





# РАСПОЗНАВАНИЕ РУКОПИСНЫХ ЦИФР НЕЙРОННЫМИ СЕТЯМИ

Алексей Кузьмин

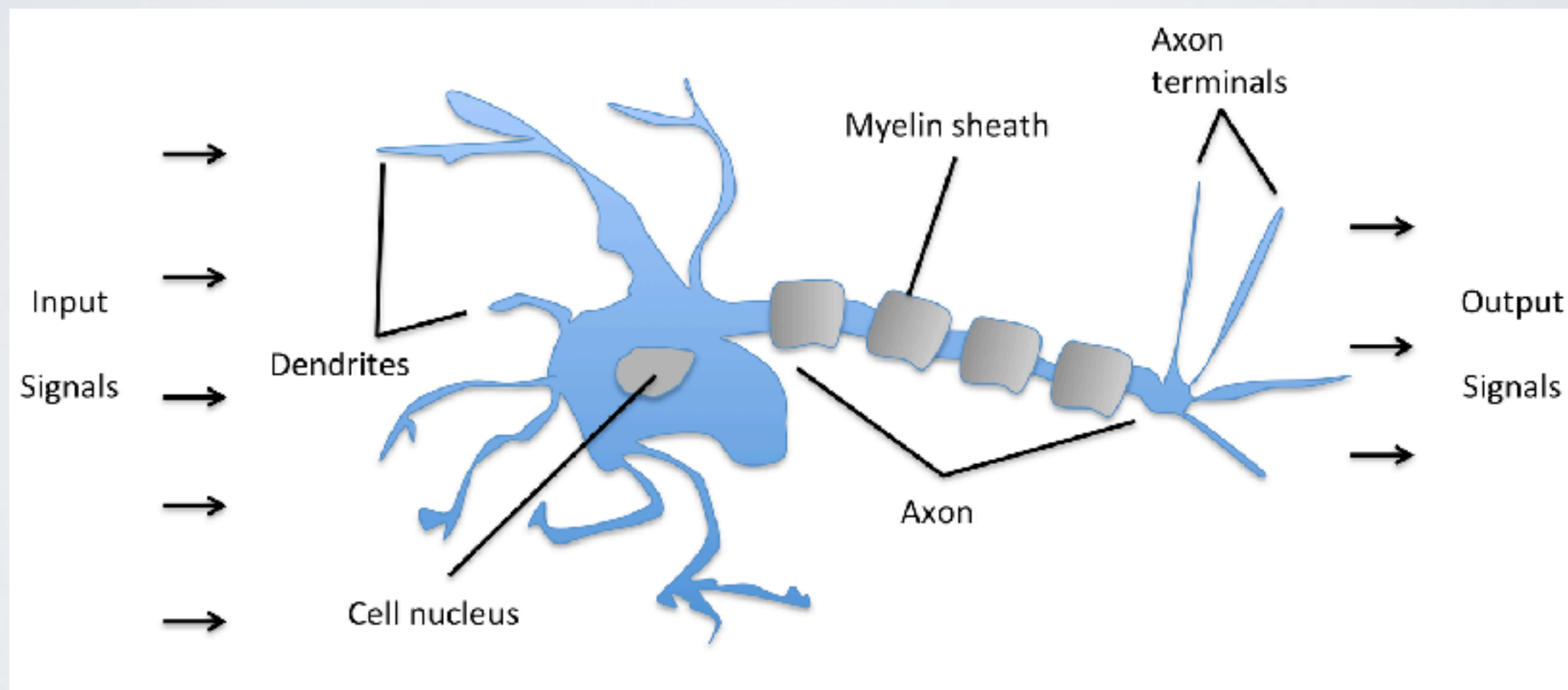
# ЗАДАЧА МАШИННОГО ОБУЧЕНИЯ

- Конкретно: Обучение с учителем
  - Есть обучающая выборка для которой что-то известно
  - Задача - построить алгоритм, который сумеет предсказывать это самое что-то по ранее неизвестным данным

