



دانشگاه صنعتی اصفهان

دانشکده مهندسی برق و کامپیوتر

تمرین عملی سری دوم

مبانی هوش محاسباتی

بهار 1404

استاد درس: دکتر سمانه حسینی

دستیار آموزشی: آرش آژند

موعد تحویل تکلیف: 20 اردیبهشت 1404

نکات تحویل تکلیف:

1. پاسخ های خود را حتما در سامانه یکتا آپلود کنید و از ارسال در بستر تلگرام خودداری کنید.
2. انجام تکلیف بصورت تک نفره است و در صورت مشاهده تقلب نمرات هم مبدأ و هم مقصد صفر میشود.
3. ساختار نامگذاری تکلیف باید بصورت روبرو باشد HWX_StudentID_LastName_Programming.zip که X شماره تمرین، StudentID شماره دانشجویی و LastName نام خانوادگی میباشد.
4. تکالیف را در محیط jupyter notebook یا google colab پیاده سازی کنید و فایل ipynb را ارسال کنید.
5. توضیح کدی که نوشته اید، بررسی و تحلیل نتایج آن و بیان علت نتایج و نیز مقایسه نتیجه با آنچه مورد انتظارتان بوده است، از اهمیت بالایی برخوردار است. شما می توانید گزارش پروژه را در همان محیط jupyter notebook بنویسید و نیازی به فایل pdf جداگانه نیست. هم چنین اگر برای حل سوال فرضیات خاصی مدنظر دارید حتما آن را قید کنید.

آیدی تلگرام:

@arash_azhand

سوال اول (50 نمره)

برای حل این سوال یک فایل ژوپیتتر (Q1.ipynb) در اختیار شما قرار گرفته که باید آن را کاملش کنید. هدف این سوال آشنایی با کتابخانه ی `pytorch` و مفاهیم مهم یادگیری عمیق است. مکان هایی که شما باید برای آن کد بنویسید با `# TODO` مشخص شده اند و توضیحات مورد نیاز هر قسمت داده شده است. با توجه به توضیحات و خواسته ی سوال کد خواسته شده را بنویسید و خروجی بگیرید. در این سوال مجموعه داده را خودمان میسازیم و با آن کار میکنیم. از این تمرین لذت ببرید و در آخر، آزمایش های مختلف را انجام دهید تا کاربرد مفاهیمی که در درس یادگرفتید را بصورت عملی نیز ببینید :

سوال دوم (100 نمره)

معرفی دیتاست:

دیتاست `Covertypes` توسط سازمان منابع طبیعی آمریکا تهیه شده و در [UCI Machine Learning Repository](https://mlr.cs.umich.edu/projects/mlr/data-repository/uci-repository/) موجود است. هدف اصلی این دیتاست، پیش‌بینی نوع پوشش گیاهی (`cover type`) مناطق مختلف جنگلی در ایالت کلرادو آمریکاست، بر اساس ویژگی‌های جغرافیایی و محیطی.

این دیتاست شامل 581,012 نمونه و 54 ویژگی است و متغیر هدف آن یک مقدار گسسته با 7 کلاس مختلف می‌باشد، که هر کلاس نمایانگر نوع خاصی از پوشش گیاهی مثل جنگل کاج، درختان صنوبر و... است که می‌توانید با مراجعه به سایت مقادیر دقیق آنها را پیدا کنید.

برخی از ویژگی‌های مهم این دیتاست:

- **Elevation**: ارتفاع از سطح دریا (بر حسب فوت)، یک ویژگی عددی مهم که تأثیر زیادی روی نوع پوشش گیاهی دارد.
- **Aspect**: جهت شیب زمین (۰ تا ۳۶۰ درجه).
- **Slope**: میزان شیب زمین.
- **Horizontal & Vertical Distance to Hydrology**: فاصله افقی و عمودی تا نزدیک‌ترین منبع آب مثل رودخانه یا دریاچه.
- **Hillshade features**: سه ویژگی مربوط به میزان نور خورشید در ساعات مختلف روز (ساعت ۹، ۱۲ و ۳ بعدازظهر).
- **Soil Type**: نوع خاک (۴۰ ویژگی باینری).
- **Wilderness Area**: ناحیه‌های حفاظت‌شده جنگلی (۴ ویژگی باینری).

