به نام خدا

نام دانشگاه: پردیس فارابی دانشگاه تهران

نام درس: پردازش زبان های طبیعی

نام تمرین: News Classification

دانیال فرهنگی ۲۲۰۷۹۸۰۷۸

ورودی ۹۸

تمرین **News Classification**:

آدرس ریپازیتوری گیتهاب: https://github.com/danfarh/news-classification

.....

در ابتدا فایل Hamshahri.zip را از گوگل درایو در گوگل کولب آپلود میکنیم و سپس با کمک کتابخانه‌هایی که در پایتون وجود دارد فایل را extract میکنیم.

اکنون به فایل corpus و stopwordها دسترسی داریم.

کتگوری‌ها و متن‌ها را از corpus جدا کردیم و مرتب درون یک فایل csv ریختیم و تمام stopwordها را از آن حذف کردیم سپس شبکه عصبی طراحی کردیم که به دقت ۹۰ درصد و لاس ۰.۳ رسیدیم. که درباره نتایج و پیش‌بینی‌های مدل‌ها، در ادامه توضیح خواهم داد.

شرح:

در ابتدا فایل زیپ شده را از گوگل درایو در گوگل کولب آپلود میکنیم و آن را extract میکنیم سپس به فایل corpus دسترسی داریم؛ کتگوری ها و متن‌ها را از هم جدا میکنیم و تمام stopwordها را از آن حذف میکنیم و سپس درون یک فایل csv به صورت مرتب‌شده ذخیره میکنیم.

فایل csv را با pandas می‌خوانیم.

از آنجایی که شبکه عصبی حروف نمیپذیرد بنابراین باید تمام این کلمات را به عدد تبدیل کنیم یعنی به هر کلمه یک عدد نسبت می‌دهیم برای این کار از Tokenizer کراس استفاده کرده‌ایم با این کار تمام کلمات یک آیدی به خود می‌گیرند و همچنین میتوانیم تعداد تکرار هر کلمه در متن و تعداد کلمات یکتا را بدست آوریم.

پس متن‌ها را تبدیل به عدد میکنیم و چون که ممکن است طول متن‌ها با هم متفاوت باشد پس باید یک padding درنظر بگیریم تا طول آنها یکسان شود اکنون این ماتریس را به عنوان ورودی به شبکه عصبی می‌دهیم.

و به عنوان خروجی باید کتگوری‌های هر متن را بدهیم اما چون که شبکه کلمه نمی‌پذیرد پس کتگوری‌ها را وان‌هات میکنیم و به شبکه می‌دهیم و سپس شبکه را train میکنیم.

معماری شبکه:

همان‌طور که در تصویر مشاهده می‌کنید به روش sequential مدل را نوشته‌ام در لایه اول از لایه Embedding استفاده کرده‌ام که برای هر کلمه یک وکتور ۵۰تایی در نظر گرفته‌ام و در لایه‌ی بعدی یک LSTM دوطرفه در نظر گرفته‌ام و بعدش max_pooling زده‌ام و برای جلوگیری از overfit از لایه Dropout استفاده کرده‌ام و در لایه آخر هم یه لایه‌ی Dense گذاشتم به اندازه‌ی تعداد

کتگوری‌ها و AF آن را softmax گذاشته‌ام که به عنوان خروجی یک تابع احتمال از کتگوری‌ها می‌دهد و آن کتگوری که بیشترین احتمال را دارد مربوط به آن متن ورودی است.

```
model = Sequential()
model.add(Embedding(vocab_size, embedding_dim, input_length=maxlen))
model.add(Bidirectional(LSTM(200, return_sequences=True)))
model.add(GlobalMaxPooling1D())
model.add(Dropout(0.3))
model.add(Dense(200, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_unique_categories, activation='softmax'))
```

```
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

بعد از ۵ ایپاک به دقت ۹۰ درصد و لاس ۰.۳ رسیدم، اکنون به سراغ تست و ارزیابی مدل میرویم.

تست و ارزیابی مدل:

اکنون یک تابع prediction همان‌طور که در تصویر مشاهده می‌کنید برای مدل ترین شده می‌نویسیم که یک آرایه‌ای از textها از کاربر می‌گیرد و کتگوری آن متن‌ها را پیش‌بینی می‌کند.

پس دقیقا همان مراحلی که در preprocess روی متن‌ها انجام دادیم دوباره روی متنی که به عنوان ورودی دریافت میکنیم انجام می‌دهیم یعنی تمام stopword ها را حذف میکنیم و تبدیل به عدد میکنیم و padding درنظر می‌گیریم، اکنون می‌توانیم به مدل بدهیم تا پیش‌بینی کند، پس از model.predict استفاده می‌کنیم و به عنوان خروجی یک تابع احتمال از کتگوری را برمی‌گرداند، باید max را پیدا کنیم و index آن را در لیست کتگوری‌ها پیدا کنیم و کتگوری مربوطه را برگردانیم.

Prediction

```
def preprocess(texts, stopwords):
  cleaned_texts = []
  for text in texts:
    word_tokenized = word_tokenize(text)
    word_tokenized_filtered = [w for w in word_tokenized if w not in stopwords]
    cleaned_text = ' '.join(map(str, word_tokenized_filtered))
    cleaned_texts.append(cleaned_text)
  return cleaned_texts
```

```
MAX_LENGTH = 1843
def prediction(texts, stopwords):
  result = []
  cleaned_texts = preprocess(texts, stopwords)
  for cleaned_text in cleaned_texts:
    text_to_sequences = tokenizer.texts_to_sequences([cleaned_text])
    pad_text_to_sequences = pad_sequences(text_to_sequences, maxlen=MAX_LENGTH)
    category = model.predict([pad_text_to_sequences])[0]
    category_index = np.where(category == max(category))
    result.append(category_types[category_index])
  return result
```

نتایج:

```
example1 = 'اعتبارنامه کمیسیون تحقیق مطرح نشد وحمید بهرامي سخنان موردي اشاره نکرد حالي اعتبارنامه نمایندگان رفسنجان جلسه علني موارد رد'
example2 = 'سال گذشته فصل جدید تیمي حدیث تیم درجه اول کشور بینیم شکست پیروزي هفته اول دلیل ادعا بازي استقلال ماشین سازي دقیق بنگریم'
```

```
prediction([example1, example2], stopwords)
```

```
[array(['siasi'], dtype=object), array(['vrzsh'], dtype=object)]
```

```
example_texts = ["این نقشها در تقابل بصري با  مورچه - انسانهاست اما در يك هماهنگي دروني با مفهوم مورچه - انسانها به تصوير كشيده شده اند"
```

```
prediction(example_texts, stopwords)
```

```
[array(['adabh'], dtype=object)]
```

```
prediction([data['text'][1]], stopwords)
```

```
[array(['adabh'], dtype=object)]
```

```
prediction([data['text'][1000]], stopwords)
```

```
[array(['siasi'], dtype=object)]
```

```
data['category'][1000]
```

```
'siasi'
```