Математика полносвязной сети: геометрия, интерпретируемость, SLAP атака

Перминов Андрей

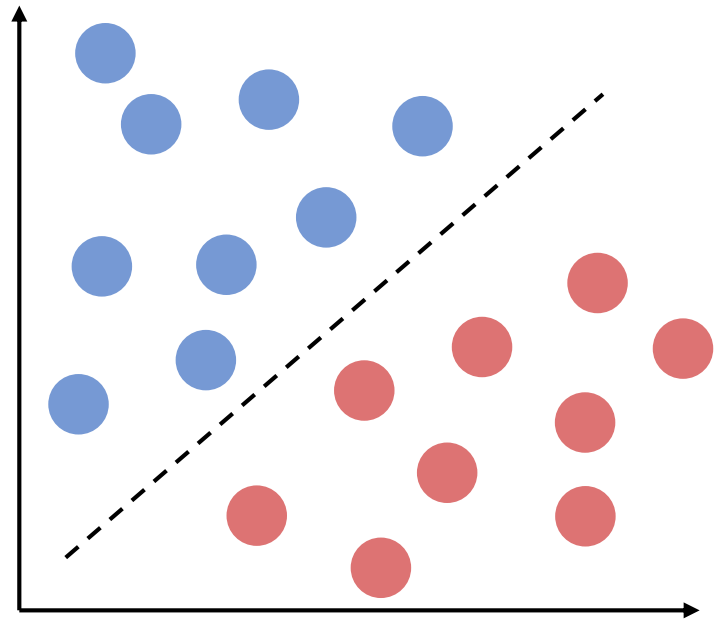
01 октября 2025

ИСП РАН

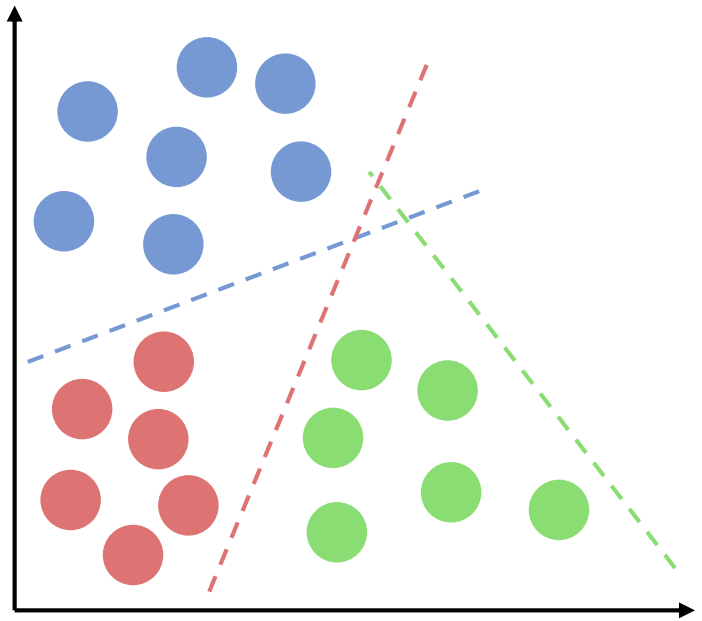
Задача классификации

Классификация – задача, в которой множество **объектов** необходимо **разделить** некоторым образом на **классы**.

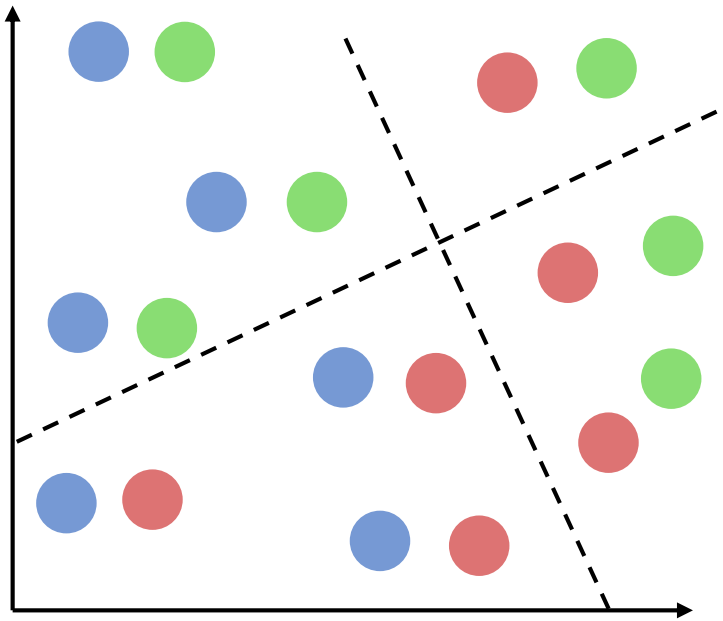
Выборка – конечное множество объектов, для которых известно, к каким классам они относятся.



бинарная



многоклассовая

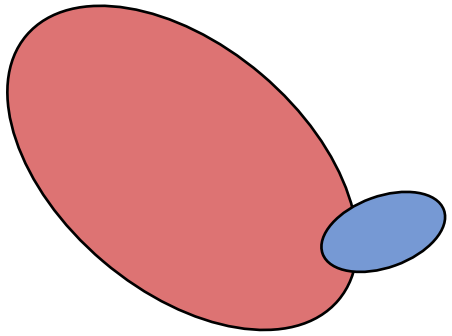


многометочная

Основные вызовы

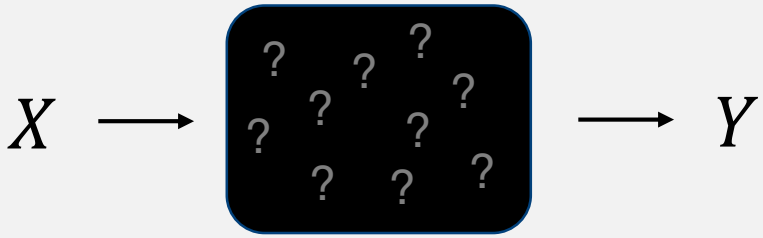
Дисбаланс классов

- Существенное **различие** априорных вероятностей.
- Деградация точности при классификации **маломощных** классов.
- **Критично** в медицине, финансах, промышленности (редкие события/дефекты).



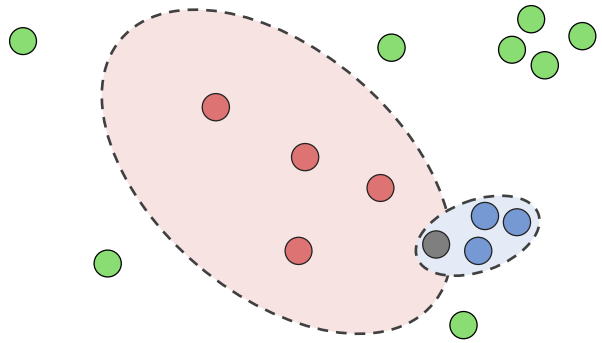
Интерпретируемость и доверие к модели

- Большинство моделей работают как **чёрный ящик**.
- Снижение **доверия** к модели из-за невозможности анализа принимаемых решений.
- В отличие от деревьев решений, нейросетевые модели **являются плохо интерпретируемыми**.



Неопределённости в данных

- Несоответствие наблюдений **носителю** распределения и **двусмысленность** в данных.
- Отсутствие **статистической** основы для классификации.
- Необходимость внедрения **отказа** от классификации.



Данные для классификации персептроном

Данные, заданные в виде матрицы $n \times (d + 1)$

$X \in R^d$ – **признаки** (первые d столбцов);

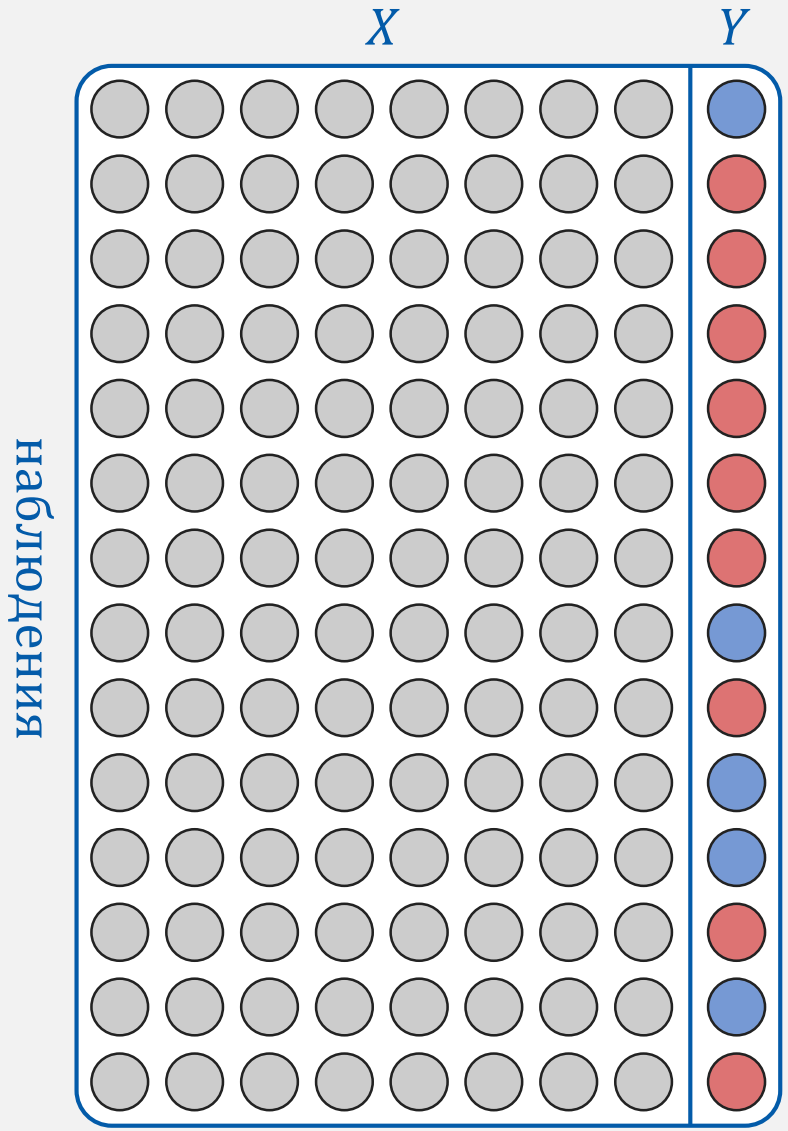
$Y \in \{1, \dots, C\}$ – **метки классов**;

$D = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ – **набор данных**.

d – размерность **входного** пространства;

n – количество **наблюдений** в наборе;

C – количество **классов**.

