



به نام خدا
درس مبانی یادگیری عمیق
پاسخنامه تمرین سری اول
استاد درس : دکتر مرضیه داوودآبادی
دستیاران : حسن حماد، مرتضی حاجی آبادی
دانشگاه علم و صنعت ایران، دانشکده مهندسی کامپیوتر
نیمسال اول تحصیلی ۱۴۰۲ - ۱۴۰۳

پاسخ های مناسب برای هر سوال، لزوماً یکتا نیستند

۱. الف) در این سوال می خواهیم کلاس متن یک خبر را تشخیص دهیم (کلاس ۰ یا کلاس ۱).
اطلاعات جدول زیر را در نظر بگیرید و با استفاده از آنها احتمال تعلق خبرهای تست به هر کلاس
را محاسبه کنید و کلاس عنوان خبر را با استفاده از مدل های احتمالاتی آموخته شده تشخیص
دهید (۱۰ نمره)

t_1 = فناوری فرهنگی علمی اجتماعی

t_2 = فناوری فرهنگی علمی اجتماعی ورزشی

T = فناوری

C = فرهنگی

Sc = علمی

So = اجتماعی

S = ورزشی

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$

$$P(d|c) = p(w_1|c).p(w_2|c)...p(w_n|c)$$

$$P(c=0) = \frac{1}{2}, P(c=1) = \frac{1}{2}$$

$$P(c=0|t_1) = P(c=0).P(T|0).P(C|0).P(Sc|0).P(So|0)$$

$$= \frac{1}{2} \cdot \frac{2}{9} \cdot \frac{2}{9} \cdot \frac{2}{9} \cdot \frac{1}{9} = \frac{4}{6561} = 6.09 * 10^{-4}$$

$$P(c = 1|t_1) = P(c = 1).P(\textcolor{red}{T}|1).P(\textcolor{green}{C}|1).P(Sc|1).P(\textcolor{brown}{So}|1)$$

$$= \frac{1}{2} \cdot \frac{1}{8} \cdot \frac{1}{8} \cdot \frac{1}{8} \cdot \frac{2}{8} = \frac{1}{4096} = 2.4 * 10^{-4}$$

احتمال کلاس ۰ بیشتر از احتمال کلاس ۱ است. پس این متن به کلاس ۰ تعلق دارد.

مطابق با راه حل برای تست اول، احتمال تعلق تست دوم به هر دو کلاس ۰ می شود. از این رو باید با روش دیگری این کار را انجام دهیم. روشی که در قسمت ب آمده است.

ب (در صورتی که در متن خبر تست، داده جدید داشته باشیم (داده تست دوم) چگونه می توان این داده را تحلیل کرد؟(راهنمایی : استفاده از هموارسازی لاپلاس با ضریب آلفای ۱)(۱۰ نمره امتیازی)

چون احتمال $P(S|0)$ و $P(S|1)$ هر دو ۰ هستند، برای رفع این اشکال از روش هموارسازی لاپلاس استفاده می کنیم

$$P(w_i|c_j) = \frac{Count(w_i, c_j) + 1}{\sum_{w \in v} Count(w, c_j) + |v|}$$

$$P(c = 0|t_2) = P(c = 0).P(\textcolor{red}{T}|0).P(\textcolor{green}{C}|0).P(Sc|0).P(\textcolor{brown}{So}|0).P(\textcolor{brown}{S}|0)$$

$$= \frac{1}{2} \cdot \frac{2+1}{9+6} \cdot \frac{2+1}{9+6} \cdot \frac{2+1}{9+6} \cdot \frac{2}{9+6} \cdot \frac{0+1}{9+6} = \frac{1}{2} \cdot \frac{3}{15} \cdot \frac{3}{15} \cdot \frac{3}{15} \cdot \frac{2}{15} \cdot \frac{1}{15} = \frac{27}{759375} = 3.555 * 10^{-5}$$

$$P(c = 1|t_2) = P(c = 1).P(T|1).P(C|1).P(Sc|1).P(So|1).P(S|1)$$

$$= \frac{1}{2} \cdot \frac{1+1}{8+6} \cdot \frac{1+1}{8+6} \cdot \frac{1+1}{8+6} \cdot \frac{2+1}{8+6} \cdot \frac{0+1}{8+6} = \frac{1}{2} \cdot \frac{2}{14} \cdot \frac{2}{14} \cdot \frac{2}{14} \cdot \frac{3}{14} \cdot \frac{1}{14} = \frac{12}{537824} = 2.231 * 10^{-5}$$

احتمال کلاس ۰ بیشتر از احتمال کلاس ۱ است. پس این متن به کلاس ۰ تعلق دارد.

