

# МОДЕЛИРОВАНИЕ НЕЙРОННОЙ СЕТИ НА ПРИМЕРЕ ПЕРСЕПТРОНА В ИССЛЕДОВАНИИ КЛАССИФИКАЦИИ ОБЪЕКТОВ

## NEURAL NETWORK MODELING ON THE EXAMPLE OF THE PERCEPTRON IN THE STUDY OF THE CLASSIFICATION OF OBJECTS

*A. Vasiliev*

*Summary.* The article is devoted to the neural network modeling using the perceptron as an example in the study of object classification. The objective of the study is to classify images of certain colors by distinguishing parameters and adjust the algorithm in such a way as to minimize erroneous classification. Like the human brain, neural networks are made up of a large number of connected elements that mimic neurons. Deep neural networks are based on such algorithms, thanks to which computers learn from their own experience, forming in the learning process multi-level, hierarchical representations of the world. The perceptron is the most primitive neural network that has two input and one output cell. Perceptrons make it possible to create a set of “associations” between input stimuli and the desired response at the output. In a biological context, this is the conversion, for example, of visual communication into physiological actions from motor neurons. Perceptrons are grouped as artificial neural networks: with one hidden layer; with threshold transfer function; with direct signal propagation. Perceptrons can be used in solving problems of binary classification, when the sample must be identified as belonging to one of the predefined two classes. In this research paper, the perceptron operation algorithm, the learning algorithm, and the algorithm correction were written.

*Keywords:* neural network, neurons, perceptrons, algorithm, function, model, weights.

*Васильев Александр Владимирович*  
Независимый исследователь  
astrgan@gmail.com

*Аннотация.* Статья посвящена вопросу моделирования нейронной сети на примере перцептрона в исследовании классификации объектов. Задача исследования — классифицировать изображения определенных цветов по отличительным параметрам и скорректировать алгоритм таким образом, чтобы свести ошибочную классификацию к минимуму. Как и человеческий мозг, нейронные сети состоят из большого количества связанных элементов, которые имитируют нейроны. Глубокие нейронные сети основаны на таких алгоритмах, благодаря которым компьютеры учатся на собственном опыте, формируя в процессе обучения многоуровневые, иерархические представления о мире. Перцептрон является самой примитивной нейронной сетью, имеющая две входных и одну выходную клетки. Перцептроны дают возможность создавать набор «ассоциаций» между входными стимулами и нужной реакцией на выходе. В биологическом контексте это преобразование, к примеру, зрительного сообщения в физиологические действия от двигательных нейронов. Перцептроны группированы как искусственные нейронные сети: с одним скрытым слоем; с пороговой передаточной функцией; с прямым распространением сигнала. Перцептроны могут быть применены при решении задач бинарной классификации, когда выборка должна быть идентифицирована как принадлежащая к одному из predefined двух классов. В данной исследовательской работе был охарактеризован алгоритм работы перцептрона, алгоритм обучения, написана коррекция алгоритма.

*Ключевые слова:* нейросеть, нейроны, перцептроны, алгоритм, функция, модель, веса.

**Н**ейрон или нервная клетка — это электрически возбудимая клетка, которая взаимодействует с другими клетками через специализированные соединения, называемые синапсами. Нейроны обычно разделяют на три типа в зависимости от их функции. Сенсорные нейроны реагируют на такие раздражители, как прикосновение, звук или свет, которые воздействуют на клетки органов чувств, и посылают сигналы в спинной или головной мозг. Моторные нейроны получают сигналы от головного и спинного мозга, чтобы контролировать все — от мышечных сокращений до выхода желез. Интернейроны соединяют нейроны с другими нейронами в той же области головного или спинного мозга. Группу связанных между собой нейронов называют нейронной цепью [7].

Нейрон состоит из тела — сомы, аксонов и дендрита. Сигнал одного нейрона передаётся через синапс к другому нейрону, эти цепочки нейронов собираются в сети. С точки зрения биологии нейронная сеть — это совокупность нейронов, соединенных между собой синапсами для выполнения определенной функции при активации. Нейронные цепи соединяются друг с другом, образуя крупномасштабные мозговые сети. Биологические нейронные сети вдохновили разработку искусственных нейронных сетей, но искусственные нейронные сети обычно не являются строгими копиями своих биологических аналогов.

