



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

ВВЕДЕНИЕ В НЕЙРОННЫЕ СЕТИ

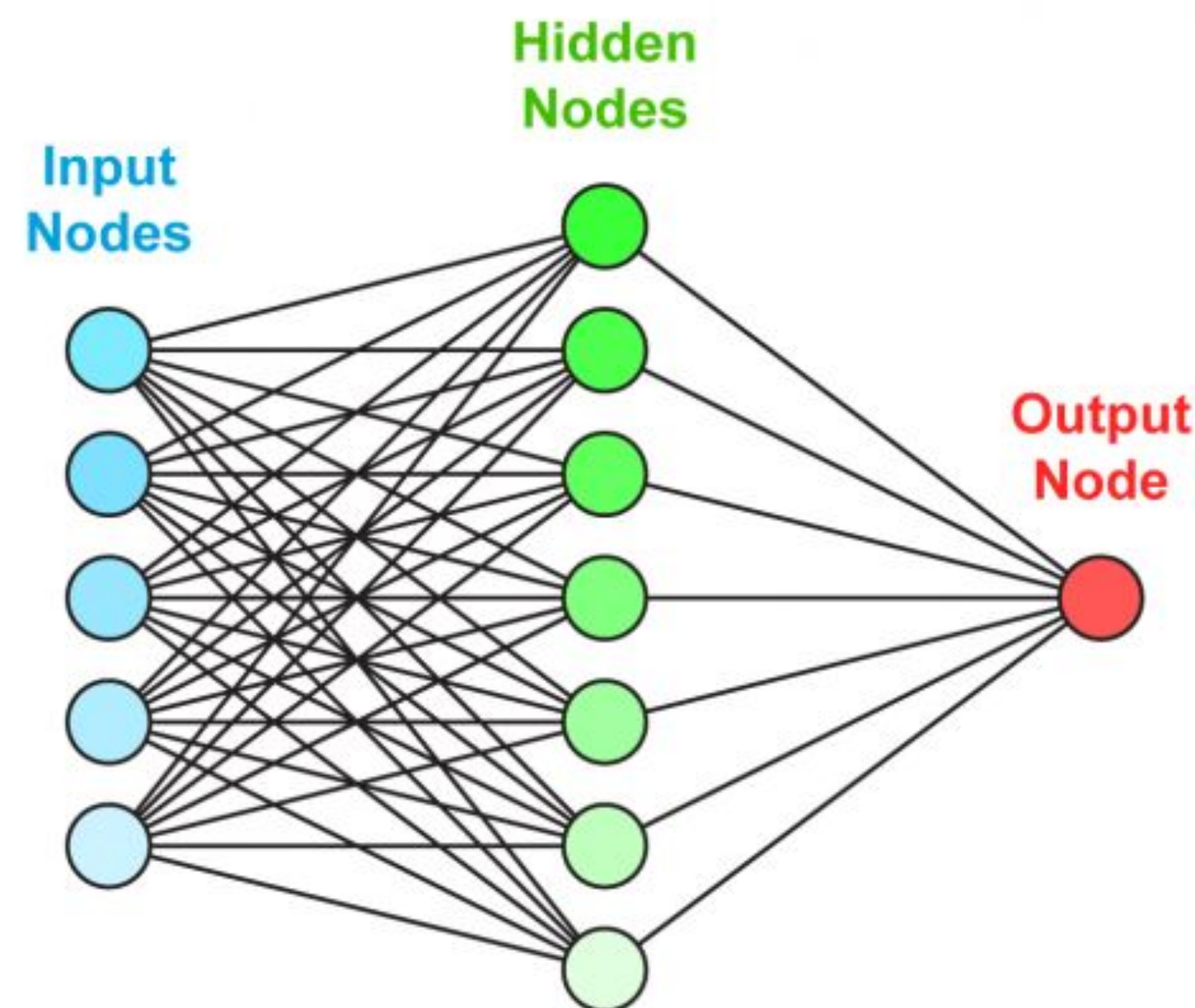
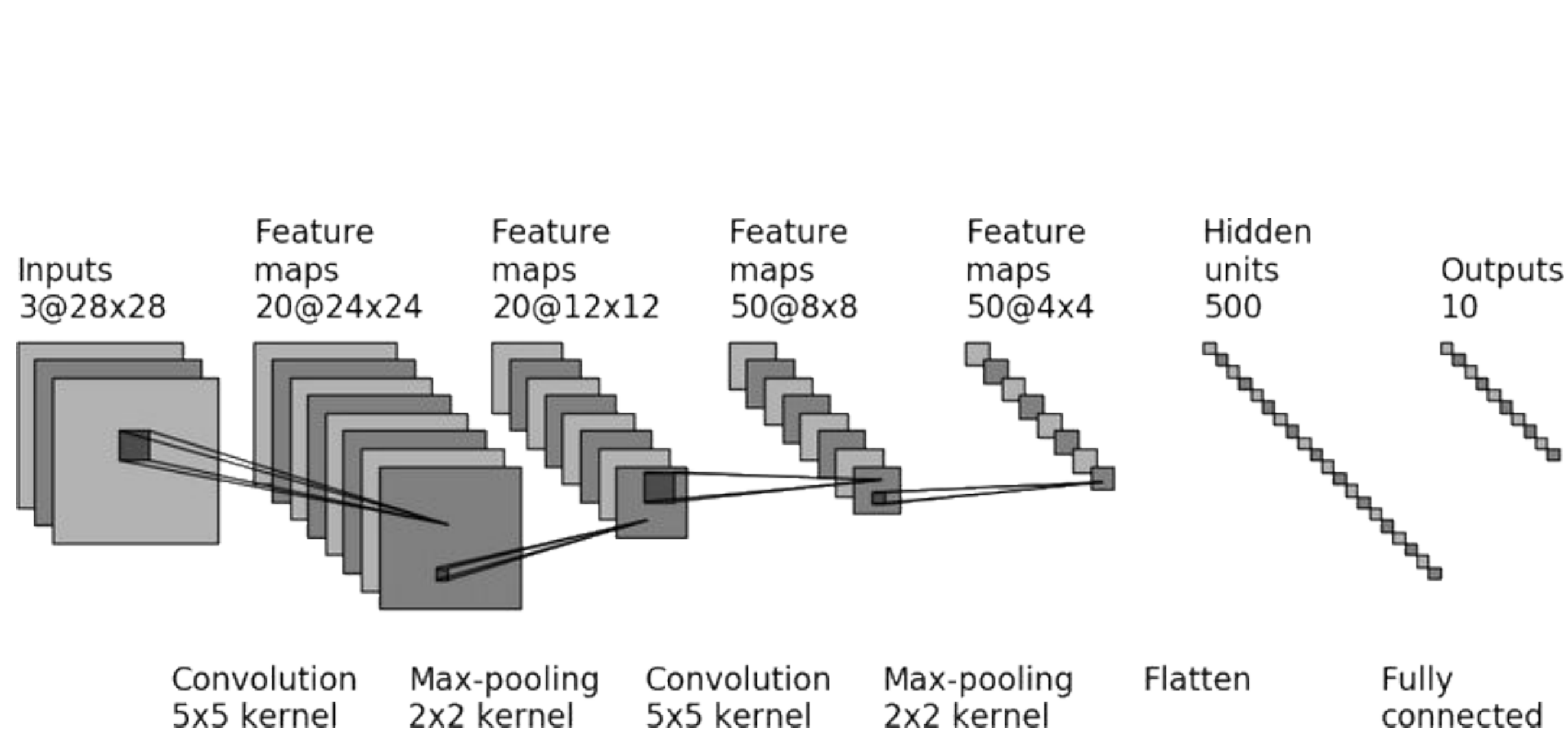
Файзулин Максим

Москва, 2023



ЧТО ТАКОЕ НЕЙРОННЫЕ СЕТИ

Нейронная сеть — это метод в искусственном интеллекте, где обучается алгоритм с целью обрабатывать данные таким же способом, как и человеческий мозг. Это тип процесса обучения, называемый глубоким обучением, который использует взаимосвязанные узлы или нейроны в слоистой структуре, напоминающей человеческий мозг.