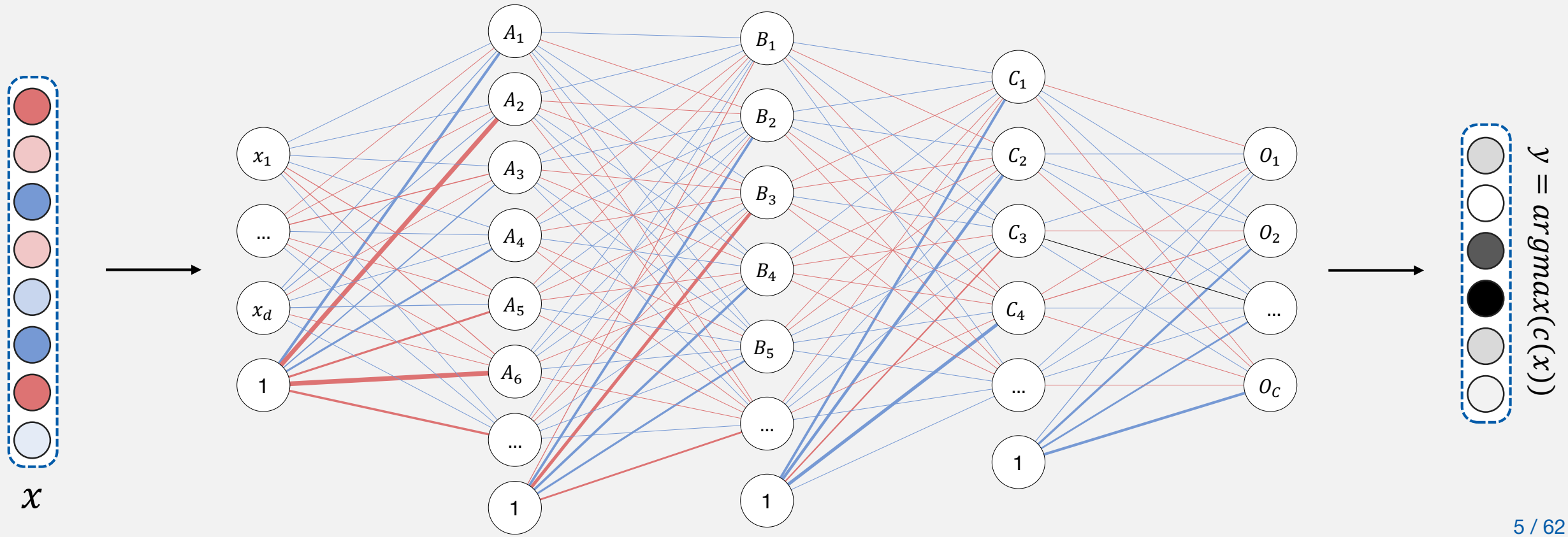


Многослойный персептрон

Равномерно непрерывная функция $c(x)\colon R^d \to R^C$ вида:

$$c(x) = \sigma(W_O\sigma(\dots\sigma(W_1x + b_1)\dots) + b_O),$$

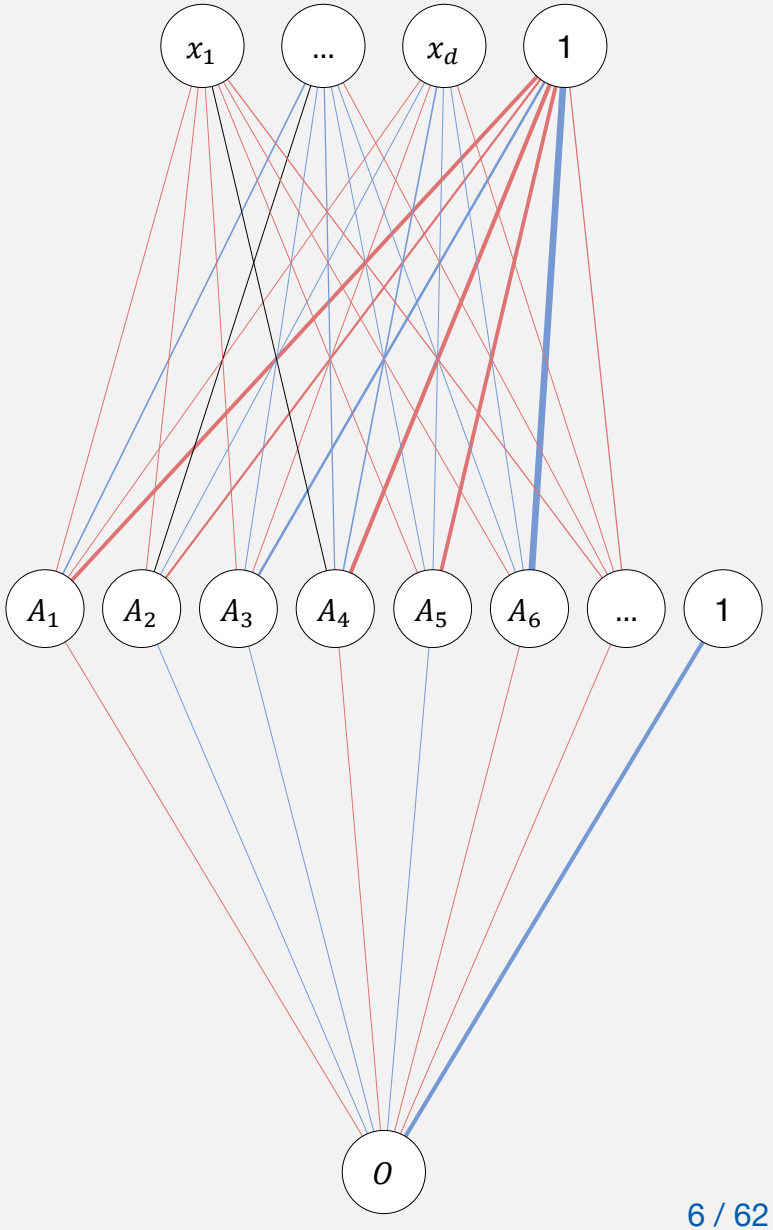
где σ – **нелинейная** функция активации.



Основная аппроксимационная теорема

Для любой **непрерывной** функции $f: [a, b]^d \to R$ и любого $\varepsilon > 0$ существует такая полносвязная нейронная сеть $c(x)$ с **одним скрытым слоем** и нелинейной функцией активации σ, что

$$\sup_{x \in [a, b]^d} |f(x) - c(x)| < \varepsilon.$$

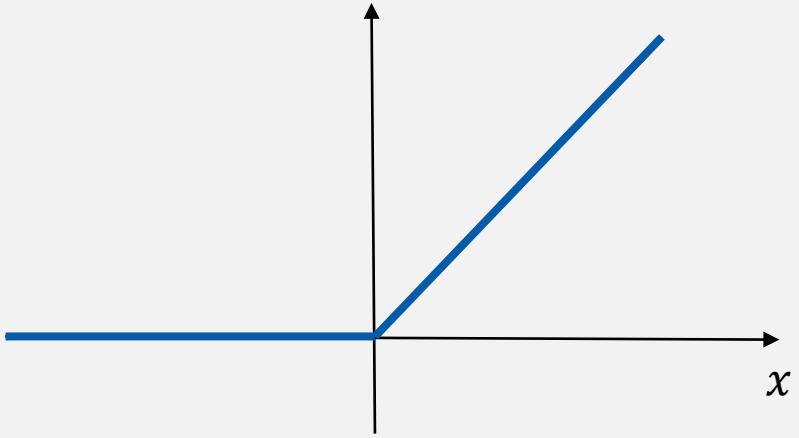


Следует ли из этой теоремы существование персептрона с более чем одним скрытым слоем?

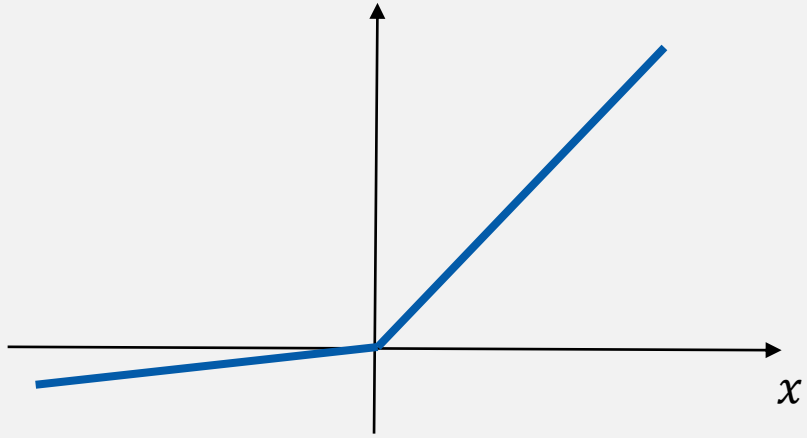
Кусочно-линейные функции активации

Кусочно-линейная функция – функция, линейная на каждом из интервалов, составляющих область определения.

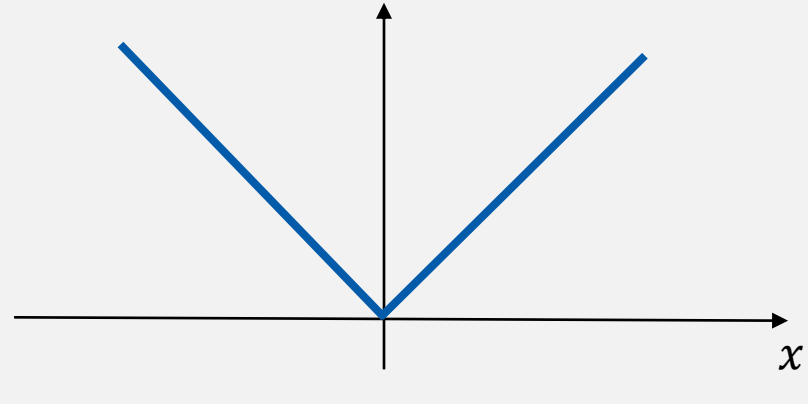
Активно используются в нейросетевых моделях благодаря **быстрому вычислению** значений.



$\boldsymbol{ReLU}(x)$:
$\max(0, x)$



$\boldsymbol{LeakyReLU}(x)$:
$\max(0.1 \cdot x, x)$



$\boldsymbol{Abs}(x)$:
$|x|$

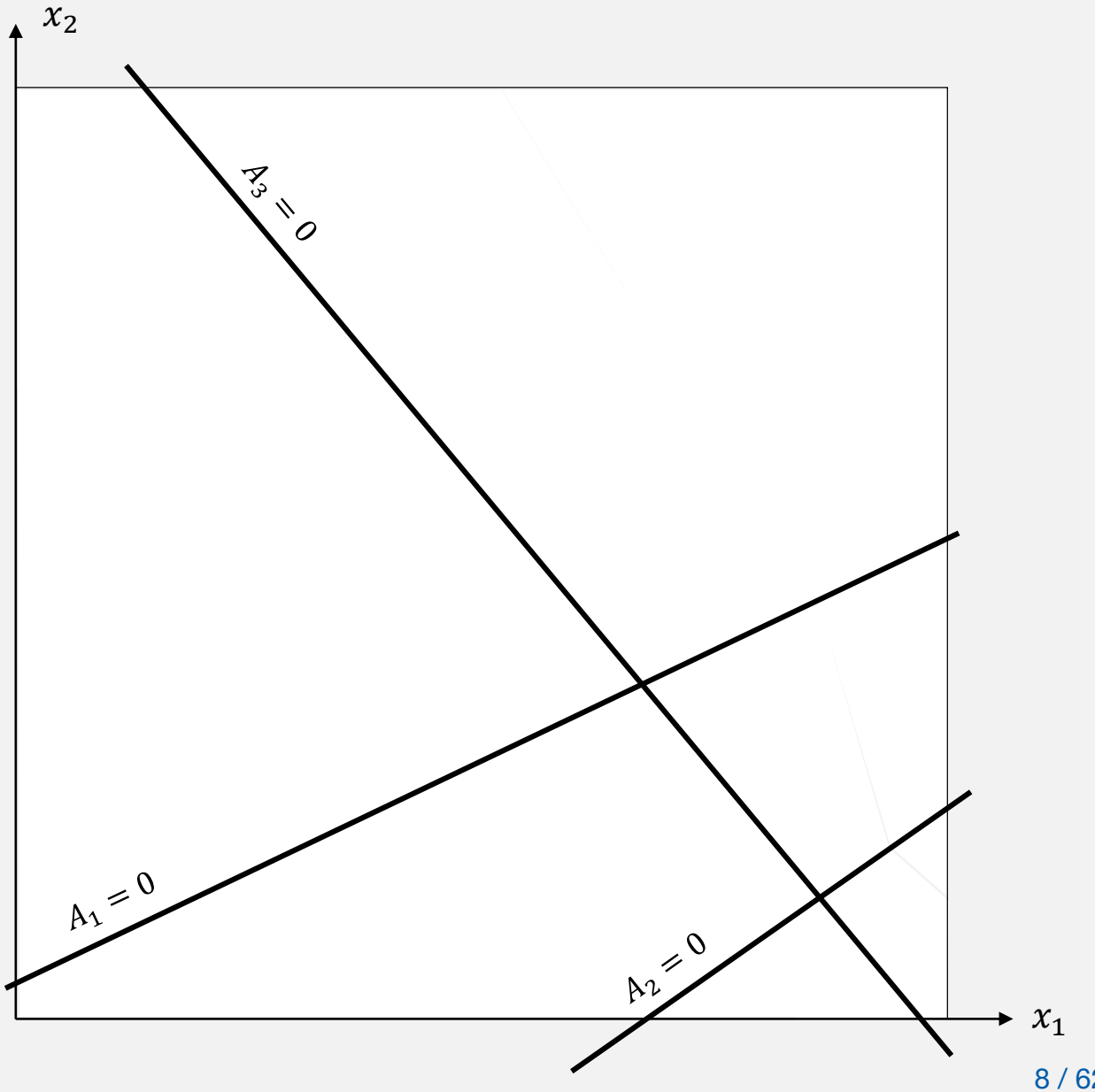
Геометрическая интерпретация: первый слой

$$A_1 = -0.4x_1 + 0.8x_2$$
$$A_2 = -1.2x_1 + 1.8x_2 + 0.8$$
$$A_3 = -1.5x_1 - 1.3x_2 + 1.5$$

$$B_1 = 0.6|A_1| - 0.5|A_2| + 2.1|A_3| - 0.5$$

Первый слой персептрона разбивает входной компакт на непересекающиеся ячейки.

