دانشگاه صنعتی امیرکبیر
دانشکده مهندسی برق

به نام خدا

پروژه درس مقدمه ای بر هوش محاسباتی

پردازش زبان طبیعی

فاز اول

تعریف پروژه: تحلیل و امتیازدهی نظرات کاربران نتفلیکس به فیلم‌ها
شرح پروژه:
این پروژه در حوزه پردازش زبان طبیعی (NLP) است و هدف آن این است که با استفاده از نظرات کاربران در یک مجموعه داده نتفلیکس، امتیازهای آن‌ها به فیلم‌ها را پیش‌بینی کنیم. در این مجموعه داده، یک ستون شامل متن نظرات کاربران وجود دارد، و هدف مدل این است که براساس این نظرات، امتیازی بین ۱ تا ۵ را به هر نظر اختصاص دهد.

مراحل پروژه:

جمع‌آوری و بررسی داده‌ها

مجموعه داده حاوی ستون نظرات کاربران و امتیازهای واقعی آن‌ها است. این مجموعه باید بررسی شود تا از کیفیت داده‌ها و کامل بودن آن اطمینان حاصل شود.

لینک مجموعه داده در این قسمت قرار داده می شود:
https://drive.google.com/file/d/1Da19V1Q4sml2T2DtxbZM5h3S5t-hRJMx/view?usp=sharing

پیش‌پردازش متن

برای آماده‌سازی متن نظرات کاربران جهت تحلیل، باید مراحل زیر طی شود:

- حذف کاراکترهای اضافی و نویزها (در این مرحله ممکن است فقط دو ستون از داده ها را نگه دارید(comments and score))

دانشگاه صنعتی امیرکبیر
دانشکده مهندسی برق

- حذف علائم نگارشی بی‌ربط (مانند "!", "#", "@") که تأثیری بر معنا ندارند.
- حذف اعداد و کاراکترهای خاصی که اهمیت معنایی ندارند.

1. تبدیل متن به حروف کوچک:
   - برای هماهنگی در تحلیل، تمام متن‌ها به حروف کوچک تبدیل شوند.
2. حذف کلمات توقف:(Stop Words)
   - کلماتی مثل "and"، "as"، "then"، "for"، که در زبان انگلیسی رایج هستند اما اطلاعات مهمی ارائه نمی‌دهند، حذف شوند.
3. ریشه‌یابی (Stemming) یا (Lemmatization)

- **ریشه‌یابی :**کلمات به ریشه خود تقلیل می‌یابند. برای مثال، "کتاب‌ها" به "کتاب" تبدیل می‌شود.
- Lemmatization **:** کلمات به شکل پایه خود برمی‌گردند، به گونه‌ای که معنای کلمه حفظ شود. مثلاً "رفتم" به "رفتن" تبدیل می‌شود.

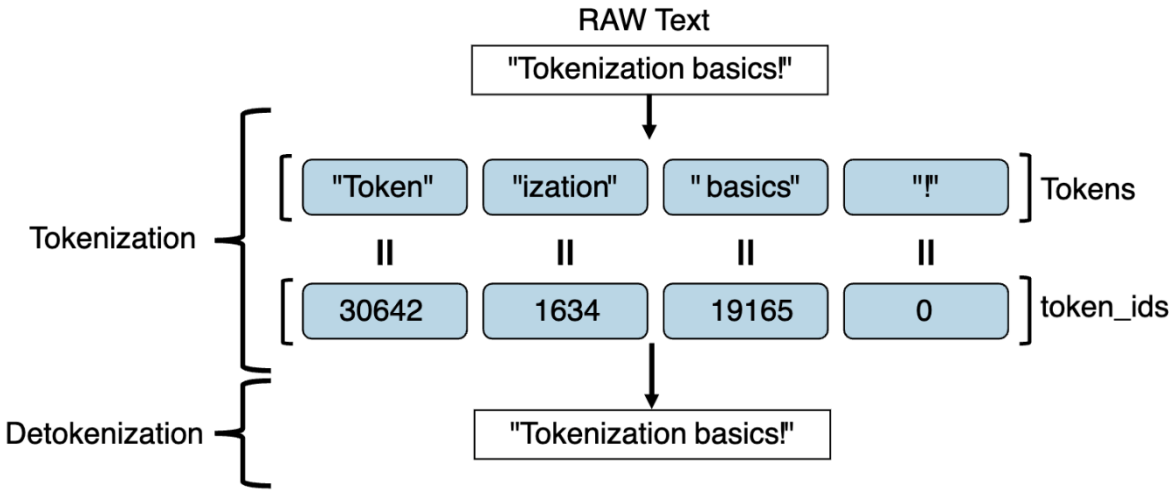
4. (Tokenization) :