# ПОЧЕМУ НЕЙРОСЕТИ?

- Хорошо и очень хорошо подходят для некоторого круга задач (обработка картинок, звуков, некоторые задачи работы с текстом)
- SIMD, благодаря этому - отлично считаются на GPU
- Много туториалов для начинающих
- Много готовых архитектур

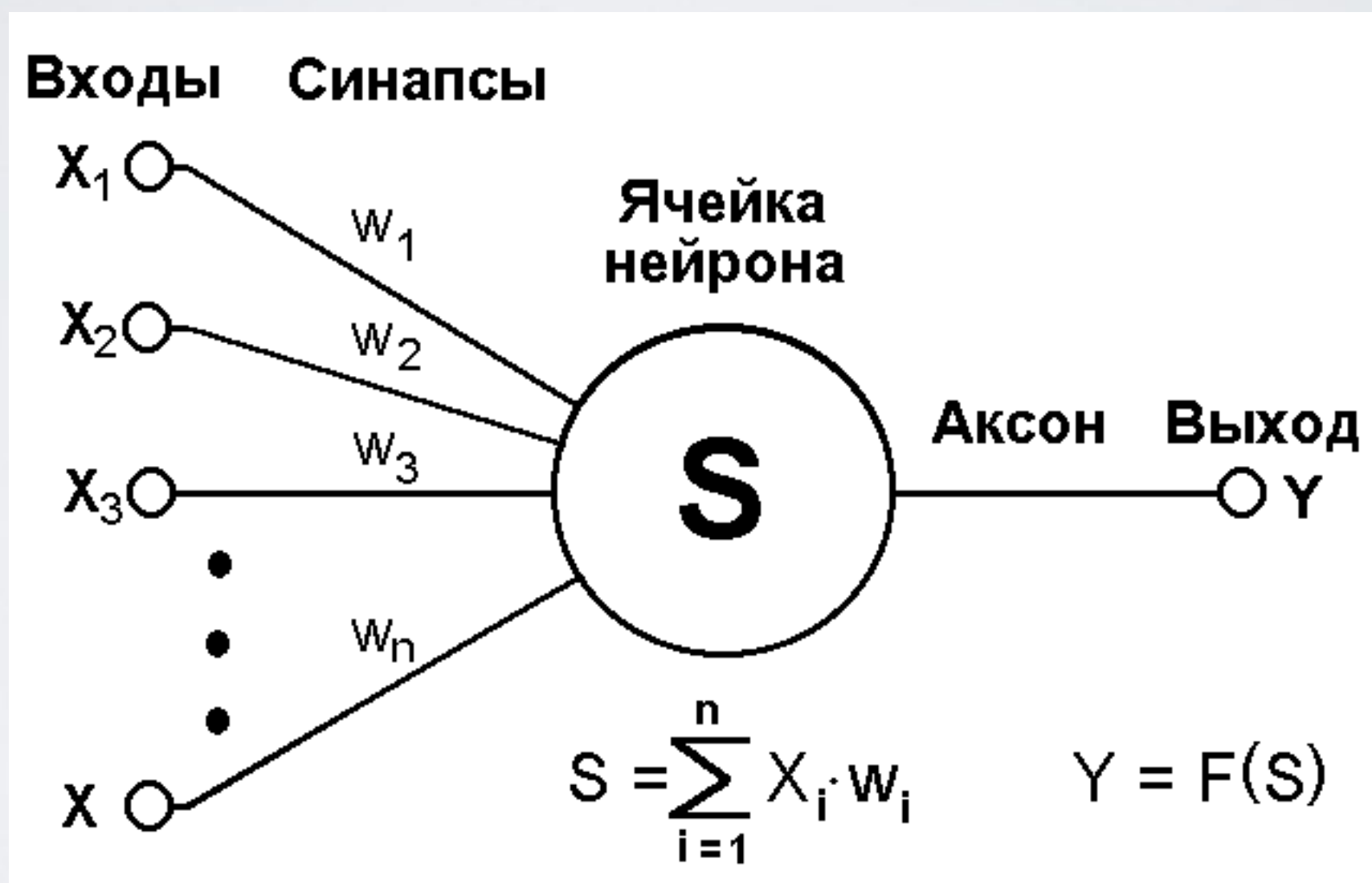
# НЕЙРОН



# ИСКУССТВЕННЫЙ НЕЙРОН

- 1943 г. - Маккалок и Питтс (упрощенная концепция)