Как определить конкретную ячейку?



Геометрическая интерпретация: первый слой

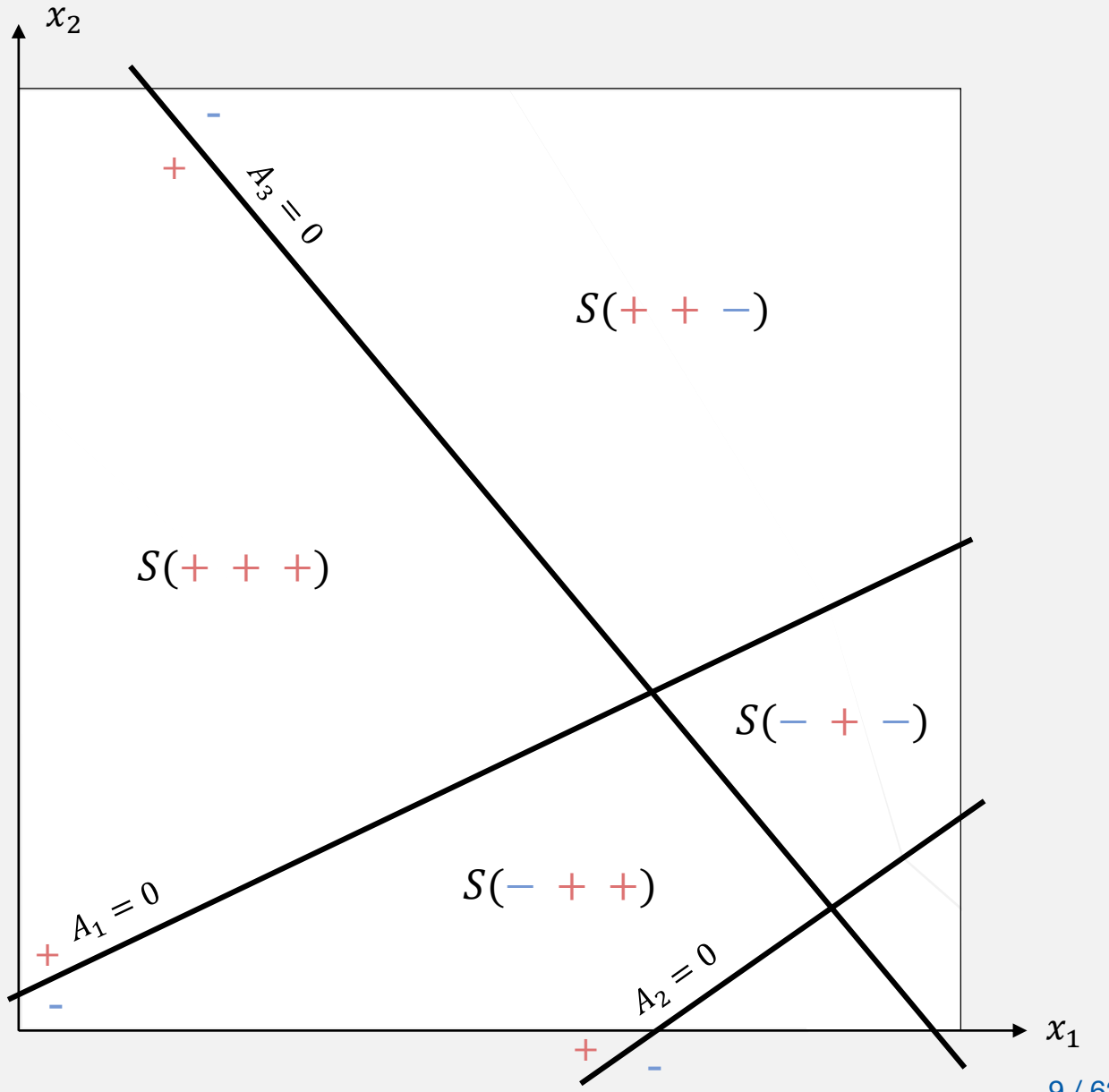
$$A_1 = -0.4x_1 + 0.8x_2$$
$$A_2 = -1.2x_1 + 1.8x_2 + 0.8$$
$$A_3 = -1.5x_1 - 1.3x_2 + 1.5$$

$$B_1 = 0.6|A_1| - 0.5|A_2| + 2.1|A_3| - 0.5$$

Первый слой персептрона разбивает входной компакт на непересекающиеся ячейки.

Каждая ячейка определяется **набором знаков** соответствующих неравенств нейронов $A_1, \ldots, A_k$.

Что будет на выходном слое?



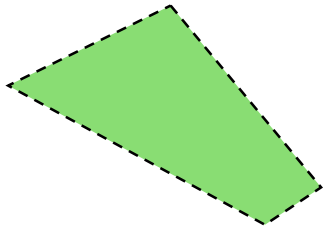
Геометрическая интерпретация: выходной слой

$$A_1 = -0.4x_1 + 0.8x_2$$
$$A_2 = -1.2x_1 + 1.8x_2 + 0.8$$
$$A_3 = -1.5x_1 - 1.3x_2 + 1.5$$

$$B_1 = 0.6|A_1| - 0.5|A_2| + 2.1|A_3| - 0.5$$

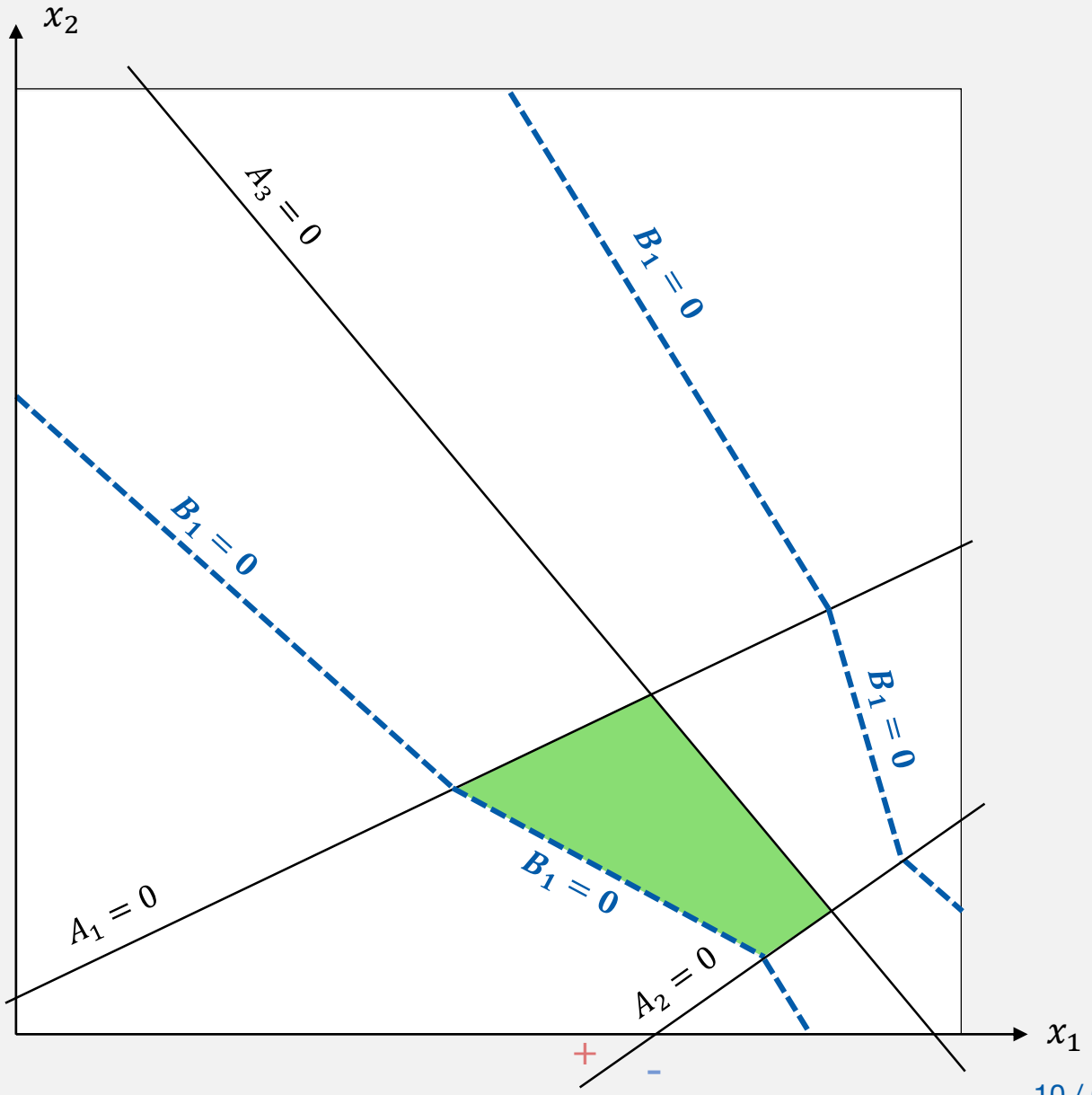
Выходной слой персептрона разбивает полученные предыдущим слоем ячейки.

