

Классификация текстов

+ gentle reminder on ML course

Задача классификации

- Есть заранее известные классы (2 - бинарная классификация, больше - мультиклассовая классификация)
- Задача - построить алгоритм, который относит объекты в свой класс

Примеры: анализ тональности, определение интента в диалоге, выявление токсичного поведения и много всего другого.

Как будем оценивать качество?

- Как и в любой ML задаче, у нас есть train и test множества: обучаемся на train, проверяем качество на test.
- Метрики качества для бинарной классификации:
 - Доля правильных ответов (accuracy)
 - Точность (precision)
 - Полнота (recall)
 - F-мера
 - AUC-ROC

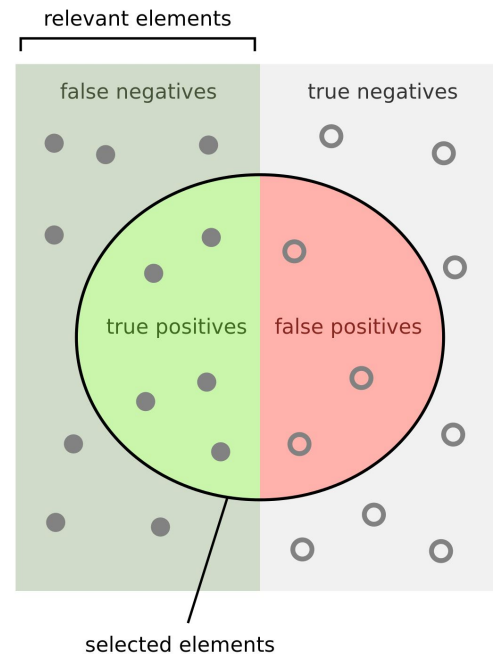
Accuracy, precision, recall

	Y = 0	Y = 1
f(x) = 0	True Negative	False Positive
f(x) = 1	False Negative	True Positive

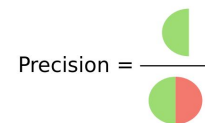
$$\text{Полнота (recall)} = \frac{T_p}{P}$$

$$\text{Точность (precision)} = \frac{T_p}{T_p + F_p}$$

$$\text{F-мера} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

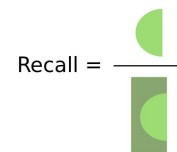


How many selected items are relevant?



$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

How many relevant items are selected?

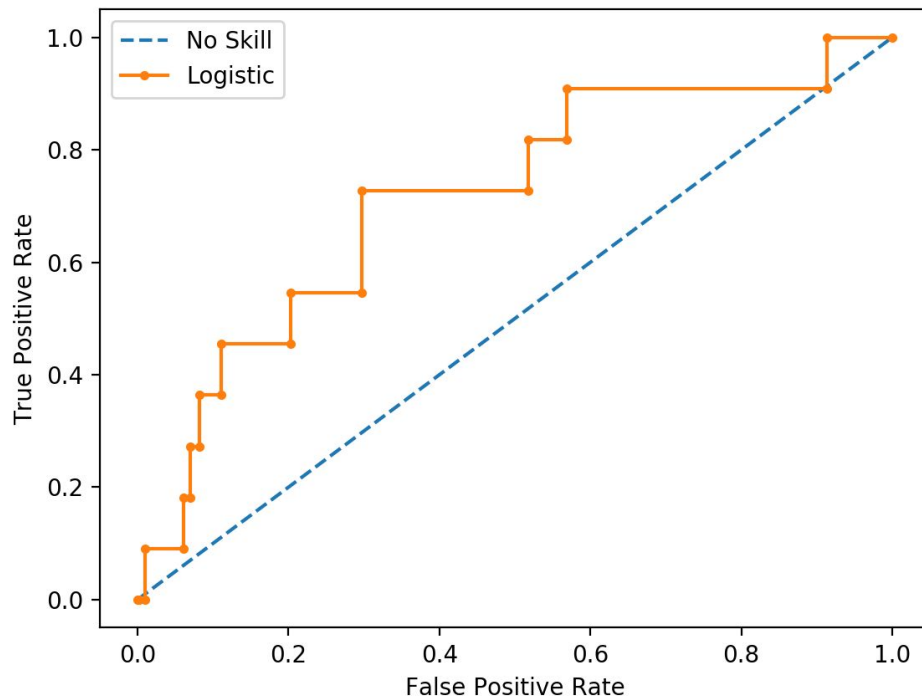


$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

AUC-ROC

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}};$$

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$



Многоклассовая классификация: One-VS-All

Обучим K (K = число классов) линейных классификаторов: $b_1(x), \dots, b_K(x)$

Каждый классификатор (бинарный!) будет отличать $b_k(x) = \langle w_k, x \rangle + w_{0k}$.
один класс от всех остальных.