کلاس	متن خبر	نوع داده
۰	فناوری فرهنگی علمی اقتصادی	آموزش
۰	فناوری فرهنگی علمی اجتماعی سیاسی	آموزش
۱	فناوری فرهنگی اجتماعی سیاسی	آموزش
۱	علمی اجتماعی سیاسی اقتصادی	آموزش
؟	فناوری فرهنگی علمی اجتماعی	تست
؟	فناوری فرهنگی علمی اجتماعی ورزشی	تست

۲. نوتبوک های آموزشی *Keras.ipynb* و *Pytorch.ipynb*، *Numpy.ipynb*، *Basic.ipynb* را اجرا کنید، دقت کنید این نوتبوکها صرفا برای یادآوری هستند و نمره ای به آن ها تعلق نخواهد گرفت.

۳. *logistic-regression* از تابع سیگموئید $\sigma(a) = \frac{1}{1+\exp(-a)}$ برای مدل سازی توزیع شرطی $p(y|x)$ استفاده می کند و سپس تخمین *maximum likelihood* را اعمال می کند. می توان از تابع *probit* (به جای تابع سیگموئید) استفاده کرد:

$$\Phi(a) = \int_{-\infty}^a N(\theta | 0, 1) d\theta$$

که در آن $N(\theta|0, 1)$ توزیع نرمال استاندارد است. برای رگرسیون *probit*، منفی ضرر شرطی *log-likelihood* را محاسبه کنید. نیازی به ساده سازی عبارت نیست. (۱۰ نمره)

$$P(y|x) = \phi(w^\top x) \rightarrow \begin{cases} w \rightarrow \text{پارامترهای مدل} \\ x \rightarrow \text{ویژگی های ورودی} \end{cases}$$

$$\begin{aligned} \text{Likelihood}(w) &= \prod_{i=1}^N P(y^i | x^i) \\ &= \prod_{i=1}^N \phi(w^\top x_i) \\ \Rightarrow -\log -\text{likelihd}(w) &= -\log \left(\prod_{i=1}^N \phi(w^\top x_i) \right) \\ &= -\sum_{i=1}^N [y_i \log(\phi(w^\top x_i)) + (1 - y_i) \log(1 - \phi(w^\top x_i))] \end{aligned}$$

۴. الف) دلیل استفاده از توابع فعال سازی در شبکه های MLP چیست؟

همه قدرت شبکه های عصبی به استفاده از توابع غیر خطی آنها وابسته است. اگر از توابع خطی استفاده شود، شبکه عصبی قدرت خاصی ندارد زیرا ترکیب چند تابع خطی را میتوان به صورت یک تابع خطی جدید نوشت و این یعنی قدرت محاسبات تفاوتی نخواهد کرد.

ب) آیا هر تابع غیرخطی را به عنوان تابع فعالسازی میتوان استفاده کرد؟ (۱۰ نمره)

تابع فعالسازی باید غیرخطی، کراندار، یکنوا، پیوسته و مشتق پذیر باشد پس نمیتوان از هر تابع غیر خطی به عنوان تابع فعالسازی استفاده کرد.

۵. الف) توابع فعال سازی زیر را توضیح دهید (با ذکر مزایا و معایب هر کدام) و باهم دیگر مقایسه کنید.