- متن به واحدهای کوچک‌تر مانند کلمات یا عبارات تقسیم می‌شود. این مرحله به مدل کمک می‌کند که هر کلمه را به عنوان یک واحد مستقل بررسی کند. این امر به این دلیل است که در ادامه برای ساختن واژه‌نامه دسترسی به هر کلمه یا عبارت الزامی است.

به نام خدا

پروژه درس مقدمه ای بر هوش محاسباتی

پردازش زبان طبیعی



5. (Vectorization):

- برای نمایش متن به صورت عددی می توان از روش‌هایی مانند TF-IDF یا مدل های تعبیه کلمات (مانند Word2Vec) استفاده کرد.

مدل سازی

برای پیش‌بینی امتیاز کاربران از مدل‌های یادگیری ماشین و عمیق استفاده می‌شود. مراحل این بخش شامل:

- این مجموعه داده شامل 113,000 نمونه است که ممکن است برای مدل‌های RNN زمانبر باشد. برای صرفه جویی در وقت از 50,000 نمونه استفاده کنید. ارایه نتایج این 50000 نمونه الزامی است.
- درصورتی که وقت داشتید می‌توانید برای 113000 نمونه هم امتحان کنید و بطور جدا ارایه دهید. این قسمت به عنوان **نمره اضافی** تلقی خواهد شد.

به نام خدا

پروژه درس مقدمه ای بر هوش محاسباتی

پردازش زبان طبیعی

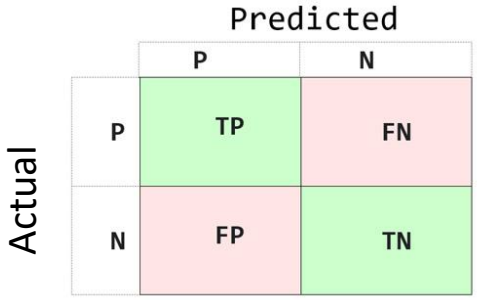
- تقسیم داده‌ها به مجموعه‌های آموزشی و آزمایشی و ارزیابی (train:80%, test:10%, validation:10%)
- استفاده از مدل های RNN, LSTM, GRU مجاز است. اما استفاده از transformer نمره نخواهد داشت.
- تنها در بخش آخر هر شبکه ای که زدید یک SoftMax قرار دهید تا خروجی نهایی را بین ۱ تا ۵ تحویل دهد.

ماتریس سردرگمی(Confusion Matrix)

بر اساس پیش بینی مدل برای داده هایی با کلاس باینری(همچنین قابل تعمیم به سایر کلاس های بیشتر) می‌توان چهار مفهوم تعریف کرد (داده ها با دو کلاس منفی و مثبت طبقه بندی می‌شوند):

1. True Negative (TN): تعداد نمونه هایی که به درستی منفی پیش بینی شده اند.
2. False Negative (FN): تعداد نمونه هایی که به اشتباه منفی پیش بینی شده اند (مقدار حقیقی آن ها مثبت می‌باشد.).
3. True Positive (TP): تعداد نمونه هایی که به درسی مثبت پیش بینی شده اند.
4. False Positive (FP): تعداد نمونه هایی که اشتباه مثبت پیش بینی شده اند (مقدار حقیقی آن ها منفی می‌باشد.).

