

# **Глубинное обучение для текстовых данных (NLP)**

# Информация по курсу

## Формула оценки :

Итог = Округление( $0.4 * ДЗ + 0.3 * КР + 0.3 * Э$ )

- 6 домашних заданий
- Контрольная работа (письменная) пройдет в середине семестра
- Экзамен устный
- Если оценка  $\geq 8$  **до** округления, можно получить автомат

## Ссылки:

Чат в тг: <https://t.me/+vC-nISGYBwpjMDMy>

GitHub: <https://github.com/ashaba1in/hse-nlp>

# Про домашки

Все дз будут требовать написания кода и отчета к нему

## Оформление:

- Строгих правил нет
- Лучше оформлять в виде проекта
  1. Проще не запутаться в коде, когда его становится много
  2. Нам проще проверять структурированные домашки
  3. Проект можно выложить на гит и хвастаться им

## Отчет:

- PDF документ с описанием проделанной работы
- Нужен *нам* для упрощения проверки
- Нужен *вам* для умения рассказывать о работе

# Классификация текста

# План

- Виды задач классификации
- Генеративные и дискриминативные модели
- Нейронные сети для текста

# Виды задачи классификации

## Бинарная классификация

- Сообщение спам или не спам?

## Многоклассовая (multi-class) классификация

- Насколько срочно надо дать ответ клиенту?

## Многоклассовая классификация с пересекающимися классами (multi-label classification)

- Какая тематика у новости?

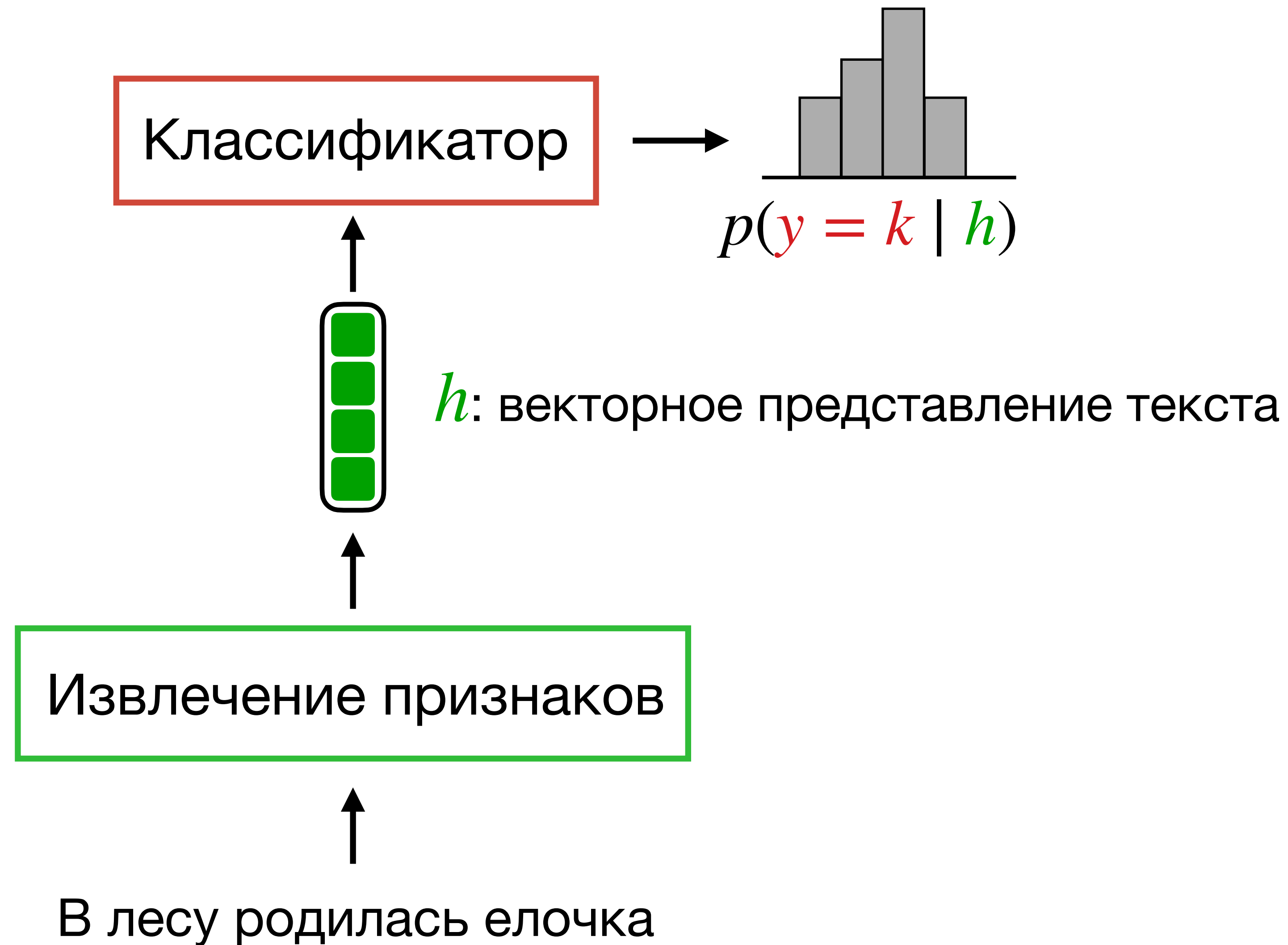
## Классификация слов

- Распознавание именованных сущностей (NER)
- Генерация текста (спойлеры)

# Датасеты для классификации

Название	Задача	Таргет	Размер	Средняя длина	Метрика
SST	тональность	5 или 2	11,855	19	Accuracy
Yelp	тональность	5 или 2	280,000	179	Accuracy
IMDb	тональность	2	50,000	271	Accuracy
QQP	перепаразирование	2	404,291	22	F1 / Accuracy
CoLA	грамматичность	2	10,657	9	Matthew's Corr
AG News	тема	4	120,000	44	Accuracy
Yahoo! Answers	тема	10	1,400,000	131	Accuracy
DBpedia	тема	14	560,000	67	Accuracy