• تابع سیگموئید ($Sigmoid$):

تابع $Sigmoid$ غیرخطی است و خروجی آنالوگ میدهد. (بر خلاف تابع پله ای). گرادیان همواری دارد و برای یک طبقه بند مناسب است. خروجی تابع فعال ساز همیشه بین ۰ و ۱ بوده که در مقایسه با توابع خطی و یا $ReLU$ که رنج نامحدود دارند محدود تر است و به همین دلیل فعال سازی شدیدی نخواهد داشت و کنترل شده است. در طول طرف تابع واکنش به تغییرات خیلی کم بوده و تقریباً هموار است و مشکل ناپدید شدن گرادیان را داریم. خروجی اش با مرکز صفر نیست (مرکزش مقدار ۰.۵ است) و بروزرسانی ها بر اساس گرادیان خیلی کم

در دو جهت متفاوت خواهند شد. این تابع اشباع میشود و کارآیی اش را در مقادیر بالا از دست میدهد. همگرایی کندی دارد و محاسبات نسبتاً هزینه بری هم دارد.

- تابع $softmax$:

تابع $Softmax$ احتمالات را محاسبه میکند و برای هر کلاس از کلاسه‌های مورد نظر یک احتمال رخداد محاسبه میکند که مشخصاً مجموع آنها باید ۱ شود. میتوان گفت از ترکیب چند $Sigmoid$ ساخته میشود یکی از مناسب‌ترین توابع برای انجام کلاس بندی چند کلاسه است از این تابع بیشتر برای نورونهای لایه آخر (خروجی) استفاده میشود. لازم به ذکر است که مقدار گرادیان این تابع برای مقادیر منفی صفر است؛ به این معنا که وزن‌ها در حین عملیات پس‌انتشار به‌روزرسانی نمی‌شوند و این می‌تواند مشکل مرگ نورون را ایجاد کند.

- تابع $ReLU$:

تابع $ReLU$ بسیار ساده‌تر از توابع قبل است و محاسبات هزینه بری ندارد و مشکل ناپدید شدن گرادیان را هم حل میکند. فقط قابل استفاده در لایه‌های میانی شبکه است. برخی گرادیانها برای مقادیر منفی که همواره صفر هستند میتوانند شکننده باشند و یک نورون را به کلی از کار بیندازند و به اصطلاح نورون بمیرد. این تابع از سمت مثبتها محدود نیست و امکان فعال سازی بسیار شدید را دارد تابع $ReLU$ در صفر مشتق پذیر نیست سرعت همگرایی این تابع بیشتر است.

- تابع $Tanh$:

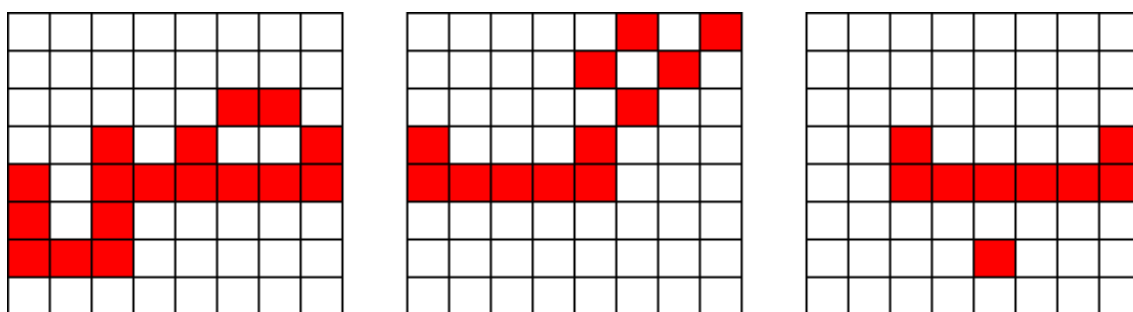
این تابع هم همانند $Sigmoid$ غیر خطی بوده و گرادیان نسبتاً همواری دارد البته به نسبت $Sigmoid$ کمی شیب بیشتری دارد خروجی تابع فعال ساز همیشه بین -1 و 1 بوده که در مقایسه با توابع خطی و یا $ReLU$ که رنج نامحدود دارند محدود تر است و به همین دلیل فعال سازی شدیدی نخواهد داشت و کنترل شده است. در طول طرف تابع واکنش به تغییرات خیلی کم بوده و تقریباً هموار است و مشکل ناپدید شدن گرادیان را داریم. این تابع اشباع میشود و کارآیی اش را در مقادیر بالا از دست میدهد همگرایی کندی دارد و محاسبات نسبتاً هزینه بری هم دارد.

