Natural Language Processing

Яковенко Ольга

Задачи NLP

Задачи NLP

Без учителя

- Моделирование тематик
- Языковое моделирование
- Поиск дубликатов

С учителем:

- Распознавание тематик
- Распознавание сентиментов
- Обнаружение и исправление опечаток
- Машинный перевод
- Распознавание намерений (интентов)
- Обнаружение спама

Примеры

Сегодня отличная погода!

Отправь это письмо трём людям!

Здравствуйте, как сделать перевод?

Примеры

Сегодня отличная погода!

Отправь это письмо трём людям!

1

Здравствуйте, как сделать перевод?

0

Примеры

Сегодня отличная погода!

Отправь это письмо трём людям!

Здравствуйте, как сделать перевод?

Примеры

Сегодня отличная погода! Отправь это письмо трём людям!

Bugungi kunda ajoyib ob-havo! Ushbu maktubni uch kishiga yuboring!

Здравствуйте, как сделать перевод?

Salom, qanday qilib tarjima qilish kerak?

Примеры

Сегодня отличная погода!

Отправь это письмо трём людям!

Здравствуйте! Не пропустите свой шанс выиграть много денег!

Здравствуйте, как сделать перевод?

Здравствуйте, как зачислить деньги на карту?

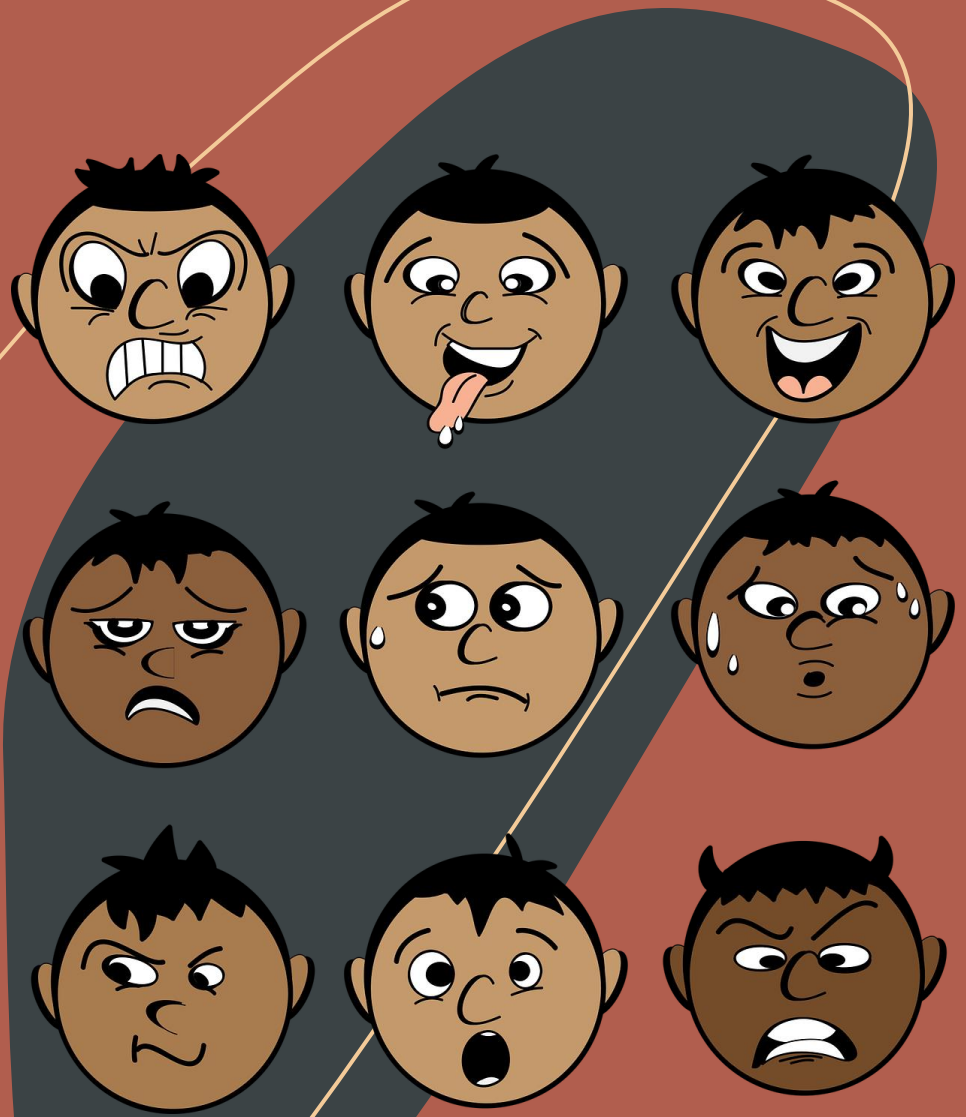
Примеры

Сегодня отличная погода! Отправь это письмо трём людям!

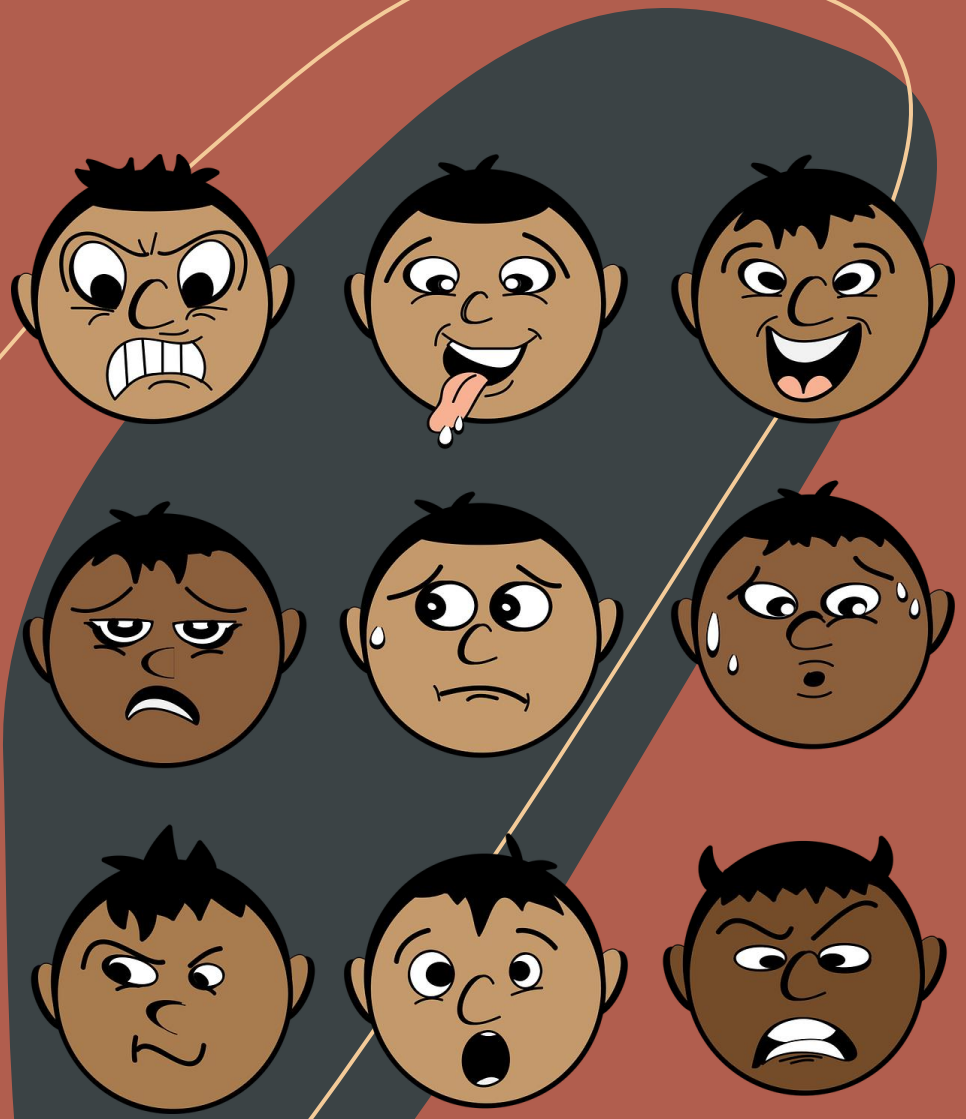
Здравствуйте! Не пропустите свой шанс выиграть много денег!