Связи между нейронами в головном мозге гораздо сложнее, чем у искусственных нейронов, используемых

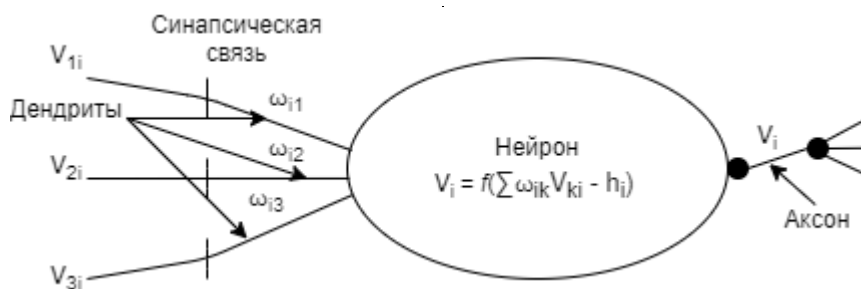


Рис. 1. Модель нейрона

в нейросетевых вычислительных моделях искусственных нейронных сетей. Основными видами связей между нейронами являются синапсы: как химические, так и электрические синапсы [5]. Создание синапсов позволяет соединять нейроны в миллионы перекрывающихся и взаимосвязанных нейронных цепей. Модель нейрона, взаимодействующего с другими нейронами в нейросети, показана на рисунке 1.

Здесь  $V_{ki}$  — энергетические доли импульсов  $V_i$ , которые генерируются другими нейронами и принимаются дендритами нейрона  $i$ .  $\omega_{ik}$  — вес дендрита;  $h_i$  — порог. Генерируемый импульс  $V_i$  в свою очередь, распространяется вдоль дендритов нейронов, с которыми нейрон связан через разветвленный аксон. Согласно закону распределения энергии, значение  $V_i$  делится пропорционально весу дендритов «принимающих» нейронов. Такое распределение энергии обычно не реализуется в практических моделях.

Каждый нейрон контролируется либо внешне, либо по принципу самоуправления с использованием обратных связей. Вес синаптической связи

$\{\omega_j\}$  и пороговые значения  $h_i$  можно регулировать. Такое регулирование, реализованное в разных моделях, определяет возможность обучения и самообучения в сети. Он устанавливает пути прохождения возбуждений через сеть, самым простым способом формируя отношения «предпосылка — эффект».

Функции  $f$  бывают разными, но простыми по объему вычислений. В самом простом случае  $f$  совпадает с линейной формой — заданным аргументом, то есть для всех дендритов с учетом их веса проводится суммирование и сравнение с пороговым значением.

Величина превышения порога — это величина возбуждения нейрона, иначе говоря определение значения величины возбуждения [1]. Например, в некоторых моделях величина возбуждения всегда равна единице, а отсутствие возбуждения соответствует нулю. В других

моделях также допускается отрицательное значение возбуждения.

В машинном обучении перцептрон — это алгоритм контролируемого обучения бинарных классификаторов. Двоичный классификатор — это функция, которая может решить, принадлежит ли вход, представленный вектором чисел, к какому-то определенному классу или нет. Это тип линейного классификатора, то есть алгоритм классификации, который делает свои предсказания на основе линейной функции предиктора, объединяющей набор весов с вектором признаков.

Перцептрон — это упрощенная модель биологического нейрона. Хотя сложность биологических моделей нейронов часто требуется для полного понимания нейронного поведения, исследования показывают, что подобная перцептронной линейной модели может вызвать некоторое поведение, наблюдаемое в реальных нейронах.

Идея перцептрона аналогична принципу действия основной обрабатывающей единицы мозга-нейрона. Нейрон состоит из множества входных сигналов, переносимых дендритами, телом клетки и одного выходного сигнала, переносимого по аксону. Нейрон запускает сигнал действия, когда клетка достигает определенного порога. Это действие либо происходит, либо нет; нет такого понятия, как «частичное» срабатывание нейрона [7].