Внутри каждой ячейки персептрон $c(x)$ является **линейной функцией**:



$$S(- \; + \; + \; -)$$

$$B_1 = -2.31x_1 - 4.11x_2 + 2.25$$



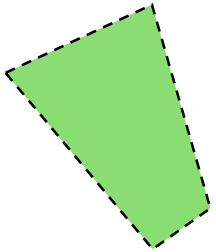
Геометрическая интерпретация: выходной слой

$$A_1 = -0.4x_1 + 0.8x_2$$
$$A_2 = -1.2x_1 + 1.8x_2 + 0.8$$
$$A_3 = -1.5x_1 - 1.3x_2 + 1.5$$

$$B_1 = 0.6|A_1| - 0.5|A_2| + 2.1|A_3| - 0.5$$

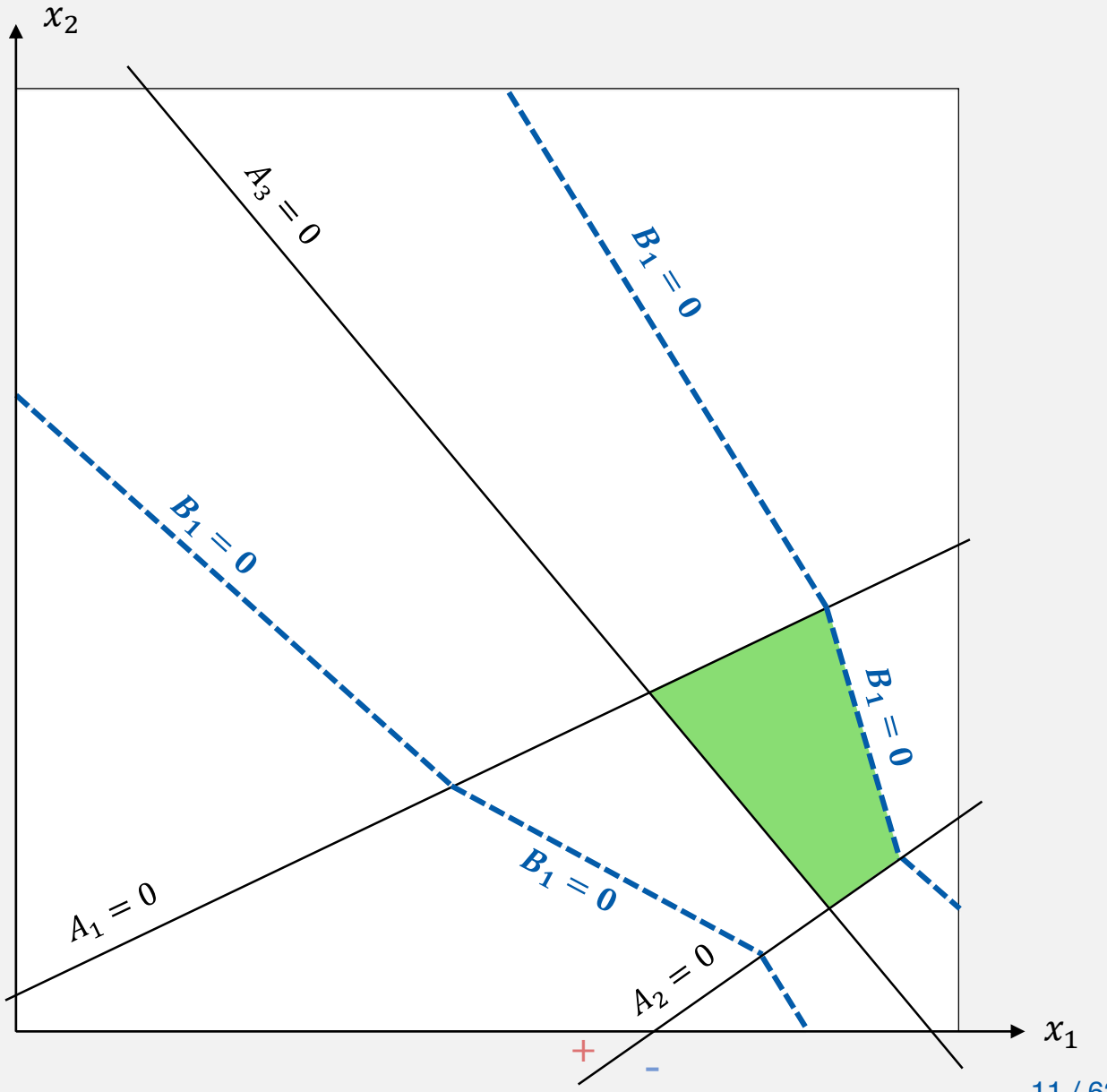
Выходной слой персептрона разбивает полученные предыдущим слоем ячейки.

Внутри каждой ячейки персептрон $c(x)$ является **линейной функцией**:



$$S(- \ + \ - \ -)$$

$$B_1 = 3.99x_1 + 1.35x_2 - 4.05$$



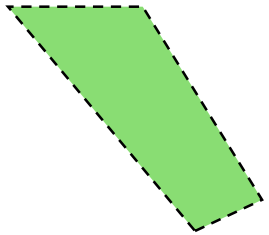
Геометрическая интерпретация: выходной слой

$$A_1 = -0.4x_1 + 0.8x_2$$
$$A_2 = -1.2x_1 + 1.8x_2 + 0.8$$
$$A_3 = -1.5x_1 - 1.3x_2 + 1.5$$

$$B_1 = 0.6|A_1| - 0.5|A_2| + 2.1|A_3| - 0.5$$

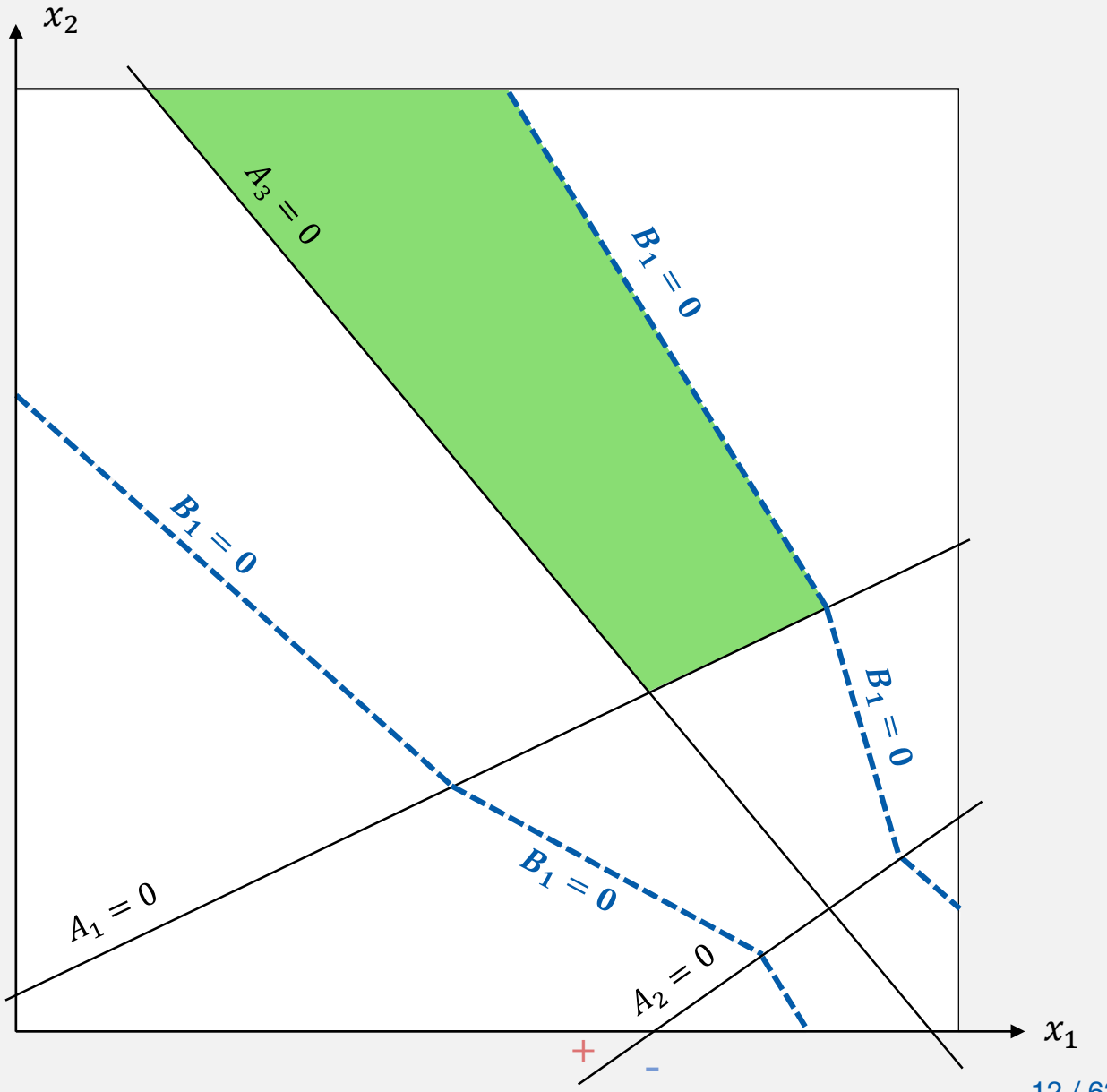
Выходной слой персептрона разбивает полученные предыдущим слоем ячейки.

Внутри каждой ячейки персептрон $c(x)$ является **линейной функцией**:



$$S(+\ +\ -\ -)$$

$$B_1 = 3.51x_1 + 2.31x_2 - 4.05$$



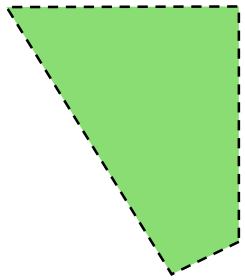
Геометрическая интерпретация: выходной слой

$$A_1 = -0.4x_1 + 0.8x_2$$
$$A_2 = -1.2x_1 + 1.8x_2 + 0.8$$
$$A_3 = -1.5x_1 - 1.3x_2 + 1.5$$

$$B_1 = 0.6|A_1| - 0.5|A_2| + 2.1|A_3| - 0.5$$

Выходной слой персептрона разбивает полученные предыдущим слоем ячейки.

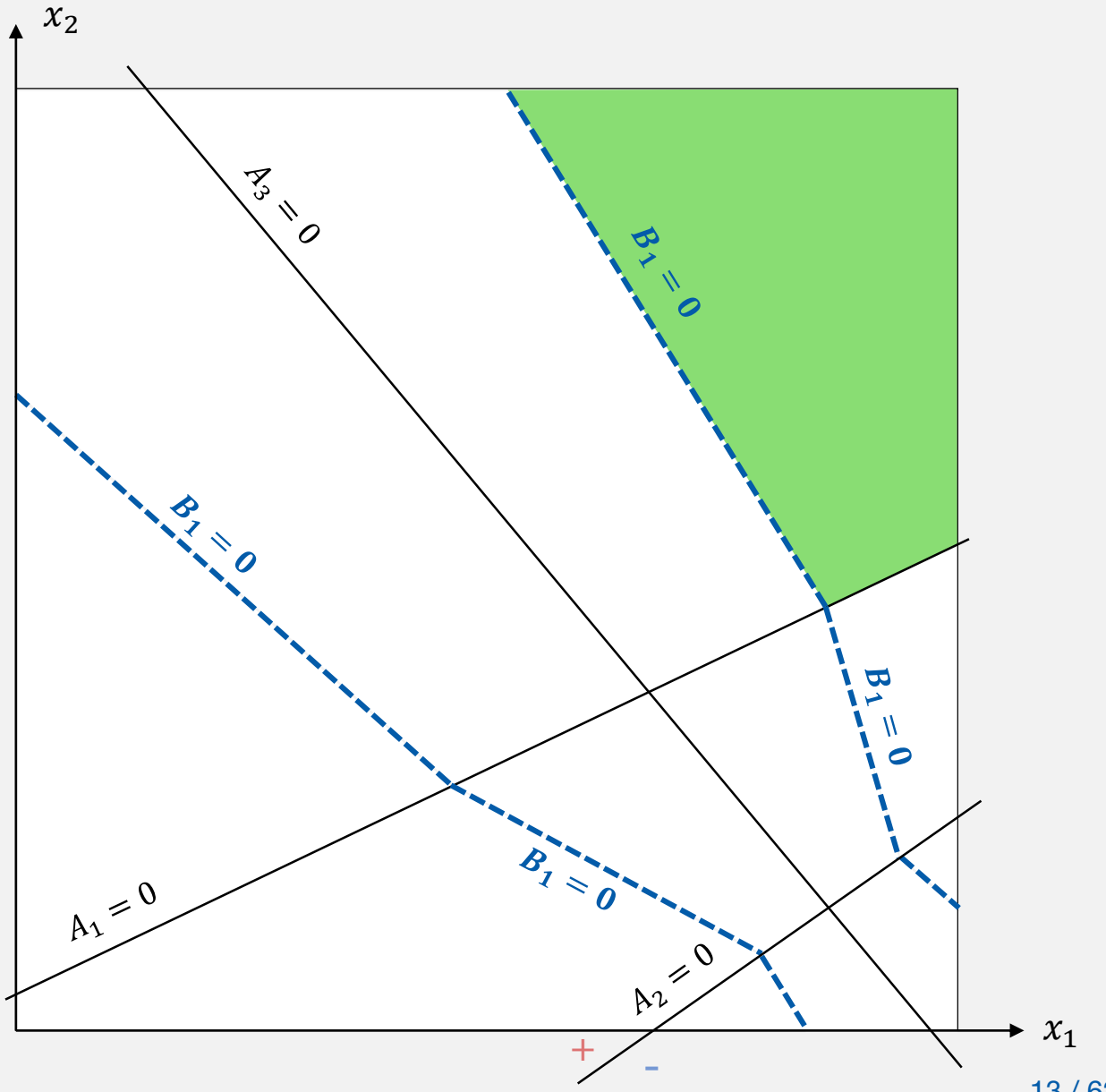
Внутри каждой ячейки персептрон $c(x)$ является **линейной функцией**:



$$S(+\ +\ -\ +)$$

$$B_1 = 3.51x_1 + 2.31x_2 - 4.05$$

Что делать, если слоёв больше одного?



