

Искусственный интеллект и машинное обучение

Руководитель группы разработки ООО «Квантрон групп»,
к.т.н., доцент кафедры ЭВМ РГРТУ, Тарасов А.С.



РЯЗАНСКИЙ
ГОСУДАРСТВЕННЫЙ
РАДИОТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ

kvantron

Искусственный интеллект

Artificial Intelligence

Искусственный интеллект (ИИ) — это область компьютерной науки, которая занимается созданием систем и алгоритмов, способных выполнять задачи, традиционно требующие человеческого интеллекта.

Несмотря на широкое применение данного термина, нет чёткой границы между тем, что можно отнести к ИИ, а что нет. В разные годы развития данной области в неё включались (а некоторые и исключались) такие задачи как: решение дифференциальных уравнений, шахматные задачи, генерация речи, нейронные сети, техническое зрение и многие другие.

Искусственный интеллект

Ключевые даты

1943 год – Уоррен Мак-Каллок и Уолтер Питтс создают первую математическую модель нейрона

1950 год – Алан Тьюринг пишет статью под названием «Может ли машина мыслить?»

1956 год – Дартмутская конференция: термин «искусственный интеллект» впервые официально введён (Джон Маккарти, Марвин Мински и др.)

1974 – 1980 Первая «Зима искусственного интеллекта»

1980 год – появление LISP-машин, рост заинтересованности в экспертных системах, развитие оптимизаторов

1987 год – достижение границы возможностей современных вычислителей

1987 – 1993 Вторая «Зима искусственного интеллекта»

1995 год – резкое развитие процессорной техники

2012 год – Появление первой глубокой сверточной сети AlexNet

2018 год – сети трансформеры, GPT 1.0

Настоящее время – взрывной рост развития генеративных сетей, работающих «на лету» с различными видами информации

Машинное обучение

Machine Learning

Машинное обучение является подразделом искусственного интеллекта.

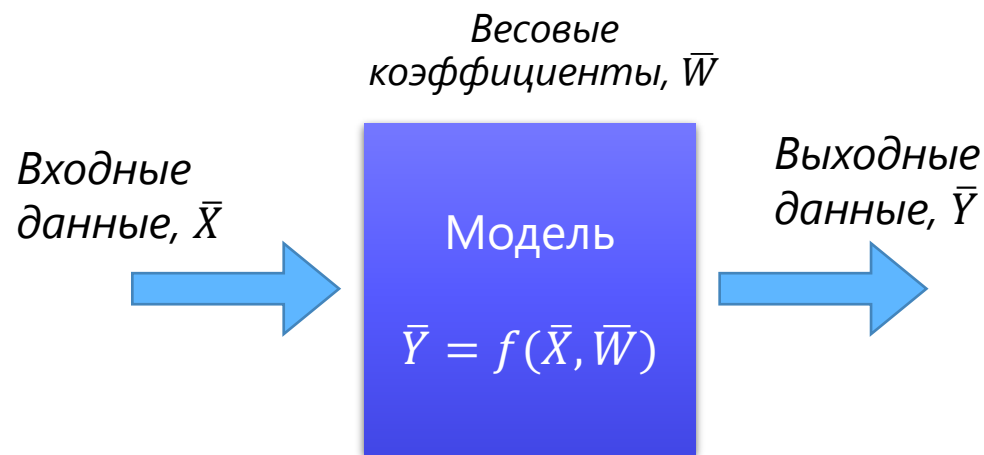
Оно предназначено для обучения компьютеров тому, как учиться на основе данных и совершенствоваться при помощи опыта, а не работать на основе явно запрограммированных алгоритмов. В процессе машинного обучения алгоритмы учатся поиску закономерностей и корреляций в больших наборах данных, а также принятию оптимальных решений и созданию прогнозов на основе этого анализа.

Приложения машинного обучения улучшаются по мере использования и становятся точнее по мере роста объема доступных данных.

Уровни взаимодействия с ML

1. Эксплуатация готовых решений (*ChatGPT, MidJourney, Kandinsky, Github Copilot*)
2. Разработка своих решений на основе обученных моделей (*DLIB, YOLO Pose Detector, Каскады Хаара*)
3. Обучение существующих моделей (*LR, VGG19, MobileNet, YOLO*)
4. Построение своих моделей на основе существующих
5. Построение своих фундаментальных архитектур

Принцип работы алгоритмов ML



```
model = Sequential()  
  
model.add(Dense(3))  
model.add(Activation('sigmoid'))  
  
model.add(Dense(2))  
model.add(Activation('sigmoid'))  
  
model.add(Dense(3))  
model.add(Activation('sigmoid'))  
  
model.add(Dense(1))  
model.add(Activation('softmax'))
```

1. Модель представляет собой некоторую неизменяемую математическую функцию, которая принимает значения входных данных фиксированной размерности и формирует выходной результат, также фиксированной размерности.
2. В модели содержатся также весовые коэффициенты – значения, которые будут подобраны в ходе обучения модели, и зафиксированы в процессе её эксплуатации.

Опыт. Набор данных

Dataset

```
data = np.loadtxt('data.csv', delimiter=';', dtype=np.float32)
```

```
X = data[:, [4, 5, 6]]
```

```
Y = data[:, [10]]
```

Датасет, опыт, выборка, набор данных — это структурированная коллекция данных, используемая для обучения, тестирования и оценки моделей машинного обучения.