Аналогично, перцептрон имеет множество входов (часто называемых признаками), которые подаются в линейную единицу, которая производит один двоичный выход. Таким образом, перцептроны могут быть применены при решении задач бинарной классификации, когда выборка должна быть идентифицирована как принадлежащая к одному из predeterminedных двух классов. Алгоритм работы перцептрона представлен на рисунке 2 [6].

Поскольку перцептроны являются бинарными классификаторами (0/1), мы можем определить их вычисление по линейной функции:

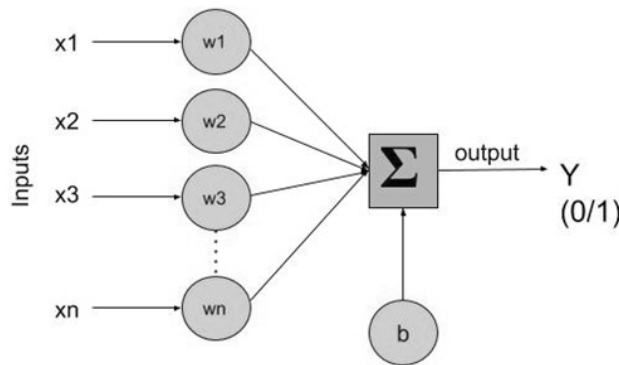


Рис. 2. Схема перцептрона

$$f_{\Phi}(\mathbf{x}) = \begin{cases} 1 & \text{if } b + \mathbf{w} \cdot \mathbf{x} > 0 \\ 0 & \text{otherwise} \end{cases}$$

Следует напомнить, что точечное (скалярное) произведение двух векторов длины  $n$  ( $1 \leq i \leq n$ ) равно  $w \cdot x = \sum_i w_i \cdot x_i$ . Функция  $f(x) = b + w \cdot x$  представляет собой линейную комбинацию векторов веса и признаков. Таким образом, перцептрон является линейным классификатором — алгоритмом, который предсказывает с помощью линейной функции предиктора [4].

Веса показывают эффективность каждой функции  $x_i$  в  $x$  в поведении модели. Чем выше вес  $w_i$  элемента  $x_i$ , тем выше его влияние на результат. С другой стороны, смещение «b» похоже на пересечение в линейном уравнении. Это константа, которая помогает модели адаптироваться таким образом, чтобы наилучшим образом соответствовать данным. Член смещения предполагает мнимый коэффициент входного признака  $x_0 = 1$ .

Для моделирования нейронной сети на примере перцептрона в исследовании классификации объектов классифицируем изображения определенных цветов по отличительным параметрам, а затем скорректируем алгоритм таким образом, чтобы свести ошибочность к минимуму.

### Алгоритм обучения на основе кода на языке Python

Модель может быть обучена по следующему алгоритму:

1. set  $b = w = 0$
2. for N iterations, or until weights do not change

- (a) for each training example  $x^k$  with label  $y^k$ 
  - I. if  $y^k - f(x^k) = 0$ , continue
  - II. else, update  $w_i, \Delta w_i = (y^k - f(x^k)) x_i$

Набор данных, который мы рассматриваем для реализации перцептрона, — это набор данных «цветок ири-са». Этот набор данных содержит 4 объекта, которые описывают цветок и классифицируют их как принадлежащие к одному из 3 классов. Мы удаляем последние 50 строк набора данных, который принадлежит классу «Iris-virginica», и используем только 2 класса «Iris-setosa» и «Iris-versicolor» потому, что эти классы линейно разделимы, и алгоритм сходится к локальному минимуму, в конечном итоге находя оптимальные веса:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
def load_data():
    URL_='https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'
    data = pd.read_csv(URL_, header = None)
    print(data)
    # сделаем набор данных линейно разделяемым
    data = data[:100]
    data[4] = np.where(data.iloc[:, -1] == 'Iris-setosa', 0, 1)
    data = np.asmatrix(data, dtype = 'float64')
    return data
data = load_data()
```

### Визуализация набора данных

Визуализируя набор данных с двумя объектами, мы видим, что этот набор данных может быть четко отделен путем рисования прямой линии между ними.

