Секція: Технічні науки

Білий Михайло Дмитрович

Харківський національний університет радіоелектроніки

м. Харків, Україна

МЕТОДИ МАШИННОГО НАВЧАННЯ ДЛЯ КЛАСИФІКАЦІЇ ТЕКСТУ НА НЕЗБАЛАНСОВАНИХ ДАНИХ (АЛГОРИТМ SMOTE ТА WORD EMBEDDINGS)

The given work is devoted to the modern developments in the field of machine learning. Particularly, methods to train model on the unbalanced dataset to classify tweets on 13 emotional classes. In the article is consider algorithm SMOTE (Synthetic Minority Over-sampling Technique) and word embedding GloVE.

Однією з великих та складних задач машинного навчання – це обробка природної мови (NLP). Такою являється класифікація тексту, але проблема ще в тому, що якщо класів багато, то за частіше даних не вистачає деяких класів, але зробити наївно та природно, так як с картинками: повертання, дзеркальне відображення, додавання шуму тощо, - з текстом такого не зробити. Тому можемо застосувати алгоритм SMOTE.

Ця стратегія заснована на ідеї створення певної кількості штучних прикладів, які були б «схожі» на наявні в міноритарному класі, але при цьому не дублювали їх. Для створення нового запису знаходять різницю $d = X_b - X_a$, де X_a, X_b- вектори ознак «сусідніх» прикладів a і b з міноритарного класу. Їх знаходять, використовуючи алгоритм найближчого сусіда (KNN). В даному випадку необхідно і достатньо для прикладу b отримати набір з k сусідів, з якого в подальшому буде обрана запис b. Решта кроки алгоритму KNN не потрібні.

Далі з d шляхом множення кожного його елемента на випадкове число

в інтервалі (0, 1) отримують $\hat{d}$. Вектор ознак нового прикладу обчислюється шляхом додавання X_a і $\hat{d}$. Алгоритм SMOTE дозволяє задавати кількість записів, яке необхідно штучно згенерувати. Ступінь подібності прикладів a і b можна регулювати шляхом зміни значення k (числа найближчих сусідів). На рисунку 1 схематично зображено те, як в двовимірному просторі ознак можуть розташовуватися штучно згенеровані приклади.

Тепер можемо повернутися до нашої задачі – класифікації твітів на 13 емоційних класів. Є декілька підходів: мішок слів або послідовність слів з використанням Word Embedding GloVe, - ми будемо йти другим шляхом.

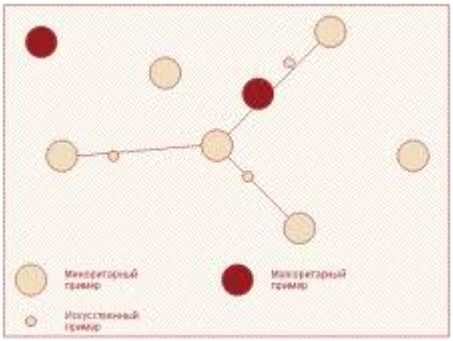

Рис. 1. Розташування штучно генерованих прикладів

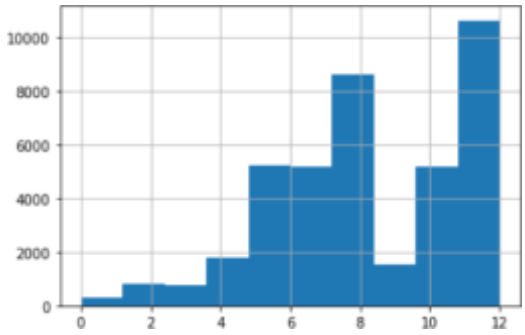

Рис. 2. Розподіл класів

Word Embedding – це векторне подання кожного слова, перевага його над «мішок слів» - це те що слова можуть корелювати між собою та синоніми у просторі будуть лежати близько.

Тепер можемо всі наші дані перетворити на послідовність з індексів слів, які відповідають індексу вектору слова в двовимірній таблиці Word Embeddings. Але розподіл даних по класам дуже не збалансований (рис. 2) . Тому використовуємо алгоритм SMOTE.

Все готово, щоби навчити модел, будемо використовувати Word Embedding, двонапревлену LSTM та повно зв'язаний шар для класифікації (код моделі на рис. 3)

```
inp = Input(shape=(maxlen,))
x = Embedding(max_features, embed_size, weights=[embedding_matrix],
trainable=False)(inp)
x = Bidirectional(LSTM(50, return_sequences=True, dropout=0.2,
recurrent_dropout=0.1))(x)
x = GlobalMaxPool1D()(x)
x = Dense(50, activation="relu")(x)
x = Dropout(0.2)(x)
x = Dense(13, activation="softmax")(x)
model = Model(inputs=inp, outputs=x)
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Рис. 3. Код моделі

В результаті після першої ж епохи отримано 0.923 точність. (табл. 1).

Таблиця 1

Порівняння методів

Input data	Accuracy
Bag of words	0.12-0.19
Word Embeddings	0.29-0.41
SMOTE + Word Embeddings	0.923

Повний код моделі з найліпшою метрикою доступний за посиланням: https://github.com/mishabeliy15/sentiment-analysis-tweets

Література

1. Stanford University: NLP Group. URL: https://nlp.stanford.edu/pubs/glove.pdf
2. Cornel University. URL: https://arxiv.org/pdf/1106.1813.pdf