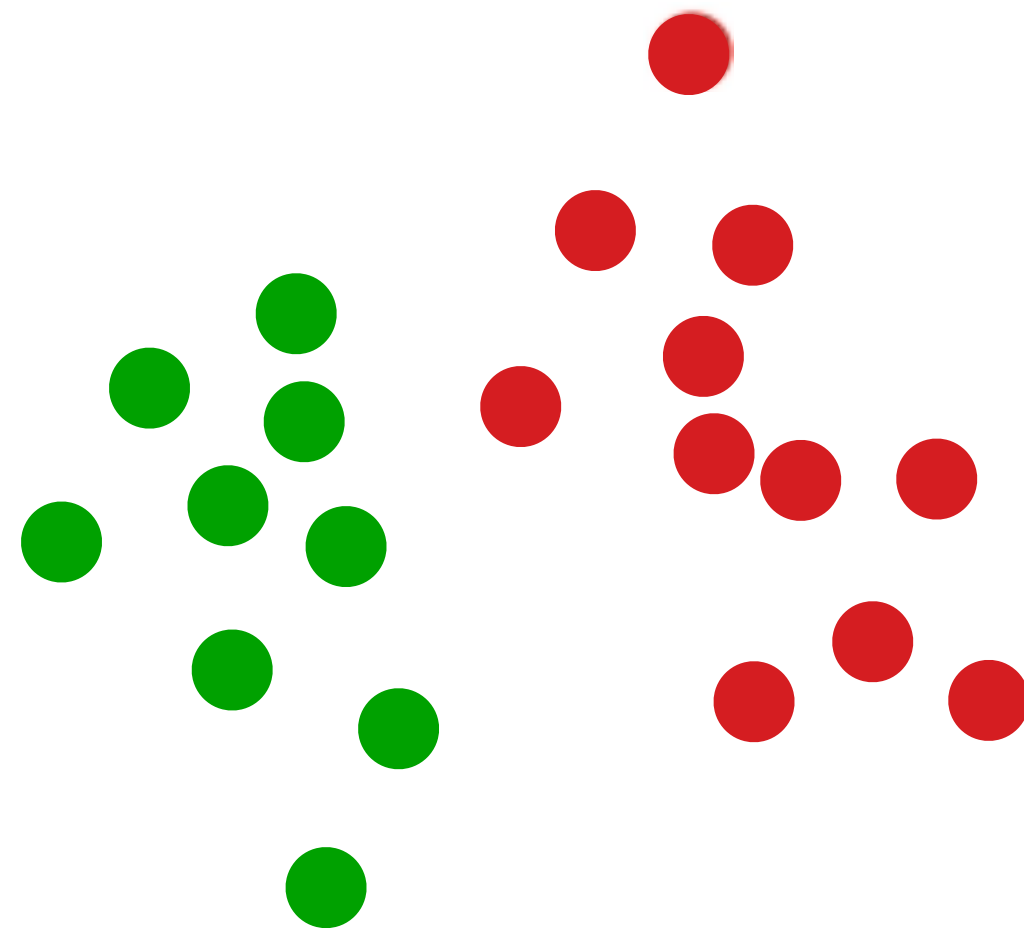
# Общая схема решения





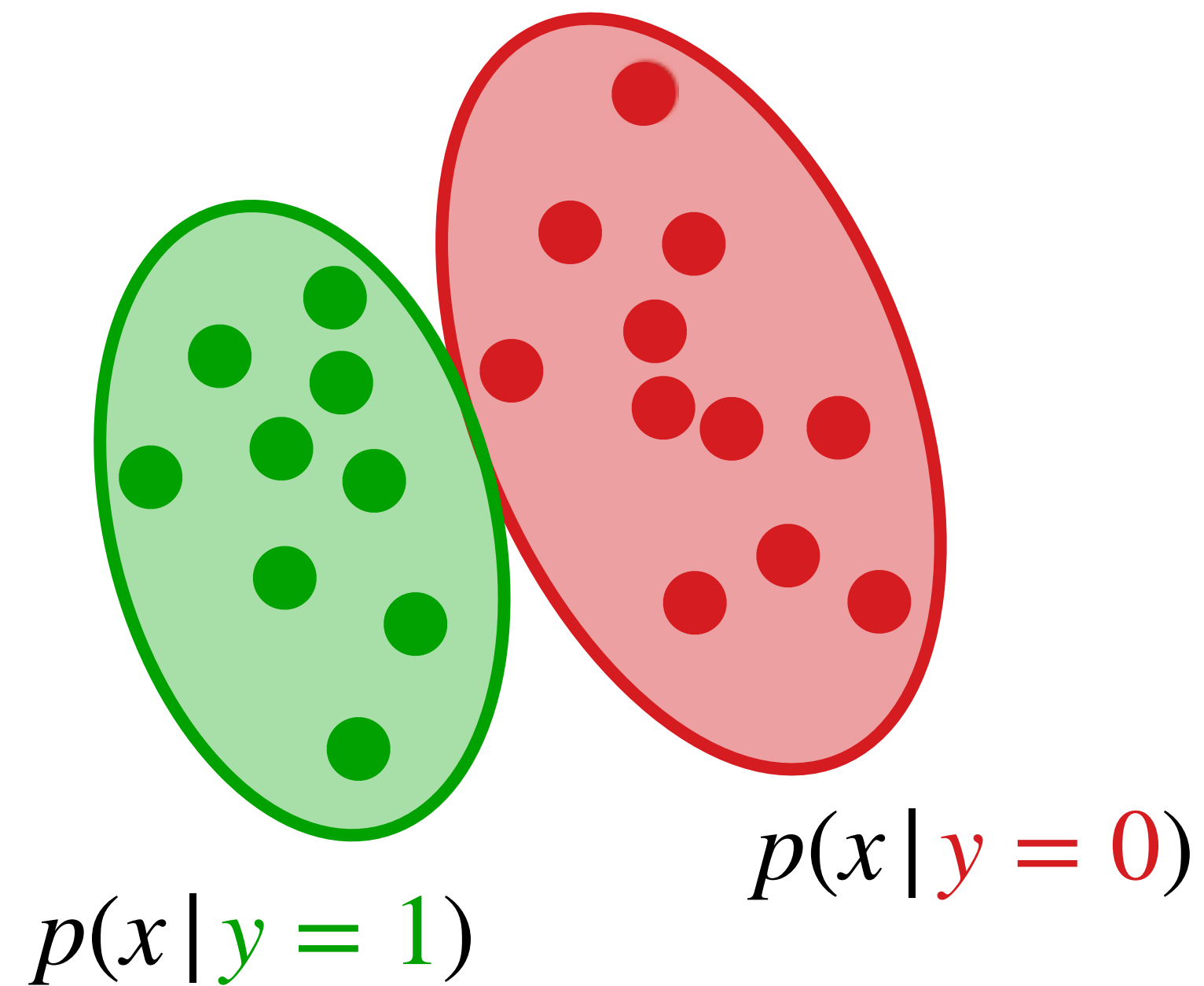
# Генеративные и дискриминативные модели