Бинарный байесовский классификатор

(X, Y) – вектор из **распределения** P;

$X \in [0,1]^d$ – признаки из **распределения** P_x с **носителем** S;

$Y \in \{\pm 1\}$ – **бинарные** метки;

$f : [0,1]^d \to \{\pm 1\}$ – **дискриминантная** функция.

Решение задачи регрессии:

$$g(x) = \mathbb{E}(Y \mid X = x), \qquad x \in [0,1]^d$$

Классический байесовский классификатор:

$$s(x) = \begin{cases} +1, & g(x) > 0 \quad \text{и} \quad x \in S \\ \pm 1, & g(x) = 0 \text{ или } x \notin S \\ -1, & g(x) < 0 \quad \text{и} \quad x \in S \end{cases}$$

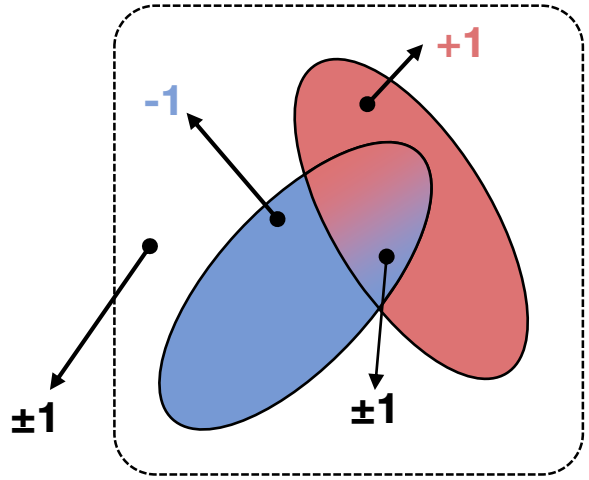
Общая задача

Классификация:

$$\mathbb{P}\big(Y \neq f(X)\big) \to \min_f$$

Регрессия:

$$\mathbb{E}\big(Y - f(X)\big)^2 \to \min_f$$



Модификация байесовского классификатора

Добавим данные с **нулевыми** метками и **равномерно распределёнными** на $[0,1]^d$ признаками («фон»):

Смесь двух распределений:

$P_\alpha = (1-\alpha)P + \alpha\mathcal{P}, \quad \alpha \in (0,1),$

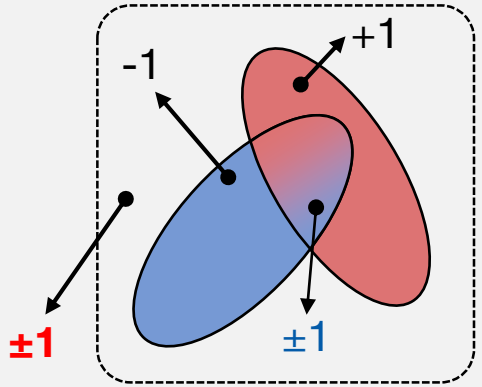
$\mathcal{P}$ – распределение «фона».

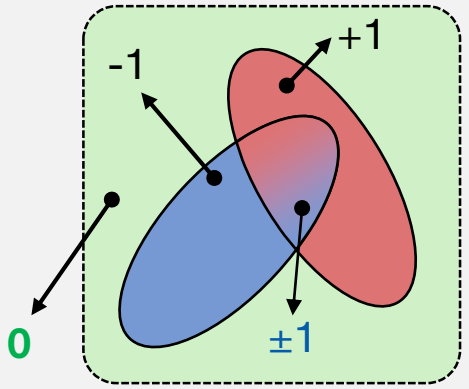
Классический классификатор $s(x)$:

$$s(x) = \begin{cases} +1, & g(x) > 0 \text{ и } x \in S \\ \pm 1, & g(x) = 0 \text{ и } x \in S \\ -1, & g(x) < 0 \text{ и } x \in S \\ \pm 1, & x \notin S \end{cases}$$

Модифицированный классификатор $s_\alpha(x)$:

$$s_\alpha(x) = \begin{cases} +1, & g_\alpha(x) > 0 \text{ и } x \in S \\ \pm 1, & g_\alpha(x) = 0 \text{ и } x \in S \\ -1, & g_\alpha(x) < 0 \text{ и } x \in S \\ \mathbf{0}, & x \notin S \end{cases}$$





Аппроксимация байесовского классификатора

$D: (X, Y) = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ – размеченный **обучающий** набор, $X_i \in [0,1]^d$, $Y_i \in \{-1, +1\}$;

$c(X): [0,1]^d \to R$ – равномерно непрерывная функция.

Задача средне-квадратичной аппроксимации:

$$E(c(X) - Y)^2 \to \min_{c(X)}$$

Сводится к аппроксимации функции регрессии:

$$E\big(c(X) - g(X)\big)^2 \to \min_{c(X)}$$

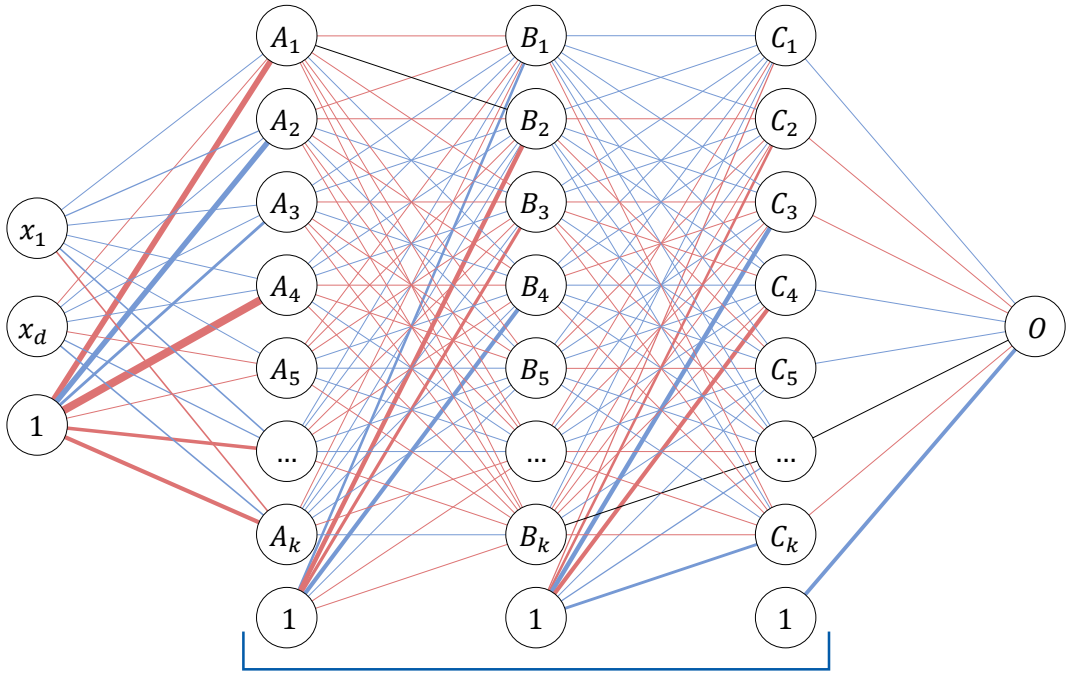
Универсальная теорема аппроксимации: для любого $\varepsilon > 0$ существуют такие k и L, что для любого $x \in [0,1]^d$:

$$\sup_{x \in [0,1]^d} |c(x) - g(x)| < \varepsilon$$

Конфигурация персептрона:

01 L **скрытых** слоёв по k нейронов с кусочно-линейной активацией

02 **Один** нейрон в **выходном** слое с линейной функцией активации



Функция нейросетевой регрессии

$\{(X_{n+1}, Y_{n+1}), \ldots, (X_{2n}, Y_{2n})\}$ – сгенерированные **фоновые** данные, $X_i \in [0,1]^d$, $Y_i = 0$;

$f(X)$ – равномерно непрерывная плотность выборки (X, Y) на S;

$p(X)$ – плотность равномерно распределённых на $[0,1]^d$ фоновых векторов.

$C(L,k)$ – класс многослойных персептронов с активацией $\boldsymbol{Abs}$ и заданными $\boldsymbol{L}$, $\boldsymbol{k}$.

Выборочная оценка решения задачи аппроксимации

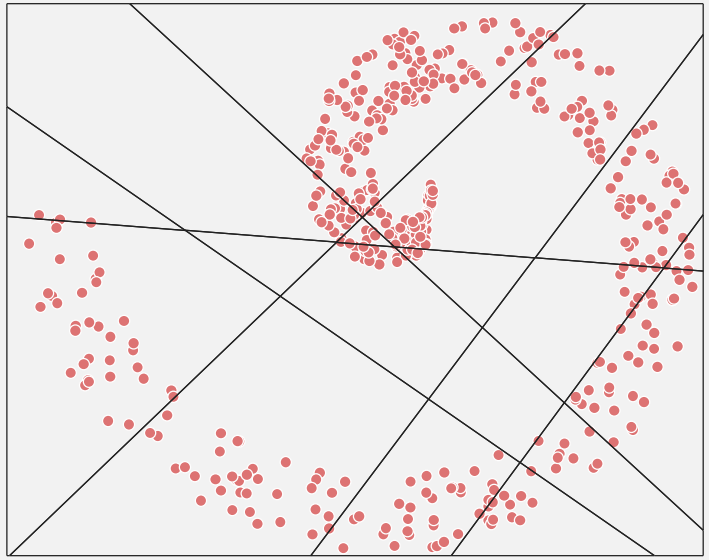
$$\sum_{i=1}^{2n} (c_n(X_i) - Y_i)^2 \to \min_{c_n(X) \in C(L,k)}$$

$c_n^*(X)$ – решение задачи, называемое **функцией нейросетевой регрессии**.