Здравствуйте, как сделать перевод?

Здравствуйте, как зачислить деньги на карту?



Sentiment recognition

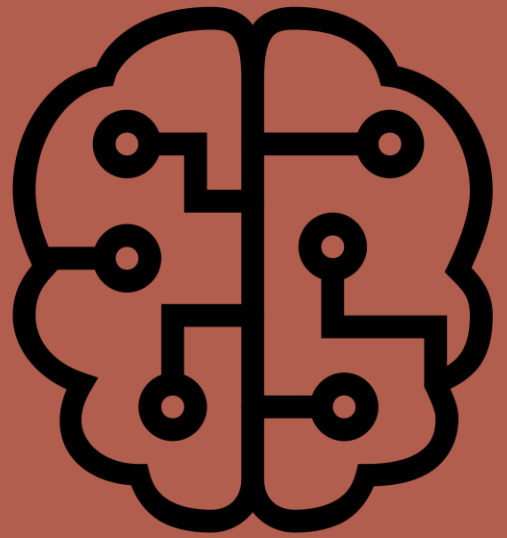


Sentiment recognition

Распознавание эмоциональности высказывания:

- Позитивное/негативное/нейтральное;
- Разновидности негативного (расизм, политика, уничижение соц и нац меньшинств, …)/нейтральное…