Итоговый класс будем вычислять как наиболее вероятный,
исходя из прогнозов всех алгоритмов:

$$a(x) = \arg \max_{k \in \{1, \dots, K\}} b_k(x).$$

Многоклассовая классификация: All-VS-All

Обучим C_K^2 классификаторов (для всех возможных пар классов):

$$b_k(x) = \text{sign}(\langle w_k, x \rangle + w_{0k}).$$

Каждый классификатор (бинарный!) обучаем на подвыборке, содержащей только 2 класса.

Для классификации нового объекта подадим его на вход всем построенным классификаторам; в качестве ответа выберем наиболее “частый” среди ответов класс.

$$a(x) = \arg \max_{k \in \{1, \dots, K\}} \sum_{i=1}^K \sum_{j \neq i} [a_{ij}(x) = k]$$

Микро- и макро-усреднение

Рассмотрим K двухклассовых задач (one-VS-all),
для каждой вычислим матрицу ошибок:

$$TP_k, FP_k, FN_k, TN_k$$

Микро-усреднение:

$$\text{precision}(a, X) = \frac{\overline{TP}}{\overline{TP} + \overline{FP}}, \quad \overline{TP} = \frac{1}{K} \sum_{k=1}^K TP_k$$

Макро-
усреднение:

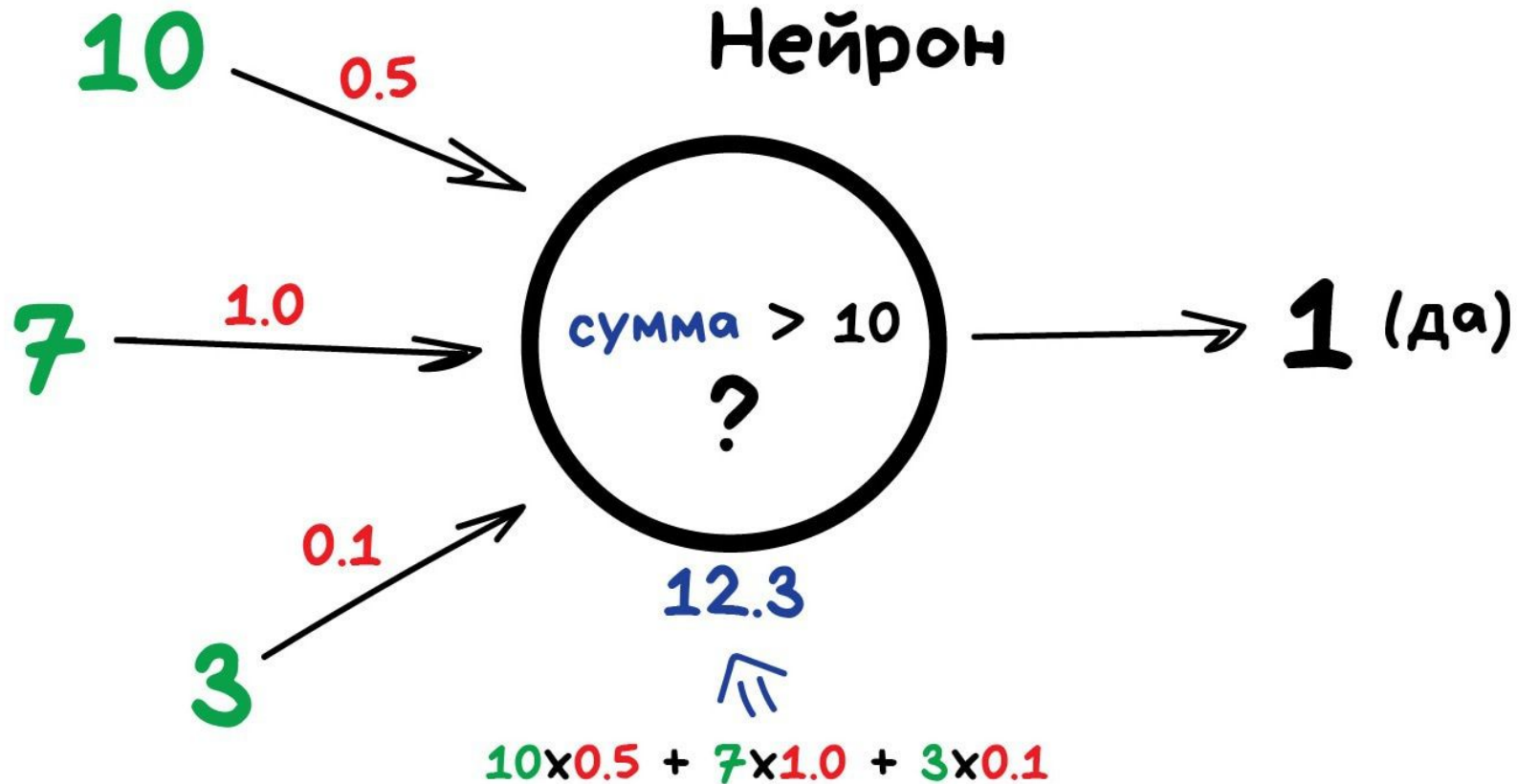
$$\text{precision}(a, X) = \frac{1}{K} \sum_{k=1}^K \text{precision}_k(a, X); \quad \text{precision}_k(a, X) = \frac{TP_k}{TP_k + FP_k}.$$