- 1957 г. - Фрэнк Розенблатт (правило обучения)

# ИСКУССТВЕННЫЙ НЕЙРОН





# ИСКУССТВЕННЫЙ НЕЙРОН

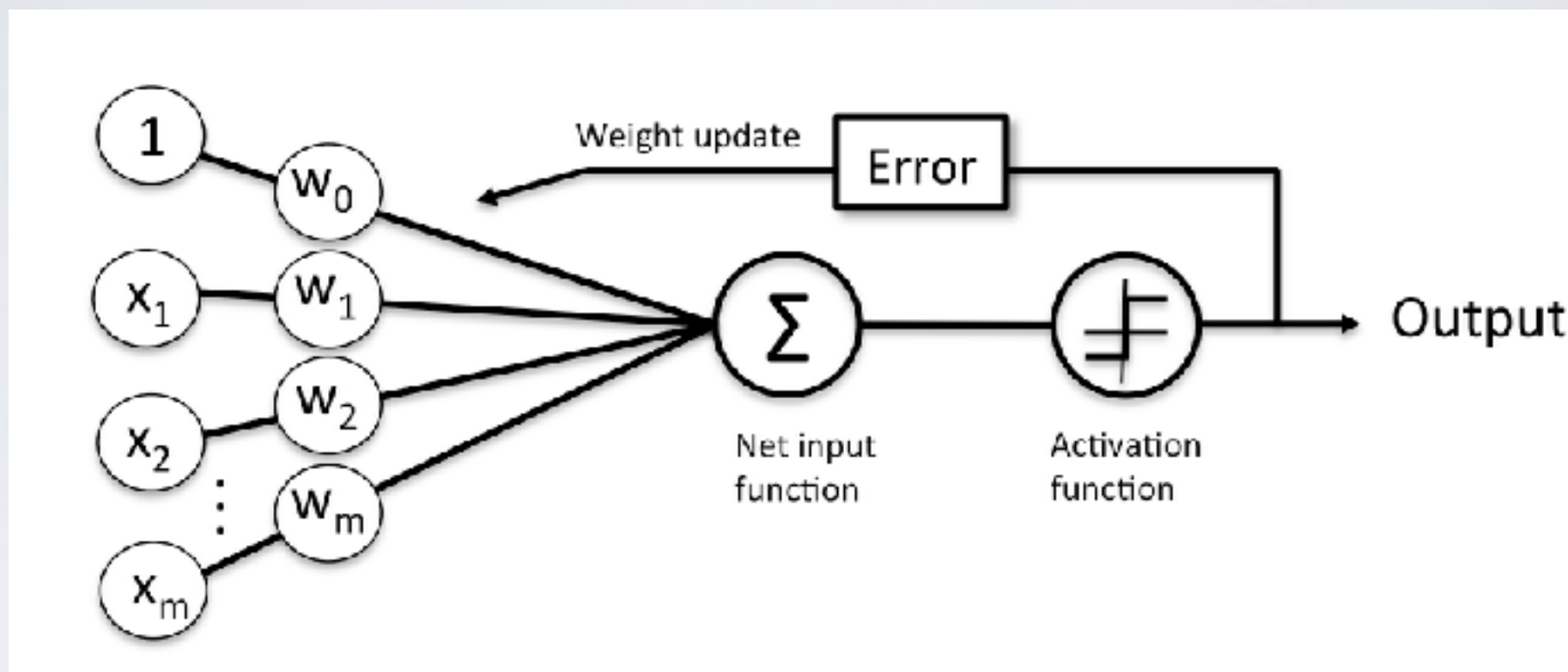
$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix}, x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

$$z = w_1 x_1 + w_2 x_2 + \dots + w_m x_m = w^T x$$

$$\phi(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ -1 & \text{else} \end{cases}$$