ЧТО ТАКОЕ НЕЙРОННЫЕ СЕТИ

С точки зрения машинного обучения, нейронная сеть представляет собой частный случай методов распознавания образов;

С точки зрения математики, обучение нейронных сетей — это многопараметрическая задача нелинейной оптимизации;

С точки зрения кибернетики, нейронная сеть используется в задачах адаптивного управления и как алгоритмы для робототехники;

С точки зрения искусственного интеллекта, ИНС является основой философского течения коннекционизма и основным направлением в структурном подходе по изучению возможности построения (моделирования) естественного интеллекта с помощью компьютерных алгоритмов.



ЗАЧЕМ НУЖНЫ НЕЙРОННЫЕ СЕТИ

Диагностировать заболевания сельскохозяйственных культур и других растений по фото с точностью до 99,35% (пример с комбайном). [Источник](#)

Прогнозировать погоду. Этим занимается, например, «Яндекс.Погода»

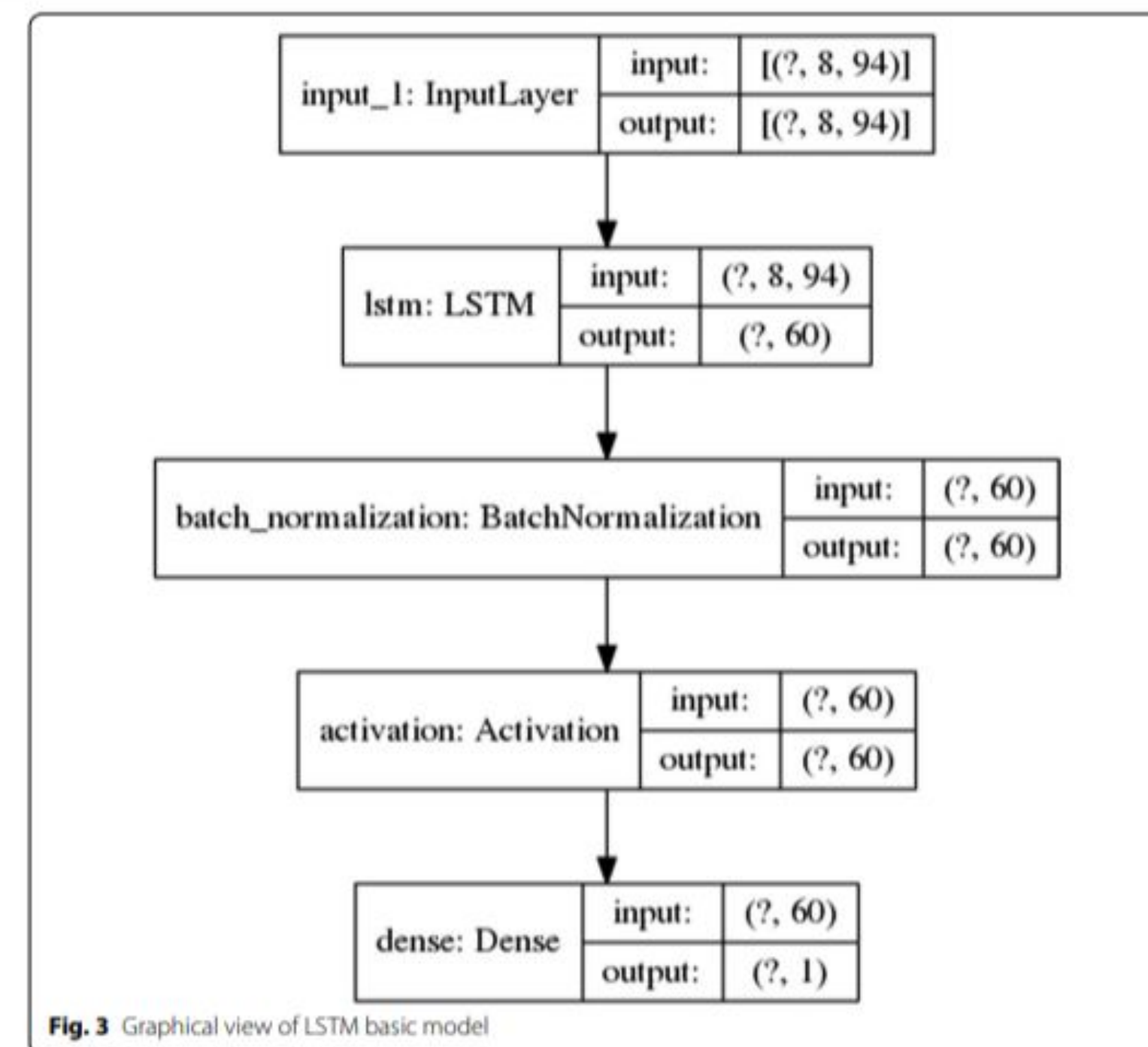
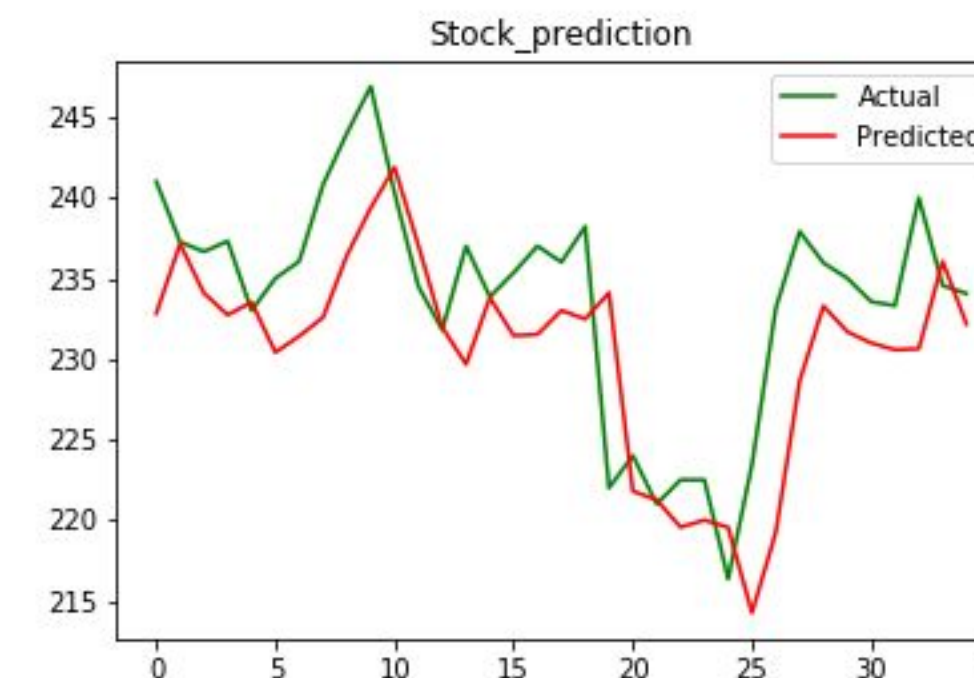
Перевод изображение в другие цветовые палитры (из ч/б в цветную), улучшение качества изображений

Использование классификаторов для нахождения древесных пород на основе снимков ГИС

Проведение sentiment-анализа: классификация тональности текстовых корпусов

Предсказание будущего движения цен финансовых активов, их ликвидности, бид-аск спрэд

И куча всего еще другого...