```
plt.scatter(np.array(data[:50,0]), np.array(data[:50,2]),
            marker='o', label='сетоза')
plt.scatter(np.array(data[50:,0]), np.array(data[50:,2]),
            marker='x', label='виргиника')
plt.xlabel('длина лепестка')
plt.ylabel('длина чашелистиков')
plt.legend()
plt.show()
```

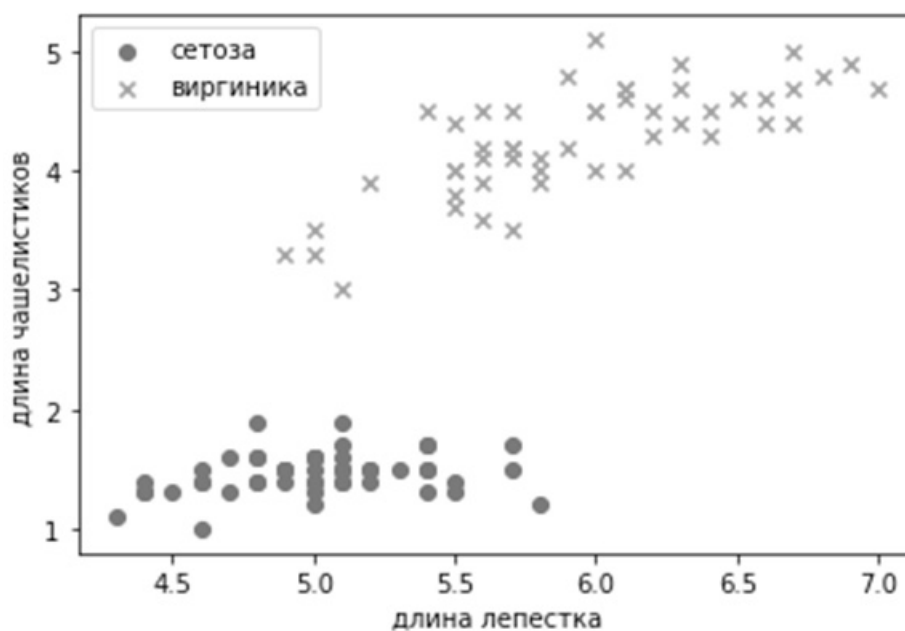


Рис. 3. Визуализация набора данных

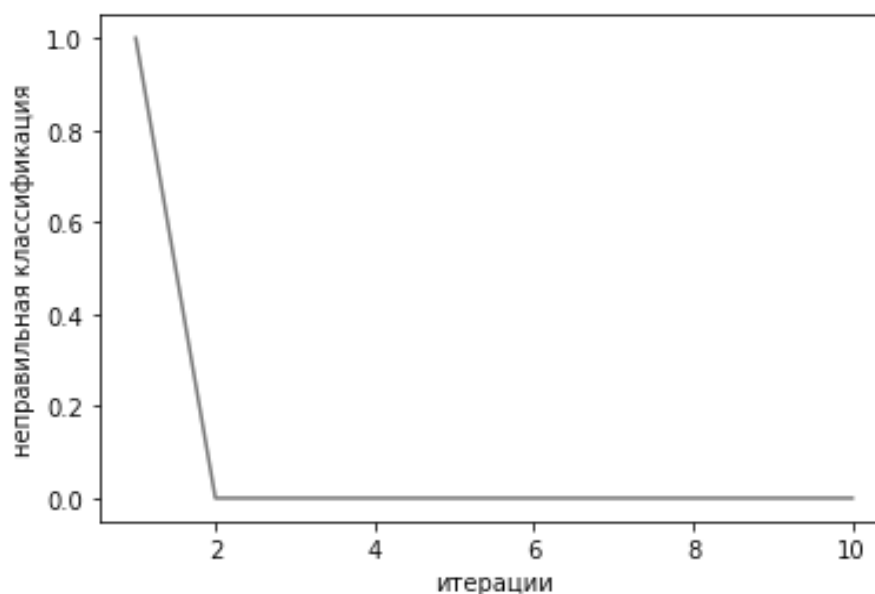


Рис. 4. Визуализация ошибочных классификаций

Наша цель — написать алгоритм, который находит эту строку и правильно классифицирует все эти точки данных.

Теперь реализуем упомянутый выше алгоритм как есть и посмотрим, как он работает. Есть 4 функции и, следовательно, 4 веса, связанные с каждой функцией. Не-

обходимо помнить, что был определён вес смещения  $w_0$ , который предполагает, что  $x_1 = 1$ , что в сумме составляет 5 весов.

Было определено число итераций равное 10. Это один из гиперпараметров, в отличие от системных па-

раметров, таких как  $w$ , которые изучаются алгоритмом. На каждой итерации алгоритм вычисляет класс (0 или 1) для всех точек данных и обновляет веса с каждой ошибочной классификацией.

### Коррекция алгоритма

Если выборка неправильно классифицирована, то веса обновляются дельтой, которая смещается в противоположном направлении. Таким образом, если образец будет классифицирован снова, результат будет «менее неправильным». Мы классифицируем любой ярлык  $\leq 0$  как «0» (*Iris-setosa*), а все остальное — как «1» (*Iris-versicolor*).

