

به نام خدا



دانشکده علوم کامپیوتر

هوش مصنوعی و سیستم های خبره
پروژه دوم (شبکه های عصبی)

دکتر آرش عبدی

بهار ۱۴۰۳

طراحان:

الهام غلامی

پروژه دوم درس هوش مصنوعی و سیستم های خبره

- در صورت وجود هرگونه ابهام در پروژه به طراح آن پیام دهید.
- علاوه بر آپلود داخل کوئرا، پروژه ها به صورت شفاهی ارائه گرفته میشوند؛ بنابراین تسلط شما هنگام ارائه شفاهی نسبت به گزارش پروژه اهمیت بیشتری دارد. برای توضیح بیشتر، اگر پروژه 100 نمره داشته باشد، نمره ارائه شفاهی 90 و نمره گزارش، 10 خواهد بود؛ بنابراین سعی کنید قبل از ارائه شفاهی نسبت به کد خود و مبحث پروژه آمادگی لازم را داشته باشید.
- با توجه به تنظیم شدن ددلاین توسط خود شما امکان تمدید وجود ندارد.
- زبان برنامه نویسی پیشنهادی پایتون است ولی مجاز به استفاده از هر زبان دلخواهی هستید. نمره هر زبان با هم برابر است.
- پروژه تک نفره است لذا از هرگونه تقلب در انجام پروژه خودداری کنید.
- برای این پروژه شما باید حداقل یک صفحه توضیحات ارائه دهید. این توضیحات در زمان ارائه شفاهی به شما کمک می کنند. نیاز به توضیح کد نیست؛ روند پیاده سازی پروژه، چالش هایی که به آن خوردید و خلاقیت های شما ارزش بیشتری دارند.

آیدی تلگرام طراح:

- @elham_gholamim

هدف پروژه: در این پروژه قصد داریم با شبکه های پرسپترون چند لایه و روش آموزش و یادگیری در این شبکه ها به صورت عملی آشنا شویم. در این پروژه استفاده از ابزارها و توابع آماده مربوط به شبکه های عصبی (MLP) بلامانع است و توصیه میشود از این ابزارهای آماده (مثلا در پایتون) استفاده کنید.

شرح پروژه:

○ بخش اول

۱. سوال اول: (ساخت و تنظیم یک شبکه عصبی چند لایه)

تعدادی تابع (حداقل ۳ تابع) با ورودی یک بعدی از ساده (مانند معادله یک خط ساده) تا خیلی پیچیده (مثلا یک تابع مثلثاتی خاص) در یک دامنه مشخص در نظر بگیرید. تعدادی نقطه از روی این توابع تولید کنید و بخشی از این نقاط را به عنوان مجموعه آموزشی در نظر بگیرید. با کمک این مجموعه داده، یک MLP را طوری آموزش دهید که بتواند تابع ورودی شما را تشخیص دهد. با استفاده از کتابخانه های موجود (مانند Matplotlib) تابع خود را در کنار تابع تخمین زده شبکه تان رسم کنید و میزان خطای شبکه را با هر معیاری نمایش دهید. در این سوال نکته حائز اهمیت، تحلیل و درک شما از چگونگی کارکرد شبکه عصبی در شرایط مختلف است. برای اینکار، می توانید پارامترهای زیر را چندین بار تغییر دهید تا تغییرات شبکه را مشاهده نمایید. بهتر است نتیجه تحلیلتان را گزارش کنید.

- تعداد نقاط ورودی
- میزان پیچیدگی تابع مورد نظر
- تعداد لایه های هر شبکه و تعداد نورون های هر لایه
- تعداد چرخه های شبکه برای تکمیل یادگیری
- وسعت دامنه ورودی خصوصا در توابع پیچیده تر
- افزایش ابعاد تابع

و هر پارامتر دیگری که فکر می کنید موثر است.