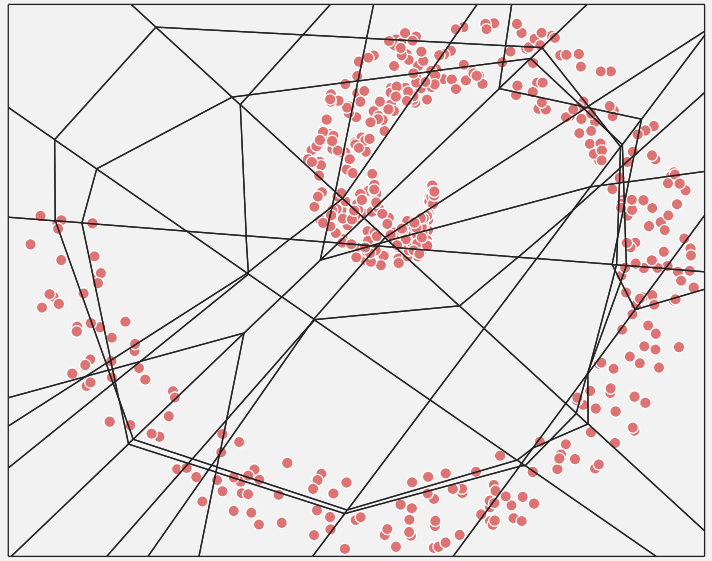
Иерархическое разбиение на ячейки

$c_n^*(x)$ строит **иерархическое разбиение** $[0,1]^d$ на N непересекающихся ячеек: $K = \{K_1, K_2, \ldots, K_N\}$

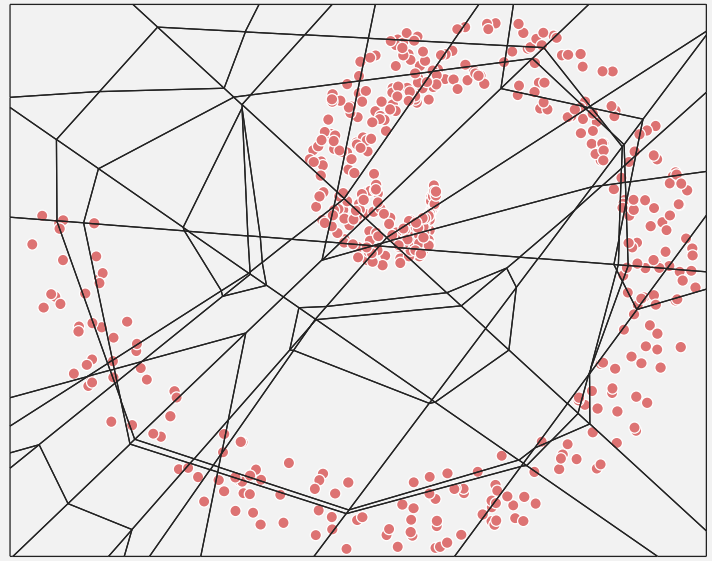
Пример разбиения некоторым MLP с $L = 2$, $k = 6$:



Первый скрытый слой: 18 ячеек



Второй скрытый слой: 121 ячейка



Выходной слой: 148 ячеек

Функция гистограммной регрессии

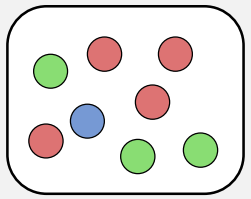
$h_n(X)$ – **кусочно-постоянная** функция **гистограммной регрессии**, определённая путём минимизации:

$$\sum_{i=1}^{2n} (h_n(X_i) - Y_i)^2 \to \min_{h_n(X)}$$

Пусть $X \in K_r \to$ для каждой ячейки K_r **оптимальное решение** имеет вид:

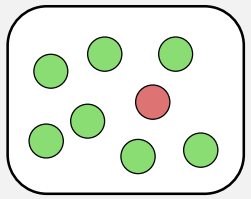
$$h_n^*(X) = \frac{n_1(X) - n_{-1}(X)}{n_{-1}(X) + n_0(X) + n_1(X)}$$

$n_j(X)$ – количество наблюдений с меткой j в ячейке.



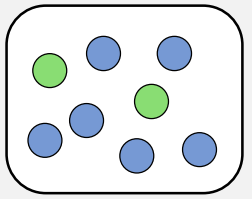
$n_{-1} = 1, n_1 = 4, n_0 = 3$

$h_n^*(X) = \frac{4-1}{1+3+4} = 0.375$



$n_{-1} = 0, n_1 = 1, n_0 = 7$

$h_n^*(X) = \frac{1-0}{0+7+1} = 0.125$



$n_{-1} = 6, n_1 = 0, n_0 = 2$

$h_n^*(X) = \frac{0-6}{6+2+0} = -0.75$

eXBTree: объясняющее дерево решений

Идея:

Раскрытие модулей в каждом нейроне – это переход по ветвям двоичного дерева (по знаку неравенства).

Узлы: линейные неравенства нейронов.

Листья: линейная функция и гистограмма классов.

Адрес ячейки (листа): знаки нейронов до активации.

Интерпретируемость и доверие:

Геометрия ячейки: выпуклый многогранник (система линейных неравенств).

Условия принятия **решения**: путь по дереву.

Прецеденты: объекты из той же или соседних ячеек.

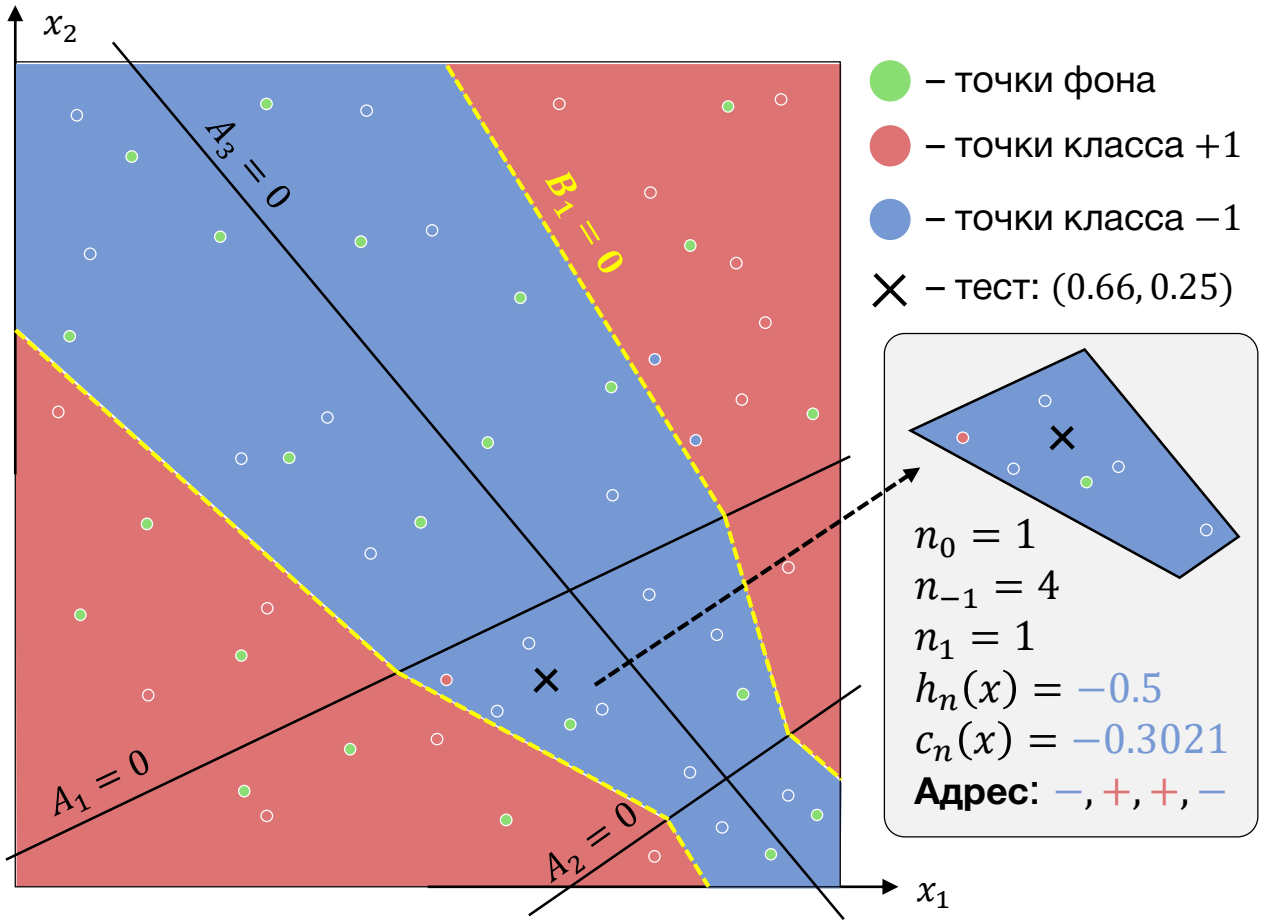
Малое число точек: адаптивное укрупнение области (подъём на уровень выше) или отказ от решения.

Сложность

Построение (полное): $O\left(k^{dL}\right)$.

Построение (по выборке): $O(n)$.

Применение (1 наблюдение): $O(Lkd) = O(1)$.



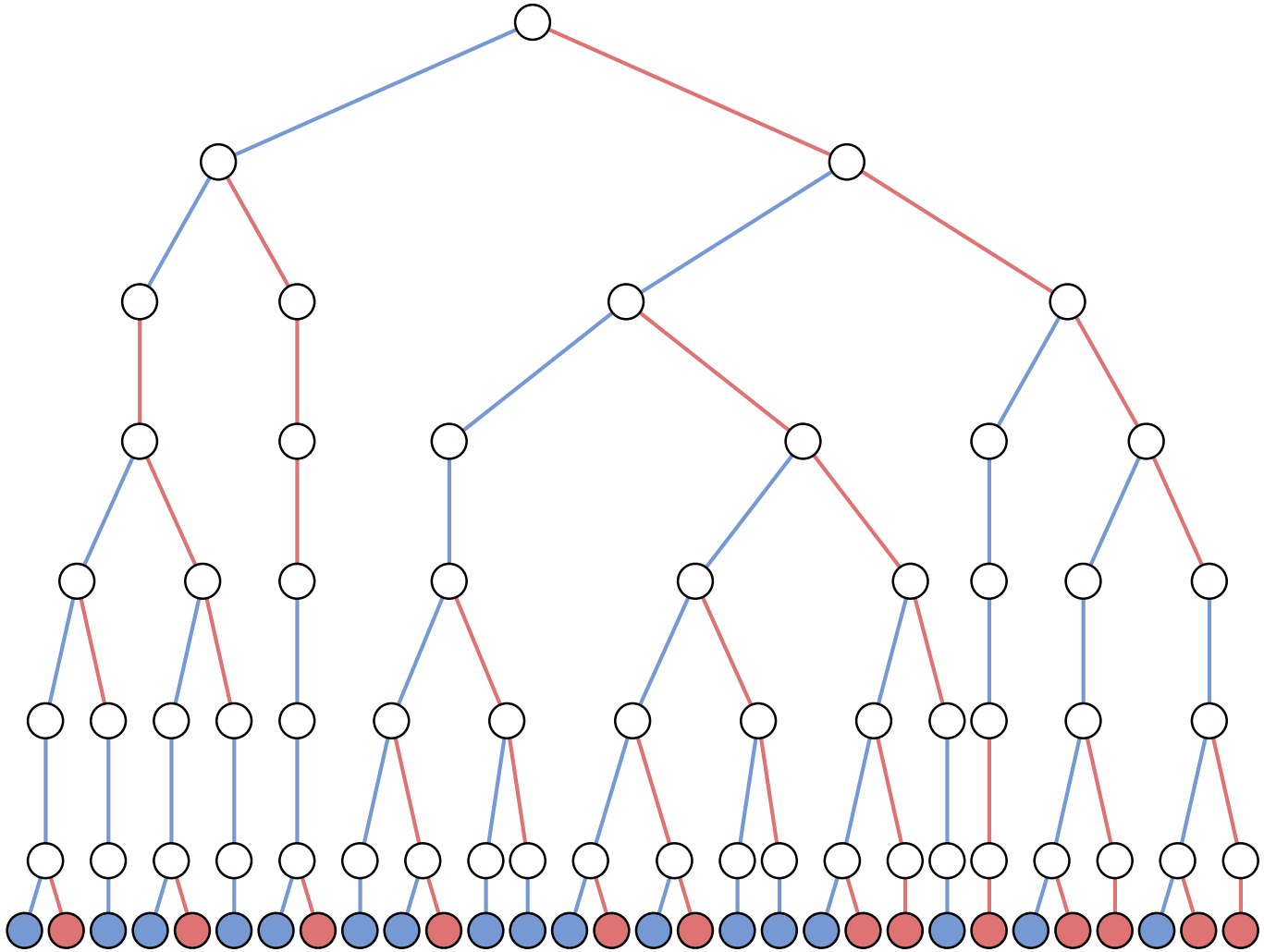
$$A_1 = -0.4x_1 + 0.8x_2 \qquad A_1(x) = -0.064 < 0 \to -$$
$$A_2 = -1.2x_1 + 1.8x_2 + 0.8 \qquad A_2(x) = 0.458 > 0 \to +$$
$$A_3 = -1.5x_1 - 1.3x_2 + 1.5 \qquad A_3(x) = 0.185 > 0 \to +$$