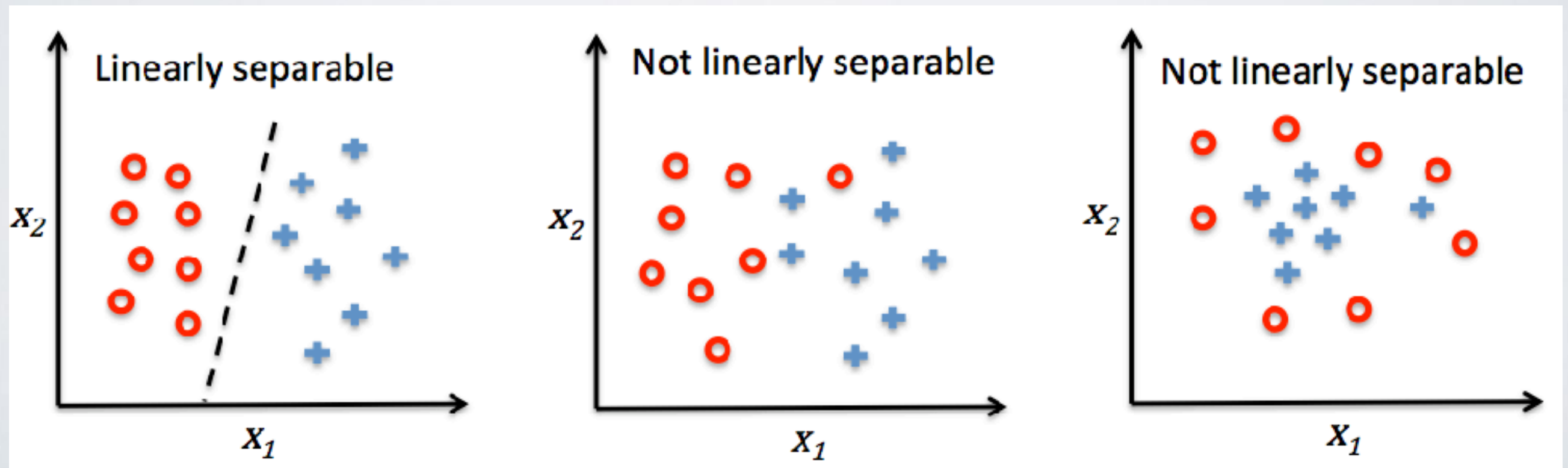
# ПРАВИЛО ОБУЧЕНИЯ



$$\Delta w_j = \eta (y^i - \hat{y}^i) x_j^i$$

# СХОДИМОСТЬ

- Гарантируется, только если классы линейно-разделимы



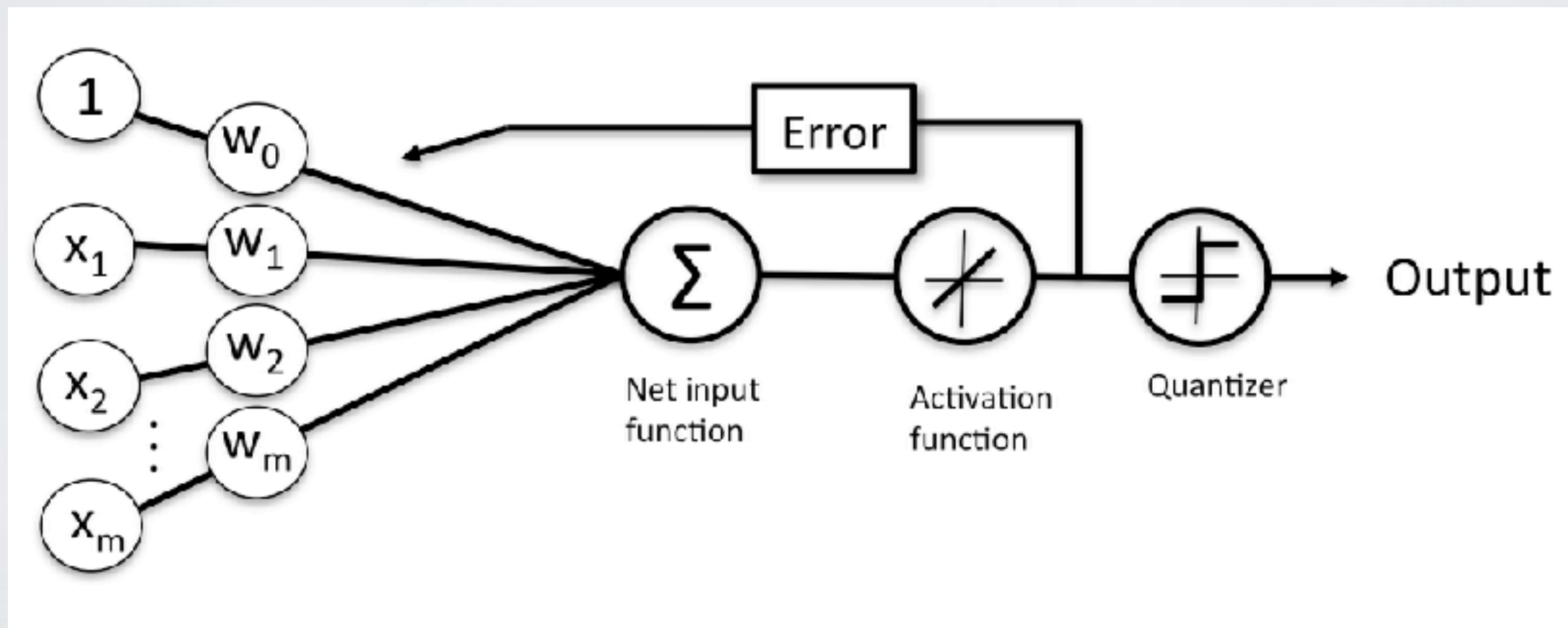
# БЛИЖЕ К ДЕЛУ

```
def fit(self, X, y):  
    self.weights_ = np.zeros(1 + X.shape[1])  
  
    for _ in range(self.n_iter):  
        for xi, target in zip(X, y):  
            update = (target - self.predict(xi))  
            self.weights_[1:] += update * xi  
            self.weights_[0] += update  
    return self
```