```
def perceptron(data, num_iter):
    features = data[:, :-1]
    labels = data[:, -1]
    # установим веса на ноль
    w = np.zeros(shape=(1, features.shape[1]+1))
    misclassified_ = []
    for epoch in range(num_iter):
        misclassified = 0
        for x, label in zip(features, labels):
            x = np.insert(x, 0, 1)
            y = np.dot(w, x.transpose())
            target = 1.0 if (y > 0) else 0.0
            delta = (label.item(0,0) — target)
```

```
if(delta): # неправильная классификация
    misclassified += 1
    w += (delta * x)
misclassified_.append(misclassified)
return (w, misclassified_)
num_iter = 10
w, misclassified_ = perceptron(data, num_iter)
```

Теперь можно посмотреть количество ошибочно классифицированных выборок в каждой итерации.

```
epochs = np.arange(1, num_iter+1)
plt.plot(epochs, misclassified_)
plt.xlabel('iterations')
plt.ylabel('misclassified')
plt.show()
```

Видно, что алгоритм сходится на 2-й итерации, то есть все образцы правильно классифицируются на 2-м проходе по данным.

Таким образом, перцептрон является упрощенной моделью биологического нейрона. Хотя сложность биологических моделей нейронов часто требуется для полного понимания поведения нейронов, исследования показывают, что перцептроноподобная линейная модель может вызывать поведение, наблюдаемое в реальных нейронах.

### ЛИТЕРАТУРА

1. Васильев А. Н. Принципы и техника нейросетевого моделирования / А. Н. Васильев, Д. А. Тархов. — М.: Высшая школа, 2014. — 218 с.
2. Головки В. А. От многослойных перцептронов к нейронным сетям глубокого доверия: парадигмы обучения и применение. В сб.: Нейроинформатика-2015. XVII Всероссийская научно-техническая конференция с международным участием. Лекции по нейроинформатике, НИЯУ МИФИ, 2015. С. 47–84
3. Дли М. И. Алгоритмы формирования изображений состояний объектов для их анализа глубокими нейронными сетями / М. И. Дли. — М.: Синергия, 2019. — 13 с.
4. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская. — М.: Горячая линия, 2013. — 384 с.
5. Тархов Д. А. Нейросетевые модели и алгоритмы. Справочник / Д. А. Тархов. — М.: Радиотехника, 2014. — 643 с.
6. Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных. — М.: ДМК Пресс, 2015. — 400 с.
7. Хайкин С. Нейронные сети: полный курс. — М.: Вильямс, 2006. — 1104 с.
8. Tarkhov D.A., Vasilyev A. N. Semi-empirical Neural Network Modeling and Digital Twins Development // Elsevier, Academic Press, 2019, 288 p.

© Васильев Александр Владимирович ( astrgan@gmail.com ).

Журнал «Современная наука: актуальные проблемы теории и практики»