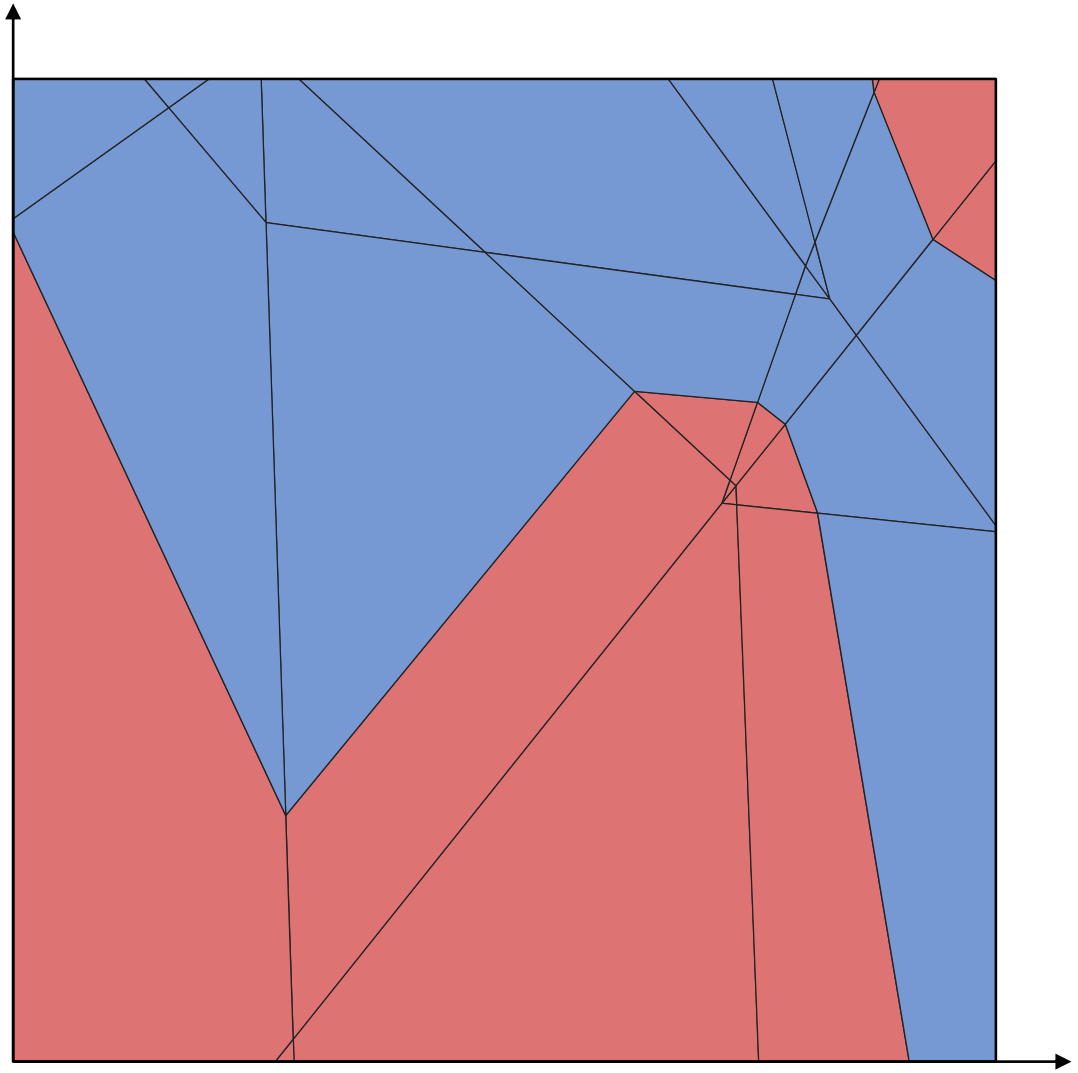
$$B_1 = 0.6|A_1| - 0.5|A_2| + 2.1|A_3| - 0.5 \qquad B_1(x) = -0.3021 < 0 \to -$$

Линейная функция ячейки: $-2.31x_1 - 4.11x_2 + 2.25$

eXBTree: подробный пример

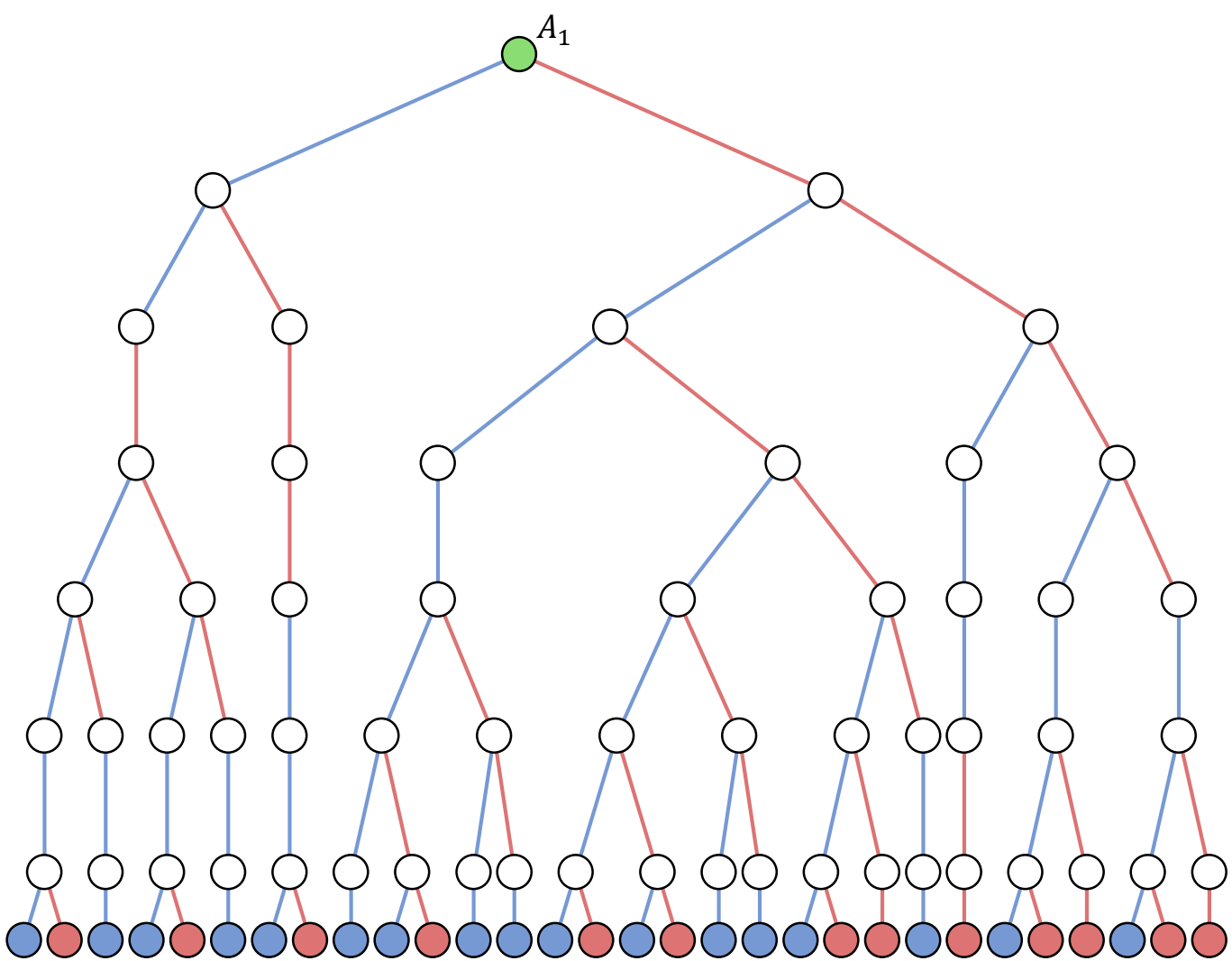
Модель с двумя скрытыми слоями по 3 нейрона: полное дерево