۲. سوال دوم: (بررسی نویز در عملکرد شبکه)

اکنون به نقاطی که در بخش قبل تولید کرده‌ای چندین نویز (داده پرت یا خراب) اضافه کنید. از نویز نزدیک به صفر تا نویز زیاد را آزمایش کنید و نتیجه شبکه را گزارش کنید. شبکه شما باید بتواند تا حد خوبی نویزها را در نظر نگیرد و دقت خود را حفظ کند.

۳. سوال سوم: (تخمین تابع با استفاده از نقاط ورودی)

یک قلم و کاغذ آماده کنید(!). هرچه خشم از پروژه‌های درسی و نمره‌ها و اساتید(!) و کلاس‌ها و عیدی و ... دارید در خود جمع کنید و یک خط خطی فجیع روی کاغذ بکشید و بخش بالایی (یا پایینی) شکل حاصله را به عنوان یک تابع در نظر بگیرید و نقاطی روی این تابع به صورت تقریبی ایجاد کنید (مقدار خروجی تابع را به صورت چشمی تقریب بزنید) و آن‌ها را به عنوان مجموعه آموزشی حاصل از عصبانیت خود به شبکه عصبی بدهید. مانند شکل‌های زیر:



مدل خود را با استفاده از تکنیک‌های cross-validation و محاسبه خطا (مانند MSE) مورد بررسی قرار دهید و سعی کنید به حداقل دقت مطلوب (حدوداً 70 درصد) برسید. هرچند عملکرد کلی شما و درستی الگوریتم از اهمیت بیشتری برخوردار است.

○ بخش دوم

۱. سوال اول: (استفاده از شبکه‌های عصبی در دسته‌بندی)

ابتدا بهتر است درباره کاربرد وسیع شبکه‌های عصبی در طبقه‌بندی داده‌ها تحقیق کنید. سپس لازم است یک پایگاه داده برای طبقه‌بندی انتخاب کنید. بهتر است از پایگاه داده‌های تصویری موجود در فریم‌ورک‌هایی مانند Keras یا پایگاه داده‌های صوتی استفاده کنید و سعی کنید طبقه‌بندی‌تان بیشتر از دو کلاس را در بر بگیرد. انتخاب پایگاه داده، آشنایی با ویژگی‌های آن، نحوه کار و استخراج داده و .. جزئی از پروژه است.

به علت حجم زیاد داده‌ها این بخش زمانبر است؛ انجام آنرا به روزهای آخر موکول نکنید. هدف از این بخش این است که تنظیم یک شبکه عصبی برای تفکیک داده را یاد بگیرید و سعی کنید دقت آن را بالا ببرید.

یک نمونه داده پیشنهادی برای شما Fashion-MNIST است. دیتاستی است که شامل تصاویر سیاه و سفید از ۷۰,۰۰۰ محصول مد در ۱۰ دسته‌بندی می‌باشد، که هر تصویر دارای ابعاد ۲۸x۲۸ پیکسل است.

دسته‌بندی‌ها در Fashion-MNIST شامل اقلام مختلفی مانند پیراهن، کفش، کیف و غیره می‌باشد. از دانشجویان انتظار می‌رود که یک مدل یادگیری ماشین را آموزش دهند تا بتواند این تصاویر را به درستی طبقه‌بندی کند. راهنمایی:

ابتدا کتابخانه‌های مورد نیاز خود را نصب کنید و سپس داده‌های خود را بارگذاری کنید. (یا می‌توانید دیتاست را دانلود کنید و آنرا به صورت دستی ایمپورت نمایید). یک راه برای بارگذاری داده:

```
from tensorflow.keras.datasets import fashion_mnist
(train_images, train_labels), (test_images, test_labels) =
fashion_mnist.load_data()
```

داده‌ها را حتما پیش پردازش و نرمال‌سازی کنید تا مقادیر پیکسل‌ها بین صفر و یک قرار بگیرند. در نهایت بنابر داده‌های آموزشی خود مدل را بسازید و تعداد دوره‌ها و اندازه دسته‌بندی‌ها را تنظیم کنید.

