی ها**

با استفاده از کتابخانه seaborn و تابع heatmap میزان همبستگی (correlation) میان ویژگی های انتخاب شده را رسم نمایید.

(ه) **طراحی معماری شبکه**

با استفاده از nn.Module یک شبکه با سه لایه پنهان (hidden layer) تعریف کنید و از تابع (relu) به عنوان activation function استفاده نمایید. همچنین لایه پنهان نخست را دارای ۱۰ ورودی، لایه پنهان دوم را دارای ۲۰ ورودی و لایه پنهان سوم را دارای ۸ ورودی در نظر بگیرید.

(و) **تست کردن مدل**

مدلی که در بخش قبل تعریف کردید با train کنید و دقت آن را روی test set دست آورید. حال با تغییر مدل به دلخواه (حتی میتوانید از ویژگی های دیگر موجود در فایل csv نیز استفاده کنید) تلاش کنید تا دقت مدل روی test set را افزایش دهید و این مقدار را گزارش دهید. حال دقت مدل را خود را روی test set بدست آورید و آن را گزارش کنید. (باید دقت شما بالای ۵۵ درصد باشد.)

* توجه داشته باشید که رسیدن به دقت ۵۵ درصد برای گرفتن نمره این سوال کافی می باشد و نیازی به تلاش برای رسیدن به درصد های بالاتر وجود ندارد.

(ز) **محاسبه نتیجه**
با استفاده از نتیجه بخش قبل، احتمال برد ایران در هر یک از بازی های دور گروهی را پیشبینی کنید. میتوانید مشخصات ایران و هم گروهی های آن را در این لینک مشاهده کنید.

۳. (۸۰ نمره) **الفبای لاتین** Arhp2000@gmail.com

در این سوال قصد داریم که شبکه ایی برای تشخیص حروف لاتین با توجه به حرکت دست انسان آموزش دهیم. دو فایل csv برای آموزش و تست در اختیار شما قرار گرفته است. ستون اول در هر فایل نشان دهنده ی برچسب آن سطر می باشد و ۷۸۴ ستون بعدی، نشان دهنده مقادیر پیکسل های هر تصویر می باشد. برچسب هر کلاس هم به این صورت می باشد که عدد ۰ معادل کلاس A ، عدد ۱ معادل کلاس B و به همین ترتیب عدد ۲۴ معادل کلاس Y می باشد. توجه داشته باشید که تصاویر بصورت تک کانال می باشند.

(آ) ابتدا ۹ داده رندوم از دیتاست Q3_train.csv بعد از استاندارد سازی در قالب تصویر به همراه کلاس هر تصویر نمایش دهید.

(ب) حال داده های validation را از داده های train جدا کرده و سپس یک شبکه MLP را آموزش دهید. می توانید از تابع هزینه cross entropy برای آموزش استفاده کنید. در انتخاب تعداد لایه و تعداد نورون های هر لایه آزاد هستید. همچنین استفاده از کتابخانه های tensorflow یا pytorch یا keras بلامانع می باشد.
آموزش را با استفاده از دو الگوریتم ADAM و SGD انجام دهید. تفاوت این دو الگوریتم را ذکر کنید و بگویید در کدام حالت نتیجه بهتری حاصل می شود. همچنین تغییرات Loss را در هر کدام از حالت ها هنگام آموزش شبکه نشان دهید.

(ج) حال لایه dropout را به شبکه خود اضافه کنید و سپس آموزش دهید. تاثیر آن را بر عملکرد شبکه روی داده های آموزش و تست بیان کنید.

(د) مشخصات و ویژگی های بهترین شبکه ایی را که در قسمت های قبل آموزش دادید را بیان کنید (حداقل درصد قابل قبول برای گرفتن نمره بخش عملکرد این سوال ۷۵ درصد روی داده های تست می باشد).

(ه) (امتیازی / به اندازه ۲۰ درصد نمره این سوال از تمرین) حال این بهترین شبکه را انتخاب کنید. برای تست شبکه سه ورودی را با استفاده از وبکم لب تاب خودتان به شبکه بدهید و نتیجه آن را نمایش دهید. می توانید اینکار را بصورت آنلاین(یعنی با استفاده از کتابخانه های لازم بصورت انلاین وقتی که دست شما مقابل وبکم قرار می گیرد این تشخیص صورت بگیرد) یا آفلاین(تصویر دست خودتان را بگیرید و سپس به شبکه برای تشخیص بدهید) انجام دهید. توجه کنید که تصویری که به شبکه می دهید باید نرمال شده باشد. همچنین دقت کنید که بک گراند تصویرتان ساده و بدون شی اضافه ایی باشد تا نتیجه بهتری حاصل بشود.