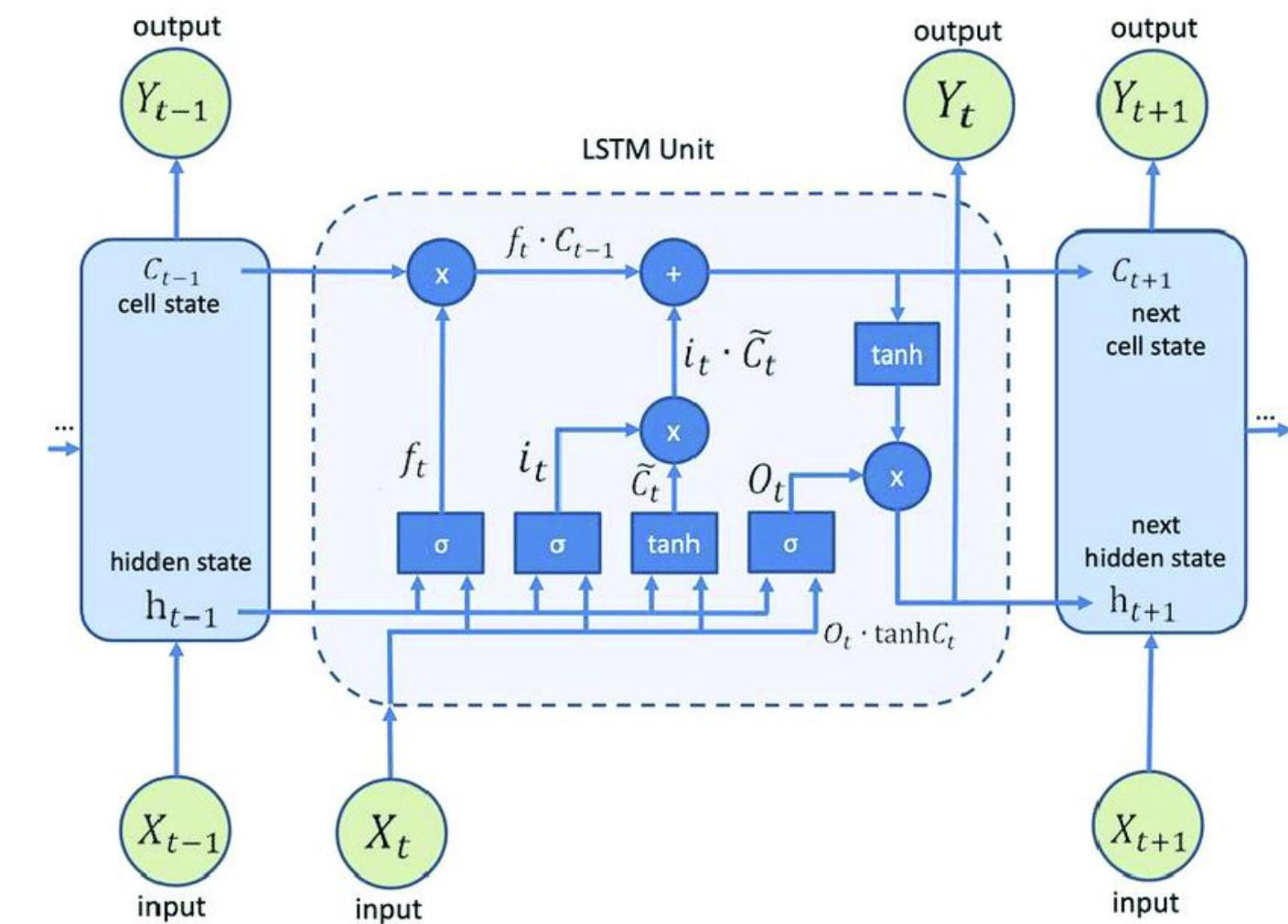
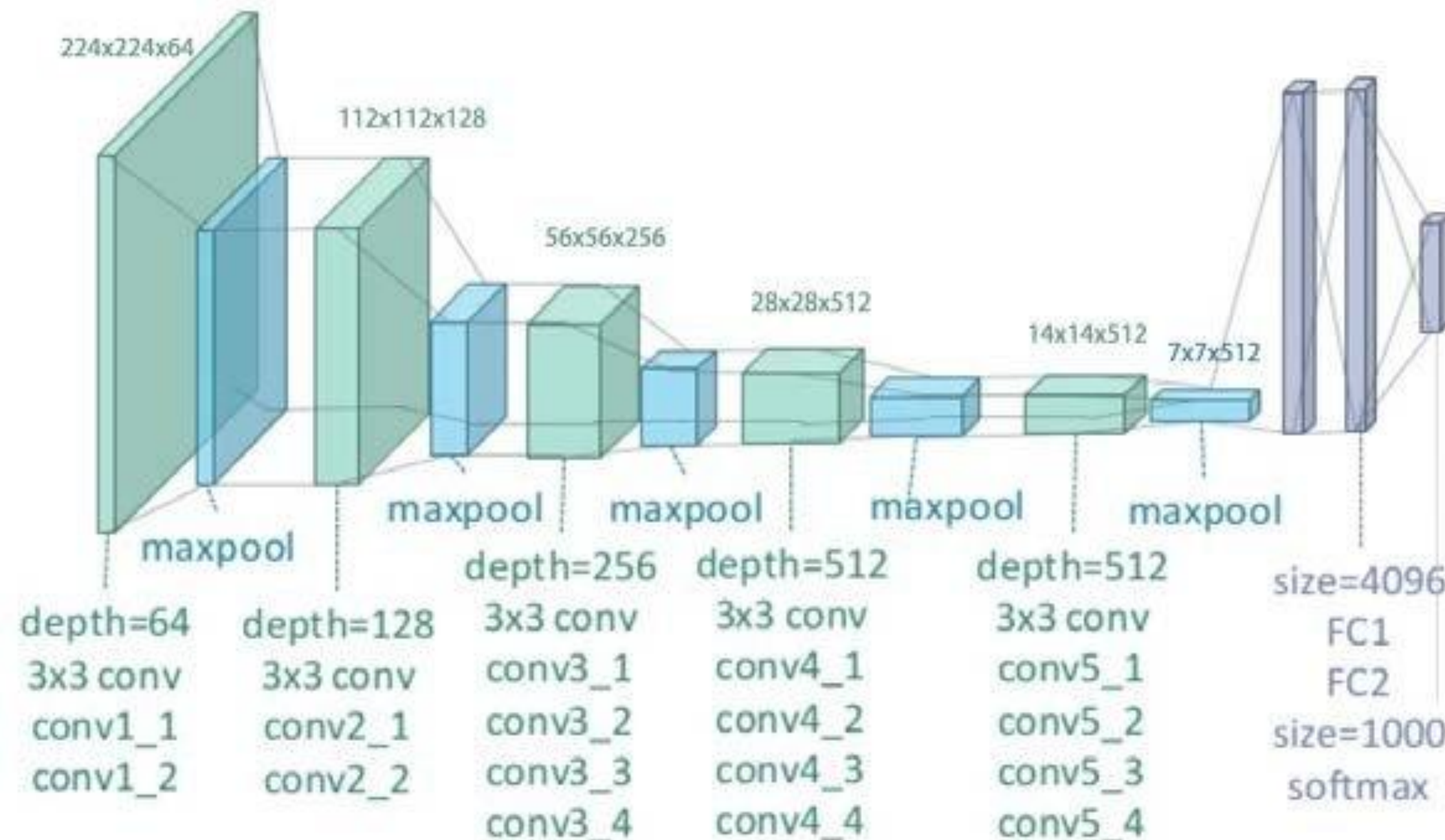




ЧТО ИСПОЛЬЗУЕМ МЫ?