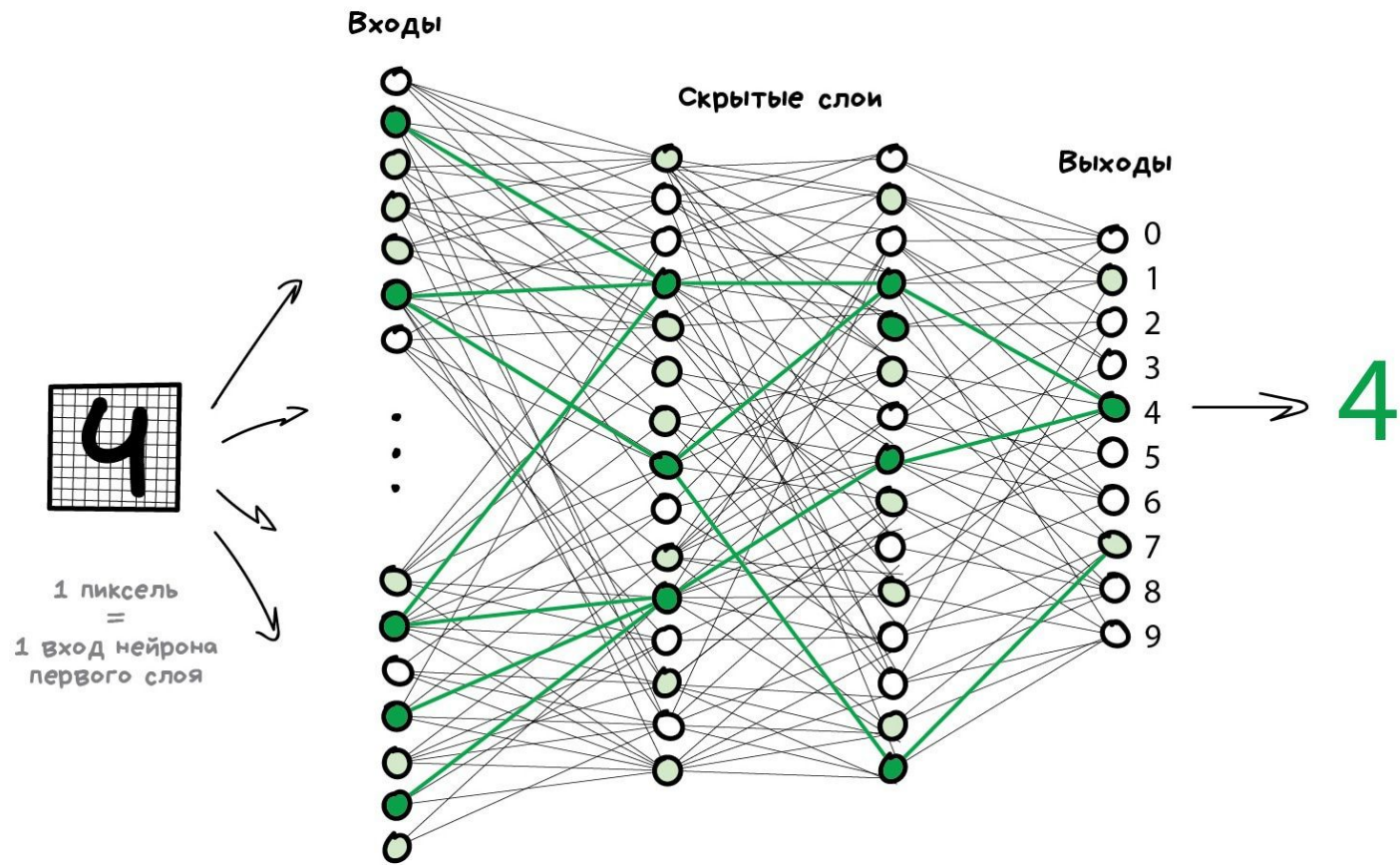
Классические алгоритмы классификации

- Метод опорных векторов (Support Vector Machine, SVM)
- Логистическая регрессия
- Метод К ближайших соседей (K Nearest Neighbors, KNN)
- Нейросетевые алгоритмы



