

به نام خدا



دانشکده‌ی مهندسی کامپیوتر

پاییز ۱۴۰۲

تمرین ۳: شبکه‌ی عصبی و لاجستیک رگرسیون

مدرس: مهدی جعفری سیاوشانی

- سررسید بخش تئوری این تمرین چهارشنبه ۸ آذر ماه ساعت ۵۹ : ۲۳ است.
- سررسید بخش عملی این تمرین جمعه ۱۰ آذر ماه ساعت ۵۹ : ۲۳ است.
- در صورت کشف تقلب، بار اول برای افراد درگیر تقلب، نمره‌ی همان سوال(های) خاص صفر در نظر گرفته می‌شوند. در صورت تکرار، نمره کل تمرین صفر در نظر گرفته می‌شود و در صورت تکرار، درس برای افراد حذف خواهد شد.
- تمامی پاسخ‌های خود را در یک فایل با فرمت (HW2-[SID]-[Fullname].zip (.pdf) روی کوئرا قرار دهید.

پرسش‌ها

۱ بخش نظری (۶۰ نمره)

۱.۱ پرسش اول (۱۰ نمره)

فرض کنید ما داده‌ی مربوط به دسته‌ای از دانشجویان درس جبرخطی را جمع‌آوری کرده‌ایم به طوری که این داده شامل دو ویژگی می‌شود. ویژگی اول، x_1 ، برابر مجموع ساعاتی است که دانشجو به مطالعه‌ی جبرخطی پرداخته است. ویژگی دوم، x_2 ، معدل کل نمرات دانشگاه دانشجو تا قبل از اخذ درس جبرخطی است. حال قصد داریم با استفاده از یک مدل logistic regression با داشتن ساعت مطالعه و معدل یک دانشجو، پیش‌بینی کنیم که احتمال اینکه در این درس نمره ۲۰ بگیرد چقدر است. پس از یادگیری بر اساس این داده‌ها، ضرایب بدست آمده برابر $\beta_0 = -5$ و $\beta_1 = 0.1$ و $\beta_2 = 0.25$ است.

۱. احتمال اینکه یک دانشجو با ۸۰ ساعت مطالعه و معدل ۱۸ بتواند در این درس نمره‌ی ۲۰ بگیرد را محاسبه کنید. (۵ نمره)

۲. دانشجوی دیگری را در نظر بگیرید که معدلش ۱۶ است و می‌خواهد در این درس نمره‌ی ۲۰ کسب کند. طبق مدلی که آموزش داده‌ایم، او چند ساعت باید جبرخطی تمرین کند تا با احتمال ۹۰ درصد بتواند به این نمره دست یابد؟ (۵ نمره)

۲.۱ پرسش دوم (۲۰ نمره)

در logistic regression، می‌توان مسئله را به صورت یک متغیر تصادفی برنولی به نام Y به شرط بردار ویژگی‌ها، x ، دید:

$$p(Y = 1|x) = \frac{1}{1 + e^{-x \cdot w}}$$

در اصل به این کلاسه‌بندی binary logistic regression نیز می‌گوییم. حال فرض کنید به جای اینکه Y برنولی باشد، بتواند K مقدار مختلف به خودش بگیرد. بنابراین احتمال شرطی این متغیر تصادفی را به صورت زیر تغییر می‌دهیم:

$$p(Y = k|x) = \frac{e^{x \cdot w_k}}{\sum_{j=1}^K e^{x \cdot w_j}}$$

که در این تعریف، x مجدداً بردار ویژگی‌های ما به طول d است و بردار w_j نیز یک بردار به طول d است.