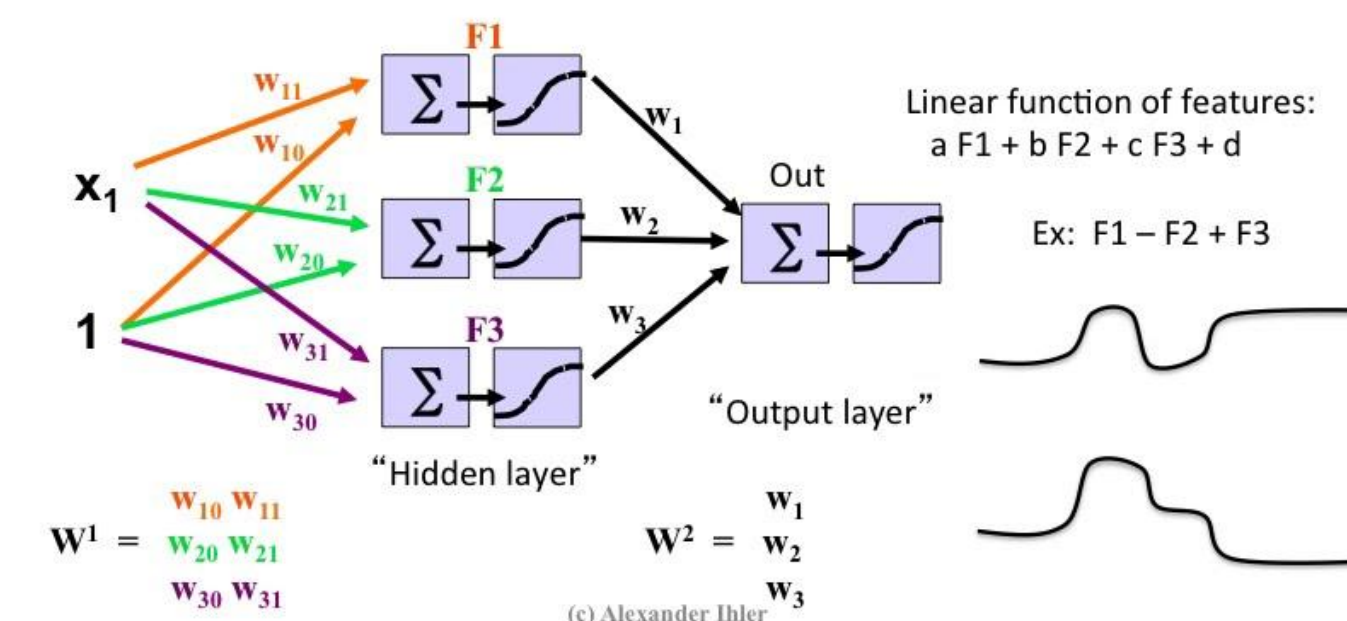
В рамках курса мы будем использовать всего несколько основных нейросетей:

- 1) Многослойный перцептрон (Multilayered perceptron)
- 2) Сеть долгой краткосрочной памяти (Long short-term memory; LSTM)
- 3) Сверточная нейронная сеть (Convolutional neural network, CNN)
- 4) Гибридная нейронная сеть (LSTM-CNN)



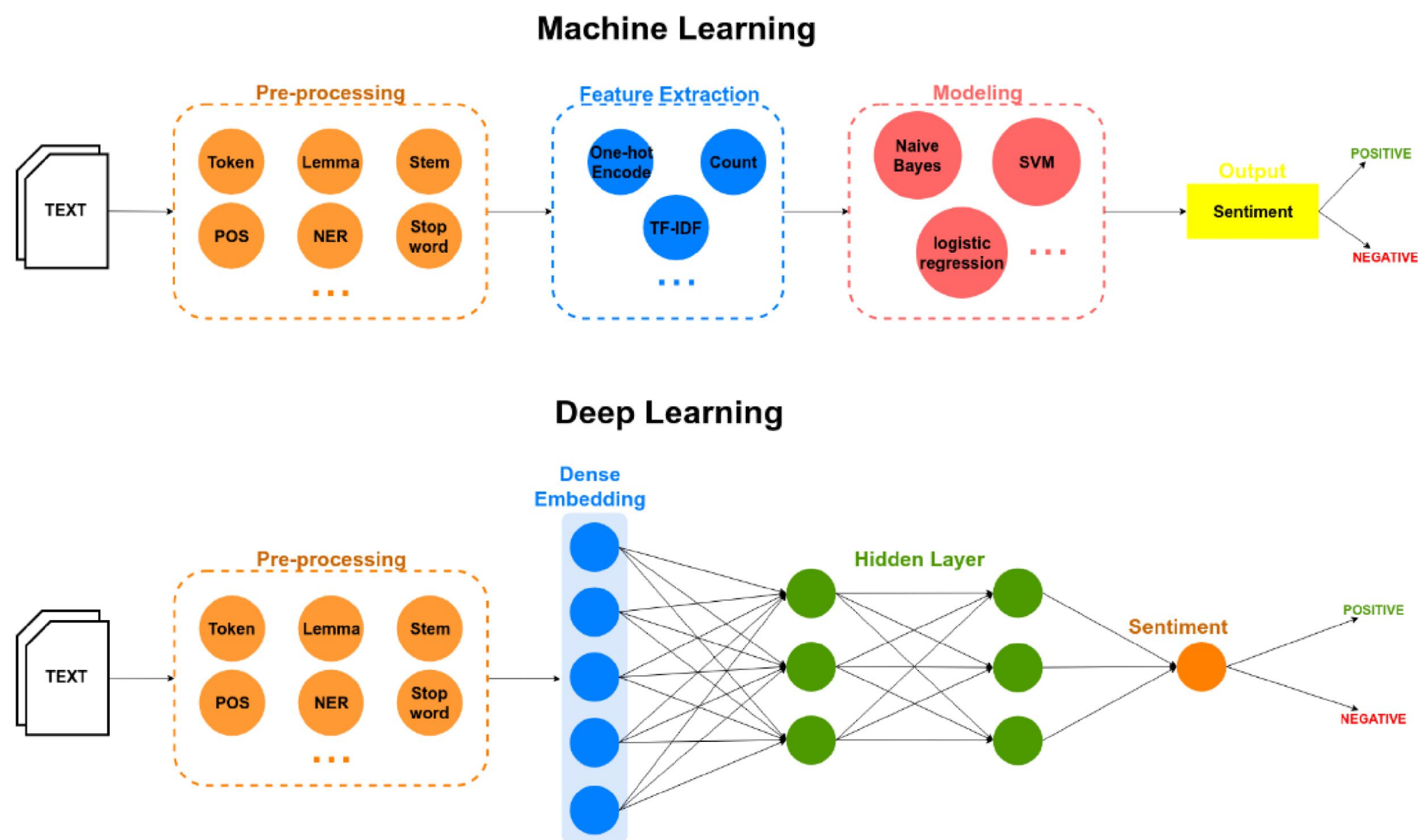
Multi-layer perceptron model

- Step functions are just perceptrons!
 - “Features” are outputs of a perceptron
 - Combination of features output of another





КАКИЕ ДАННЫЕ БУДЕМ ИСПОЛЬЗОВАТЬ



Для реализации задач классификации или предсказания будущих значений доходности / котировок криптоактивов мы будем использовать следующие виды данных:

- 1) Биржевые характеристики активов (дневные данные): цены закрытия и открытия, максимальные и минимальные значения цен, объем торгов
- 2) Технические индикаторы: MACD, RSI, Полосы Боллинджера, MA, EMA и др.
- 3) Текстовые корпуса: сообщения и комментарии с различных веб-ресурсов. Перевод текстовых данных в векторное пространство.