Пример распределения данных для двух классов

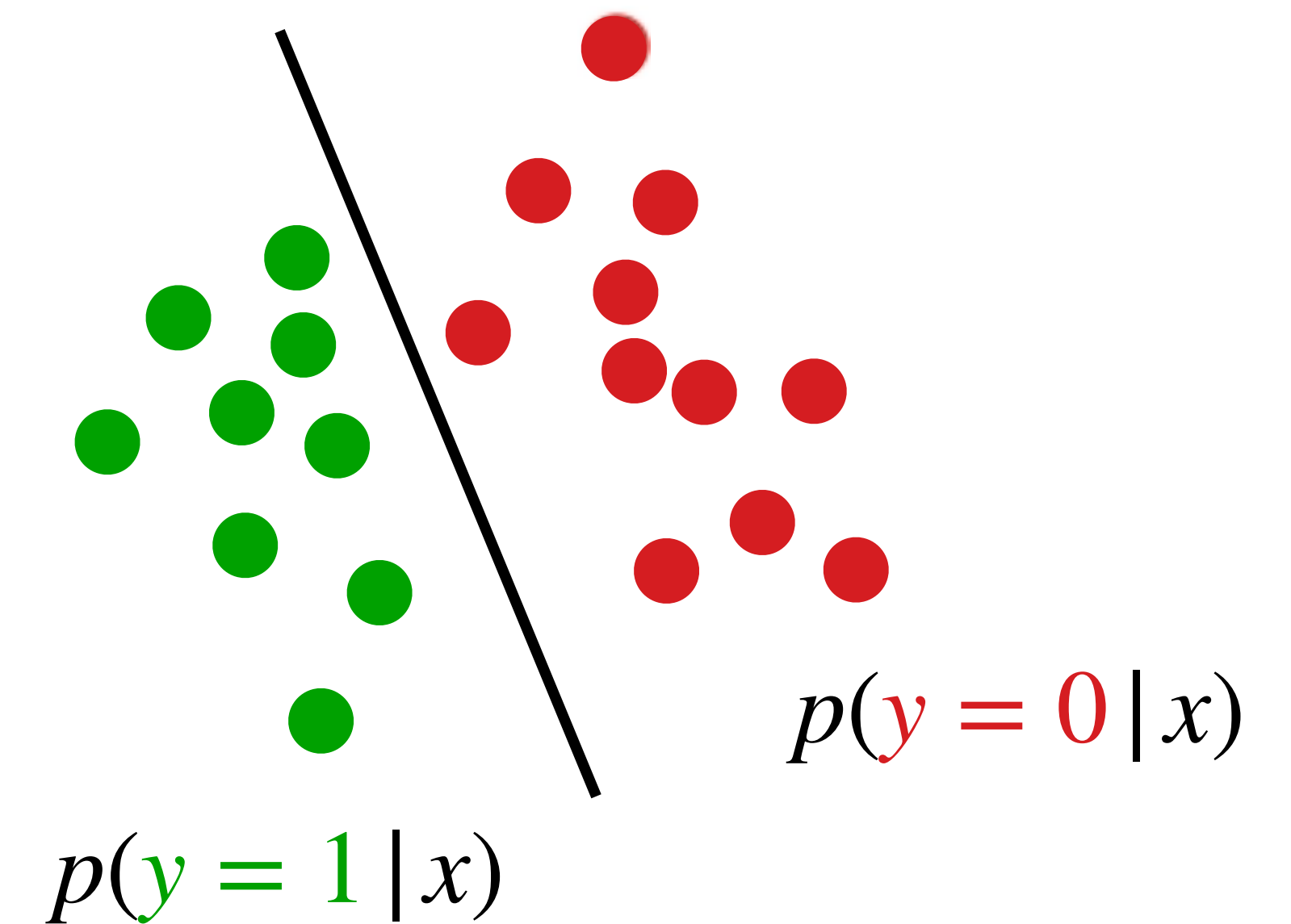


# Генеративные и дискриминативные модели

Генеративные

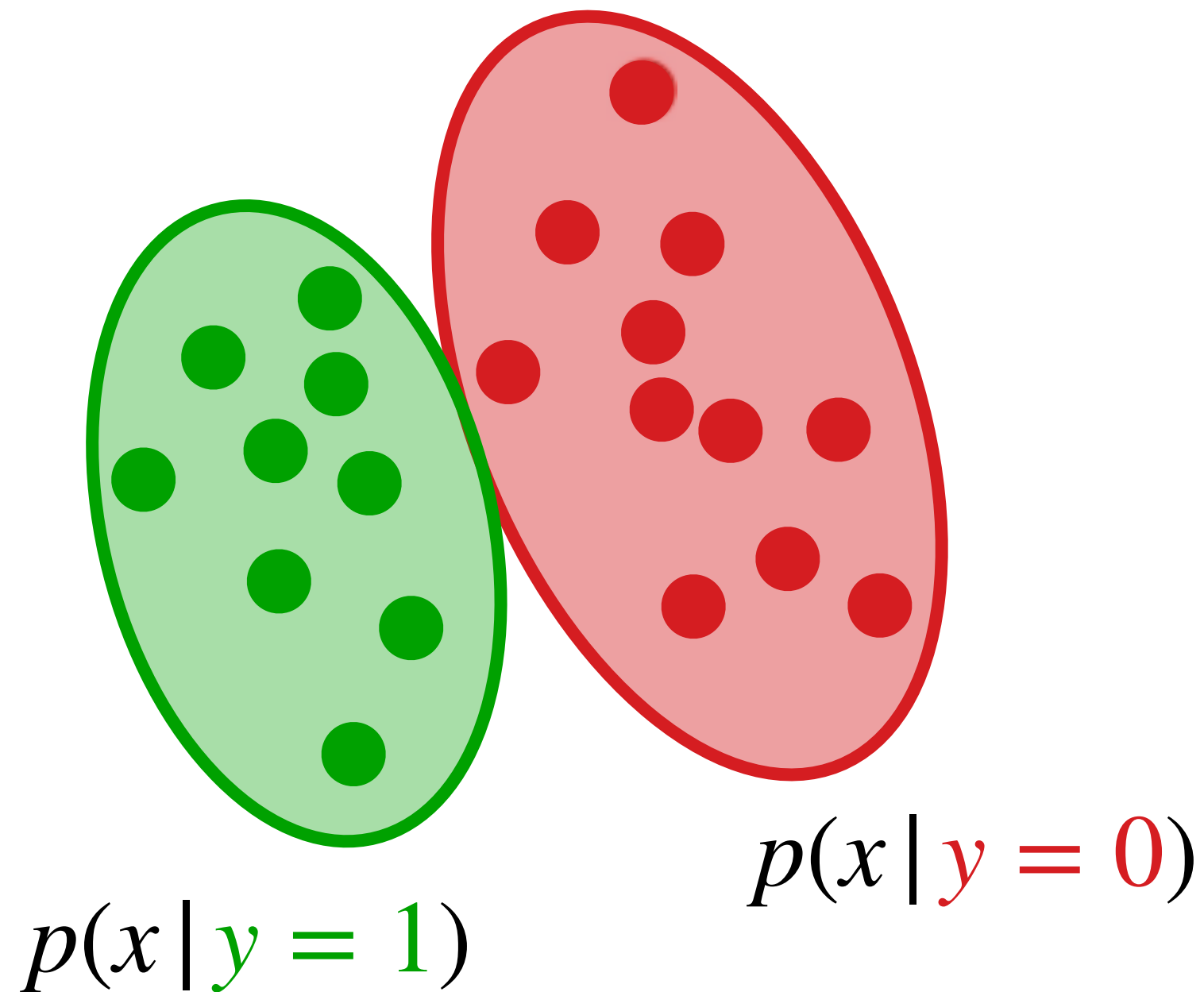


Дискриминативные



# Генеративные и дискриминативные модели

Генеративные

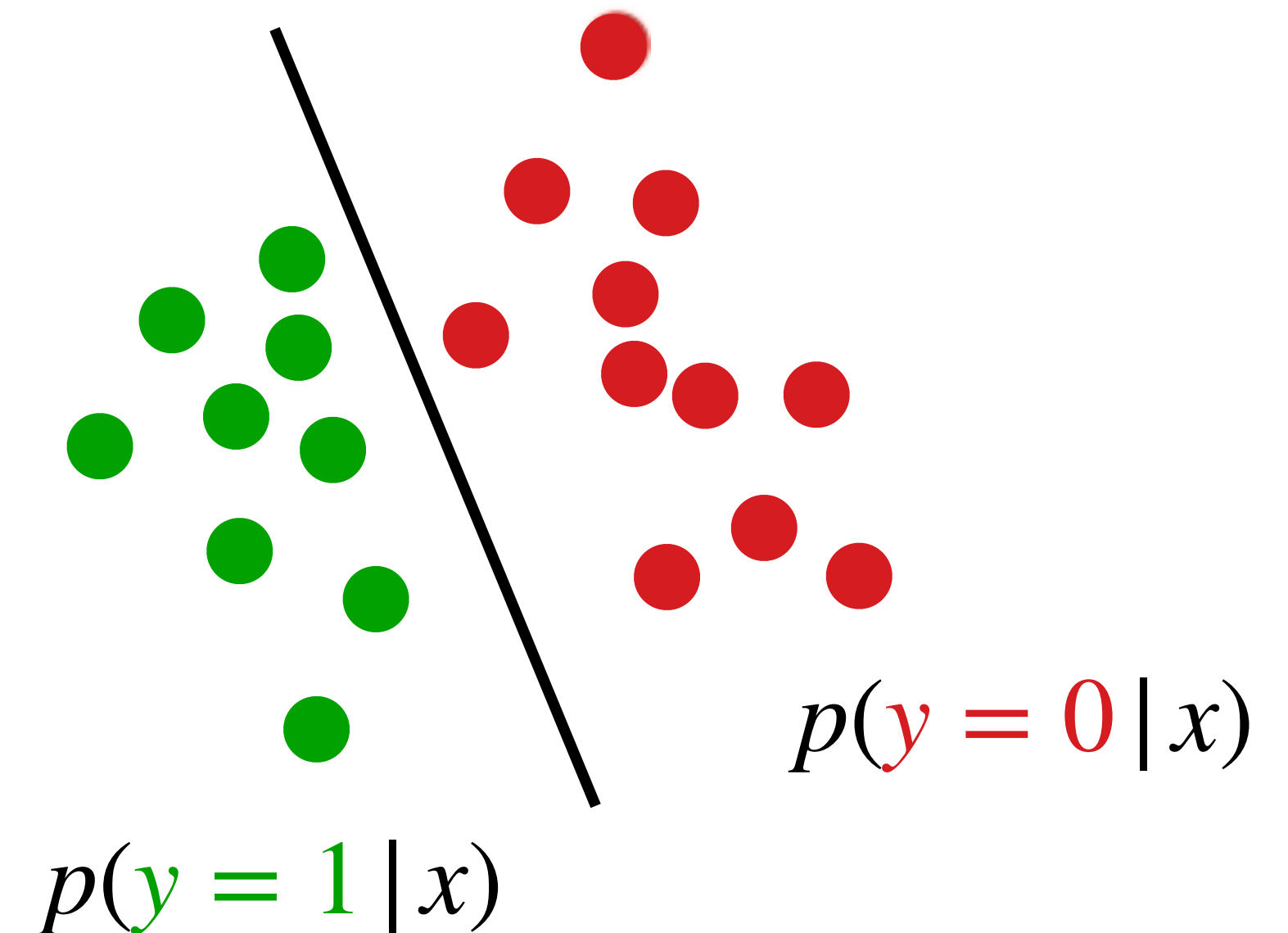


Обучаем :  $p(x | y = k)$

Предсказываем:

$$\hat{y} = \arg \max_y p(y, x) = \arg \max_y p(x | y)p(y)$$

Дискриминативные



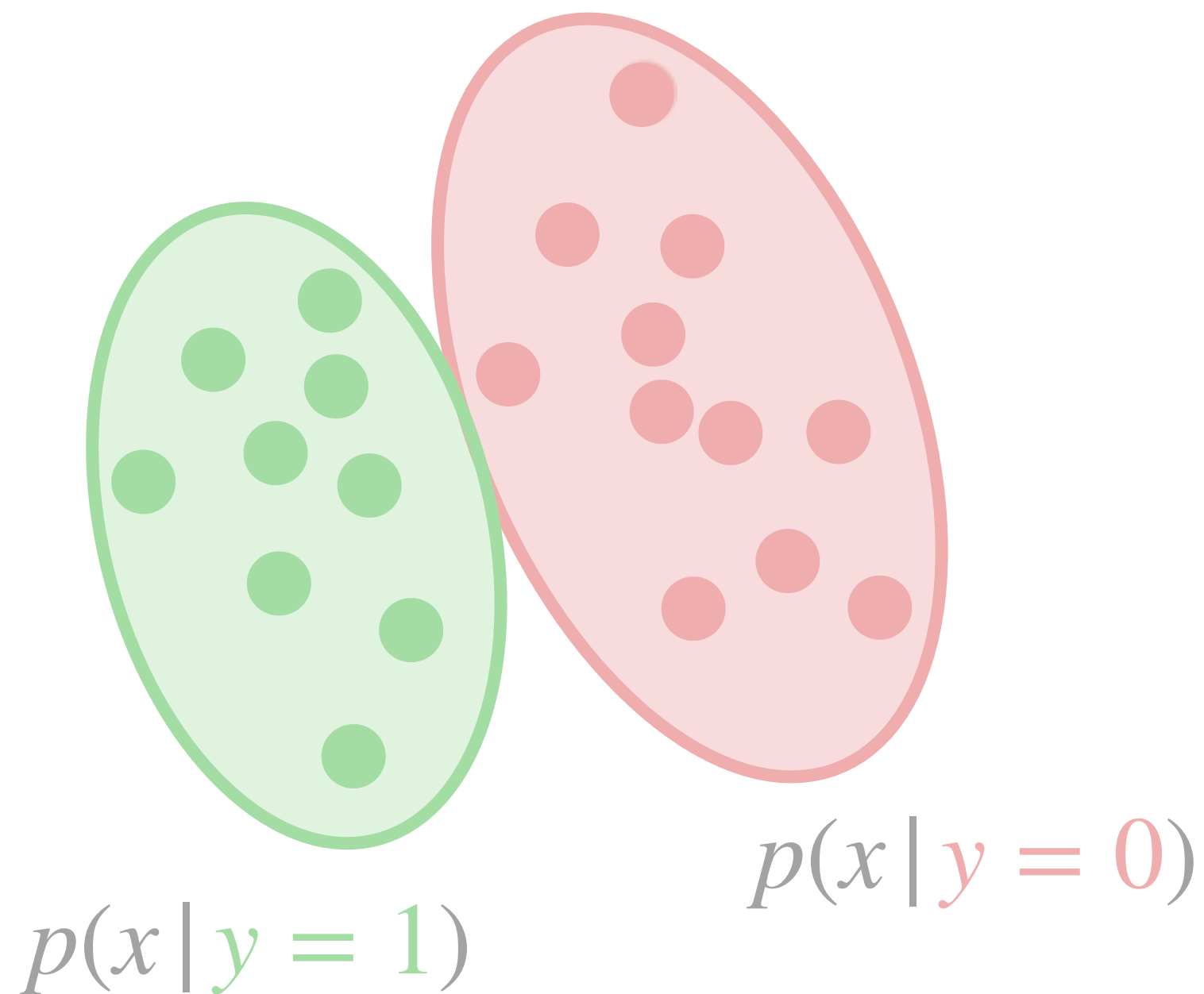
Обучаем :  $p(y = k | x)$

Предсказываем:

$$\hat{y} = \arg \max_y p(y | x)$$

# Генеративные и дискриминативные модели

Генеративные

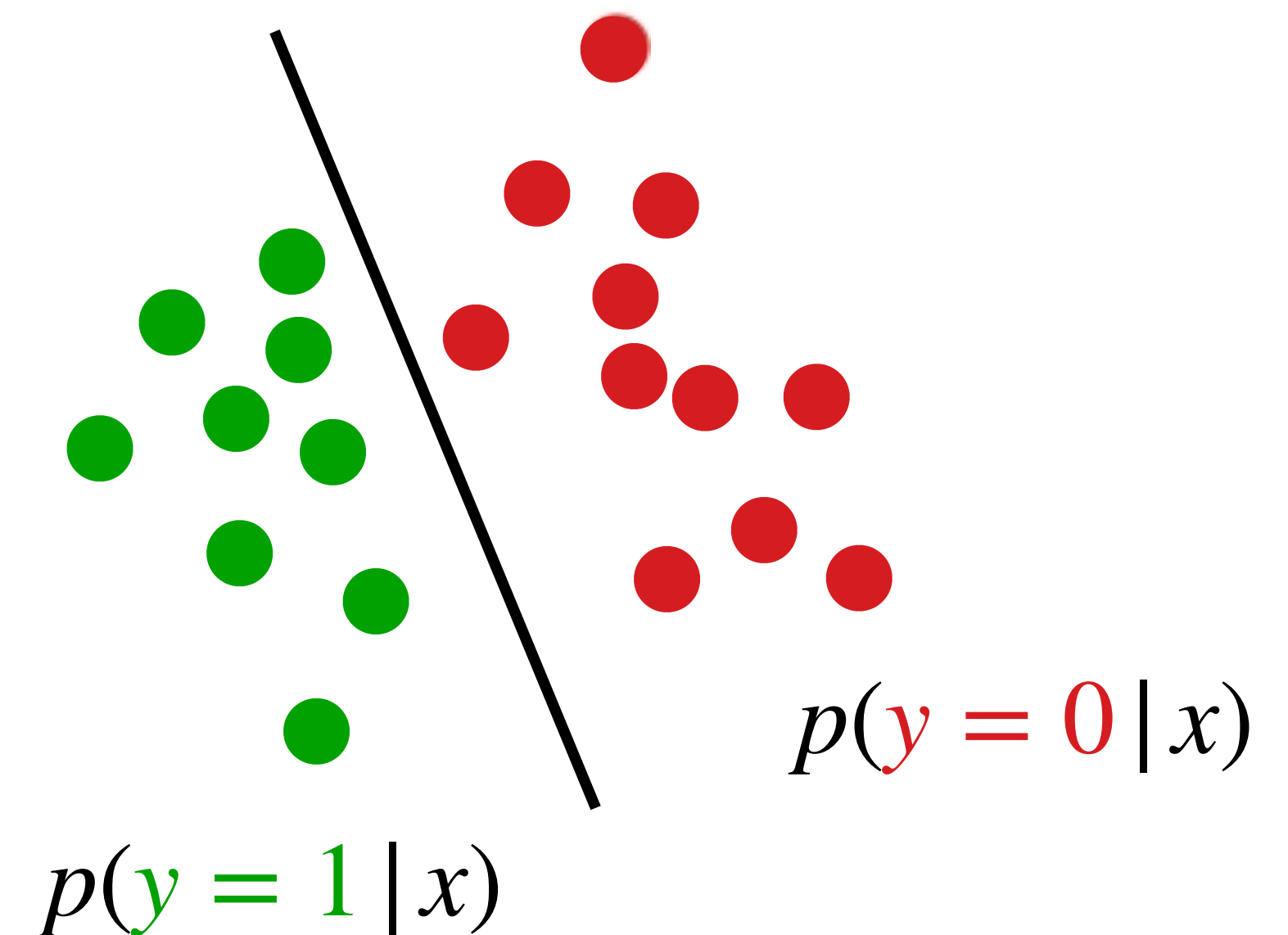


Обучаем :  $p(x | y = k)$

Предсказываем:

$$\hat{y} = \arg \max_y p(y, x) = \arg \max_y p(x | y)p(y)$$

Дискриминативные



Обучаем :  $p(y = k | x)$

Предсказываем:

$$\hat{y} = \arg \max_y p(y | x)$$

Почти все модели  
дискриминативные

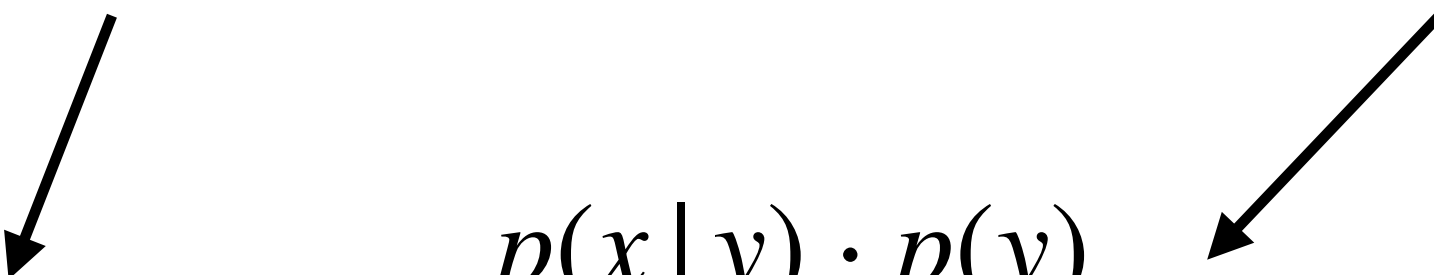
# Когда полезны генеративные модели?

- Когда в данных есть выбросы
- Когда распределение тестовой выборки отличается
- Когда данных мало и дискриминативная модель переобучается

# Наивный Байес

Теорема Байеса

$p(x)$  не зависит от  $y$

$$y = \arg \max_y p(y | x) = \arg \max_y \frac{p(x | y) \cdot p(y)}{p(x)} = \arg \max_y p(x | y)p(y)$$


Как найти  $p(x | y)$  и  $p(y)$ ?

# Как найти $p(x | y)$ и $p(y)$ ?

Посчитаем доли каждого класса в выборке

$$p(y = k) = \frac{1}{N} \sum_{i=1}^N [y_i = k]$$

Предполагаем, что:

- Порядок слов не важен
- Вероятность слова не зависит от соседей при заданном классе

$$p(x | y = k) = p(x_1, \dots, x_n | y = k) \approx \prod_{i=1}^n p(x_i | y = k)$$

# Почему это работает?

$$p(x|y) = \prod_{i=1}^n p(x_i|y)$$

Для несложных задач такое предположение не лишено смысла!

$$\begin{aligned} p(\text{очень вкусная еда} \mid y = -) \\ &= p(\text{очень} \mid y = -) \\ &\times p(\text{вкусная} \mid y = -) \\ &\times p(\text{еда} \mid y = -) \end{aligned}$$

$$\begin{aligned} p(\text{очень вкусная еда} \mid y = +) \\ &= p(\text{очень} \mid y = +) \\ &\times p(\text{вкусная} \mid y = +) \\ &\times p(\text{еда} \mid y = +) \end{aligned}$$



# Почему это работает?

$$p(x|y) = \prod_{i=1}^n p(x_i|y)$$

Для несложных задач такое предположение не лишено смысла!

$$\begin{aligned} & p(\text{очень вкусная еда} \mid y = -) \\ &= p(\text{очень} \mid y = -) \\ &\times \frac{p(\text{вкусная} \mid y = -)}{p(\text{еда} \mid y = -)} \end{aligned}$$

<

$$\begin{aligned} & p(\text{очень вкусная еда} \mid y = +) \\ &= p(\text{очень} \mid y = +) \\ &\times \frac{p(\text{вкусная} \mid y = +)}{p(\text{еда} \mid y = +)} \end{aligned}$$

Ключевые слова

$$p(\text{вкусная} \mid y = -) < p(\text{вкусная} \mid y = +)$$

# Как оценить $p(x_i | y)$ ?

Сколько раз слово  $x_i$  встречалось  
в текстах с меткой  $k$

$$p(x_i | y = k) = \frac{N(x_i, y = k)}{\sum_{j=1}^{|V|} N(x_j, y = k)}$$

Что если  $N(x_i, y = k) = 0$ ?

# Как оценить $p(x_i | y)$ ?

Сколько раз слово  $x_i$  встречалось  
в текстах с меткой  $k$

$$p(x_i | y = k) = \frac{N(x_i, y = k)}{\sum_{j=1}^{|V|} N(x_j, y = k)}$$

Что если  $N(x_i, y = k) = 0$ ?

$$\begin{aligned} & p(\text{самый вкусный Bratwurst} \mid y = +) \\ &= p(\text{самый} \mid y = +) \\ &\times p(\text{вкусный} \mid y = +) \\ &\times \underline{p(\text{Bratwurst} \mid y = +)} = 0 \\ &= 0 \end{aligned}$$

# Сглаживание Лапласа

$$p(x_i | y = k) = \frac{N(x_i, y = k) + \delta}{\sum_{j=1}^{|V|} N(x_j, y = k) + |V| \cdot \delta} \quad \delta \in [0,1]$$

Если  $\delta = 1$ , то сглаживание называется сглаживанием Лапласа

# Как предсказывать?

$$\hat{y} = \operatorname{argmax}_y p(x|y) \cdot p(y)$$

$x$  = очень вкусная еда

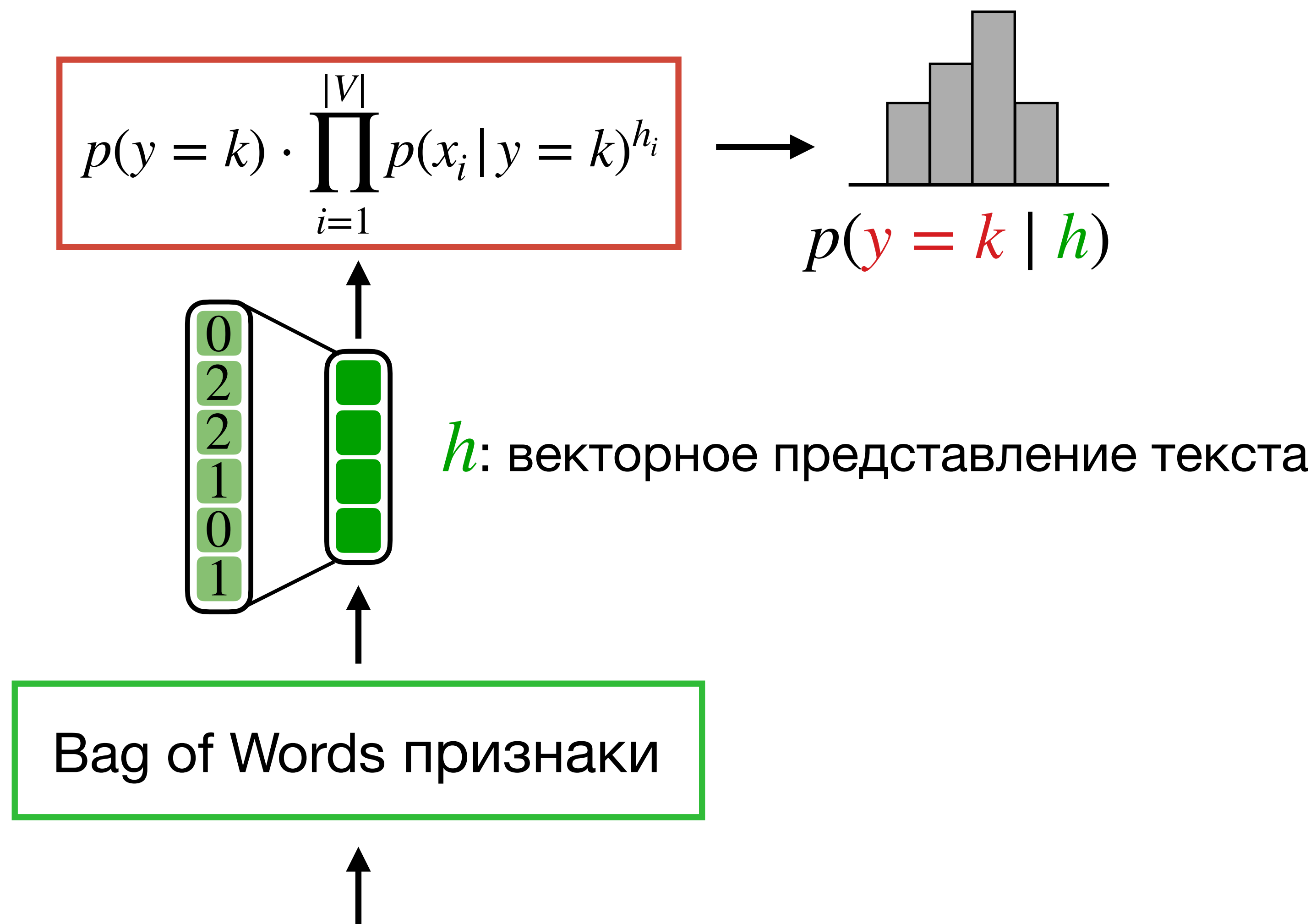
$$\begin{aligned} & p(\text{очень вкусная еда} \mid y = -) p(y = -) \\ &= p(\text{очень} \mid y = -) \\ &\times p(\text{вкусная} \mid y = -) \\ &\times p(\text{еда} \mid y = -) \\ &\times p(y = -) \end{aligned}$$

<

$$\begin{aligned} & p(\text{очень вкусная еда} \mid y = +) p(y = +) \\ &= p(\text{очень} \mid y = +) \\ &\times p(\text{вкусная} \mid y = +) \\ &\times p(\text{еда} \mid y = +) \\ &\times p(y = +) \end{aligned}$$

