یادگیری ماشین برای بیوانفورماتیک

نیم‌سال دوم ۹۸ـ۹۹

دانشکده‌ی مهندسی کامپیوتر
متین خواجوی ـ سروش عمران‌پور ـ احسان هنری

تمرین پنجم
یادگیری عمیق ـ مدل پنهان مارکوف

تئوری

مسئله‌ی ۱. Hidden Markov Models

میخواهیم وضعیت سلامت روحی یک فرد عادی را برحسب فعالیتش در بازه های زمانی مختلف با HiddenMarkovModel مدلسازی کنیم. فرض کنید که وضعیت سلامت روحی فرد (متغیرهای پنهان مساله) "سالم" و "مریض" باشد. فعالیت‌های فرد (مشاهدات)شامل "خوابیدن:S"، "غذا خوردن:E" و "ورزش کردن:A" هستند. فرض کنید چهار دنباله از رفتار فرد در زمان های مختلف به صورت "S،A،E" و "A،E،E" و "S،S،E" و "E،A،E" مشاهده شده است. با استفاده از الگوریتم forward-backward مدل بهینه برای وضعیت فرد را پیدا کنید.
پس از بدست آوردن توزیع های مدل، برای اطمینان از صحت مدل خود می‌توانید از کتابخانه hmmlearn در پایتون استفاده کنید.

Deep Learning

مسئله‌ی ۲. Loss Function

در این مسئله به بررسی خواصی از توابع هزینه می‌پردازیم. فرض کنید خروجی شبکه بردار $\hat{y}$ است.(که برداری است به اندازه‌ی تعداد دسته‌ها، و هر درایه‌ی آن نشان دهنده‌ی احتمال آن دسته است) و بردار y نشان دهنده‌ی دسته‌ی درست داده است که به صورت one-hot کد شده است. دو تابع هزینه را در نظر می‌گیریم:

$$Loss_{Cross-Entropy}(y, \hat{y}) = -\sum_i y_i log(\hat{y}_i)$$

$$Loss_{MSE}(y, \hat{y}) = \frac{1}{n}\sum_i (y_i - \hat{y}_i)^2$$

که هزینه‌ی کل، جمع تابع هزینه روی تمام داده‌ها خواهد بود. همچنین فرض می‌کنیم خروجی لایه‌ی آخر از تابع Softmax رد شده است:

$$\hat{y} = softmax(z) \qquad \hat{y}_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

الف) حساسیت تابع Softmax نسبت به یکی از خروجی ها (مانند z_i) را به‌دست آورید و نمودار آن را رسم کنید. منظور از حساسیت این است که سایر خروجی‌ها (سایر z_jها) را ثابت بگیرید و تغییرات بردار $\hat{y}$ را نسبت به z_i محاسبه کنید.

ب) توضیح دهید چرا استفاده از Softmax بهتر از حالتی است که هر z_i را مستقلا از فعالسازی مانند سیگموید عبور دهیم. یعنی داشته باشیم:

$$\hat{y}_i = \sigma(z_i)$$

پ) نشان دهید اگر به تمام ورودی‌های Softmax مقدار ثابتی اضافه شود، احتمال‌ها تغییری نمی‌کنند.

ت) هنگام محاسبه‌ی تابع نمایی در Softmax ممکن است گاهی overflow اتفاق بیفتد. با توجه به قسمت قبل بگویید که چگونه می‌توان از این اتفاق جلوگیری کرد.

ث) نشان دهید که کمینه کردن تابع هزینه‌ی Cross-Entropy روی تمام داده ها معادل با بیشینه کردن Log Likelihood مربوط به خروجی دسته‌ی درست است.

ج) هر دو تابع هزینه را بر حسب z_iها بازنویسی کنید.

چ) برای هر دو تابع هزینه $\frac{\partial L}{\partial z}$ را بر حسب y و $\hat{y}$ محاسبه کنید.