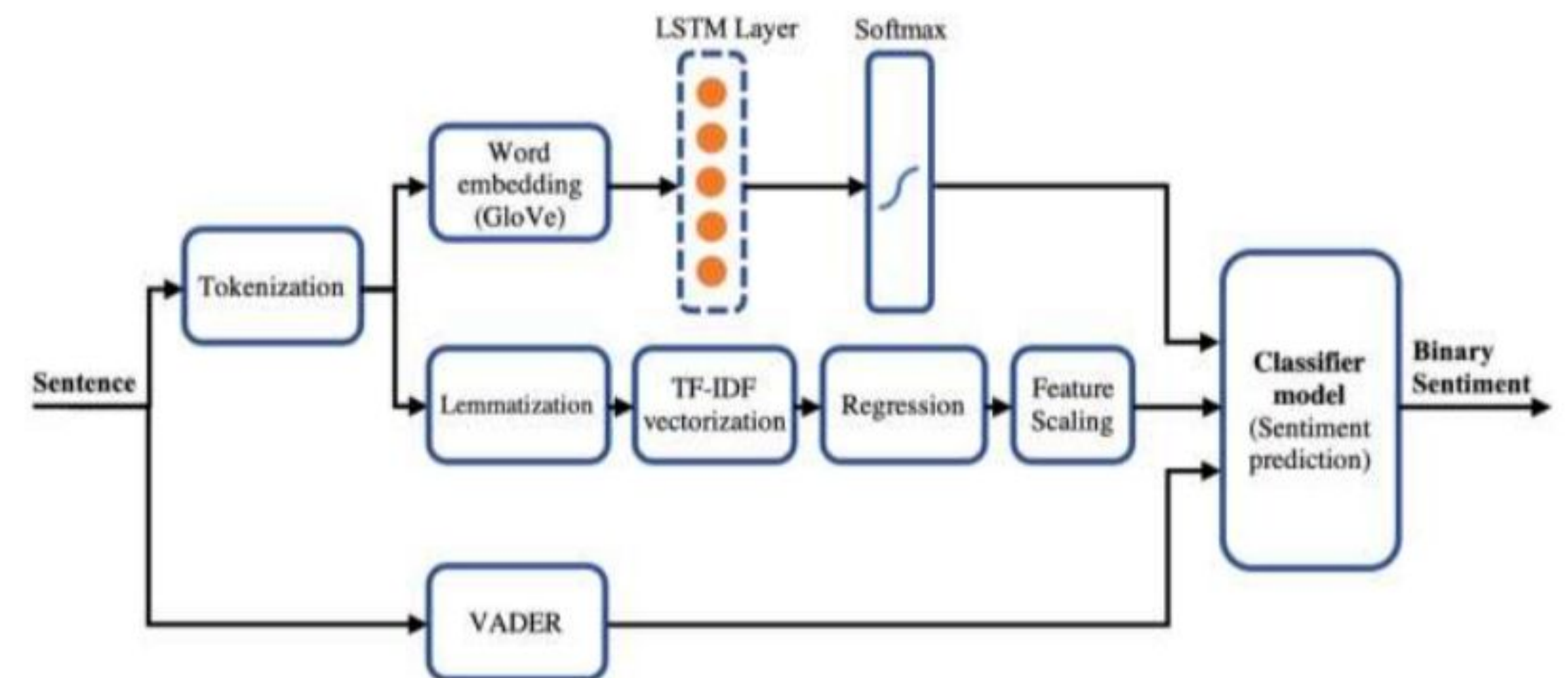
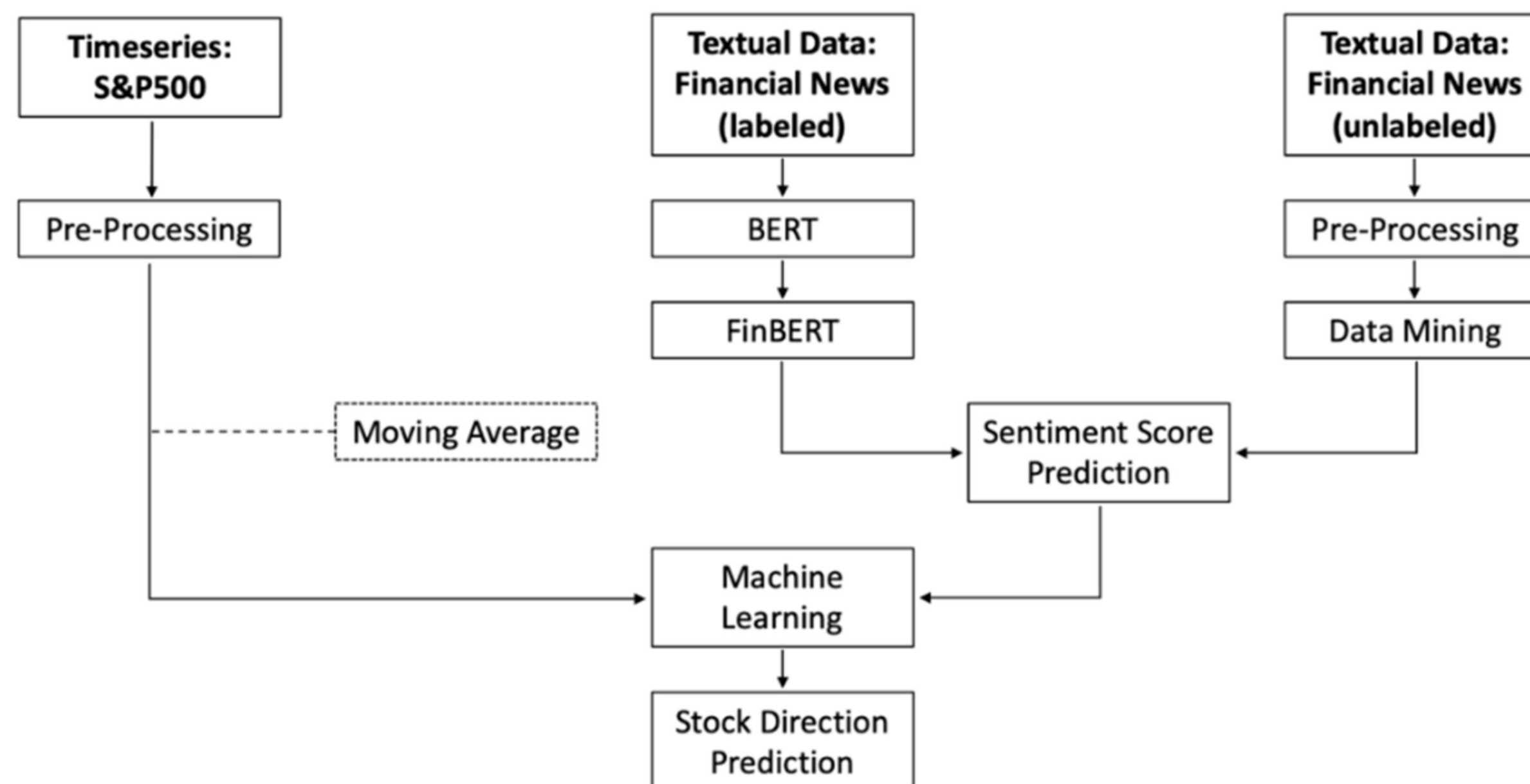
ПРЕДОБУЧЕННЫЕ МОДЕЛИ

Мы также попробуем использовать несколько предобученных моделей для решения разных задач:

1) Для решения задачи классификации текстовых данных нам помогут: VADER, DistilBERT, FinBERT

2) Для решения задачи классификации движения цен (вверх / вниз): BERT

3) Для решения задачи предсказания будущего значения цен / доходностей актива: BERT