۱. در نظر بگیرید که N داده را به صورت جفت‌های (x_i, y_i) مشاهده کرده‌ایم. $X = \{(x_i, y_i) | i \in \{1, \dots, N\}\}$ حال log-likelihood این مجموعه داده را محاسبه کنید. در اصل باید $\log(P(Y|X))$ را محاسبه کنید. این تابع در حقیقت تابع هدف ما برای یادگیری است. (۸ نمره)

۲. حال فرض کنید که یک L2 regularizing term به این تابع هدف اضافه کنیم:

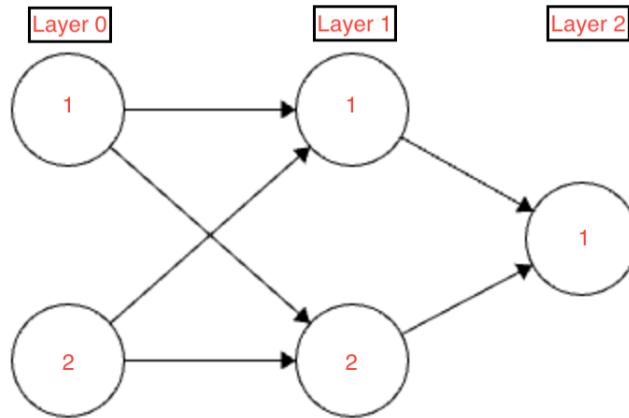
$$f(X) = \log(P(Y|X)) - \lambda \sum_{k=1}^K \|w_k\|^2$$

حال گرادینان تابع $f(X)$ را نسبت به w محاسبه کنید. (۸ نمره)

۳. اگر بردار وزن‌های کلاس k در مرحله t برابر $w_k^{(t)}$ باشد، نحوه‌ی آپدیت شدن w_k و بدست آوردن $w_k^{(t+1)}$ را بنویسید. در حقیقت با توجه به اینکه نرخ یادگیری برابر η است، update rule را برای مجموعه داده‌های X با توجه به جواب بخش قبلی بنویسید. (۴ نمره)

۳.۱ پرسش سوم (نمره ۳۰)

شبکه‌ی عصبی ۲ لایه‌ی زیر را در نظر بگیرید.



حال وزن یال بین سلول زام i لایه‌ی i و سلول k ام لایه‌ی $i-1$ را با $W_{j,k}^{(i)}$ نشان می‌دهیم. همچنین مقدار بایاس برای سلول‌های لایه‌ی i ام را با $b^{(i)}$ نشان می‌دهیم. بنابراین خروجی خطی سلول k ام لایه‌ی i را به صورت زیر تعریف می‌کنیم:

$$z_j^i = \left(\sum_k W_{j,k}^i a_k^{i-1} \right) + b_i$$

بردار کل خروجی‌های قبل از فعال‌سازی (activation) لایه‌ی i ام را به صورت زیر نشان می‌دهیم:

$$\mathbf{z}^{(i)} = \mathbf{W}^{(i)} \mathbf{a}^{(i-1)} + \mathbf{b}^{(i)}.$$

پس از اعمال تابع اکتیویشن روی z_j^i خروجی زام غیرخطی لایه‌ی i ام به صورت a_j^i نشان می‌دهیم: $a_j^{(i)} = \sigma(z_j^{(i)})$. برای حفظ یکپارچگی در تعریف مسئله‌ی خود، خروجی‌های غیرخطی لایه‌ی صفر را همان ورودی‌های مدل در نظر می‌گیریم، $a^{(0)} = \mathbf{x}$ و $a^{(1)} = \hat{y}$. تابع فعال‌سازی مورد استفاده در این مدل نیز تابع soft rectifier به صورت $\sigma(z) = \ln(1 + e^z)$ است که در اصل یک تقریب از تابع ReLU است. در نهایت تابع loss مدل خود را به صورت

$$\ell(y, \hat{y}) = \frac{1}{2} \|y - \hat{y}\|_2^2$$

تعریف می‌کنیم به طوری که y در آن لیبیل هر دیتاپوینت در دیتاست ما است. فرض کنید مقادیر وزن‌های یال‌های زیر را داریم:

$$\mathbf{W}_{1,1}^{(1)} = -0.5, \mathbf{W}_{1,2}^{(1)} = 0.5, \mathbf{W}_{2,1}^{(1)} = -0.8, \mathbf{W}_{2,2}^{(1)} = 0.8, \mathbf{W}_{1,1}^{(2)} = 1, \mathbf{W}_{2,1}^{(2)} = -1,$$