این مقادیر را می‌توان به صورت ماتریسی نمایش داد:



شکل 1-ماتریس سردرگمی

باتوجه به ماتریس سردرگمی:

دانشگاه صنعتی امیرکبیر
دانشکده مهندسی برق

$$TP + FP = Predicted\ P$$
$$FN + TN = Predicted\ N$$
$$TP + FN = Actual\ P$$
$$FP + TN = Actual\ N$$

صحت، تمامیت، دقت و F_1

مفاهیم صحت (precision)، دقت(accuracy)، و تمامیت(recall این ترجمه برای مفهوم بهتر انتخاب شده است.) را نیز می‌توان به صورت زیر تعریف کرد:

$$precision = \frac{TP}{TP + FP} = \frac{TP}{Predicted\ P}$$
$$recall = \frac{TP}{TP + FN} = \frac{TP}{Actual\ P}$$
$$accuracy = \frac{TP + TN}{TP + FN + TP + FP}$$

صحت مشخص می‌کند چه کسری از میان داده هایی که مثبت پیش بینی شده اند واقعا مثبت بوده اند.

تمامیت مشخص می‌کند چه کسری از میان کل داده های مثبت درست پیش بینی شده اند (تمامیت پیش بینی های مثبت).

دقت مشخص می‌کند چه کسری از کل داده ها درست پیش بینی شده اند.

برای ایجاد تعادلی میان دو تعریف recall و precision مفهومی به عنوان معیار F_1 را به صورت زیر تعریف می‌کنیم که میانگین همساز(Harmonic mean) از این دو مقدار می‌باشد:

$$F_1 = \frac{2}{\frac{1}{recall} + \frac{1}{precision}} = \frac{2.precision.recall}{preicision + recall}$$

ارزیابی مدل (Metrics)

عملکرد مدل با استفاده از معیارهای زیر ارزیابی می‌شود:

دانشگاه صنعتی امیرکبیر
دانشکده مهندسی برق

- از میانگین خطای مطلق (MAE) و یا میانگین مربعات خطا (MSE) به عنوان loss استفاده شود.
- درصورتی که از یک تابع loss جدید (Customized) استفاده می‌کنید، شرح تابع و توضیحات مربوطه الزامی است.

در بخش آخر توابع confusion matrix و accuracy و f1-score و precision و recall به عنوان خروجی شبکه نمایش داده شود.

خواسته های فاز ۱

۱. در ابتدا مراحل preprocess باید به شکل کامل توضیح داده شود.
۲. در بخش بعدی شبکه ای که استفاده می کنید و ساخته شده برای آموزش در گزارش کار توضیح داده شود.
۳. بعد از آموزش در یک نمودار هم train loss و هم validation loss نمایش داده شود.
۴. همچنین نمودار accuracy, f1-score, recall, precision بر حسب Epoch کشیده شود.
۵. confusion matrix ترین و تست شبکه را نمایش دهید.
۶. در نهایت، وزن‌های شبکه آموزش دیده نهایی را ذخیره نمایید. مدل در زمان ارایه ممکن است بررسی شود.

دانشگاه صنعتی امیرکبیر
دانشکده مهندسی برق

به نام خدا

پروژه درس مقدمه ای بر هوش محاسباتی

پردازش زبان طبیعی

فاز دوم

NER چیست؟

شناسایی موجودیت‌های نامدار (Named Entity Recognition) فرآیندی در پردازش زبان طبیعی است که هدف آن شناسایی و دسته‌بندی کلمات یا عبارات خاص در متن به موجودیت‌های مشخص است. این موجودیت‌ها می‌توانند شامل نام افراد، سازمان‌ها، مکان‌ها، زمان‌ها و ویژگی‌های زبانی (مانند فعل ، اسم، صفت و ...) باشند. NER به ما کمک می‌کند تا اطلاعات مهم را از متن استخراج کنیم و در کاربردهای مختلفی مانند موتورهای جستجو، پرسش و پاسخ و تحلیل داده‌ها استفاده شود.