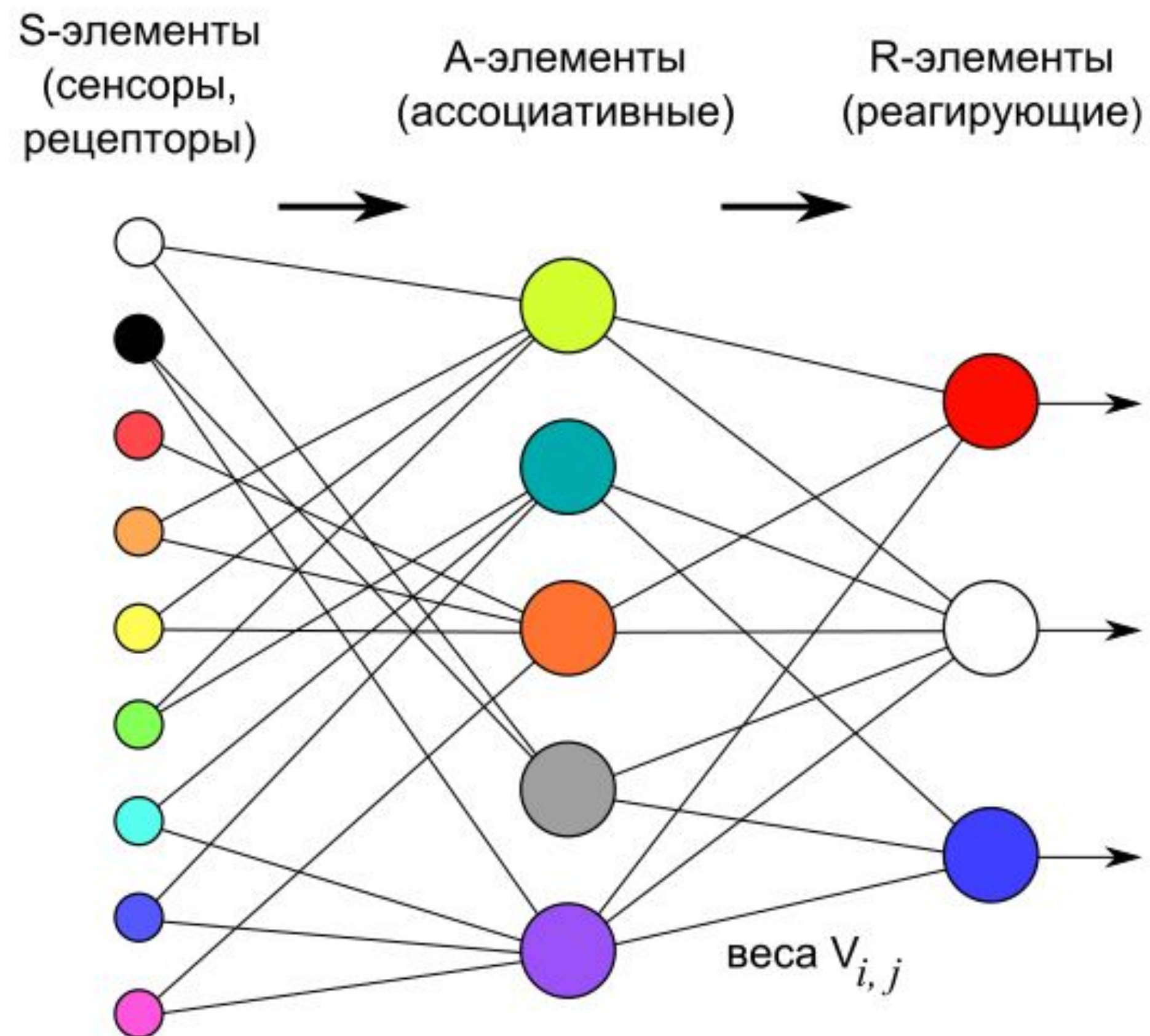


Пример гибридной архитектуры. [Источник](#)



ПЕРЦЕПТРОН РОЗЕНБЛАТТА

Перцептрон состоит из трёх типов элементов, а именно: поступающие от датчиков сигналы передаются ассоциативным элементам, а затем — реагирующим элементам.



Концептуальная модель

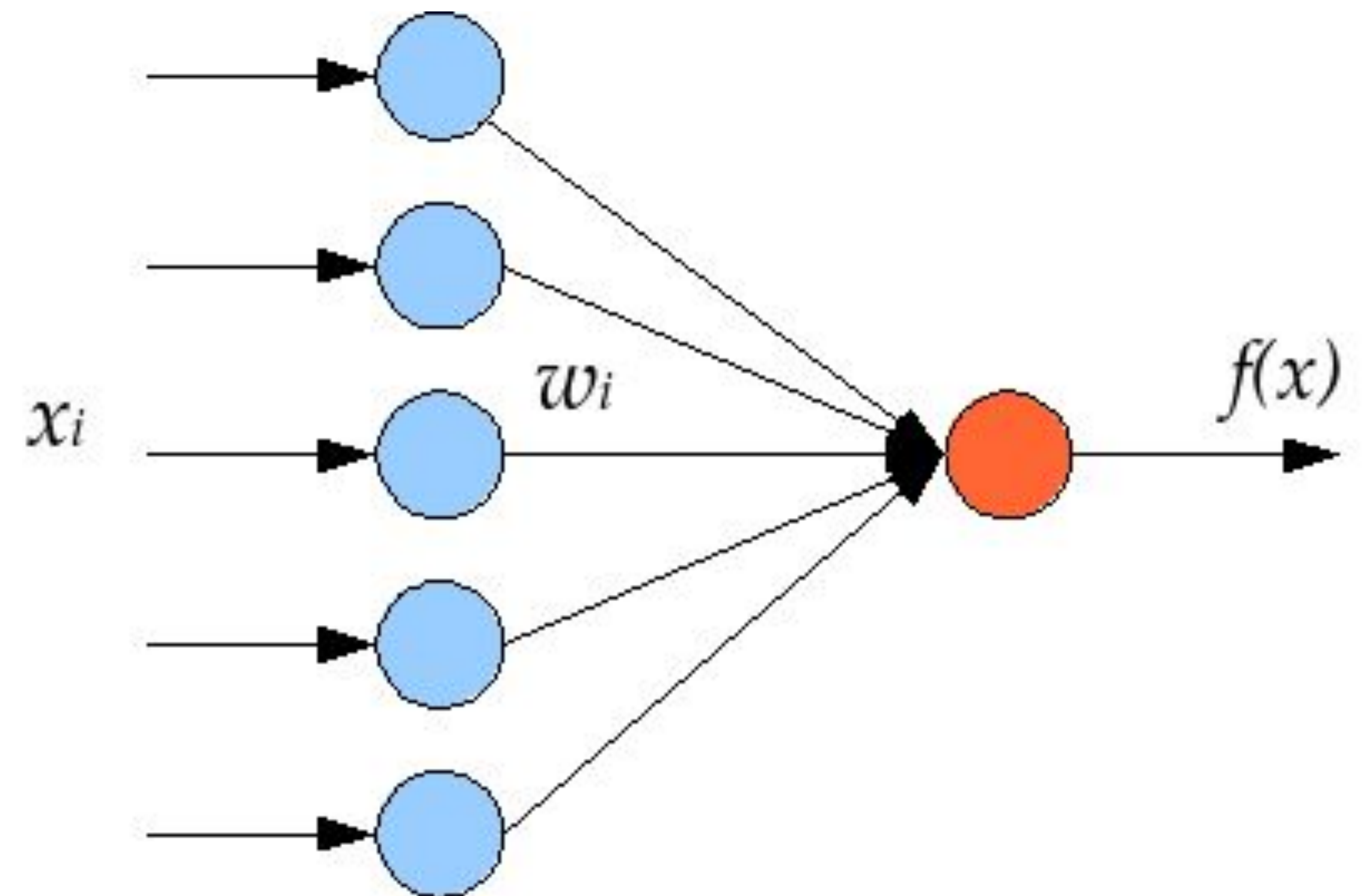


Схема искусственного нейрона



ПЕРЦЕПТРОН РОЗЕНБЛАТТА

Основное условие определения значения выходного слоя – функция, состоящая из двух условий (0 и 1):

Перцептрон:

$$f(x, w, b) = \begin{cases} 1 & \text{если } \sum_{i=1}^n w_i * x_i + b > 0 \\ 0 & \text{если } \sum_{i=1}^n w_i * x_i + b \leq 0 \end{cases}$$

Где w – вектор весов, x – вектор входных активаций, b – смещение (bias), а $f(x, w, b)$ – выходная активация перцептрона