مسئله‌ی ۳. Universal Approximation (امتیازی)

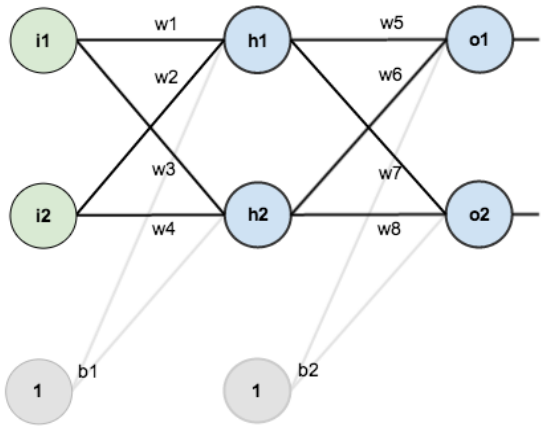
می‌خواهیم تابع $f(x) = \sin(\pi x)$ را در بازه‌ی $-1 \leqslant x \leqslant 1$ به صورت تکه‌ای خطی تقریب بزنیم.(با حداقل تعداد خط)

الف)عبارت ریاضی این تقریب را بنویسید.

ب) یک شبکه‌ی عصبی مناسب (با تعیین مقدار پارامترها و توابع فعال‌ساز مناسب) برای پیاده‌سازی این تقریب پیشنهاد نمایید.

مسئله‌ی ۴. Back-propagation

در این مسئله قصد داریم یک مرحله از الگوریتم backpropagation را به صورت دستی پیاده‌سازی کنیم. شبکه مورد نظر ما دارای ۲ ورودی ، یک لایه‌ی نهان با ۲ سلول و ۲ خروجی است.
تابع فعال‌ساز مورد استفاده نیز تابع سیگموید است.



الف) ورودی و وزن‌های اولیه را به صورت زیر در نظر بگیرید و خروجی شبکه را محاسبه کنید.

$$[w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8] = [0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5]$$

$$[b_1, b_2] = [0.3, 0.5] \qquad [i_1, i_2] = [0.05, 0.15]$$

ب) حال فرض کنید خروجی مطلوب برابر $[t_1, t_2] = [0.1, 0.9]$ است. با درنظرگرفتن $\text{learning rate} = 0.4$ و یک بار اعمال الگوریتم backprop وزن‌های جدید را به دست‌آورید.
خطا را از رابطه‌ی $E = \frac{1}{2}[(o_1 - t_1)^2 + (o_2 - t_2)^2]$ محاسبه کنید.

مسئله‌ی ۵. Back-propagation in CNN

در حالت کلی کانولوشن ورودی X با سایز (N, C, i_h, i_w) که در آن N تعداد نمونه‌ها در ورودی W با سایز (F, C, i_h, i_w) و $stride = s$ و خروجی O با سایز (N, F, O_h, O_w) را می‌دهد و برای یک نمونه b و فیلتر f به صورت زیر محاسبه می‌شود:

$$O(b, f, i, j) = \sum_{r=0}^{C-1} \sum_{k=0}^{f_h-1} \sum_{l=0}^{f_w-1} W(f, r, k, l) X(b, r, s \times i + k, s \times j + l)$$

$$O_w = \frac{i_w - f_w}{s} + 1$$

$$O_h = \frac{i_h - f_h}{s} + 1$$

الف) رابطه گرادیان تابع هزینه اسکالر L نسبت به فیلتر W را بدست آورید. (مشتق تابع هزینه نسبت به یک متغیر اسکالر $W(f', c', k', l')$ را محاسبه کنید.)

ب) رابطه گرادیان تابع هزینه اسکالر L نسبت به ورودی X را بدست آورید. (مشتق تابع هزینه نسبت به یک متغیر اسکالر $X(b', c', k', l')$ را محاسبه کنید.)

مسئله‌ی ۶. Activation Function

در این سوال به تابع های فعال‌سازی sigmoid و tanh و relu می‌پردازیم.

الف) با بررسی مقادیر خود توابع و مشتق آن‌ها معایب و مزایای هر یک را بنویسید.