به عنوان مثال، در جمله زیر:

"باراک اوباما رئیس‌جمهور ایالات متحده بوده است."

- "باراک اوباما" ⟶ **PERSON** (شخص)
- "ایالات متحده" ⟶ **LOCATION** (مکان)

در اینجا، NER با شناسایی "باراک اوباما" به عنوان یک شخص و "ایالات متحده" به عنوان یک مکان، اطلاعات مهمی را از جمله استخراج می‌کند. این نوع شناسایی به ما کمک می‌کند تا متوجه شویم که چه کسی در جمله مورد بحث است و در کجا قرار دارد.

با استفاده از روش‌های مختلف یادگیری ماشین و پردازش زبان طبیعی، NER می‌تواند به طور خودکار موجودیت‌های نامدار را شناسایی کرده و آن‌ها را دسته‌بندی کند.

برای انجام پروژه شناسایی موجودیت‌های نامدار (NER) با استفاده از RNN بر روی مجموعه داده CoNLL-2003، مراحل زیر را دنبال کنید:

دانشگاه صنعتی امیرکبیر
دانشکده مهندسی برق

به نام خدا

پروژه درس مقدمه ای بر هوش محاسباتی

پردازش زبان طبیعی

آشنایی با مجموعه داده CoNLL-2003

مجموعه داده CoNLL-2003 شامل متون خبری است که به دو زبان انگلیسی و آلمانی در دسترس است. این مجموعه داده شامل سه بخش اصلی است: داده‌های آموزشی، داده‌های توسعه و داده‌های تست.
موجودیت‌ها به چهار دسته اصلی تقسیم می‌شوند: اشخاص (PER)، مکان‌ها (LOC)، سازمان‌ها (ORG) و سایر موارد (MISC).
برای بارگیری داده های فاز دوم و فاز سوم، لطفا کتابخانه زیر را نصب کنید:

```
pip install datasets
```

در صورتی که می‌خواهید در گوگل کولب پروژه را انجام دهید کد زیر را در یک cell اجرا کنید:

```
!pip install datasets
```

پس از نصب این کتابخانه، می‌توانید بصورت زیر به داده ها دسترسی پیدا کنید:

```
from datasets import load_dataset
data = load_dataset("conll2003")
```

پس از دسترسی به داده ها لطفا از مراحل زیر اطمینان حاصل کنید.

ایجاد دیکشنری

یک دیکشنری برای نگاشت توکن‌ها به شناسه‌های عددی ایجاد کنید.

پدینگ

اطمینان حاصل کنید که تمام جملات دارای طول یکسانی هستند، بنابراین ممکن است نیاز به پدینگ داشته باشید.

طراحی مدل RNN

یک مدل RNN طراحی کنید که به فرم sequence to sequence باشد. از آنجایی که متن ورودی شبکه با خروجی هم اندازه است(یعنی با یک مساله many to many سروکار داریم)، می‌توانید از مدل encoder, decoder های RNN که برای متن هستند استفاده فرمایید. استفاده از GRU و LSTM برای encoderیا

دانشگاه صنعتی امیرکبیر
دانشکده مهندسی برق

decoder بلامانع است. (دقت فرمایید که این معماری ها به زمان بیشتری برای آموزش نیاز دارند). لازم به یاد آوری است که باید حتما قبل از دادن ورودی به RNN یک مرحله Embedding در ابتدای مدل ضروریست.
داده های ورودی به صورت زیر هستند:

```
['EU', 'rejects', 'German', 'call', 'to', 'boycott', 'British', 'lamb', '.']
```

بدیهی است که این ورودی بایستی با استفاده از دیکشنری ساخته شده به لیستی از word_id تبدیل شود تا بتواند به شبکه داده شود.
خروجی به فرم زیر خواهد بود:

```
[3, 0, 7, 0, 0, 0, 7, 0, 0]
```

این داده خروجی نیازی به دستکاری ندارد و برای آموزش شبکه آماده است.