با استفاده از این کد میتوانید دیتاست را در گوگل کولب بارگذاری کنید و متادیتای آن را مشاهده کنید:

```
!pip install ucimlrepo
from ucimlrepo import fetch_ucirep
# fetch dataset
coverttype = fetch_ucirep(id=31)
# data (as pandas dataframes)
X = coverttype.data.features
y = coverttype.data.targets
# metadata
print("\ndataset overview:")
display(coverttype.metadata)
print("\nTarget distribution:")
print(y.value_counts())
```

آماده سازی داده

- دیتاست را با استفاده از کد داده شده بارگذاری کنید. سپس EDA مختصری روی آن انجام دهید (مقادیر NULL، رکورد های تکراری، نرمال سازی ویژگی ها در صورت نیاز و ...) البته برخی از این کارها قبلا روی این دیتاست انجام شده و خیلی نیازی به تمیز کردن آن نداریم.
- با استفاده از `train_test_split` داده ها را به سه بخش آموزش (90٪)، اعتبارسنجی (5٪) و آزمون (5٪) تقسیم کنید. از آنجایی که این دیتاست بسیار `class imbalance` است نیاز هست از 'stratify' استفاده کنیم. درباره ی کارکرد آن تحقیق کنید و بنویسید اگر استفاده نکنیم به چه مشکلی برخورد خواهیم کرد.

تحلیل داده ها

- تعداد نمونه های هر کلاس را در مجموعه های آموزش، اعتبارسنجی و آزمون با نمودار هیستوگرام رسم کنید.
- توزیع یکسری از ویژگی ها را با نمودار های هیستوگرام با `barplot` نمایش دهید تا درک بهتری از ویژگی ها پیدا کنید. میتوانید از نمودار های مختلف دیگر هم مثل `heatmap` و غیره استفاده کنید.

آماده سازی داده برای PyTorch

- یک کلاس دیتاست بسازید که داده ها و برچسب ها را از `Numpy` یا `Pandas` بخواند. در صورت نیاز داده ها را به `torch.tensor` تبدیل کنید.
- برای سه مجموعه `train, val, test` دیتالودر بسازید. برای داده های آموزش، `shuffle=True` فعال باشد.

ساخت مدل شبکه عصبی

- یک کلاس برای مدل Fully Connected بنویسید. لایه‌ها می‌توانند به این شکل باشند:
• Linear → BatchNorm → ReLU → Dropout → Linear → ...
- ابعاد ورودی باید برابر با تعداد ویژگی‌ها (54) و خروجی برابر با تعداد کلاس‌ها (7) باشد. از CrossEntropyLoss برای Loss Function استفاده کنید.

پیاده‌سازی CUDA

- در ابتدا، چک کنید که CUDA در دسترس باشد. می‌توانید از gpu کولب استفاده کنید.
- مدل، داده‌ها، و برجسب‌ها را به این دیوایس منتقل کنید.

تعریف optimizer و تکنیک‌های Regularization

- از optimizer مناسب مثل Adam یا SGD استفاده کنید. اگر می‌خواهید از L2 Regularization استفاده کنید، پارامتر weight_decay را تنظیم کنید.
- می‌توانید نرخ یادگیری (learning rate) را در طول آموزش با StepLR یا ReduceLROnPlateau تنظیم کنید.

آموزش و اعتبارسنجی مدل

- تابع train() و validate() بنویسید که شامل:
 - عبور از دیتالودر
 - محاسبه Loss و Accuracy
 - استفاده از model.train() و model.eval()
 - صفر کردن گرادیان‌ها و ...
- در پایان هر اپیک، Loss و Accuracy را ذخیره کرده و در نمودار رسم کنید.

آزمون نهایی و گزارش عملکرد

- مدل آموزش‌دیده را روی داده‌های تست ارزیابی کنید. دقت نهایی را گزارش دهید.
- Confusion Matrix را رسم کنید تا ببینید مدل در تشخیص کدام کلاس‌ها بهتر یا بدتر عمل کرده است.

نکات نهایی:

- در این سوال دست شما کاملاً باز هست که از هر چیزی که بلد هستید یا فکر میکنید ممکن است کمک بکنه در بهتر شدن دقت نهایی مدل استفاده کنید. فقط اگر از مفاهیمی مثل کاهش بعد، Upsampling یا هر چیزی که خارج از مباحث تدریس شده در این کلاس هست، استفاده میکنید باید آن مفهوم را توضیح داده (هرچه توضیح کامل تر بهتر) تا مجاز به استفاده از آن شوید. در غیر اینصورت صرف گرفتن کد های پیشرفته از جیپیتی **نمره نخواهد داشت**.
- در این سوال شما آزاد هستید به هر ترتیبی که تمایل دارید و راحت تر هستید کد سوال ها را بنویسید برای همین سوالات شماره گذاری نشده اند فقط یک ترتیب پیشنهادی هستند.
- برای گرفتن نمره کامل نیاز هست که کد سوالات خواسته شده زده شود و کد ها و تحلیل های بیشتر و استفاده از مفاهیم پیشرفته تر (با شرطی که گفته شد) نمره اضافی قابل توجهی خواهد داشت.
- منابع مفید:
 - [:](#)
 - [PyTorch Training Loop](#)
 - [Working with CUDA in PyTorch](#)
 - [Matplotlib & Seaborn plotting](#)