# ПРАКТИКА I

# ADALINE

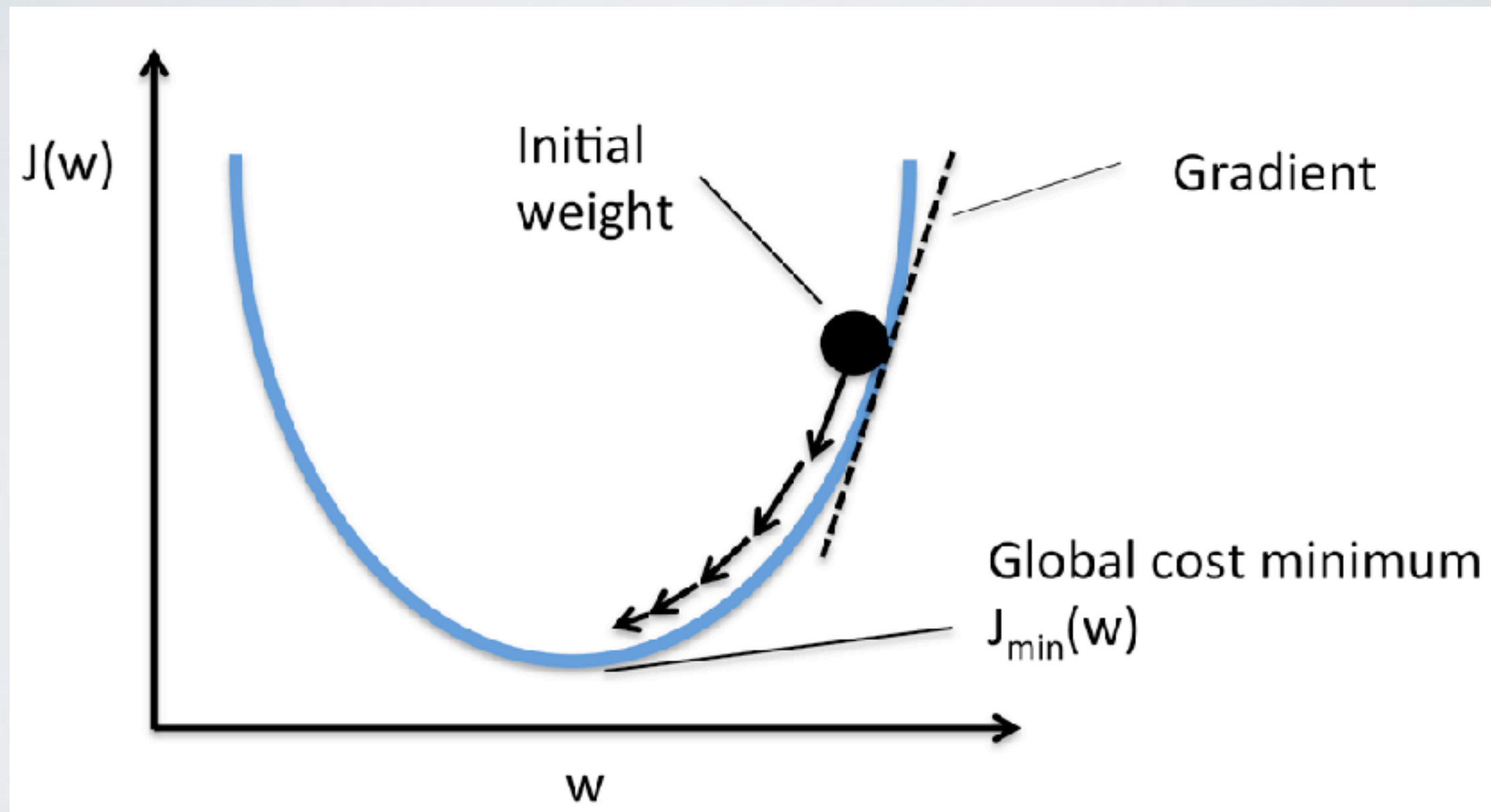
- Как понять что мы сделали хороший классификатор?