ب) این توابع را از نظر هزینه محاسباتی مقایسه کنید.

پ) این توابع را از نظر مشکل vanishing gradient مقایسه کنید.

مسئله‌ی ۷. Regularization

الف) توضیح دهید، چگونه drop-out مانند منظم‌ساز یا regularizer عمل می‌کند.

ب) توضیح دهید، چگونه drop-out عملکردی شبیه ensemble-learning دارد.

پ) آیا عملکرد drop-out در مرحله‌ی test و train یکسان است؟ توضیح دهید.

ت) توضیح دهید، چگونه batch-normalization مانند regularizer عمل می‌کند.

ث) در فرآیند آموزش یک شبکه با لایه‌ی max-pooling ،آیا موقعیت سلول دارای بیشترین مقدار باید ذخیره شود یا خیر؟ چرا؟

مسئله‌ی ۸. CNN

دو شبکه‌ی کانولوشنی زیر را برای دسته‌بندی ۱۰۰ کلاسه تصاویر در نظر بگیرید:

Network 1

$$16 \times conv(5 \times 5) \rightarrow 32 \times conv(5 \times 5) \rightarrow max\ pooling(5 \times 5, stride = (2,2)) \rightarrow$$
$$32 \times conv(5 \times 5) \rightarrow 64 \times conv(5 \times 5) \rightarrow max\ pooling(5 \times 5, stride = (2,2)) \rightarrow$$
$$64 \times conv(5 \times 5) \rightarrow 128 \times conv(5 \times 5) \rightarrow max\ pooling(5 \times 5, stride = (2,2)) \rightarrow$$
$$FC(1024) \rightarrow FC(100)$$

Network 2

$$128 \times conv(5 \times 5) \rightarrow 64 \times conv(5 \times 5) \rightarrow max\ pooling(5 \times 5, stride = (2,2)) \rightarrow$$
$$64 \times conv(5 \times 5) \rightarrow 32 \times conv(5 \times 5) \rightarrow max\ pooling(5 \times 5, stride = (2,2)) \rightarrow$$
$$32 \times conv(5 \times 5) \rightarrow 16 \times conv(5 \times 5) \rightarrow max\ pooling(5 \times 5, stride = (2,2)) \rightarrow$$
$$FC(1024) \rightarrow FC(100)$$

توجه کنید که در هر دو ساختار لایه‌های conv و FC دارای بایاس هستند و هر کدام دارای تابع فعالسازی می‌باشند. همچنین در لایه‌های pooling max و conv در هنگام اعمال فیلترها، قسمتی از حاشیه ورودی که که کاملا در ابعاد فیلتر نمی‌گنجد به دور ریخته می‌شود.(یعنی padding نداریم . معادل padding = valid در تنسورفلو) تمامی conv ها نیز با گام ۱ صورت می‌گیرند.

الف) در صورتی که ابعاد تصویر اولیه $N \times N$ ،سایز فیلتر اعمال شده برابر $F \times F$ و گام (Stride) برابر S باشد، سایز خروجی لایه را محاسبه کنید.

ب) تعداد پارامترهای هر لایه در هر دو ساختار را محاسبه کنید و مشخص کنید کدام لایه‌ها پارامتر بیشتری به شبکه تحمیل می‌کنند. (در ادامه تصویر ورودی شبکه را ۲۲۴ × ۲۲۴، grayscale در نظر بگیرید)

ت) سایز تصویر خروجی هر لایه در دو ساختار را حساب کنید.

ث) حال با توجه به اینکه سایز تصاویر، معیاری از میزان حافظه مورد استفاده در مسیر forward است، تعیین کنید که کدام یک از دو ساختار بالا مناسب‌تر هستند.