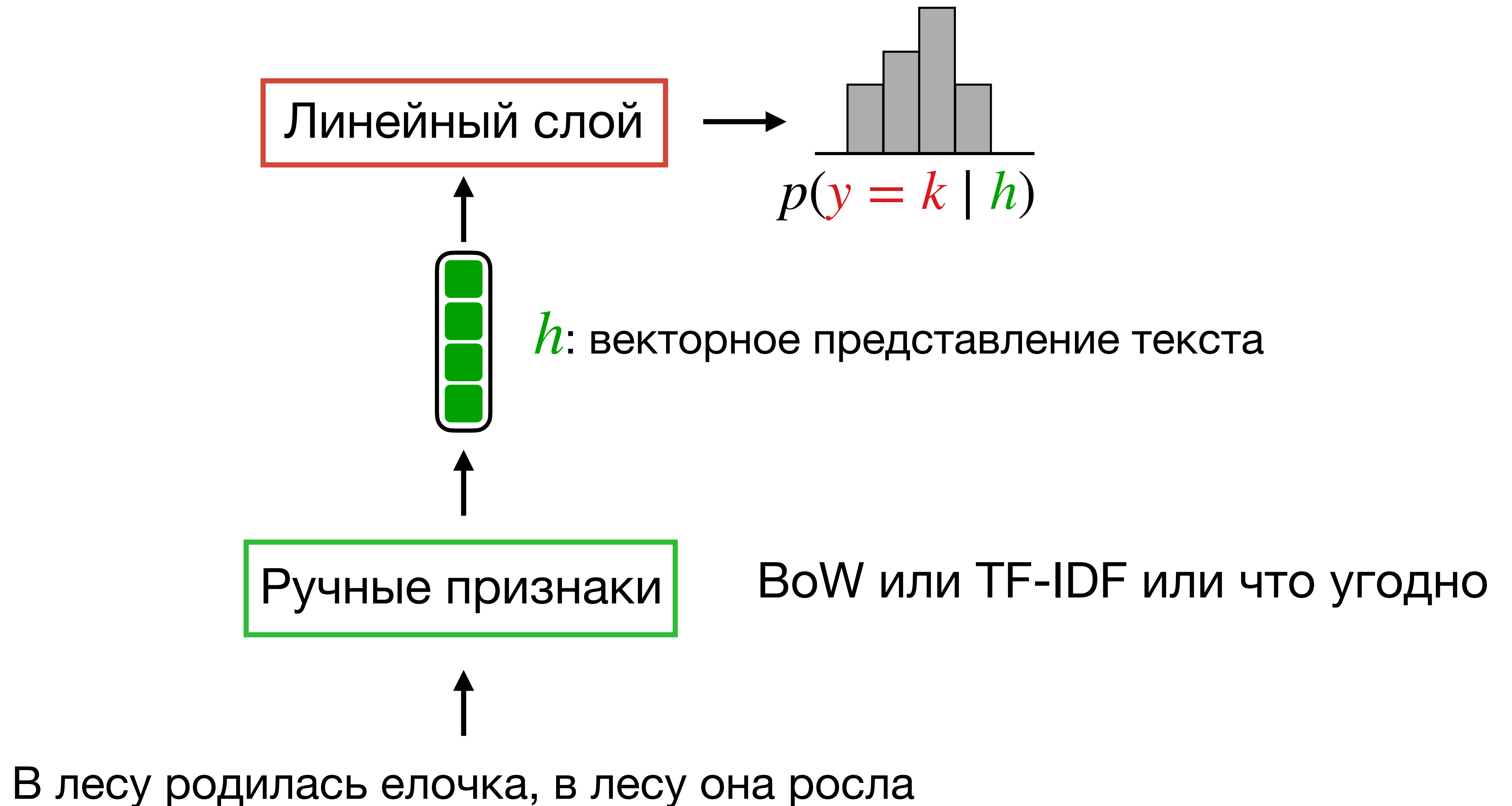
Если  $p(y = -) \approx p(y = +)$

# Наивный Байес

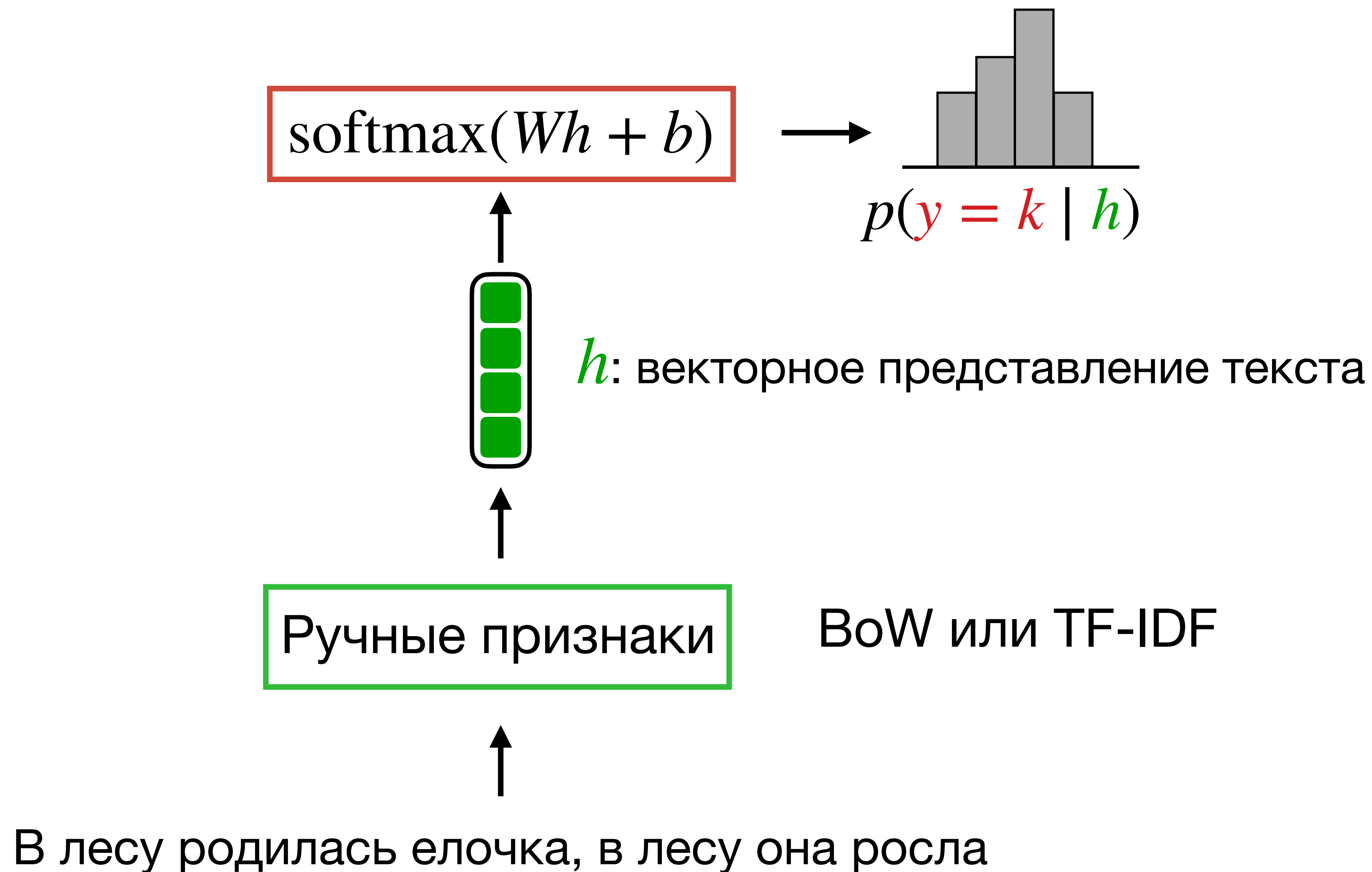


В лесу родилась елочка, в лесу она росла

# Логистическая регрессия



# Логистическая регрессия





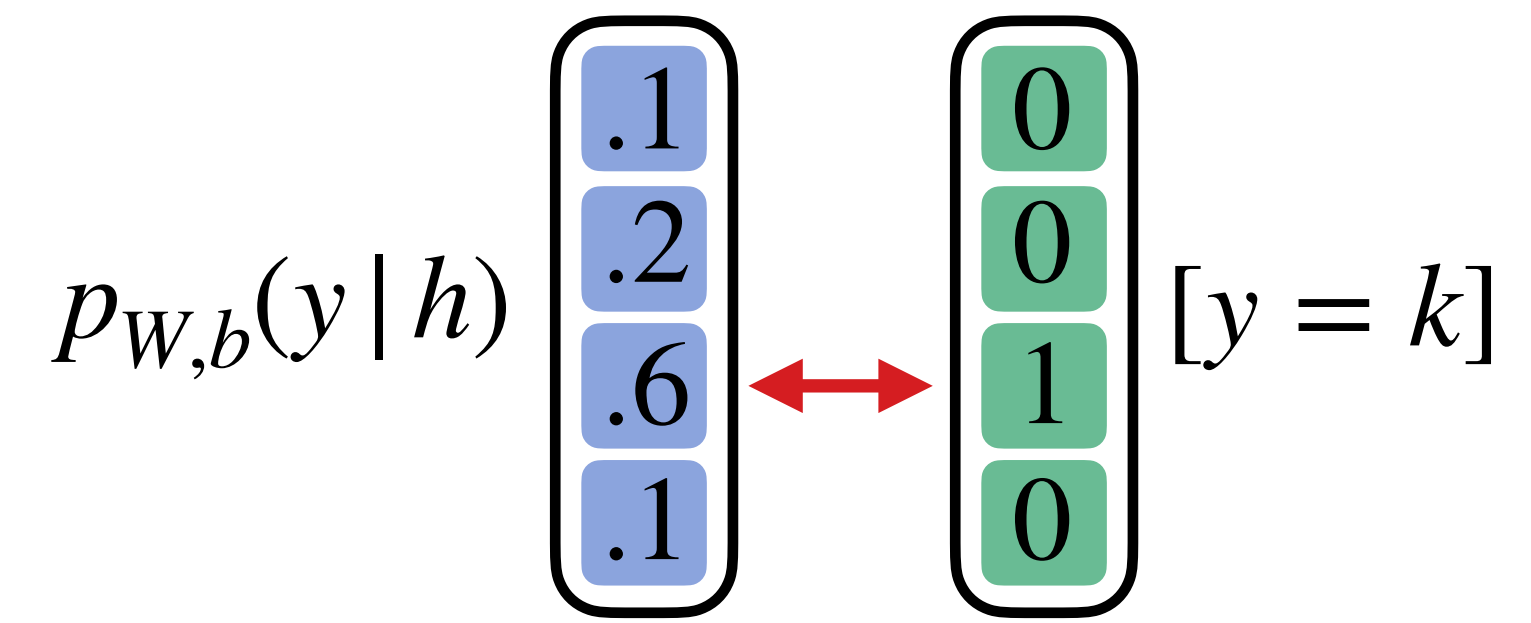
# Как обучать?