Обучающая выборка (Training set) — используется для обучения модели. *Обычно 80-85% от всего набора*

Валидационная выборка (Validation set) — для оценки модели в процессе обучения. *Обычно 10-15% от всего набора*

Тестовая выборка (Test set) — для финальной оценки модели. *Обычно от 0 до 5% от всего набора*

X (input values)

Y (ou

| PassengerId | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|-------------|--------|--------------------------|--------|------|-------|-------|-----------|---------|-------|----------|
| 892 | 3 | Kelly; Mr, James | male | 34,5 | 0 | 0 | 330911 | 7,8292 | | Q |
| 893 | 3 | Wilkes; Mrs, James (Ell | female | 47 | 1 | 0 | 363272 | 7 | | S |
| 894 | 2 | Myles; Mr, Thomas Frai | male | 62 | 0 | 0 | 240276 | 9,6875 | | Q |
| 895 | 3 | Wirz; Mr, Albert | male | 27 | 0 | 0 | 315154 | 8,6625 | | S |
| 896 | 3 | Hirvonen; Mrs, Alexand | female | 22 | 1 | 1 | 3101298 | 12,2875 | | S |
| 897 | 3 | Svensson; Mr, Johan Ce | male | 14 | 0 | 0 | 7538 | 9,225 | | S |
| 898 | 3 | Connolly; Miss, Kate | female | 30 | 0 | 0 | 330972 | 7,6292 | | Q |
| 899 | 2 | Caldwell; Mr, Albert Fra | male | 26 | 1 | 1 | 248738 | 29 | | S |
| 900 | 3 | Abraham; Mrs, Joseph (| female | 18 | 0 | 0 | 2657 | 7,2292 | | C |
| 901 | 3 | Davies; Mr, John Samue | male | 21 | 2 | 0 | A/4 48871 | 24,15 | | S |
| 902 | 3 | Ilieff; Mr, Ylio | male | | 0 | 0 | 349220 | 7,8958 | | S |
| 903 | 1 | Jonas; Mr, Charles Cro | male | 46 | 0 | 0 | 694 | 26 | | S |

Train

Valid

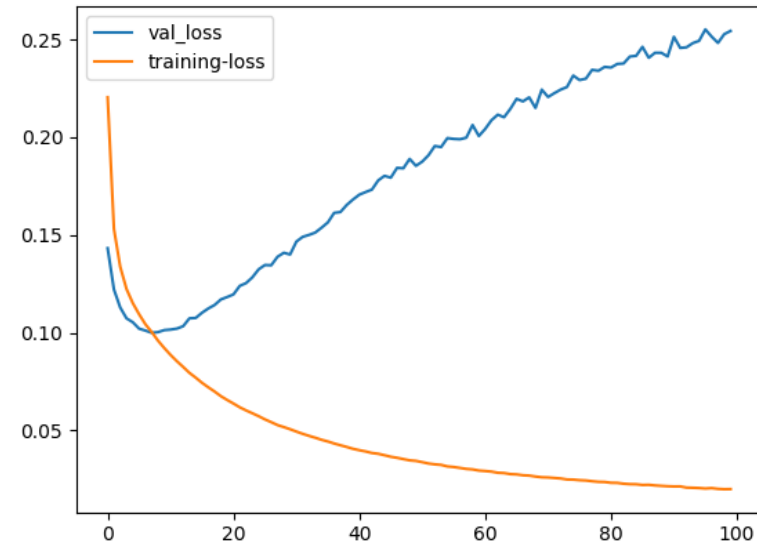
Обучение

Fitting (training) model

Обучение – процесс подстройки весов \bar{W} на основе данных **обучающей выборки**.

Алгоритм обучения принято называть **оптимизатором** (англ. *Optimizer*) (Adam, SGD, RMSProp). Задача оптимизатора – минимизировать ошибку между предсказанными данными и истинными. Определяет ошибку между данными **функция потерь** (англ. *Loss*) (MSE, MAE, BCE, CCE). **Эпоха** (англ. *Epoch*) – однократная подстройка весов модели на основе всех примеров обучающей выборки.

```
model.compile(optimizer='adam', loss='mse')  
model.fit(X, Y, epochs=50, batch_size=32, validation_split=0.2)
```



Применение

Inference model

В процессе эксплуатации модели её структура и веса не изменяются. Таким образом, при идентичных входных данных, такие модели будут всегда давать одинаковый результат. Исключение – сети с ячейками памяти и трансформеры.

```
test = np.asarray([[0.17, 1.42, 0.39]])  
model.predict(test)
```

Что нужно, чтобы начать?

Основной язык разработки ML – Python. На нём разрабатывается большинство решений

Pandas, Numpy, OpenCV, Matplotlib – полезные инструменты для загрузки и обработки данных

TensorFlow, PyTorch, Onnx – популярные платформы (Backend) для обучения нейронных сетей

Ultralytics, Keras, Dlib, Sklearn – популярные надстройки для запуска и обучения своих моделей

```
import numpy as np
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout

data = np.loadtxt('data.csv', delimiter=';',
                  dtype=np.float32)

X = data[:, [0, 1, 2]]
Y = data[:, [10]]

model = Sequential()

model.add(Dense(3))
model.add(Activation('sigmoid'))

model.add(Dense(2))
model.add(Activation('sigmoid'))

model.add(Dense(3))
model.add(Activation('sigmoid'))

model.add(Dense(1))
model.add(Activation('softmax'))

model.compile(optimizer='adam', loss='MSE')
model.fit(X, Y, epochs=50, batch_size=32)

test = np.asarray([[0.17, 1.42, 0.39]])
model.predict(test)
```

Спасибо за внимание!

Переходи по ссылке, чтобы получить доступ к материалам:

<https://github.com/vb2005/keras-quickstart>

Tarasov.a.s@kvantron.com



РЯЗАНСКИЙ
ГОСУДАРСТВЕННЫЙ
РАДИОТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ

kvantron¹¹