Test classification based on GLUE benchmark

Костенок Елизавета
kostenok.en@phystech.edu

1 Идея

В задании требуется обучить классификатор на задачах CoLA, RTE и SST-2. Краткое описание задач и используемые для них метрики качества в таблице ниже:

Data	Task	Metric
CoLA	Sentence is grammatical or not grammatical	Matthews
SST-2	Review is positive, negative or neutral	Accuracy
RTE	Sentence 1 -> Sentence 2?	Accuracy

Table 1: Tasks summary

План решения был такой: сначала для каждой задачи получить baseline, используя модели BERT и DistilBERT с дефолтными параметрами, посмотреть, какое получится качество, а затем улучшить его подбором гиперпараметров, усовершенствованием классификатора и дообучением модели.

1.1 Baseline

Для начала применила классический подход к классификации текстов: использовать предобученную BERT-подобную сетку как feature extractor, чтобы получить некое информативное представление текстовых данных, на котором уже можно обучить сам классификатор: линейную модель или нейросеть. Я использовала предобученные модели BERT и DistilBERT из Hugging Face (добавила модель DistilBERT, так как она выигрывает в скорости и размере у классического BERT, практически не проигрывая в качестве и было интересно их сравнить). В качестве классификатора использовала Logistic Regression из sklearn.

1.2 Усовершенствования

- GridSearch по сетке параметра регуляризации для логистической регрессии.
- Использование нейросети в качестве классификатора. Так как датасеты небольшие (менее 10000 примеров), то модель должна быть сравнительно простой во избежание переобучения. Я написала классификатор из трех линейных слоев и активаций ReLU между ними с Log_softmax на выходе.
- Fine-tuning для предобученных языковых моделей.

2 Эксперименты

2.1 CoLA

Model	Train Matthews correlation	Test Matthews correlation
BERT + Default Logistic Regression	0.53	0.38
BERT + Logistic Regression with applied GridSearch	0.52	0.41
BERT + Custom Classifier	0.77	0.39
BERT + Fine-tuning	0.87	0.56

Table 2: CoLA results

Значения метрики baseline модели (BERT+Logistic Regression) получились низкими, а использование нейросети как классификатора(Custom Classifier) привело к переобучению и улучшению метрики только на тренировочной выборке. Чтобы получить хорошее значение и на тесте, требовалась тонкая подстройка параметров BERT. После 4 итераций дообучения модели качество предсказания сравнялось с опубликованными результатами для fine-tuned BERT (0.56 для тестовой выборки).

2.2 SST-2

Результаты представлены в таблице ниже: Сначала я использовала для получения признакового описания

Model	Train accuracy	Test accuracy
DistilBert + Default Logistic Regression	0.88	0.85
DistilBert + Logistic Regression with applied GridSearch	0.88	0.86
DistilBert + Custom Classifier	0.97	0.85
BertForSequenceClassification + Fine-tuning	0.98	0.93

Table 3: SST2 results

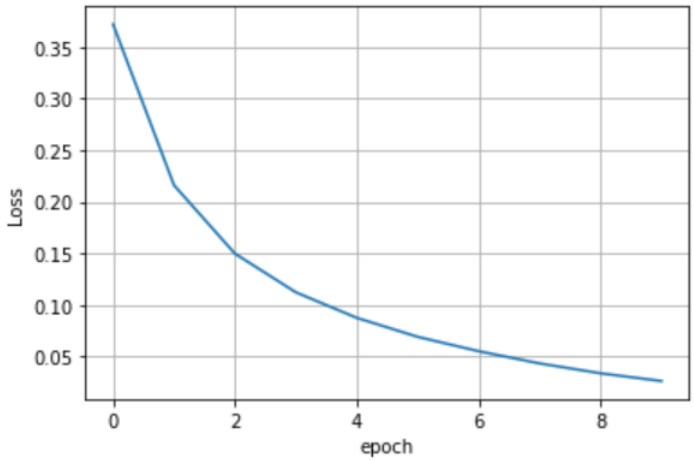

Figure 1: Training loss for Custom Classifier(SST2 task)

текстов и BERT, и DistilBERT, но т.к. на baseline моделях отличия метрик оказались в пределах сотой, продолжила эксперименты с более эффективной по времени моделью DistilBERT. Результаты, которые публиковались по этой задаче: DistilBERT достигает accuracy 0.91, BERT - accuracy 0.95. Эксперименты с классификатором показали, что его усовершенствование дает сильную прибавку в качестве для тренировочной выборки, но не для тестовой. Как и для прошлой задачи, fine-tuning улучшил признаковое описание текста и позволил добиться отличных значений по метрике: 0.98 на тренировочной и 0.93 на тестовой выборках.

2.3 RTE

Так как задача определения смыслового отношения пары предложений более сложная, чем классификации одного из них, то baseline, который сработал для предыдущих задач, не подходит, решила сразу дообучать модели. BERT дообучался 3 эпохи, пока возрастало значение метрики на валидационной выборке, на тестовой выборке получилось значение $accuracy = 0.77$. Чтобы улучшить результат, далее работала с усовершенствованной моделью - RoBERTa, которая эффективнее обучается и значительно превосходит BERT на задаче RTE по опубликованным данным (0.7 для BERT, 0.86 для RoBERTa). Так получилось и в моем эксперименте: для второй модели уменьшился эффект переобучения на первых эпохах, что позволило учить модель дольше и улучшить значения метрики и на тренировочной, и на тестовой выборках относительно первой модели (см. таблицу 3).

Model	Train accuracy	Test accuracy
BertForSequenceClassification + Fine-tuning	0.87	0.77
RobertaForSequenceClassification + Fine-tuning	0.92	0.80

Table 4: RTE results

3 Итог

Самым эффективным методом обучения моделей для классификации текстов стал fine-tuning предобученных языковых моделей, с его помощью получились значения метрики, близкие к опубликованным результатам state-of-the-art моделей на этих задачах.

4 Источники

Очень помогли примеры загрузки предобученных моделей и подробное описание моделей с сайта Hugging Face; использовала материалы с семинаров курса машинного обучения МФТИ по классификации текстов https://github.com/girafe-ai/ml-mipt для реализации baseline, потому что ранее не работала с NLP моделями, но усовершенствования, которые давали основной прирост к метрике относительно baseline, реализовывала самостоятельно.