عملی

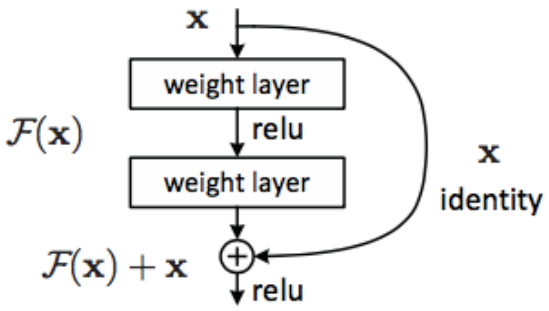
مسئله‌ی ۹. MLP

با الهام از سوال ۴ که در آن الگوریتم backpropagation را به صورت دستی پیاده‌سازی کرده‌اید یک شبکه mlp دولایه را پیاده سازی کنید که دارای ۴ ورودی، یک لایه نهان با ۳ سلول و تابع فعال‌ساز سیگموید و درنهایت ۳ خروجی است. تابع فعال‌سازی لایه آخر را softmax درنظر بگیرید. این شبکه را روی دیتاست iris از کتابخانه sklearn پایتون آموزش دهید.

نکته: برای پیاده سازی از کتابخانه‌های tensorflow numpy، pytorch، میتوانید استفاده کنید. دقت کنید که نمی‌توانید از بهینه‌ساز های آماده pytorch یا tensorflow استفاده کنید. همچنین روند تغییرات مقدار تابع هزینه در هر epoch را نمایش دهید. از آنجایی که در این سوال تنها صحت پیاده‌سازی مهم است، مقدار نهایی تابع هزینه اهمیتی ندارد و صرفا نشان‌دادن یادگیری مدل کافی‌ست.
برای این سوال از دفترچه ی موجود در پوشه سوالات به نام mlp.ipynb استفاده کنید و آنرا کامل نمایید.

مسئله‌ی ۱۰. CNN

در این سوال قصد داریم یک شبکه دسته‌بند CNN با معماری residual پیاده سازی کنیم. این معماری در سال ۲۰۱۵ در این مقاله معرفی شد. این معماری در رقابت Imagenet در سه تسک مختلف توانست مقام اول را کسب کند. در شبکه‌های خیلی عمیق با افزایش تعداد لایه‌هایی که پشت سر هم قرار گرفته اند، بهینه‌سازی مدل بسیار سخت شده و درواقع لایه‌های ابتدایی تغییر عمده‌ای در وزن‌هایشان دریافت نمی‌کنند. این مساله که به degradation معروف است باعث افزایش نرخ خطا می‌شود. برای رفع این مشکل که روند یادگیری مدل را بسیار کند میکند از skip-connection استفاده می‌شود.



همانطور که در شکل بالا می‌بینید در یک بلوک residual علاوه بر لایه‌های convolutional یک نگاشت همانی از ورودی درنظر می‌گیریم و با خروجی لایه‌ها جمع می‌کنیم. و درنهایت تابع فعال‌سازی را روی آن اعمال می‌کنیم. حال شما با استفاده از یکی از فریم‌ورک‌های tensorflow یا pytorch یک شبکه resNet را پیاده سازی کنید.

جزییات مدل:

مدل از بلوک‌های residual تشکیل شده که پشت سر هم قرار گرفته اند و درنهایت یک لایه ی fully-connected با ۱۰ سلول وجود دارد. در هر بلوک residual دو لایه convolution قرار دارد که اندازه فیلتر و تعداد کانال‌های یکسان دارند و به صورت پشته قرار گرفته اند. در صورتی که برای skip-connection با مشکل یکسان نبودن ابعاد ورودی و خروجی برای عمل جمع مواجه شدید از یک لایه convolution به عنوان downsampling استفاده کنید. بجز لایه آخر که تابع فعال‌ساز softmax دارد در بقیه قسمت‌ها از relu استفاده کنید. مدل شما از سه بلوک با مشخصات گفته شده تشکیل شده است که به ترتیب دارای فیلتر با اندازه ۷، ۵ و ۳ هستند. برای کنترل تعداد پارامترهای مدل خود از stride استفاده کنید. stride را تنها بر اولین لایه convolution هر بلوک(ودر صروت نیاز بر لایه ی downsampling) اعمال کنید. بهتر است مقدار stride هرلایه را برابر یکی از مقسوم‌علیه های اندازه ورودی لایه قرار دهید. تعداد کانال های خروجی هر لایه convolution را برابر توانی از ۲ قرار دهید. (برای مثال به ترتیب ۸، ۱۶ ، ۳۲ ، ۶۴، ...)