آموزش مدل

- پارامترهای آموزش مانند تعداد دوره‌ها (epochs)، اندازه دسته (batch size) و نرخ یادگیری (learning rate) را تعیین کنید.
- مدل RNN خود را با استفاده از داده‌های آموزشی آموزش دهید و از داده‌های توسعه برای تنظیم پارامترها استفاده کنید.

ارزیابی مدل

پس از آموزش، مدل را بر روی داده‌های تست ارزیابی کنید.

- معیارهای صحت (accuracy)،دقت (precision)، یادآوری (recall) و امتیاز F1 (F1 score) را نسبت به Epoch محاسبه کنید تا عملکرد مدل خود را بررسی کنید.

دانشگاه صنعتی امیرکبیر
دانشکده مهندسی برق

به نام خدا

پروژه درس مقدمه ای بر هوش محاسباتی

پردازش زبان طبیعی

خواسته‌های فاز ۲

نتایج پیش‌بینی شده را با برچسب‌های واقعی مقایسه کنید. در صورتی که مدل شبکه عملکرد خوبی نداشت، دلایل منطقی برای عملکرد نامناسب مدلتان را ارایه دهید.

1. در بخش بعدی شبکه ای که استفاده می کنید و ساخته شده برای آموزش در گزارش کار توضیح داده شود.
2. بعد از آموزش در یک نمودار هم train loss و هم validation loss برحسب Epoch نمایش داده شود.
3. همچنین نمودار accuracy, f1-score, recall, precision برحسب Epoch کشیده شود.
4. confusion matrix ترین و تست شبکه را نمایش دهید.
5. در نهایت، وزن‌های شبکه آموزش دیده نهایی را ذخیره نمایید. مدل در زمان ارایه ممکن است بررسی شود.

دانشگاه صنعتی امیرکبیر
دانشکده مهندسی برق

فاز سوم

Federated Learning [1]

Federated Learning یک رویکرد یادگیری ماشین غیرمتمرکز است که در آن چندین مشترک (به عنوان مثال، دستگاه های تلفن همراه) به طور مشترک یک مدل را آموزش می دهند بدون اینکه داده های محلی خود را با Server مرکزی به اشتراک بگذارند. این رویکرد به نگرانی‌ها و محدودیت‌های مربوط به حفظ حریم خصوصی مرتبط با آموزش متمرکز سنتی می‌پردازد، به‌ویژه هنگامی که با مجموعه‌های داده حساس یا بزرگ سروکار داریم.

در اینجا یک راهنمای گام به گام ارائه شده است که فرآیند را شرح می دهد:

مرحله 1: مقداردهی اولیه

Server مرکزی یک Global Model را با وزن های تصادفی یا پارامترهای از پیش آموزش داده شده مقداردهی اولیه می کند.

1. server یک مدل ایجاد می کند و آن را به همه مشترک موجود می دهد. برای هر مشترک یک نمونه مدل با پارامترهای اولیه متفاوت ایجاد نکنید. همه مشترک ها با مقدار اولیه پارامتر مدل دقیقاً مقداردهی اولیه می شوند.
2. برای این پروژه Federated NER، هر مشترک را به عنوان یک سازمان با داده های مختلف در نظر بگیرید و داده های مربوطه را بارگذاری کنید.

مرحله 2: انتخاب مشترک

- در هر Communication Round، سرور زیر مجموعه ای از مشترک را برای شرکت در آموزش انتخاب می کند. این انتخاب می تواند تصادفی یا بر اساس عواملی مانند در دسترس بودن مشترک و توزیع داده باشد.
- توجه: هر مشترک داده های خاص خود را دارد. در شبیه سازی ما، همه مشترک ها را برای همه round ها حاضر در نظر بگیرید.