Максимизируем правдоподобие правильного класса

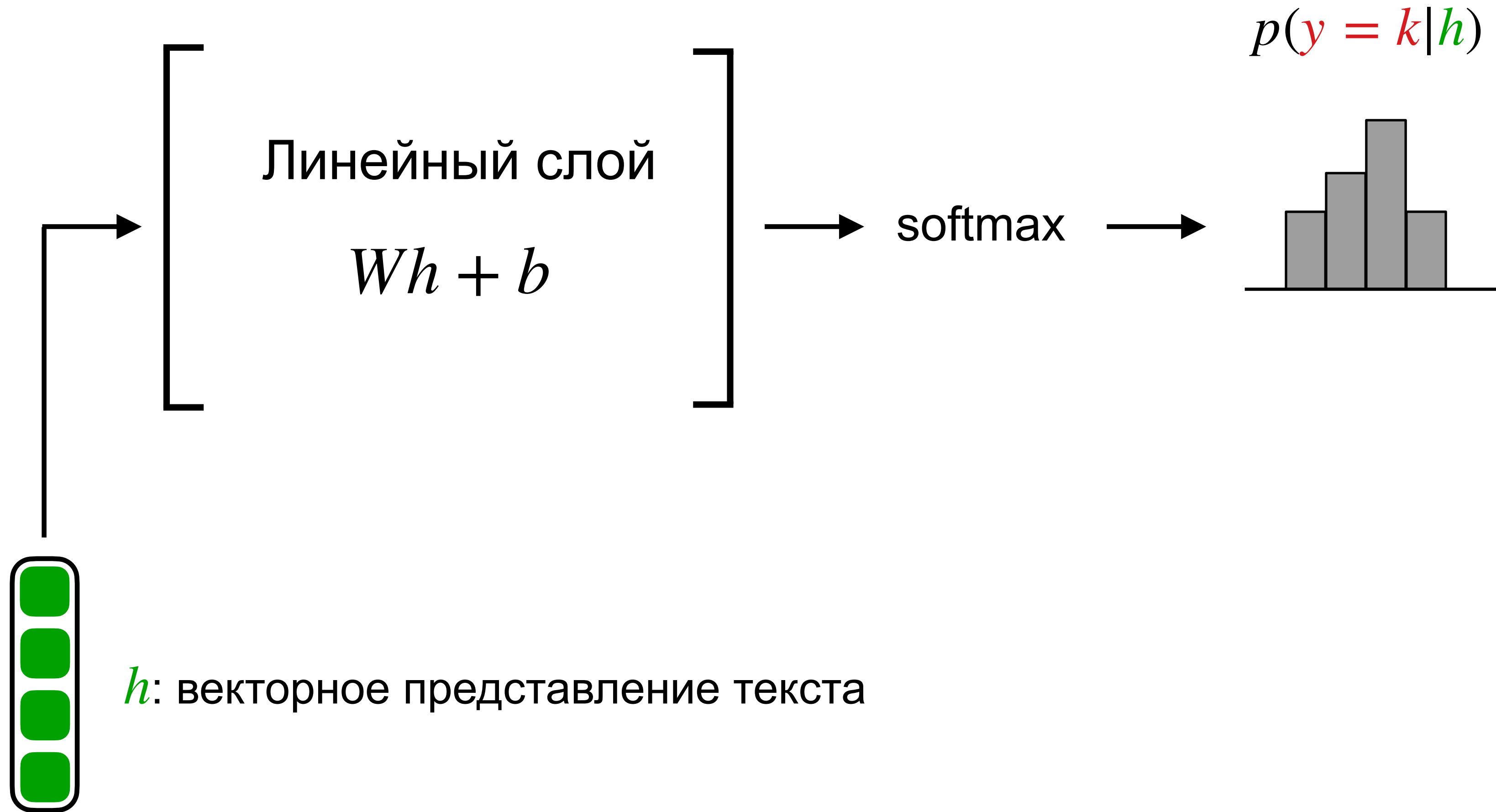
$$p_{w,b}(y | h) = \prod_{k=1}^K p_{w,b}(y = k | h)^{[y=k]}$$

Накладываем логарифм и отрицание

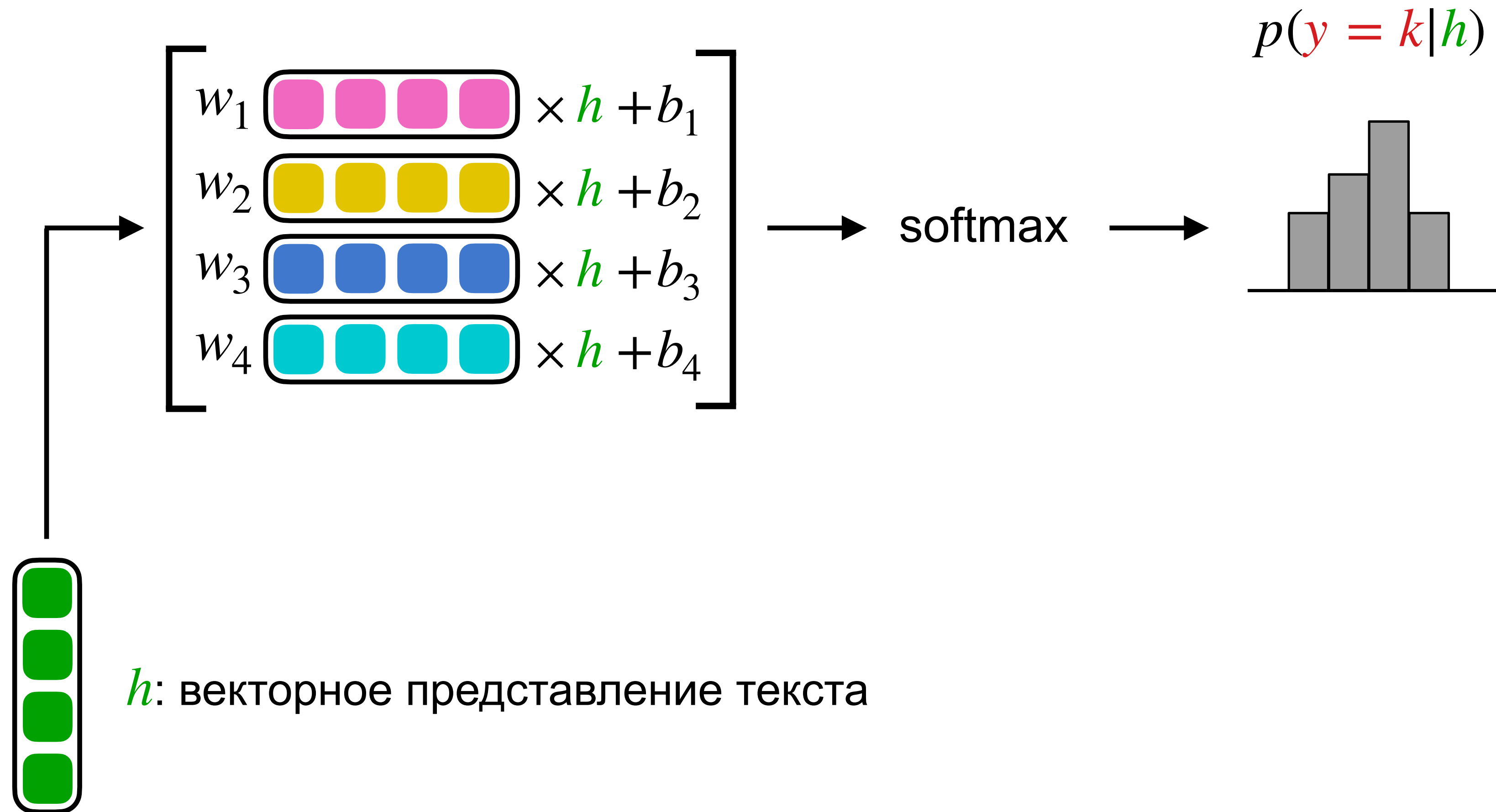
$$L(W, b) = - \sum_{i=1}^N \sum_{k=1}^K [y = k] \cdot \log p_{\theta}(y = k | h) \rightarrow \min_{W, b}$$



# Линейный слой



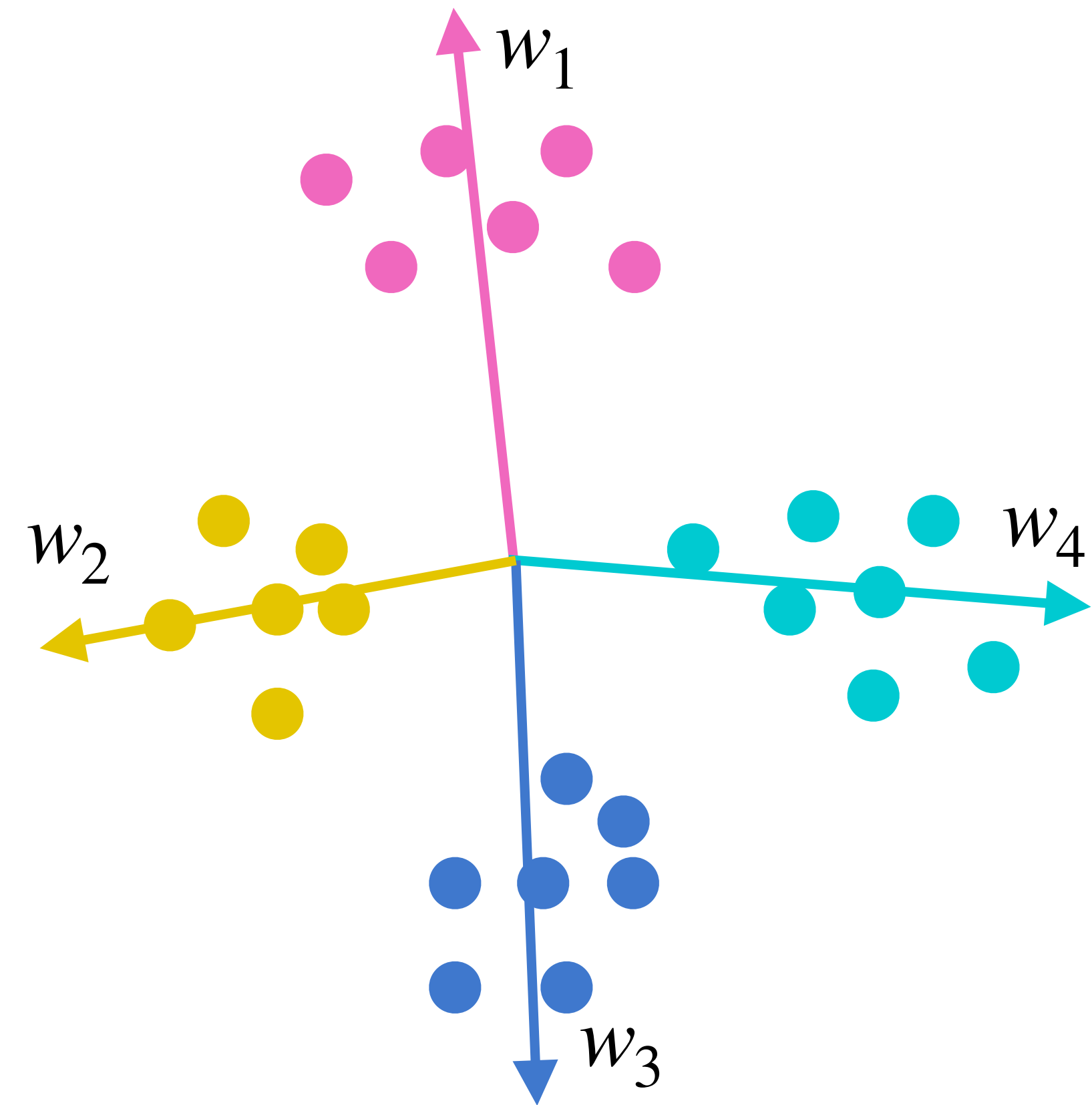
# Линейный слой



# Линейный слой

Векторы линейного слоя для каждого класса должны коррелировать с векторными представлениями элементов класса.