Основное условие

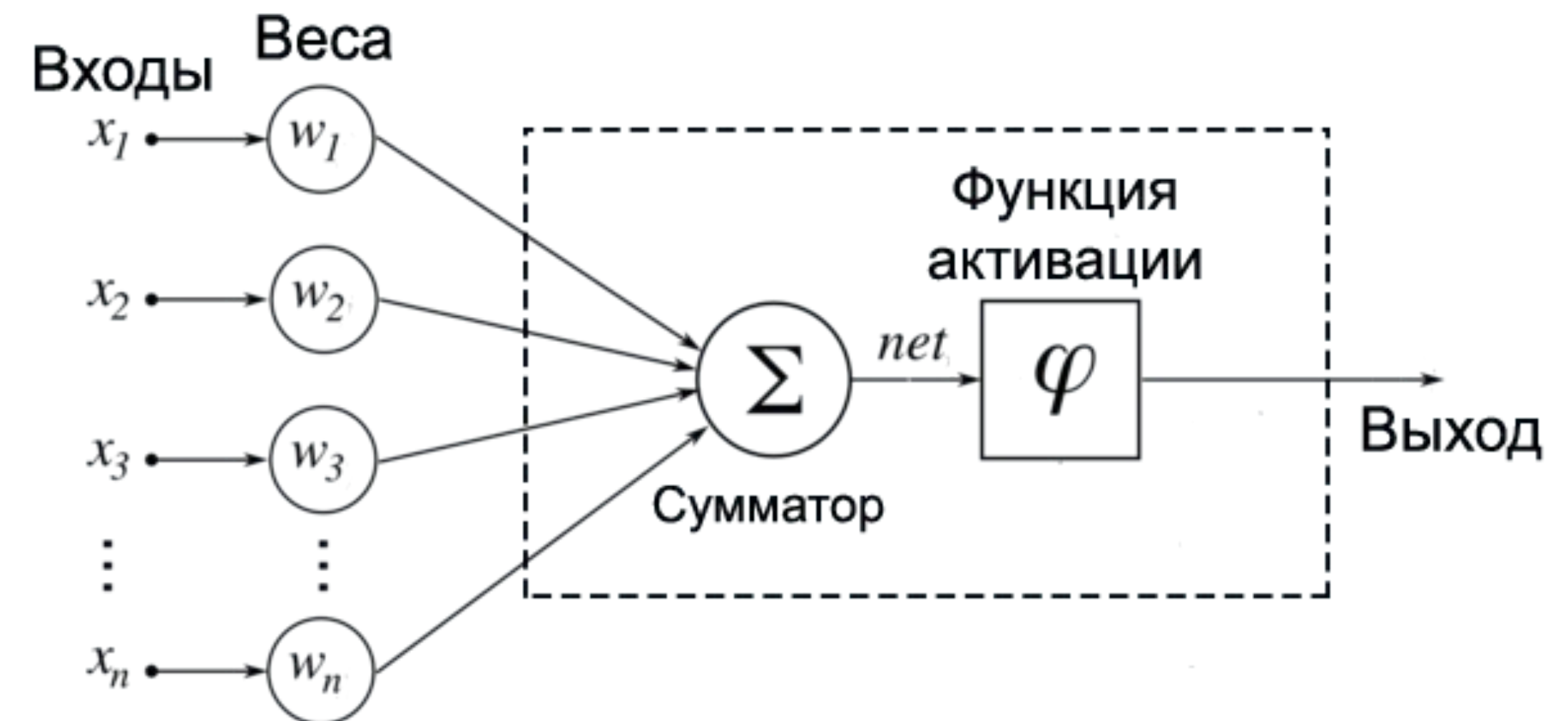


Схема искусственного нейрона

Функция активации принимает собой линейный вид, либо такая функция не используется совсем



КАК ИЩЕМ ВЕСА ДЛЯ ПЕРЦЕПТРОНА?

Случай без функций активации (с учетом нескольких функций ошибок):

- 1) Находим по каждому наблюдению выходное значение с нулевыми весами по формуле:

$$\vec{Y}_{predicted} = \vec{X}^T \vec{W} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \cdot \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = [x_1 \quad x_2] \cdot \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = [x_1 w_1 + x_2 w_2]$$

- 2) Выбираем функцию ошибки и определяем общую дельту

$$L = \frac{1}{N} \sum_{i=1}^N (y_{predicted(i)} - y_{expected(i)})^2$$

среднеквадратичная ошибка
(Mean Squared Error, MSE)

$$L = \frac{1}{N} \sum_{i=1}^N |y_{predicted(i)} - y_{expected(i)}|$$

среднее отклонение (Mean
Absolute Error, MAE)

$$L = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_{predicted(i)} - y_{expected(i)})^2}$$

среднеквадратичное
отклонение (Root MSE)

$$L = - \sum_{i=1}^N \sum_{j=1}^M y_{expected(ij)} \log(y_{predicted(ij)})$$

перекрестная энтропия
(Cross entropy)



КАК ИЩЕМ ВЕСА ДЛЯ ПЕРЦЕПТРОНА?

3) Обратное распространение ошибки (backpropagation)

Цель обучения нейронной сети – это минимизация функции ошибки (L):

$$L \rightarrow \min$$

$$\vec{w}^{(k+1)} = \vec{w}^k - \mu \nabla L(\vec{w}^k)$$

де k – k -ая итерация обучения нейронной сети;
 μ – шаг обучения (learning rate), обычно это может быть 0.1; 0.01
 ∇L – градиент функции-ошибки

X1	W1	X2	W2	Y
5	0	1	0	0
6	0	2	0	0
7	0,2	3	0,5	1



КАК ИЩЕМ ВЕСА ДЛЯ ПЕРЦЕПТРОНА?

Решить задачу калибровки весов на основе текущих данных (уровень обучения используем = 0,1; 0,01; 0,001):

$$L = \frac{1}{N} \sum_{i=1}^N (y_{predicted(i)} - y_{expected(i)})^2$$

среднеквадратичная ошибка
(Mean Squared Error, MSE)

$$L = \frac{1}{N} \sum_{i=1}^N |y_{predicted(i)} - y_{expected(i)}|$$

среднее отклонение (Mean
Absolute Error, MAE)

X1	W1	X2	W2	Y
5	0	1	0	0
6	0	2	0	0
7	0,2	3	0,5	1

Задачу требуется решить за счет двух функций ошибок с разными уровнями обучения!

Задачу требуется решить за счет двух функций ошибок с разными уровнями обучения!

Какие проблемы возникли?

Что можем сказать про уровни обучения?

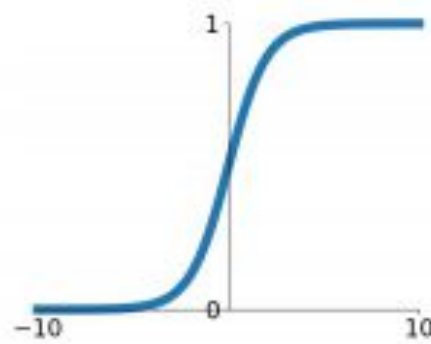
Какую функцию потерь лучше использовать?



ФУНКЦИИ АКТИВАЦИИ

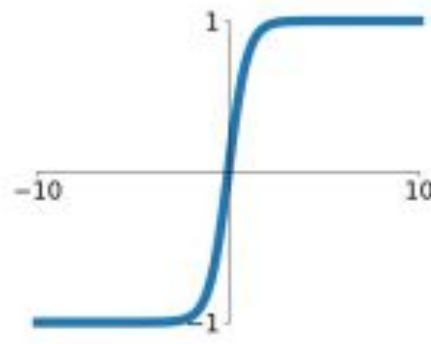
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



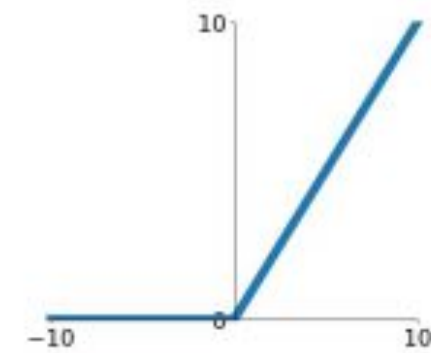
tanh

$$\tanh(x)$$



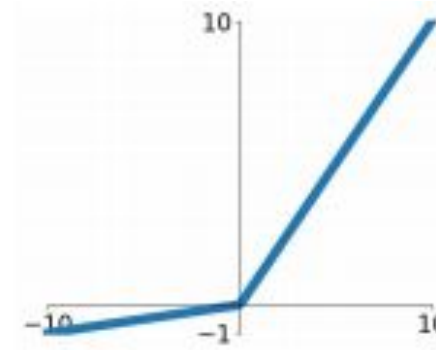
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$



Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

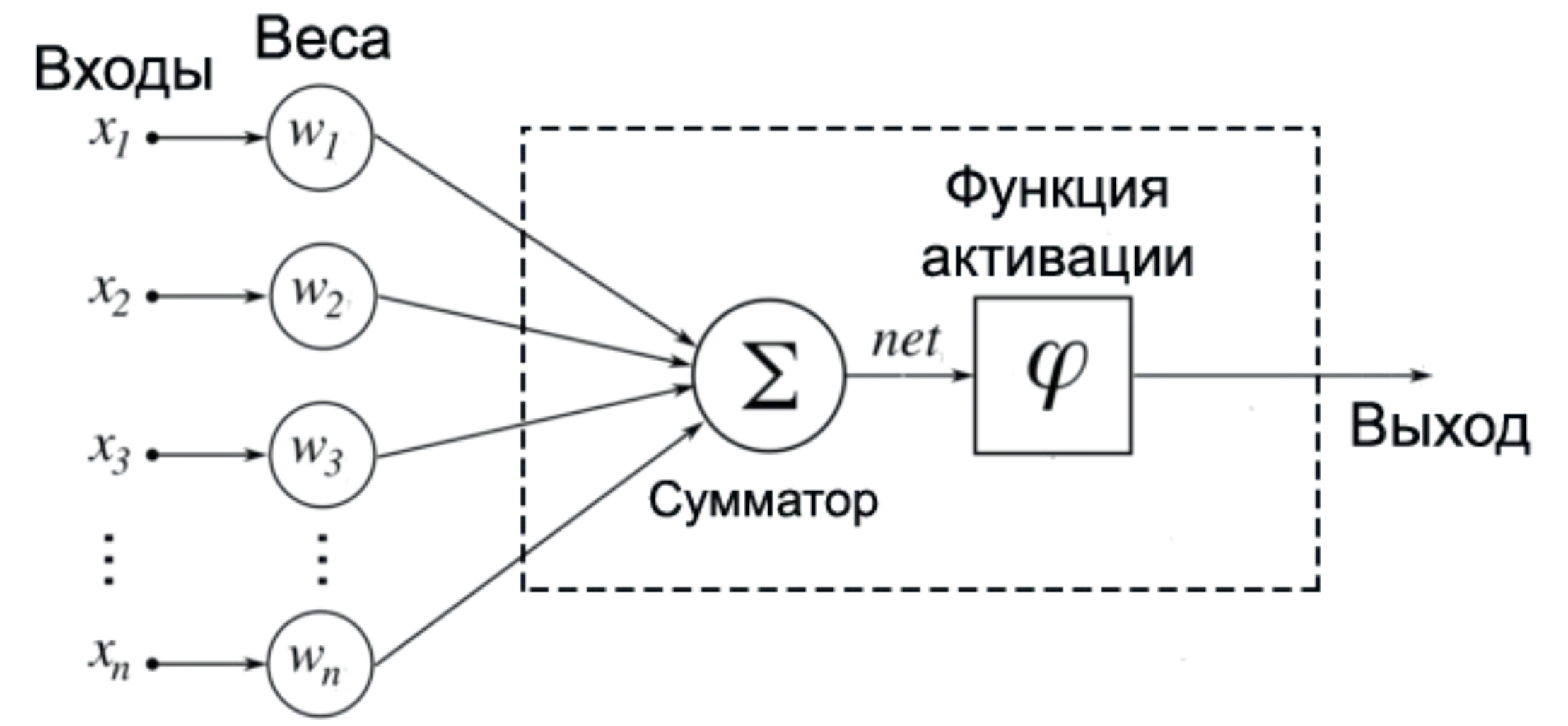
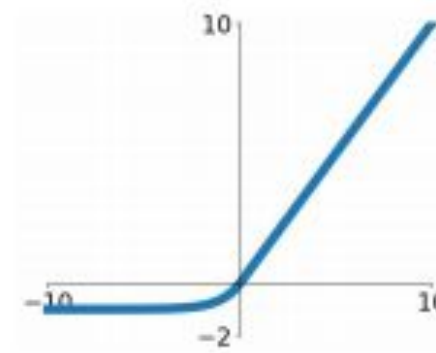


Схема искусственного нейрона

$$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$$



КАК ИЩЕМ ВЕСА ДЛЯ ПЕРЦЕПТРОНА?

Решить задачу калибровки весов на основе текущих данных (уровень обучения используем = 0,001):

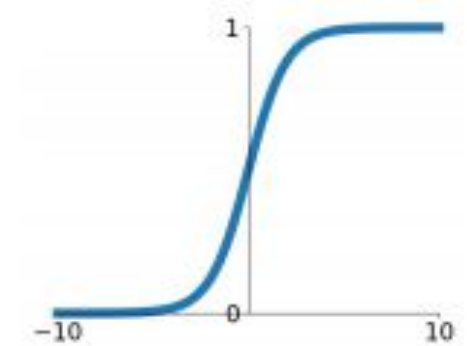
$$L = \frac{1}{N} \sum_{i=1}^N (y_{predicted(i)} - y_{expected(i)})^2$$

среднеквадратичная ошибка
(Mean Squared Error, MSE)

X1	W1	X2	W2	Y
5	0	1	0	0
6	0	2	0	0
7	0,2	3	0,5	1

Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



Задание: Найти новые веса после первой итерации за счет использования среднеквадратичной ошибки, уровня обучения 0,001 и сигмоидной функции активации.

Подсказка: градиент функции ошибки теперь зависит еще от функции активации. Требуется найти сложную производную



РЕШЕНИЕ ЗАДАЧИ

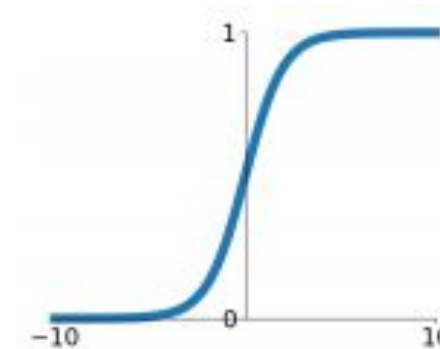
Определяем условие ошибки в зависимости от трех компонент:

- 1) конечной квадратичной ошибки;
- 2) функции активации;
- 3) значения перцептрона до функции активации

$$\frac{\partial Loss Total}{\partial W1} = \frac{\partial Loss(out)}{\partial \hat{y}} \times \frac{\partial Activation Function}{\partial x} \times \frac{\partial out}{\partial W1}$$

Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$





РЕШЕНИЕ ЗАДАЧИ

$$\frac{\partial Loss Total}{\partial W1} = \frac{\partial Loss(out)}{\partial \hat{y}} \times \frac{\partial Activation Function}{\partial x} \times \frac{\partial out}{\partial W1}$$

$$\frac{\partial Loss(out)}{\partial \hat{y}} (\hat{y} - y)^2 \Rightarrow 2(\hat{y} - y) * 1$$

$$\frac{\partial Activation Function}{\partial x} \left(\frac{1}{1+e^{-x}} \right) \Rightarrow t = 1 + e^{-x} \Rightarrow \frac{\partial t}{\partial x} (1 + e^{-x}) \Rightarrow -e^{-x}$$

$$\frac{\partial Activation Function}{\partial t} \left(\frac{1}{t} \right) \Rightarrow \frac{-1}{t^2}$$

$$\frac{\partial Activation Function}{\partial x} \left(\frac{1}{1+e^{-x}} \right) \Rightarrow \frac{-1}{(1+e^{-x})^2} * -e^{-x} \Rightarrow \frac{e^{-x}}{(1+e^{-x})^2}$$

$$\frac{\partial out}{\partial W1} (j_i * w_1 + j_h * w_2) \Rightarrow j_i$$

$$\frac{\partial Loss Total}{\partial W1} \Rightarrow 2(\hat{y} - y) * 1 \times \frac{e^{-x}}{(1+e^{-x})^2} \times j_i$$



МНОГОСЛОЙНЫЙ ПЕРЦЕПТРОН РУМЕЛЬХАРТА

Многослойный перцептрон — частный случай перцептрона Розенблатта, в котором один **алгоритм обратного распространения ошибки обучает все слои**. Название по историческим причинам не отражает особенности данного вида перцептрона, то есть не связано с тем, что в нём имеется несколько слоёв (так как несколько слоёв было и у перцептрона Розенблатта). Особенностью является наличие более чем одного обучаемого слоя (как правило — два или три).

Отличия от простого перцептрона Розенблатта:

- Использование нелинейной функции активации, как правило **сигмоидальной**.
- Число обучаемых слоев **больше одного**.
- Допускается **произвольная архитектура** связей (в том числе, и полносвязные сети).
- Обучение проводится не до отсутствия ошибок после обучения, а до стабилизации весовых коэффициентов при обучении или прерывается ранее, чтобы избежать **переобучения**.

Вопрос: Прошлая задача по нахождению весов входных параметров является простым или многослойным перцептроном?



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