Подходы к решению



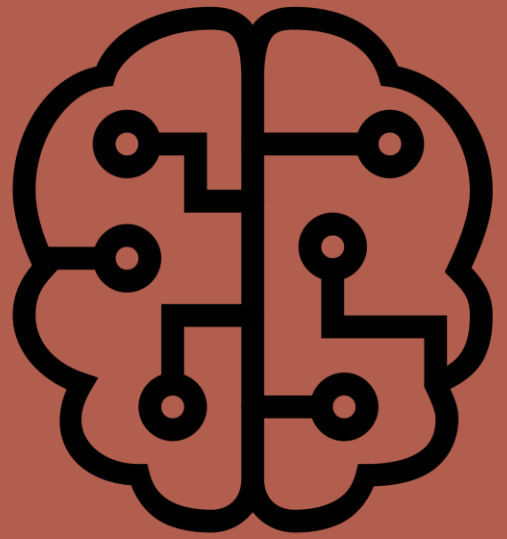
Подходы к решению

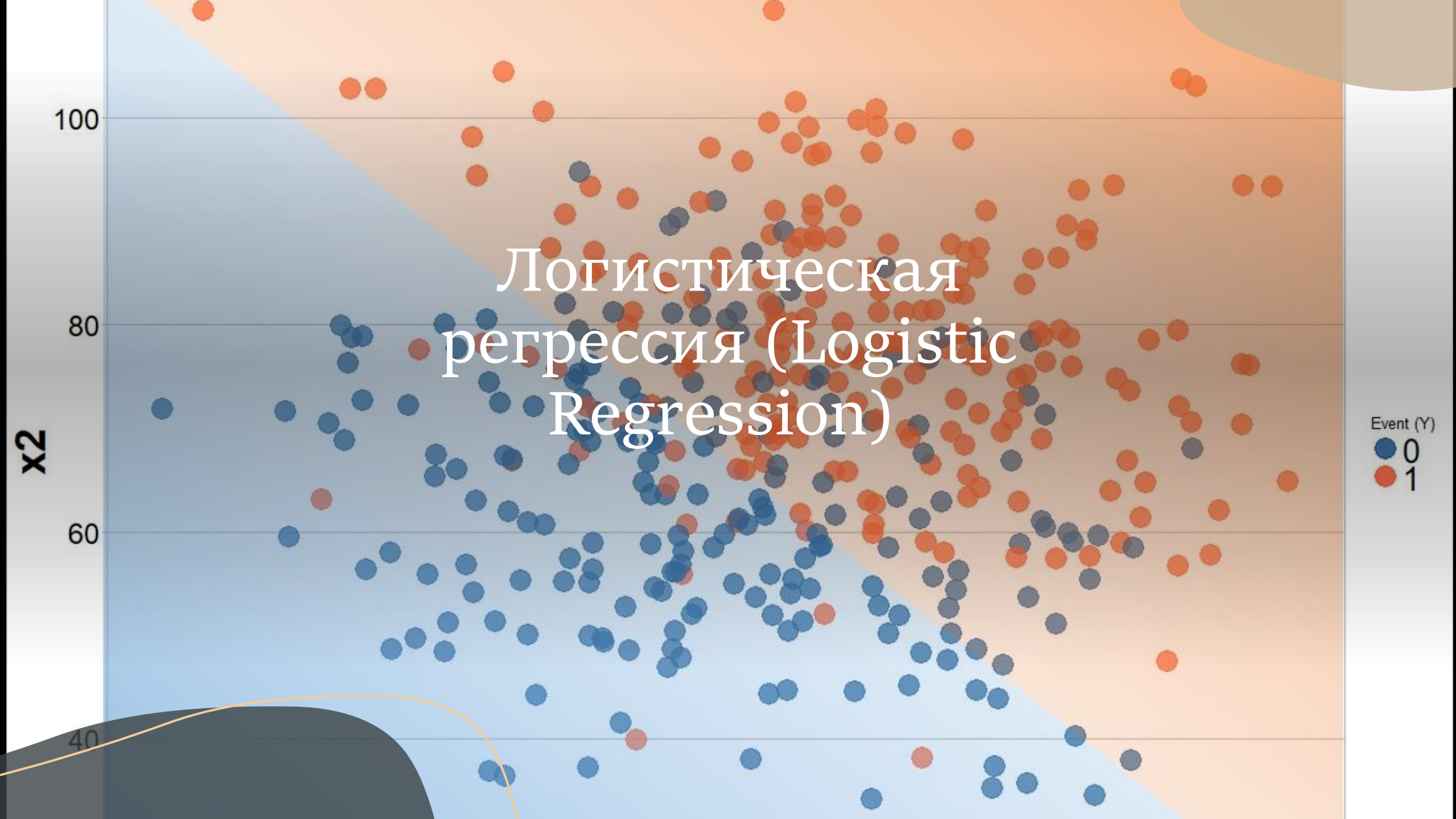
МОДЕЛЬ КЛАССИФИКАЦИИ

Логистическая регрессия (Logistic Regression) или полносвязная нейронная сеть (Multilayer Perceptron)

Свёрточная нейронная сеть (Convolutional Neural Network)

Рекуррентная нейронная сеть (Recurrent Neural Network)