۲. سوال دوم: (حذف نویز)

در دنیای واقعی دیتاست‌ها، اغلب ناقص و با نویز اند. در این سوال هدف استفاده کردن از شبکه‌عصبی برای داده‌های نویزدار است. درباره انواع نویز بخوانید و از تابع‌های آماده یا هر تابع دلخواه برای افزودن نویز به داده‌هایتان استفاده کنید. برای اینکار می‌توانید از پایگاه داده انتخابی خود در سوال قبل استفاده کنید؛ بخشی از آن را به صورت نامنظم دست‌چین کنید و به آن نویز اضافه کنید.

نسخه با نویز و بدون نویز داده‌ها را در نظر بگیرید، نسخه با نویز را به عنوان ورودی و نسخه بدون نویز داده‌ها را به عنوان خروجی شبکه بدهید. آیا شبکه می‌تواند رفع نویز را یاد بگیرد؟

با چه ساختاری؟ شبکه را در داده‌های آزمایشی و آموزشی خود بررسی کنید. رفع نویز در داده آموزشی و آزمایشی چه قدر تفاوت دارد؟

مسیر پرسش و پاسخ را ادامه دهید و تحلیل های خود را بیان کنید.

میزان نویز را از خیلی کم تا خیلی زیاد تغییر دهید (حداقل ۳ آزمایش)، و نتایج حاصله را مقایسه و ارزیابی کنید.

نکاتی در رابطه با تحویل پروژه: (بخوانید!)

- در این پروژه (به خصوص بخش دوم) استفاده از منابع آموزشی آنلاین و ابزارهای آماده مجاز است و حتی توصیه می‌شود از آنها استفاده کنید. برای مثال کتابخانه Tenserflow یا PyTorch برای بخش دوم، می‌تواند بسیار کمک کننده باشد.
- در بخش اول، استفاده از کتابخانه‌های نمایش گرافیکی برای نشان دادن نمودار خروجی لازم است و خلایق و سلیقه شما نمره دارد!
- در بخش دوم، می‌توانید نموداری از تغییرات دقت و loss در هر epoch^۱ را نمایش دهید.
- استفاده از مجموعه داده پیشنهادی در بخش دوم کاملاً اختیاری است. داده‌ای را امتحان کنید که برایتان جالب است.
- در بخش دوم، با توجه به محدودیت هایتان پیشنهاد می‌شود تا حد امکان از تکنیک های ساده سازی مانند پایین آوردن ابعاد عکس و استفاده محدود و حساب شده و مدیریت به جای داده ها استفاده کنید.
- در تمام مراحل، از شما دقت بالا انتظار نمی‌رود. تلاش شما و نحوه آزمایش و تنظیم مدل شبکه عصبی شماست که اهمیت دارد.
- تلاش شده است تا حد امکان تعریف پروژه واضح باشد، اما قسمت‌هایی از آن دقیق نیست و کاری که انجام می‌دهید بستگی به درک خودتان از موضوع دارد.
- بخش دوم نمره و وزن بیشتری دارد.
- در هر یک از بخشهای پروژه که لازم میدانستید از تکنیک Cross-Validation استفاده کنید.

فراموش نکنید هر آنچه نشان دهنده تلاش و تحلیل شما باشد (مانند استفاده از نمودارها، چندین بار آزمایش نمودن و ...) در افزایش نمره شما تاثیر دارد! ☺

موفق باشید!

^۱

Epoch : از آنجایی که یک دوره زمانی ارسال کل مجموعه داده از جلو به عقب شبکه ممکن است خیلی بزرگ باشد، آنرا به چند دسته کوچکتر به نام epoch (بخوانید اپیاک!) تقسیم می‌کنند. بهتر است در این باره در اینترنت جستجو کنید.