دانشگاه صنعتی امیرکبیر
دانشکده مهندسی برق

مرحله 3: توزیع مدل

Server پارامترهای مدل جهانی فعلی را به مشترکهای انتخاب شده ارسال می کند.

مرحله 4: آموزش محلی

·هر مشترک انتخاب شده، مدل سراسری دریافتی را در مجموعه داده محلی خود برای تعداد معینی از Epoch با Batch Size تعریف شده آموزش می دهد. فرآیند آموزش محلی شامل تکرارهای متعدد به روز رسانی وزن مدل بر اساس گرادیان های محاسبه شده از داده های محلی است.

مرحله 5: به روز رسانی

·پس از تکمیل آموزش محلی، هر مشترک به روز رسانی های مدل خود (به عنوان مثال، وزن یا گرادیان مدل) را به سرور برمی‌گرداند. این به روز رسانی ها منعکس کننده تغییرات ایجاد شده در مدل جهانی بر اساس داده های محلی مشترکها هستند.

مرحله 6: جمع آوری مدل‌ها

Server به‌روزرسانی‌های دریافتی را از همه مشترکهای شرکت‌کننده جمع‌آوری می‌کند. بدین ترتیب، یک مدل جهانی جدید ایجاد می‌شود که دانش به دست آمده از همه مشترکها را در بر می گیرد.

1. روش های تجمیع جدید مانند FedNova،Fedprox و غیره وجود دارد. همه دانش آموزان باید از FedAvgاستفاده کنند.
2. روش تجمیع اغلب برای سادگی استراتژی نامیده می‌شود.

مرحله 7: تکرار

·مراحل 2-6 برای چندین مرحله تکرار می شوند تا زمانی که مدل جهانی به سطح دقت مطلوبی همگرا شود.

این فرآیند تضمین می کند که سرور مرکزی هرگز مستقیماً به داده های آموزشی خام دسترسی پیدا نمی‌کند و خطرات حفظ حریم خصوصی را به حداقل می رساند. مدل جهانی در هر دور با ترکیب مجموعه داده های مشتری متنوع بدون به خطر انداختن حریم خصوصی داده ها بهبود می یابد.

دانشگاه صنعتی امیرکبیر
دانشکده مهندسی برق

به نام خدا

پروژه درس مقدمه ای بر هوش محاسباتی

پردازش زبان طبیعی

پیش پردازش داده‌ها

مشابه فاز دوم بصورت زیر داده ها را بازخوانی کنید:

```
from datasets import load_dataset
data_مشترک1 = load_dataset("conll2003")
data_مشترک2= load_dataset("DFKI-SLT/few-nerd", "supervised")
data_مشترک3 = load_dataset("rjac/kaggle-entity-annotated-corpus-ner-dataset")
```

دقت کنید که این داده ها اندازه های متفاوتی دارند. مثلا داده مشترک1 در حدود 15000 نمونه و داده مشترک2در حدود 130,000 نمونه دارد. در صورتی که همه داده ها را استفاده کنید زمان آموزش بسیار طولانی خواهد شد و قاعدتا شبکه نسبت به داده های مشترک2 بیشتر fit خواهد شد که مطلوب نیست.
بنابراین از هر مجموعه داده، 5000 نمونه اول را برای آموزش و 1000 نمونه بعدی را برای تست جدا کنید.
نکته: مجموعه داده دوم دو نوع موجودیت دارد. یکی موجودیت های کلی ،دوم موجودیت‌های جزیی. شما فقط از موجودیت‌های کلی استفاده نمایید.
موجودیت کلی مثلا : اسم شخص
موجودیت جزیی مثلا : اسم شخص بازیگر، اسم شخص سیاستمدار و...