ب) در این بخش از سوال می‌خواهیم توابع فعال‌سازی را پیاده‌سازی کنیم و با توابع فعال‌سازی کتابخانه $Pytorch$ مقایسه کنیم. برای این کار نوتبوک `activation_functions.ipynb` را تکمیل کنید. در نوتبوک موجود است

ج) حال می‌خواهیم یک شبکه MLP ساده طراحی کنیم که بتواند تصاویر زیر را از هم دیگر جدا کند. برای این کار یک معماری برای شبکه MLP ارائه دهید و علت انتخاب این معماری را توضیح دهید.

معماری شما باید شامل اجزای زیر می‌شود:

- تعداد لایه‌ها و علت انتخاب این تعداد
- تعداد نورون‌های هر لایه و علت انتخاب این تعداد
- تابع فعال‌سازی و علت انتخاب آن
- تابع ضرر و علت انتخاب آن



معماری پیشنهادی شبکه در این حالت

برای این حالت یک MLP با ۶۴ نورون (برابر با سائز تصویر ورودی است) در لایه ورودی و ۳ نورون در لایه خروجی (برابر با تعداد کلاس‌های مورد نظر است) پیشنهاد میشود. با توجه به وضعیت مساله و پیچیدگی کم آن نیازی به اضافه کردن لایه‌های میانی نیست و برای تفکیک سه الگوی داده شده کافی است. از تابع فعال‌سازی $softmax$ استفاده خواهیم کرد چون این مسئله یک مسئله دسته بندی چند کلاسه است. و از $crossentropy\ categorical$ به عنوان تابع ضرر استفاده خواهیم کرد چون نشان داده شده که برای تابع فعال‌سازی $softmax$ مناسب است.

د) برای اطمینان بیشتر، یک شبکه MLP طبق معماری خود را با استفاده از کتابخانه $PyTorch$ پیاده‌سازی کنید. سپس با استفاده از تصاویر مورد نظر تست کنید و نتایج را گزارش کنید. (۳۰ نمره) تست معماری در فایل نوتبوک [Test – MLP – Architecture.ipynb](#) آورده شده است که نشان میدهد شبکه همگرا می‌شود.

۶. یک شبکه عصبی چندلایه (MLP) را در نظر بگیرید که برای دسته بندی دو کلاس مورد استفاده قرار می‌گیرد. خروجی نورون آخر را z در نظر بگیرید و خروجی شبکه عصبی به صورت زیر محاسبه می‌شود:

$$y = \sigma(ReLU(z))$$

که در آن y خروجی پیش‌بینی شده است، σ تابع فعال‌ساز سیگموئید و $ReLU$ نمایانگر تابع فعال‌ساز واحد خطی بازگشتی است.

MLP از یک آستانه ۰.۵ استفاده می‌کند و خروجی‌های بزرگتر یا مساوی ۰.۵ را به عنوان کلاس ۱ در نظر می‌گیرد و خروجی‌های کمتر از ۰.۵ را به عنوان کلاس ۰ دسته‌بندی می‌کند.

در استفاده از این شبکه عصبی برای دسته‌بندی دو کلاسه در نظر گرفته شده، چه مشکلات یا چالش‌هایی با توجه به ترکیب توابع فعال‌ساز و آستانه ۰.۵ ممکن است وجود داشته باشد؟ (۲۰ نمره)