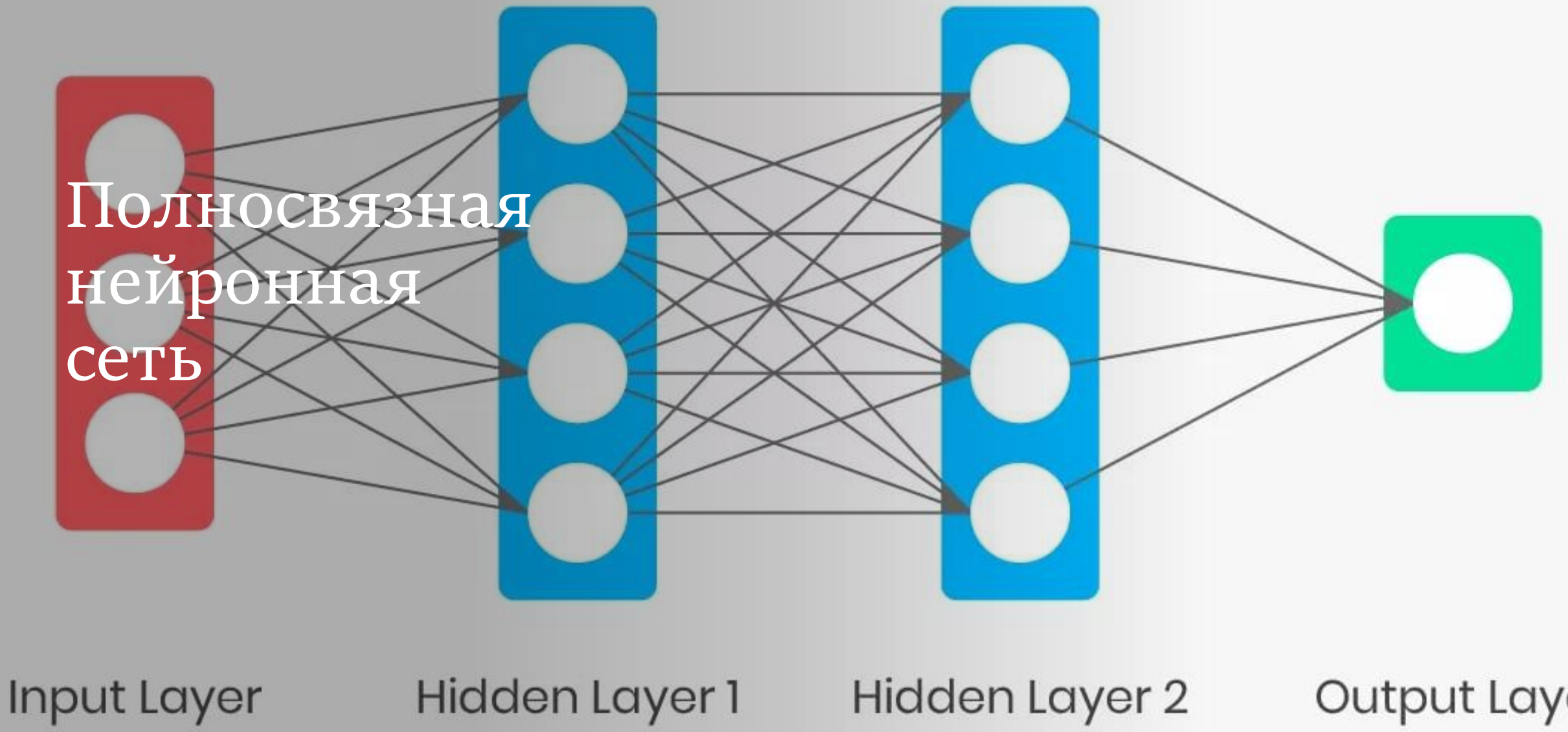

Логистическая регрессия (Logistic Regression)
100
80
60
40
x2
Event (Y)
0
1

Логистическая регрессия (Logistic Regression)