# ФУНКЦИЯ СТОИМОСТИ

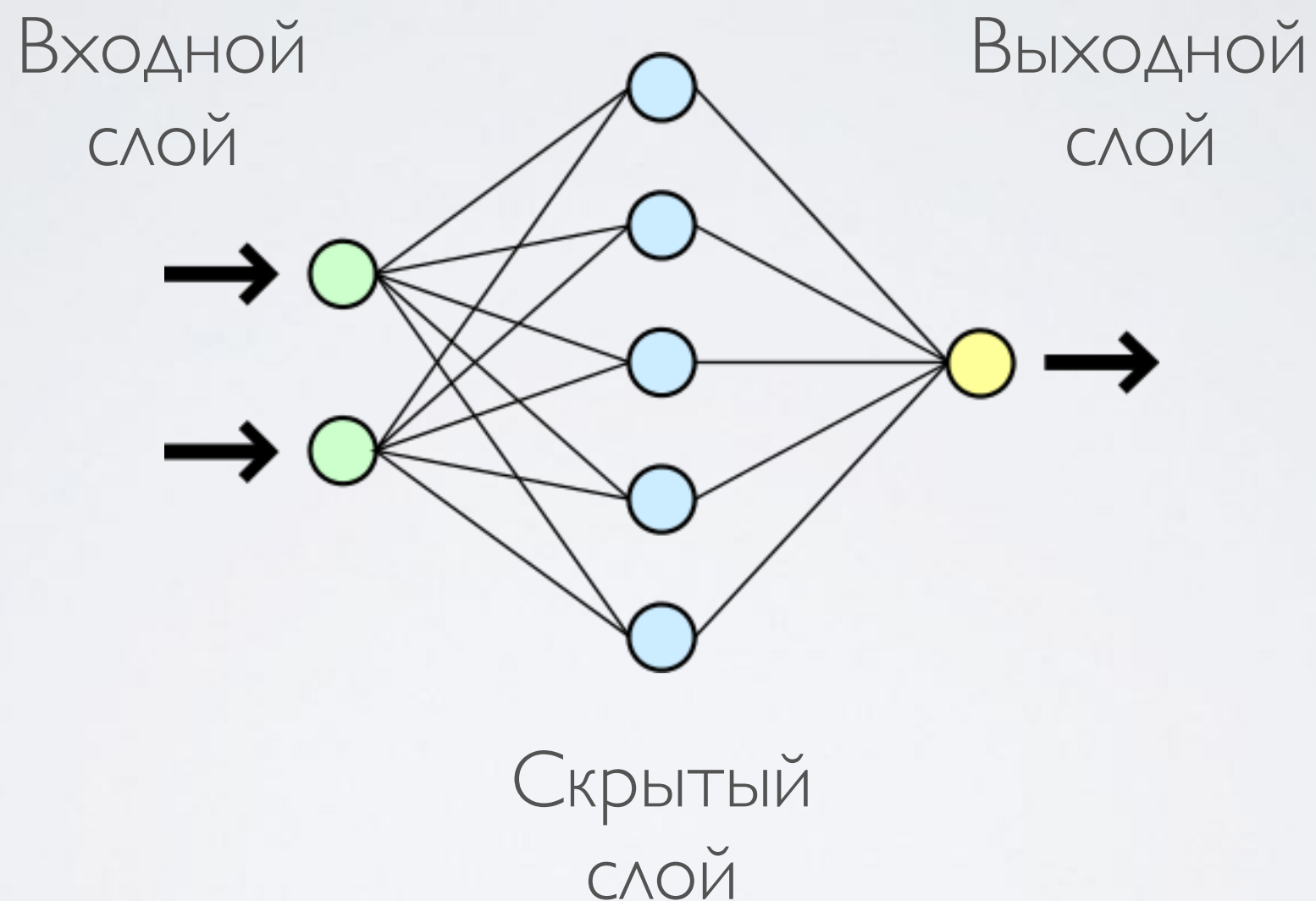
$$J(w) = \frac{1}{2} \sum_{i=1}^n (y^i - \text{activat}^i)^2$$