- **محوشدگی گرادیان (*vanishing*) در اطراف آستانه 0.5:** زمانی که خروجی نورون آخرین لایه در MLP به آستانه تصمیم‌گیری (0.5) نزدیک می‌شود، گرادیان‌ها می‌توانند بسیار کوچک شوند و باعث کند شدن آموزش می‌شوند. این مشکل اتفاق می‌افتد چرا که تابع فعال‌ساز سیگموئید که در لایه آخر استفاده می‌شود، وقتی ورودی به 0.5 نزدیک می‌شود، اشباع می‌شود.
- **طبیعت قطعه‌ای تابع $ReLU$ و آستانه 0.5:** تابع فعال‌سازی $ReLU$ به صورت تکه‌ای خطی است، به این معنی که برای مقادیر ورودی کمتر از صفر، صفر را خروجی می‌دهد. اگر ورودی به $ReLU$ به 0.5 نزدیک باشد، خروجی ممکن است صفر شود و اطلاعات از دست برود.
- **کلاس‌های نامتوازن:** توزیع نامتوازن کلاس‌ها به معنای این است که یک کلاس دارای تعداد نمونه بسیار بیشتری از کلاس دیگر است. استفاده از یک آستانه ثابت 0.5 برای دسته‌بندی ممکن است به پیش‌بینی‌های با میل به کلاس اکثریت منجر شود.
- **چالش‌های تنظیم ابرپارامترها:** تعیین آستانه (مثلاً 0.5) و تعیین معماری MLP (تعداد لایه‌ها و نورون‌ها) دشواری‌هایی دارد. انتخاب‌ها به ویژه از مشکلات خاص هر مسئله و مجموعه داده وابسته است.
- **عدم اطمینان در پیش‌بینی:** MLP تخمینات نقطه‌ای ارائه می‌دهند (به عنوان مثال، احتمال تعلق به کلاس ۱)، اما اطلاعاتی از درجه اعتماد مدل به این پیش‌بینی‌ها نمی‌دهند. در برخی موارد، تخمین عدم اطمینان حیاتی است.
- **مسائل مرتبط با آموزش:** انتخاب صحیح تابع هزینه، تکنیک‌های اصلاح، و راهبردهای آموزش (مانند برنامه‌های نرخ یادگیری) می‌تواند به طور قابل توجهی روی عملکرد MLP

تأثیر بگذارد. انتخاب روش‌های آموزش نامناسب می‌تواند به مسائل در آموزش منجر شود. مانند عدم همگرایی بهینه مدل، عدم پایداری در آموزش یا عدم بهره‌برداری از داده‌های آموزش ممکن است رخ دهد.

۷. تحلیل خود برای سوالات زیر را بنویسید (۲۰ نمره).

الف) به نظر شما مهم‌ترین تفاوت یادگیری ماشین و یادگیری عمیق در چیست؟ یکی از تفاوت‌های اصلی بین یادگیری ماشین (ML) و یادگیری عمیق (DL) نحوه کار با ویژگی‌ها (فیچرها) است: در یادگیری ماشین سنتی (ML)، مرحله مهمی به نام مهندسی ویژگی وجود دارد که توسط متخصصان انسانی به صورت دستی انجام می‌شود. در این مرحله ویژگی‌های مرتبط از داده‌ها به صورت دستی انتخاب و طراحی می‌شوند. سپس این ویژگی‌ها به عنوان ورودی به الگوریتم‌های یادگیری ماشین داده می‌شوند. مدل‌های یادگیری ماشین بر اساس این ویژگی‌های پیش‌تعریفی برای پیش‌بینی‌ها استفاده می‌کنند. در یادگیری عمیق (DL)، مدل‌ها به طور خودکار ویژگی‌ها را از داده‌های خام یاد می‌گیرند. این موضوع نیازی به مهندسی ویژگی صریح ندارد. شبکه‌های عصبی عمیق چندین لایه دارند که می‌توانند به صورت خودکار ویژگی‌های سلسله‌مراتبی را استخراج کنند. این ویژگی اصلی یادگیری عمیق است و به خصوص در وظایفی که مهندسی ویژگی ممکن است دشوار یا ناممکن باشد، بسیار مؤثر است.