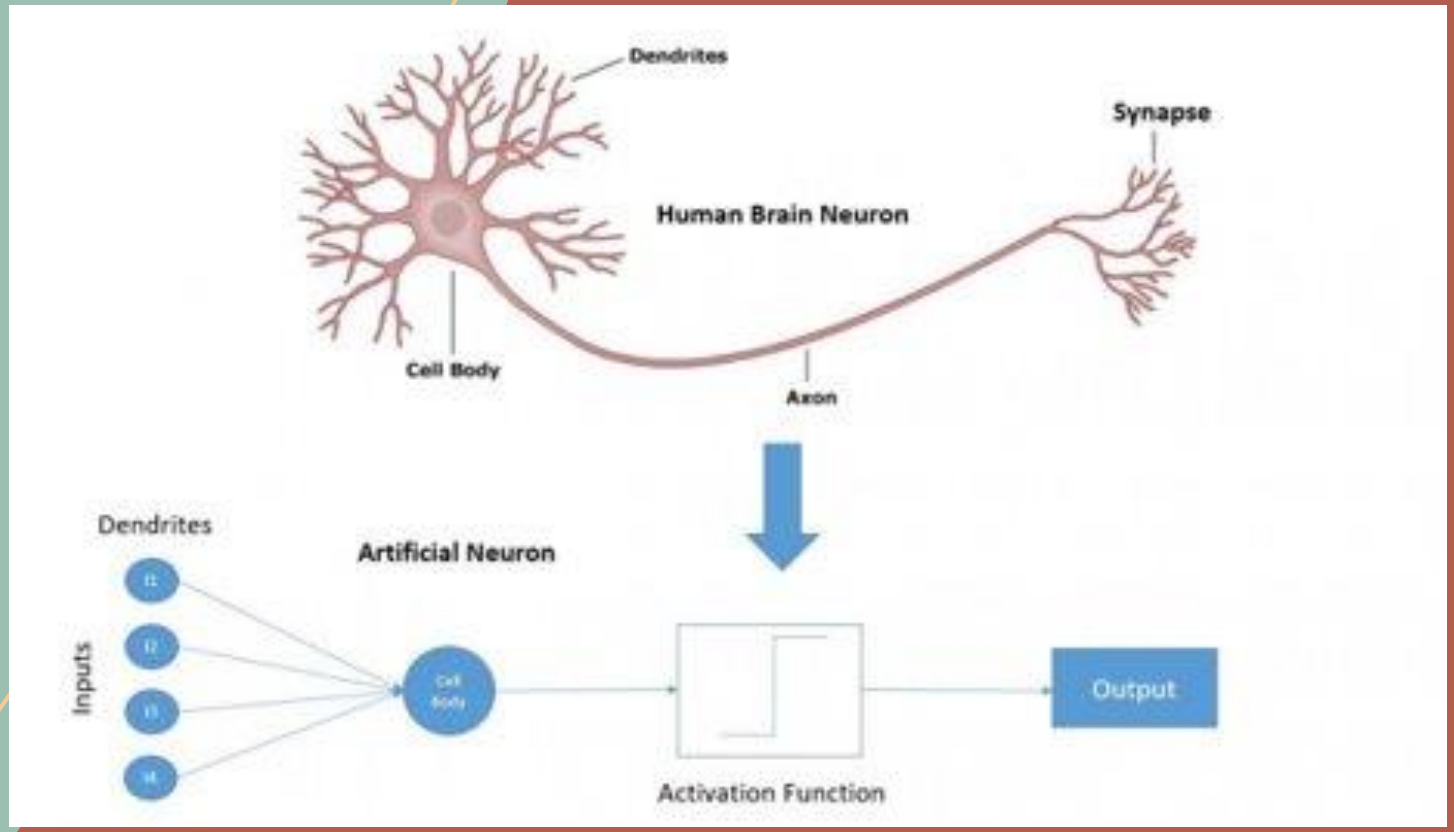
$$p(x) = \sigma(t) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

$$\min_{w,c} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \log(\exp(-y_i(X_i^T w + c)) + 1).$$