نکته بسیار مهم ۱: برای ساختن دیکشنری کلمات، لازم است که یک دیکشنری واحد برای تمام مشترک ها داشته باشید. بنابراین برای ساختن دیکشنری بایستی قبل از دادن word id از هر مجموعه داده یک دیکشنری بگیرید و از سه دیکشنری حاصل، یک دیکشنری واحد جهانی برای تمام مشترک ها بسازید.
نکته بسیار مهم ۲: از انجایی که ممکن است این سه مجموعه داده در entity های خود باهم اشتراک داشته باشند، لازم است یک لیست برای شماره گذاری تمام entityهای داده ها داشته باشید. به عنوان مثال ممکن است مجموعه داده اول موجودیت شخص را شماره ۱ در نظر داشته باشد ولی مجموعه داده ۲ شخص را به عنوان شماره 8 کد کرده باشد. بنابراین بایستی label های مجموعه داده ها را دستکاری کنید.
لینک هر سه مجموعه داده به ترتیب به صورت زیر است. برای پیدا کردن کد موجودیت ها به این لینک ها مراجعه کنید.

https://huggingface.co/datasets/tner/conll2003
https://huggingface.co/datasets/DFKI-SLT/few-nerd

دانشگاه صنعتی امیرکبیر
دانشکده مهندسی برق

به نام خدا

پروژه درس مقدمه ای بر هوش محاسباتی

پردازش زبان طبیعی

https://huggingface.co/datasets/rjac/kaggle-entity-annotated-corpus-ner-dataset

خواسته های فاز ۳

1. داده های پیش پردازش شده را تنظیم کنید و یک تابع برای بارگیری داده های test و train برای هر مشترک ایجاد کنید.
   - این تابع را <LoadDataLoaders> صدا کنید.
   - این تابع یک شناسه مشترکid_ (که می تواند یک عدد صحیح باشد) دریافت می کند.
   - بر اساس مشترکId ، تابع باید مجموعه داده های مربوطه را برای مشترک بارگذاری کند.
   - تابع یک <tuple> مانند(<train_dataLoader> ، <test_dataLoader>) را خروجی می دهد.
2. یک تابع ایجاد کنید که حاوی حلقه آموزشی باشد.
   - این تابع یک<net instance> ، <train_dataLoader>، <test_dataLoader>، تعداد <epochs>را دریافت می کند.
     - Object هایی را ایجاد می کند که برای آموزش استفاده می شوند، مانند توابع <loss> و <Optimizer>. این وابستگی ها باید در همه مشترک ها مشابه باشند.
   - تابع نمونه شبکه آموزش دیده را برمی‌گرداند.

3. یک کلاس <Server> ایجاد کنید که به عنوان <server> مرکزی عمل می کند. از آنجایی که ما فقط رفتار یادگیری فدرال را شبیه سازی می کنیم، اطمینان حاصل کنید که داده های هر> مشترک <توسط سایر مشترکها استفاده نمی‌شود. تابع‌های ایجاد شده در مراحل قبلی در این کلاس قابل استفاده هستند.

4. تمام نتایج و معیار های استفاده شده در فاز های قبلی را برای مدل جهانی در هر round ترسیم کنید.

دانشگاه صنعتی امیرکبیر
دانشکده مهندسی برق

به نام خدا

پروژه درس مقدمه ای بر هوش محاسباتی

پردازش زبان طبیعی

موارد تحویلی :

- فایل تحویل پروژه می بایست یک فایل زیپ شامل فایل word و pdf گزارش پروژه به همراه تمام کدهای اجرا شده و به فرمت ipynb یا py، مراجع مورد استفاده و دادهها باشد.
- تحویل کدهای مربوط به سلسله مراحل و جزییاتی که توضیح داده شده اند الزامی است. مثلا در فاز اول مراحل پیش پردازش توضیح داده شده بایستی به همان ترتیب توضیح داده شده، تحویل داده شوند.
- در روز ارایه حتما نسخه های آموزش دیده برای هر شبکه و هر فاز را ذخیر داشته باشید تا عملکرد هر شبکه در هر فاز با داده های جدید مورد ارزیابی قرار می گیرد.

مراجع

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, 2017: PMLR, pp. 1273-1282.