ب) فرض کنید یک شبکه یادگیری عمیق دارای ۱۶ لایه است. به نظر شما لایه ۷ام برای دستیابی به نتیجه نهایی در طبقه بندی مناسب تر است یا لایه ۱۱ام؟ چرا؟ لایه ۱۱ام. چون ویژگی‌های سطح بالاتری را استخراج کرده و برای طبقه بندی مناسب تر خواهد بود

ج) به نظر شما برای تقریب توابع استفاده از شبکه‌های عمیق تر کاراتر است یا شبکه‌های عریض تر؟ چرا؟

انتخاب بین شبکه‌های عمیق و گسترده برای تقریب توابع به عوامل متعددی بستگی دارد، از جمله پیچیدگی تابعی که سعی در تقریب آن دارید، اندازه مجموعه داده‌تان و منابع محاسباتی. در ادامه به بررسی مزایا و معایب هر یک از این گزینه‌ها می‌پردازیم:

شبکه‌های عمیق:

مزایا:

- شبکه‌های عمیق قابلیت درک و استخراج ویژگی‌های سلسله‌مراتبی و انتزاعی از داده‌ها دارند که آنها را مناسب برای وظایف پیچیده می‌کند.

- آنها در وظایفی مانند تشخیص تصاویر و پردازش زبان طبیعی به خوبی عمل کرده‌اند.
- شبکه‌های عمیق ممکن است پارامترهای کمتری نسبت به شبکه‌های گسترده داشته باشند، به ویژه وقتی با مجموعه داده‌های بزرگ سر و کار دارید.

معایب:

- آموزش شبکه‌های عمیق هزینه محاسباتی دارد و ممکن است نیاز به داده‌های بیشتری داشته باشد تا از بیش‌برازش جلوگیری کند.

- شبکه‌های عمیق ممکن است به گرادین‌های ناپدید شونده حساس باشند که آموزش را دشوارتر می‌کند.

شبکه‌های عریض:

مزایا:

- شبکه‌های گسترده می‌توانند همزمان ویژگی‌های گسترده‌ای را استخراج کنند، که آنها را برای وظایفی که ممکن است بسیاری از ویژگی‌های مختلف مرتبط داشته باشند، مناسب می‌کند.

- آنها ممکن است در طی فرآیند آموزش سریع‌تر همگرا شوند و به داده‌های کمتری نیاز داشته باشند نسبت به شبکه‌های عمیق.

معایب:

- شبکه‌های گسترده بیشتر به بیش‌برازش حساس هستند، به ویژه وقتی مجموعه داده کوچک است.

- افزایش تعداد نوروها در هر لایه ممکن است منجر به هزینه محاسباتی بالاتر شود. در عمل، انتخاب میان شبکه‌های عمیق و گسترده اغلب شامل تعادلی میان عمق و عرض است.

(د) مزایا و معایب افزودن لایه‌های بیشتر به شبکه عصبی عمیق چیست؟

افزودن لایه‌های بیشتر به یک شبکه عصبی عمیق چندین مزیت از جمله افزایش ظرفیت بازنمایی، توانایی یادگیری ویژگی‌های سلسله‌مراتبی، بهبود عملکرد در کارهای پیچیده و کارایی پارامترها را ارائه می‌دهد. شبکه‌های عمیق‌تر می‌توانند الگوهای پیچیده‌ای را در داده‌ها ثبت کنند و آنها را برای کارهایی مانند تشخیص تصویر و پردازش زبان طبیعی مناسب می‌سازد. علاوه بر این، آنها می‌توانند به عملکرد مشابه یا بهتری مانند شبکه‌های کم عمق با پارامترهای کمتر دست یابند که می‌تواند از نظر محاسباتی سودمند باشد. معایب: شبکه‌های عمیق‌تر با هزینه‌های محاسباتی بالاتری همراه هستند و به طور بالقوه از مشکلات ناپدید شدن (*vanishing*) یا انفجار (*exploding*) گرادین در طول آموزش رنج می‌برند. آنها بیشتر مستعد بیش‌برازش هستند، به خصوص با داده‌های محدود، و پیچیدگی آنها می‌تواند تنظیم ابرپارامترها را چالش برانگیز کند. علاوه بر این، تفسیر تصمیمات شبکه‌های عمیق با افزایش عمق آنها دشوارتر می‌شود. در نهایت، تصمیم برای افزودن لایه‌های بیشتر باید بر اساس ارزیابی دقیق مشکل، داده‌های موجود و منابع محاسباتی، با آگاهی از مبادلات بین افزایش پیچیدگی

و دستاوردهای بالقوه عملکرد باشد.