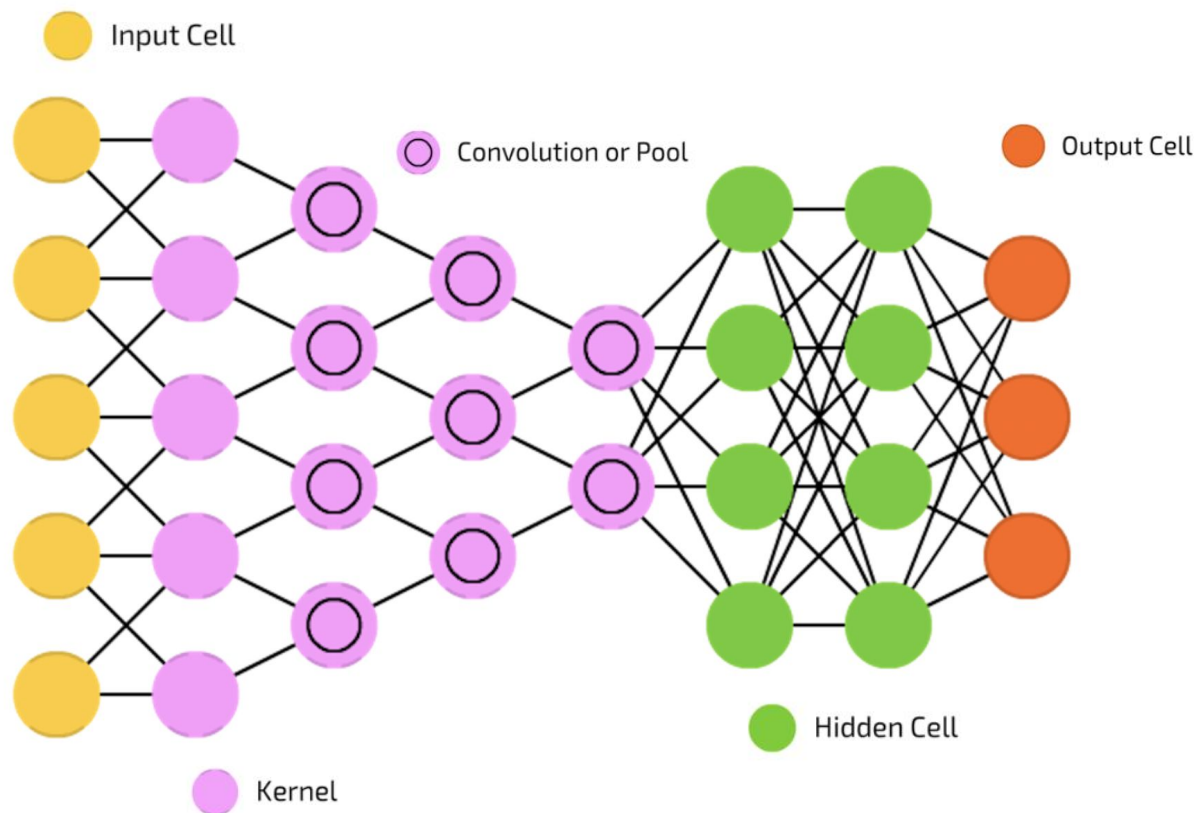
Нейрон





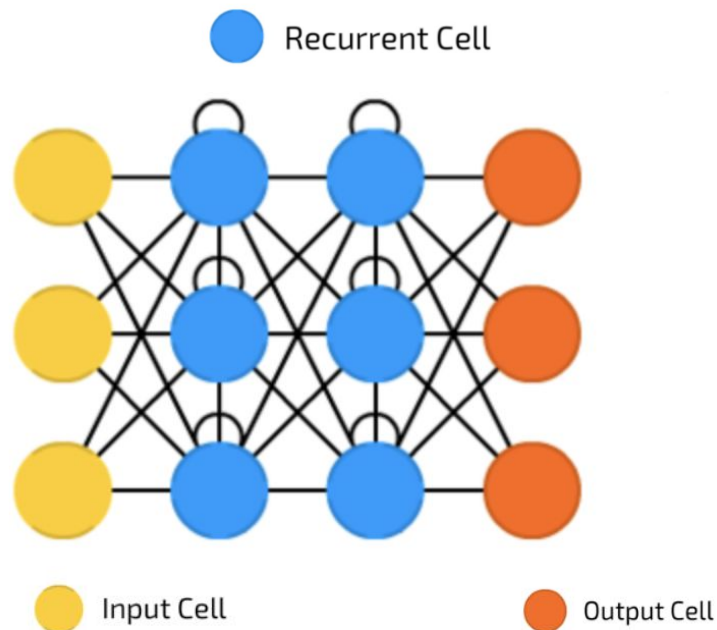
Многослойный Перцептрон (MLP)

Сверточные нейросети (convolutional neural networks, CNN)



Рекуррентные нейросети (recurrent neural network, RNN)

Все, что связано с последовательностями данных: тексты, речь, музыка, временные ряды и т.д.

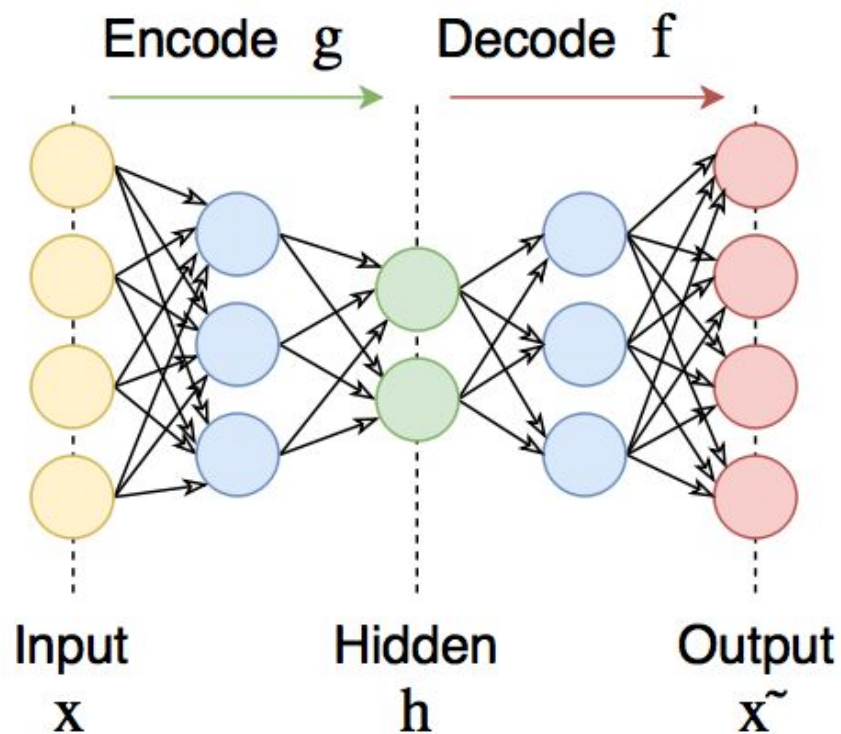


Генеративные состязательные нейросети (generative adversarial network, GAN)

Две нейросети: одна
учится генерировать
данные, вторая - отличать
настоящие от
сгенерированных.



Автоэнкодеры (Autoencoders)



Библиотеки для написания моделей

Criteria	⬆ Winner ⬆	⬆ Runner up ⬆
Community and Support	Tensorflow	Pytorch
Ease of Use	Pytorch	Tensorflow
Academia(Prototyping)	Pytorch	Torch/Caffe
Industry	Tensorflow	Caffe2
Embedded Computer vision	Caffe	Tensorflow

Пример нейросети на Keras

```
model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3),
                 activation='relu',
                 input_shape=input_shape))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))

model.compile(loss=keras.losses.categorical_crossentropy,
              optimizer=keras.optimizers.Adadelta(),
              metrics=['accuracy'])

model.fit(x_train, y_train,
          batch_size=batch_size,
          epochs=epochs,
          verbose=1,
          validation_data=(x_test, y_test))
```

Sum up

- Хороший бейзлайн для классификации - fasttext
- Если есть модель эмбедингов (или можно ее обучить) - модели классификации поверх эмбедингов
- Нейросетевые классификаторы (о них во второй лекции)