Скалярное произведение векторов **максимально**, когда они **сонаправлены**.



# Минусы подходов

- Не учитывают связь между словами
- Не учитывают порядок слов

$p(y = + \mid \text{это не хорошо, совсем плохо})$

||

$p(y = + \mid \text{это хорошо, совсем не плохо})$

- Признаки извлекаются вручную

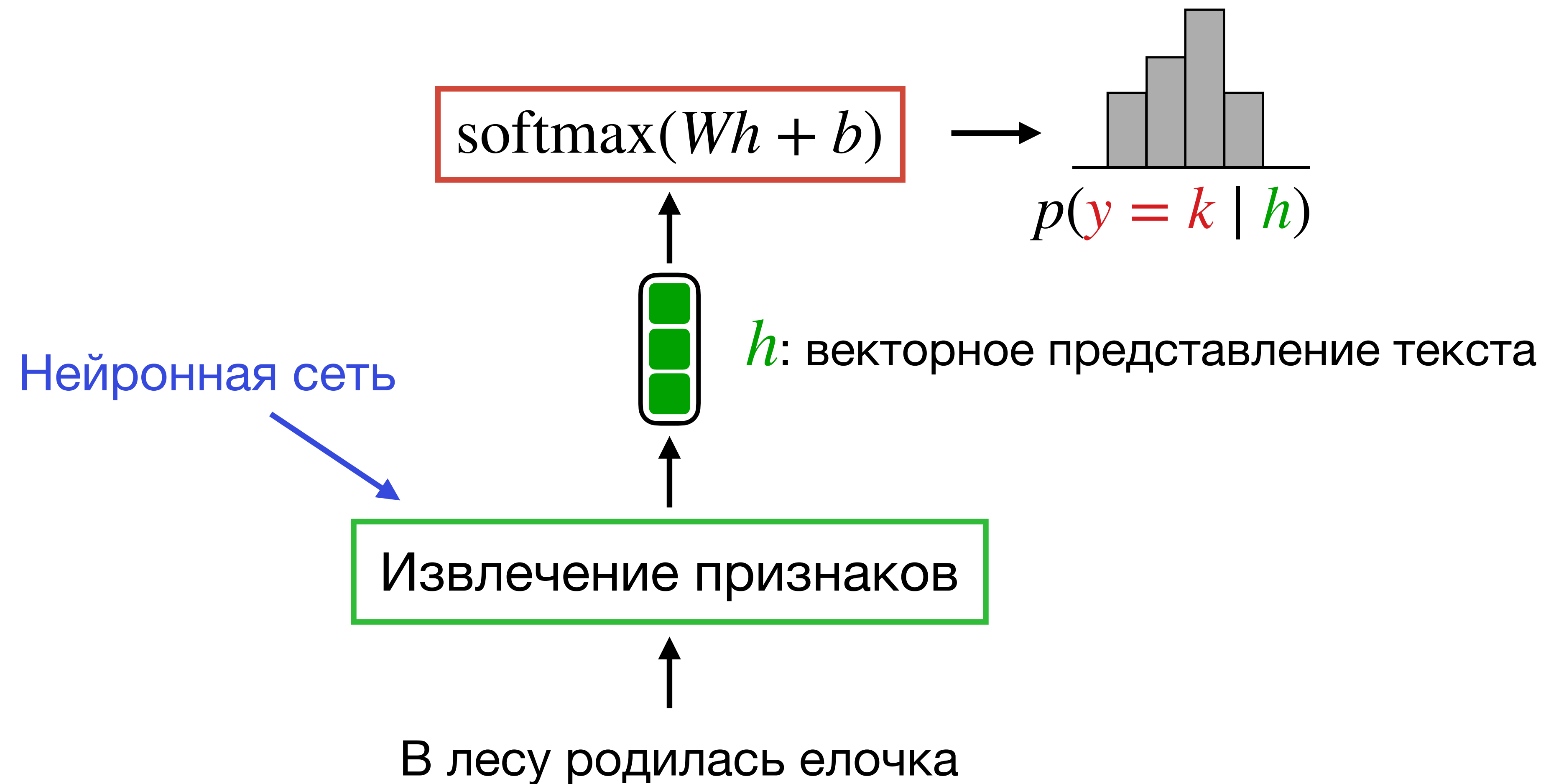
# Плюсы подходов

- Достаточно хорошо справляются с несложными задачами
- Скорость работы
- Время обучения
- Интерпретируемость

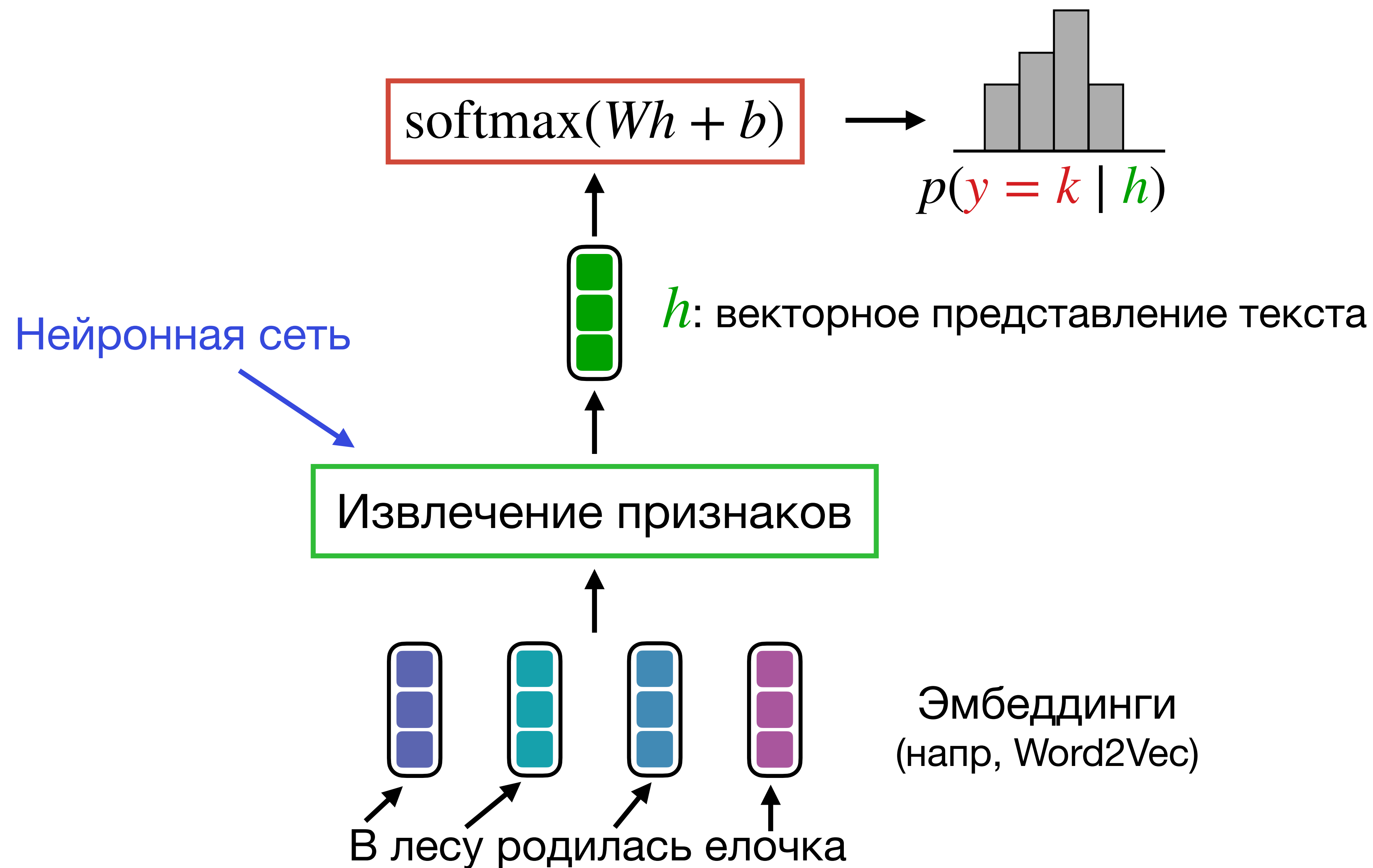
Интерпретируемость очень важна, когда цена ошибки велика