نکات پیاده‌سازی :

۱. مدل خود را روی دیتاست tumor اموزش دهید. این دیتاست در اختیار شما قرا خواهد گرفت متاسفانه تعداد سمپل‌های این دیتاست برای آموزش مدل شما کافی نیست به همین دلیل لازم است با data-augmentation دیتاست خود را بزرگ‌تر کنید. برای این کار ابتدا با استفاده از تابع نمونه‌ای که در اختیار شما قرار داده می‌شود (crop.py) عکس‌ها را کراپ کنید. سپس با استفاده از های‌transform معمول عکس تعداد سمپل‌هارا افزایش داده و سمپل‌های جدید تولید کنید و در دسته مناسب قرار دهید. برای این کار می‌توانید هم از transform های آماده موجود در torchvision استفاده کنید یا این که خودتان transform جدید تعریف کنید.

۲. پس از افزایش داده‌ها آنرا به سه دسته ،train validation و test تقسیم کنید و به ازای هر epoch مقدار تابع هزینه و دقت را روی train و validation گزارش کنید. تابع هزینه را binary-cross-entropy قرار بدهید .مدل را حداکثر برای ۱۰۰ epoch آموزش دهید.

۳. کلاس خود مدل را به تنهایی در یک فایل به نام resnet.py قرار داده و در نوت‌بوک خود آنرا import کنید. این کار به خوانایی کد شما کمک شایانی می‌کند. برای خوانایی بیشتر کلاس dataset را هم در فایلی جداگانه به نام dataset.py قرار دهید و انرا import کنید.

۴. پس از طی کردن این مراحل در قالب یک گزارش کوتاه به سوالات زیر پاسخ داده و برای هرکدام از توضیح کافی و نمودار مناسب استفاده کنید.

الف) تاثیر قرار دادن batch normalization را در هر بلوک بررسی کنید. برای افزودن این لایه به هر بلوک آنرا بلافاصله پس از لایه ی convolution و قبل از relu قرار دهید.

ب) تاثیر قرار دادن dropout را در هر بلوک بررسی کنید. برای افزودن این لایه به هر بلوک آن را پس از relu قرار دهید. پارامتر dropout را که درواقع احتمال خاموش شدن هر نورون است بین ۰/۲ تا ۰/۴ قرار دهید.

پ) نمودار نقشه ویژگی‌ها (feature map) را برای یکی از لایه های ابتدایی و یکی از لایه‌های انتهایی مدل نمایش داده و ویژگی های یادگرفته شده را مقایسه کنید. هر کدام از این لایه‌ها چه وظیفه‌ای را در پردازش عکس ورودی برعهده دارند؟ (برای مثال پیدا کردن لبه‌ها، سایه‌ها، ...)
برای اطلاعات بیشتر درمورد visualization از این لینک استفاده کنید.

****در هر دو سوال کدهای پیاده‌سازی را در اسکریپت پایتون بزنید و در فایل جوپیتر import کنید.**

****برای کسب نمره کامل سوال ۲ عملی باید به حداقل دقت ۸۵ برسید.**

نکات مهم

- پاسخ بخش تئوری را در یک فایل pdf با اسم HW5-STD-NUM قرار دهید. برای هر سوال عملی یک پوشه بسازید و فایل‌های مربوط به سوال (نوت‌بوک اصلی به همراه کدهای اضافی) را در آن قرار دهید. در نهایت همه موارد را در قالب یک فایل زیپ با اسم HW5-STD-NUM آپلود نمایید.
- ددلاین تمرین ساعت ۲۳:۵۹ روز ۲۹ خرداد می‌باشد.