# КАК ЕЕ УЧИТЬ?





# НЕЙРОННАЯ СЕТЬ



# ПРОБЛЕМА ГРАДИЕНТНОГО СПУСКА

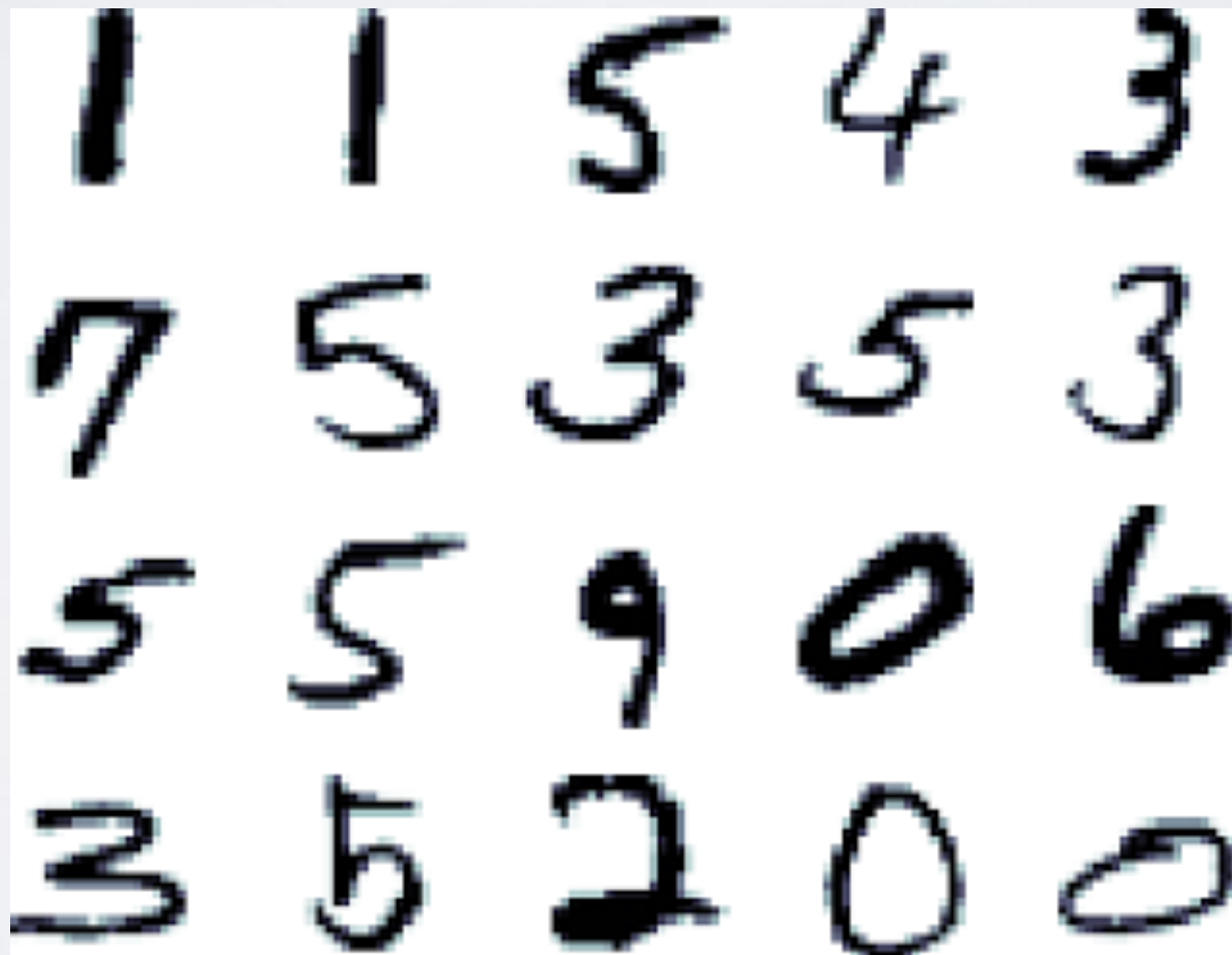
- Много весов - много частных производных, много вычислений (долго) - первый кризис нейронных сетей

# МЕТОД ОБРАТНОГО РАСПРОСТАНЕНИЯ ОШИБОК

- Просто хак, который позволяет красиво посчитать градиенты, не делая кучу прямых прогонов

# РАСПОЗНАВАНИЕ ЦИФР

- MNIST



# ИНСТРУМЕНТЫ

- keras
- numpy
- scikit-learn

# ПРАКТИКА 2

НЕУЖЕЛИ ВСЕ ТАК КРУТО?



# МИНУСЫ

- Сложно улучшать
- Нужно много данных
- Крутая кривая обучения

ВОПРОСЫ?



СПАСИБО ЗА ВНИМАНИЕ