- Постановка медицинского диагноза
- Вынесение приговора в суде

# Нейросетевые модели



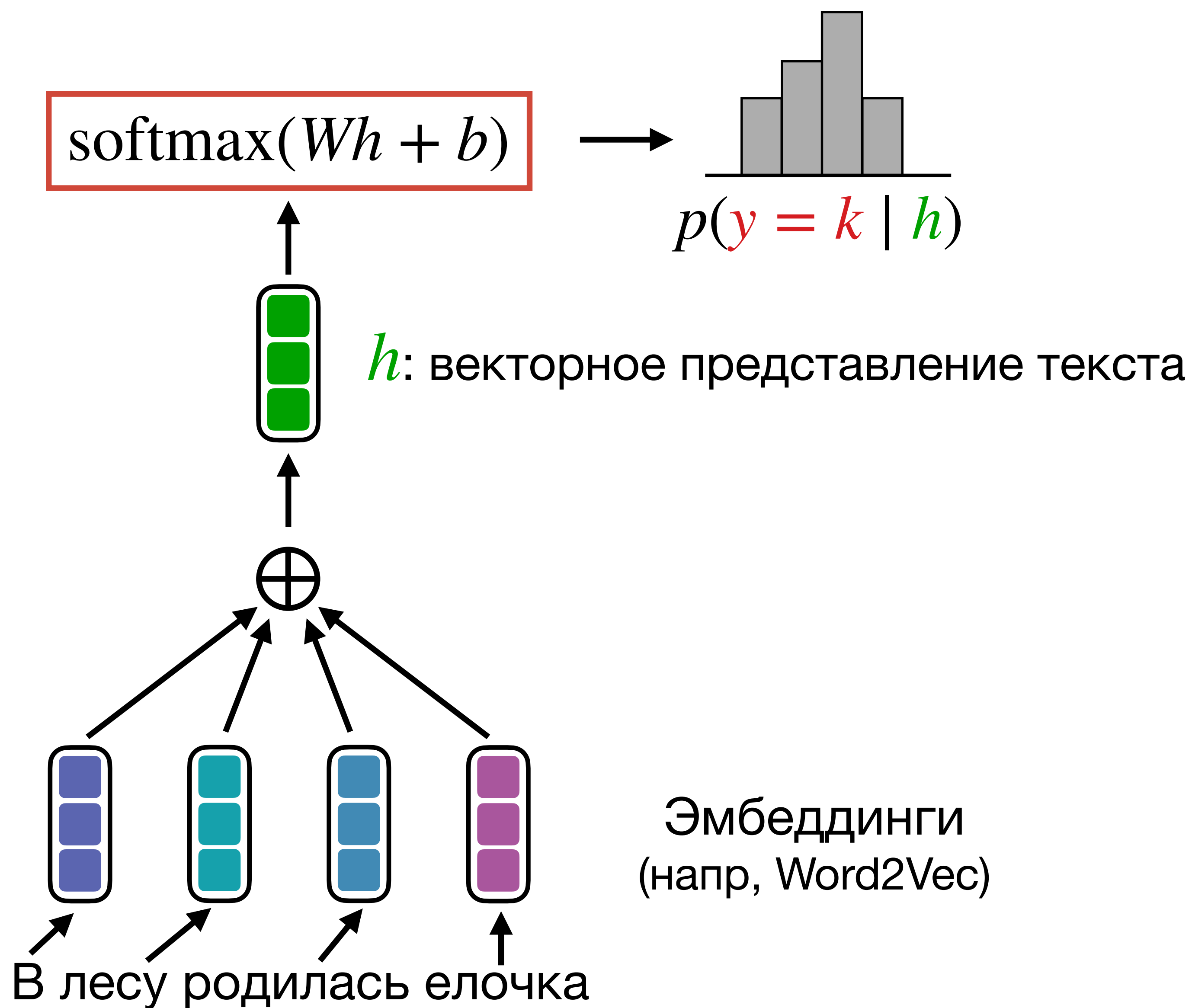
# Как извлекать признаки?





# Bag of Embeddings

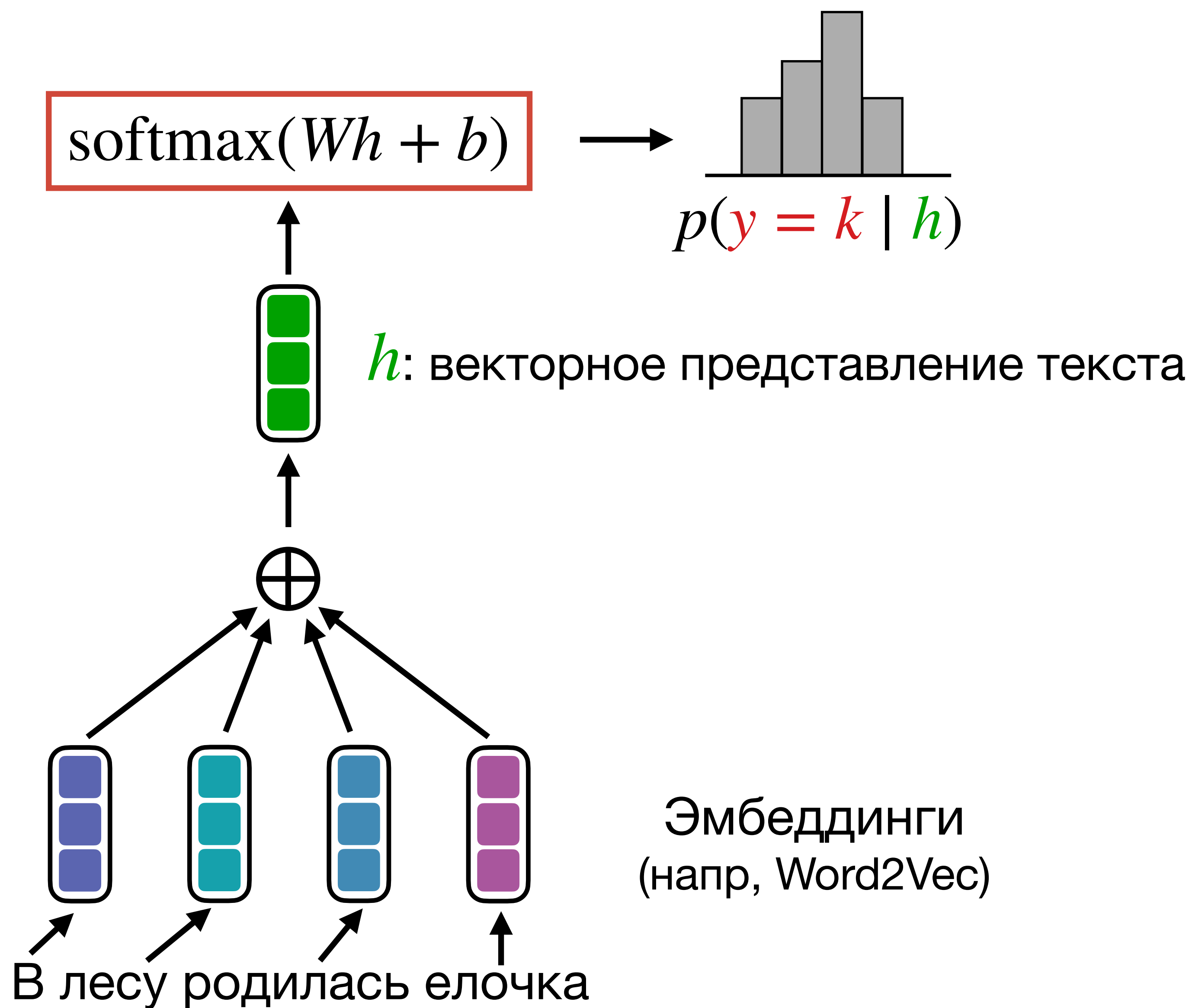
Представляем текст в виде суммы эмбеддингов



# Bag of Embeddings

Представляем текст в виде суммы эмбеддингов

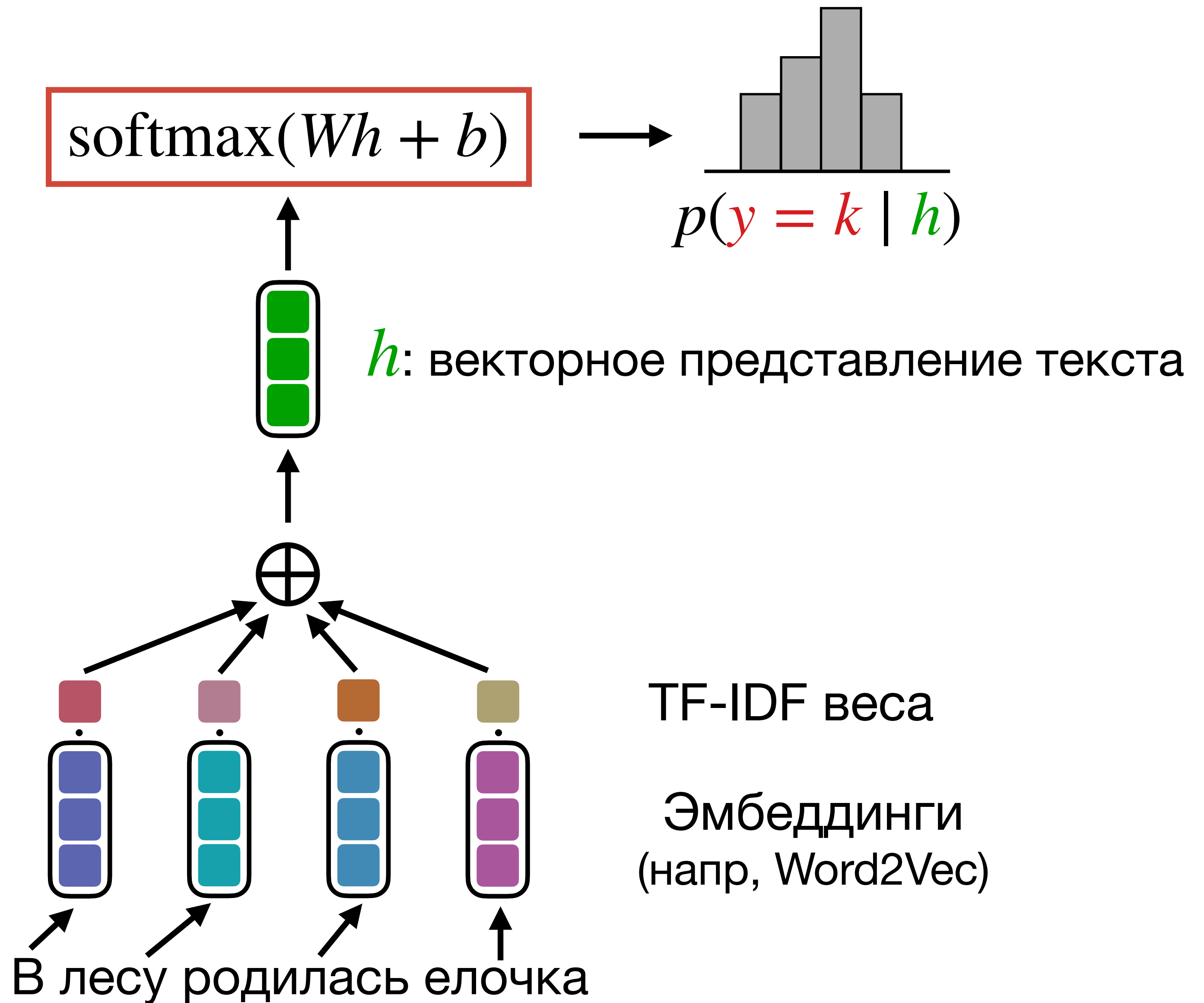
- + Очень легко реализовать
- Не учитываем связь между словами
- Нейтральные слова могут перетянуть вес на себя



# Weighted Bag of Embeddings

- Домножаем эмбединги на веса TF-IDF
- После этого складываем

- + Все еще легко реализовать
- + У менее важных слов будет меньший вес
- Не учитываем связь между словами



# Самое важное

- Классификация текста/слов – самая популярная задача
- Очень часто простые методы хорошо справляются
- Качество напрямую зависит от того, насколько хорошие признаки удалось извлечь