علاوه بر آن فرض کنید بایاس‌ها نیز دارای مقادیر زیر هستند:

$$b^{(1)} = 0.1, b^{(2)} = 0.2,$$

همچنین فرض کنید که ورودی زیر را در اختیار داریم:

$$\mathbf{x} = [0.4, 0.6]$$

۱. حال مقدار $a_1^{(1)}$ و $a_1^{(2)}$ و در نهایت خروجی نهایی، یعنی $\hat{y} = a_1^{(2)}$ را برای مقادیر ورودی و وزن‌های داده‌شده در صورت سوال محاسبه کنید. (۱۰ نمره)
۲. تابع اکتیویشنی که برای این مدل در نظر گرفته‌ایم، با نام soft rectifier، در صورتی که ورودی بزرگی در محاسبات کامپیوتری داشته باشد، overflow خواهد کرد. آن را به صورتی بازنویسی کنید که این اتفاق رخ ندهد. (۴ نمره)
۳. الگوریتم backpropagation مشتق‌های پاره‌ای تابع loss را نسبت به تک‌تک وزن‌ها به صورت $\frac{\partial \ell}{\partial W_{j,k}^i}$ محاسبه می‌کند تا وزن $W_{j,k}^i$ را به کمک gradient decent بهینه‌سازی کند. ابتدا $\frac{\partial \ell}{\partial a_1^{(2)}}$ و $\frac{\partial a_1^{(2)}}{\partial z_1^{(2)}}$ و $\frac{\partial z_1^{(2)}}{\partial W_{j,1}^{(2)}}$ را بر حسب y ، $a_j^{(1)}$ و $a_1^{(2)}$ محاسبه کنید. (۱۰ نمره)
۴. با در نظر گرفتن $y = 1, \eta = 1$ مقدار $\frac{\partial \ell}{\partial W_{1,1}^{(2)}}$ را محاسبه کنید و $W_{1,1}^{(2)}$ را به‌روزرسانی کنید. (۴ نمره)
۵. برای به‌روزرسانی $W_{1,1}^{(1)}$ مقدار $\frac{\partial \ell}{\partial W_{1,1}^{(1)}}$ را با $\eta = 1$ محاسبه کنید. (۲ نمره)

۲ بخش عملی (۶۰ نمره)

۱.۲ Logistic Regression (۲۵ نمره)

در این سوال روی دیتاستی کار می‌کنید که هدف نهایی آن تعیین مثبت یا منفی بودن تعلق بیمه به یک متقاضی است. یعنی تعدادی متغیر مستقل وجود دارد مانند درآمد متقاضی، نوع شغل متقاضی و ... که با استفاده از آنها باید مشخص کنید که آیا بیمه باید به متقاضی تعلق بگیرد یا خیر. نخست باید با داده آشنا شوید و EDA را روی آن اجرا کنید. سپس باید پیش‌پردازش‌هایی روی داده انجام دهید تا آماده استفاده شود و در نهایت تنها با استفاده از numpy به پیاده‌سازی Logistic Regression بپردازید و برای این کار باید بخش‌های loss و محاسبه مشتق loss و fit کردن داده به مدل را کامل کنید که تمامی این موارد داخل jupyter notebook مربوطه مشخص شده است و می‌توانید طبق آن جلو بروید.

۲.۲ MultiLayer Perceptron(MLP) (۳۵ نمره)

هدف از این سوال پیاده‌سازی شبکه‌ی MLP به کمک Numpy می‌باشد. شما باید بخش‌های مشخص‌شده در نوت‌بوک را طبق توضیحات تکمیل کنید. در ابتدا کد مربوط به پیاده‌سازی ماژول‌های استفاده‌شده در یک شبکه‌ی عصبی چند لایه (Linear, Sigmoid, Binary cross-entropy loss) را تکمیل خواهید کرد. سپس با کمک ماژول‌های پیاده‌سازی شده، یک شبکه‌ی MLP ۲ لایه تعریف کرده و آن را روی داده‌های دیتاست MNIST آموزش داده و آن را ارزیابی می‌کنید. برای راحت تر شدن کار، در نوت بوک خواسته شده که تنها اعداد ۳ و ۷ را از دیتاست جدا کرده و شبکه را تنها برای دسته‌بندی این دو دسته عدد آموزش دهید.

موفق باشید