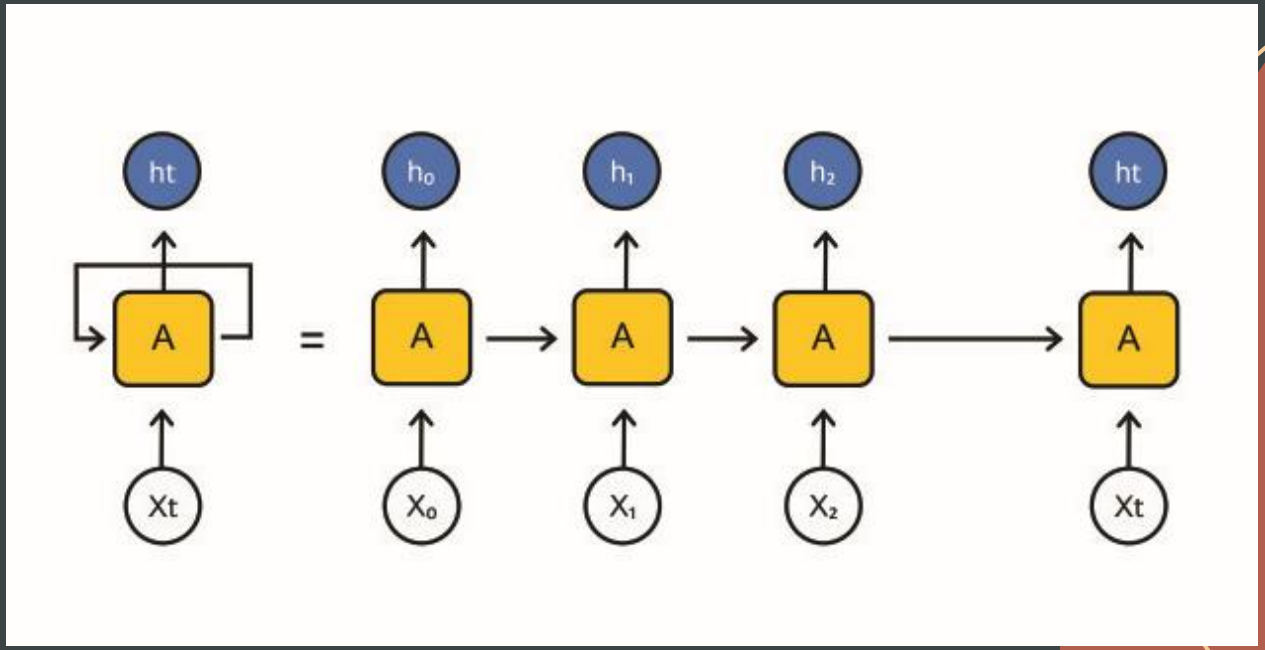
Полносвязная нейронная сеть



Полносвязная нейронная сеть



Свёрточная нейронная сеть



Рекуррентная нейронная сеть

Подходы к решению

Признаковое представление одного сэмпла	Модель классификации
1D вектор, представляющий целый документ (BoW, Tf-idf, усреднённые word2vec, bag of word2vec)	Логистическая регрессия (Logistic Regression) или полносвязная нейронная сеть (Multilayer Perceptron)
Матрица размера (max_n_words, feature_vector_size) – настаканные друг на друга эмбеддинги слов для представления текста (фиксированного размера)	Свёрточная нейронная сеть (Convolutional Neural Network)
Матрица размера (n_words, feature_vector_size) – настаканные друг на друга эмбеддингислов для представления текста (переменного размера)	Рекуррентная нейронная сеть (Recurrent Neural Network)

Задание

Как будет выглядеть матрица признаков в первом, втором и третьем случае?

Признаковое представление одного сэмпла
1D вектор, представляющий целый текст (BoW, Tf-idf, усреднённые word2vec, bag of word2vec)
Матрица размера (max_n_words, feature_vector_size)–настаканные друг на друга эмбеддинги слов для представления текста(фиксированного размера)
Матрица размера (n_words, feature_vector_size)–настаканные друг на друга эмбеддинги слов для представления текста (переменного размера)

```
model = Sequential()
model.add(Conv2D(32, kernel_size=(5, 5), strides=(1,
                 activation='relu',
                 input_shape=input_shape))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2,
model.add(Conv2D(64, (5, 5), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(1000, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))
```

Свёрточная нейронная сеть (Convolutional Neural Network)

```
model = Sequential()
model.add(Conv2D(32, kernel_si
                 activation='relu',
                 input_shape=input_shape))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2,
model.add(Conv2D(64, (5, 5), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(1000, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))
```

Определение типа конструктора модели

```
model = Sequential()
model.add(Conv2D(32, kernel_size=(5, 5), strides=(1,
                 activation='relu',
                 input_shape=input_shape))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2,
model.add(Conv2D(64, (5, 5), activation='relu'))
                                (pool_size=(2, 2)))
model.add(Dense(1000, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))
```

Добавление слоя свёртки

```
model = Sequential()
model.add(Conv2D(32, kernel_si
                 activation='relu',
                 input_shape=input_shape))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2,
model.add(Conv2D(64, (5, 5), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(1000, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))
```

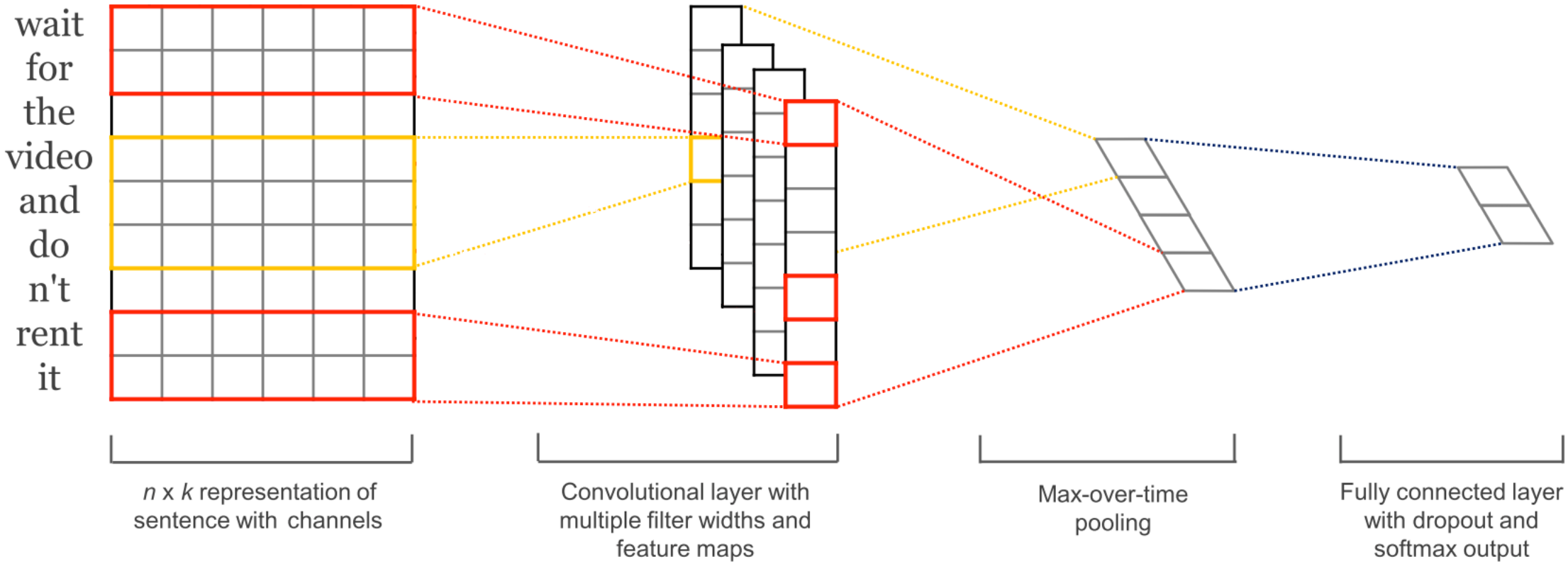
Добавление слоя подвыборки

```
model = Sequential()
model.add(Conv2D(32, kernel_size=(5, 5), strides=(1,
                 activation='relu',
                 input_shape=input_shape))
                 (pool_size=(2, 2), strides=(2,
model.add(Conv2D(64, (5, 5), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(1000, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))
```

Добавление обычного полносвязного слоя с 1000 нейронами

```
model = Sequential()
model.add(Conv2D(32, kernel_size=(5, 5), strides=(1,
                 activation='relu',
model.add                                    trides=(2,
model.add(Conv2D(64, (5, 5), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(1000, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))
```

https://adventuresinmachinelearning.com/keras-tutorial-cnn-11-lines/



https://www.aclweb.org/anthology/D14-1181

Практика

https://github.com/DinoTheDinosaur/russian_sentiment_edu/blob/master/notebooks/Logistic_Regression_BoW.ipynb