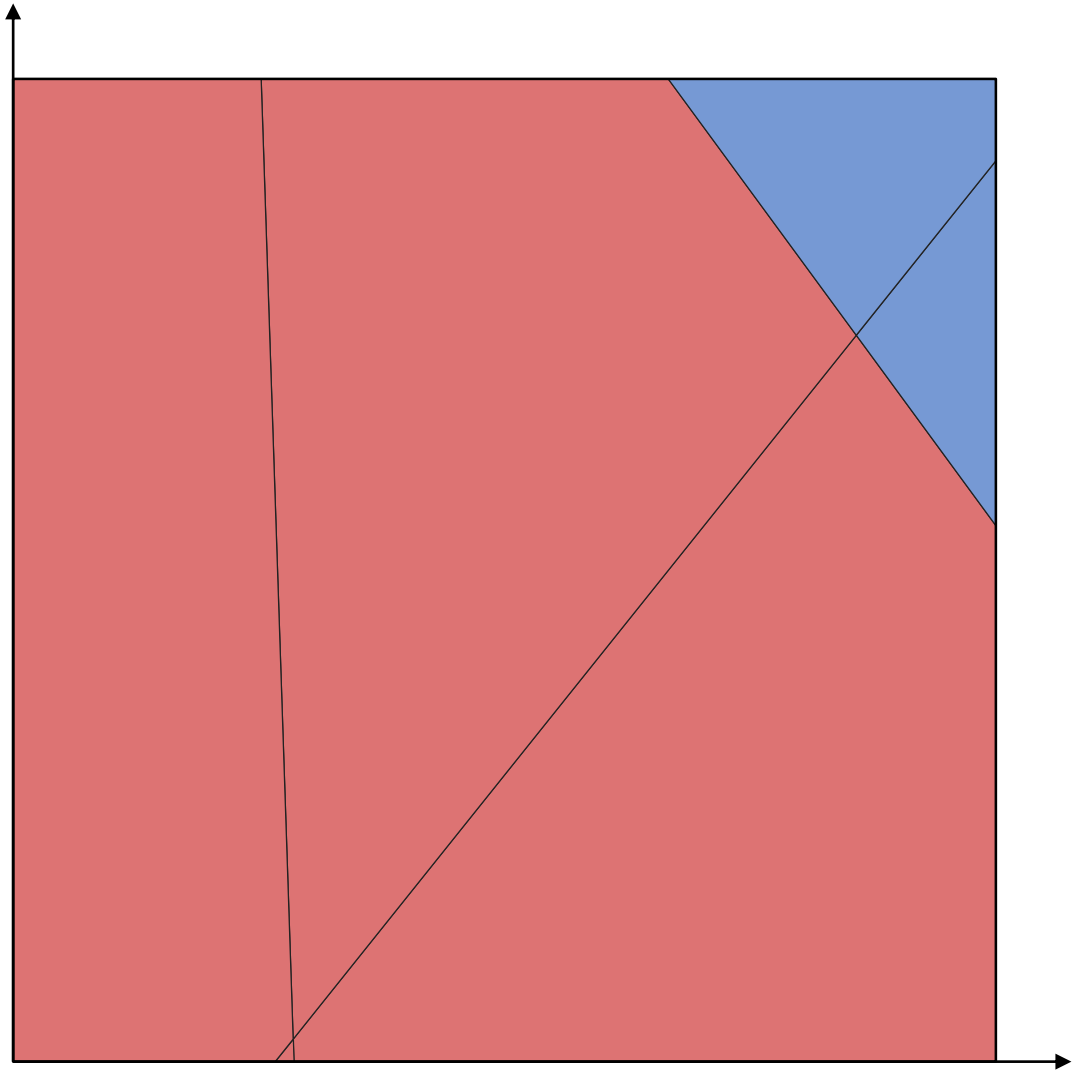




eXBTree: ячейки первого скрытого слоя

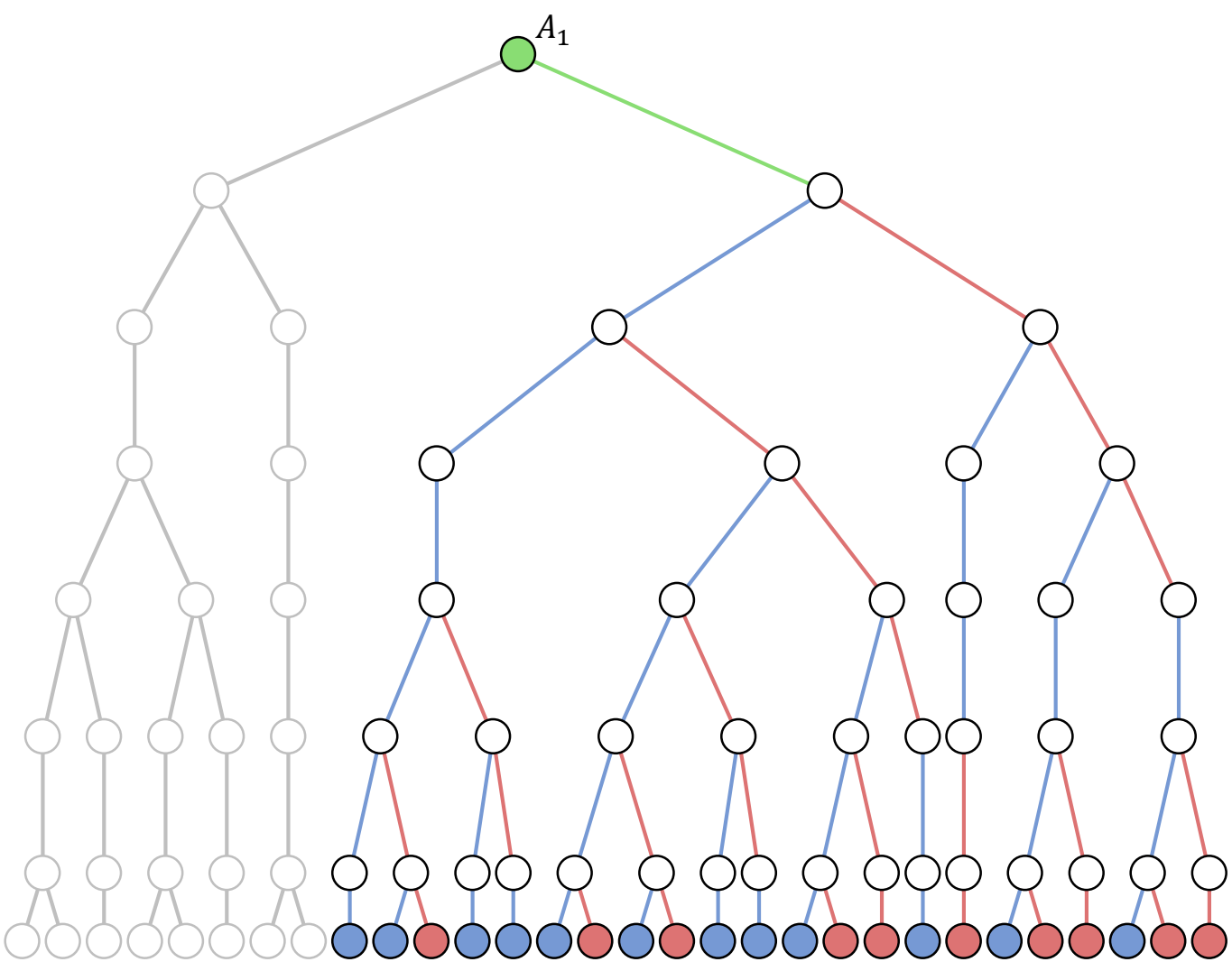
Проследим путь $(+\ -\ +\ -\ -\ +\ -)$ по дереву: нейрон A_1

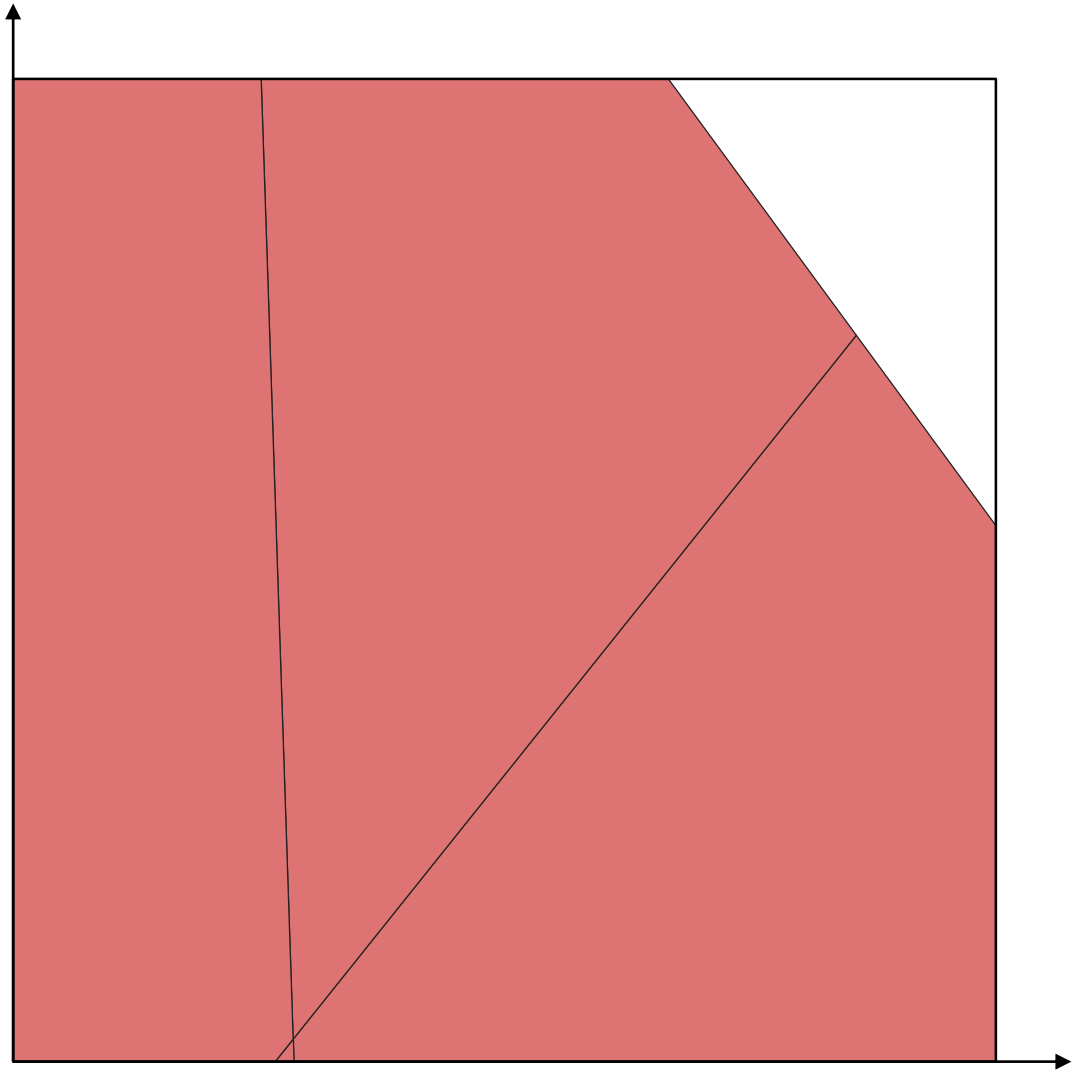




eXBTree: ячейки первого скрытого слоя

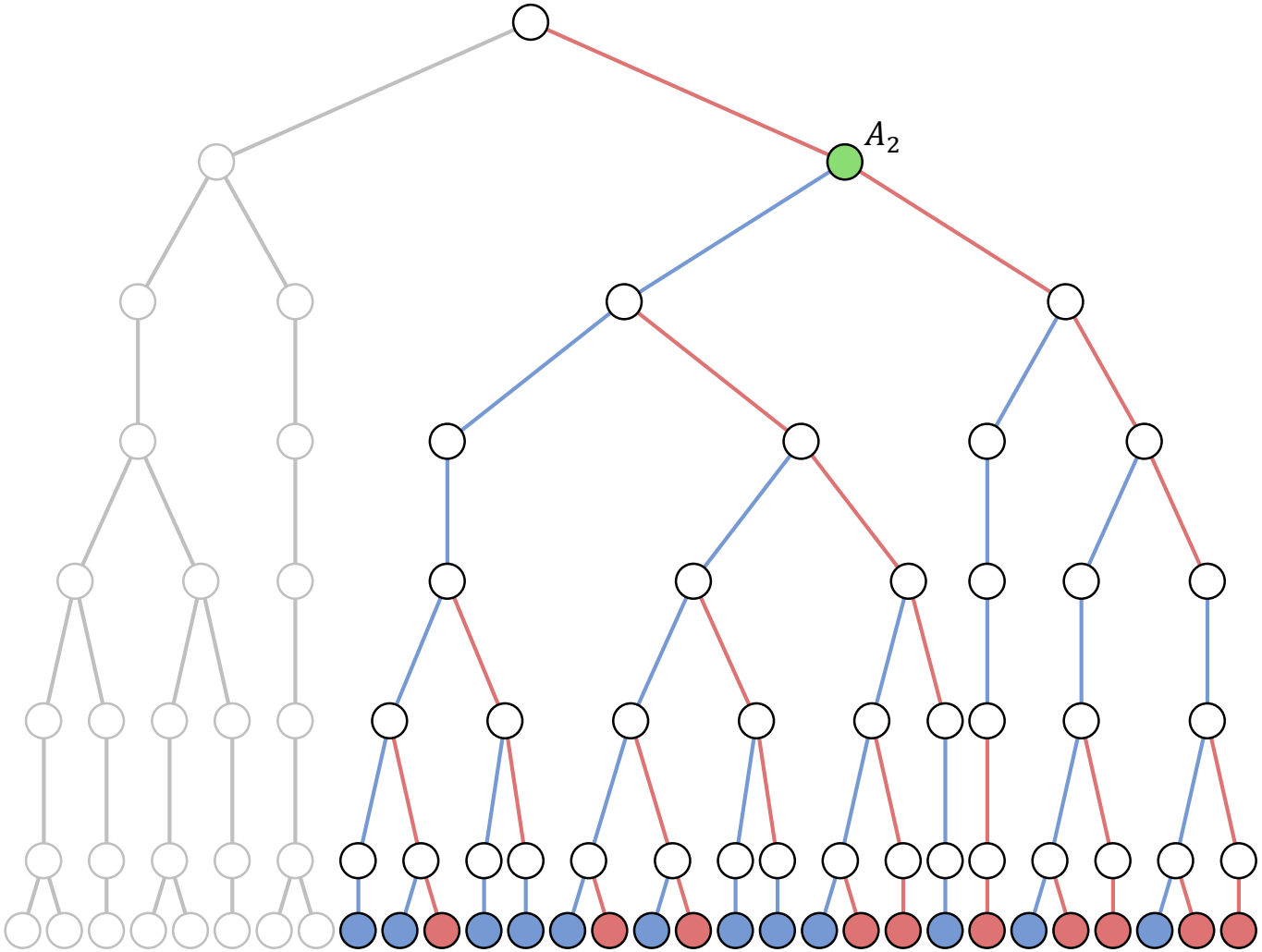
Проследим путь ($+\ -\ +\ -\ -\ +\ -$) по дереву: нейрон $A_1 > 0$

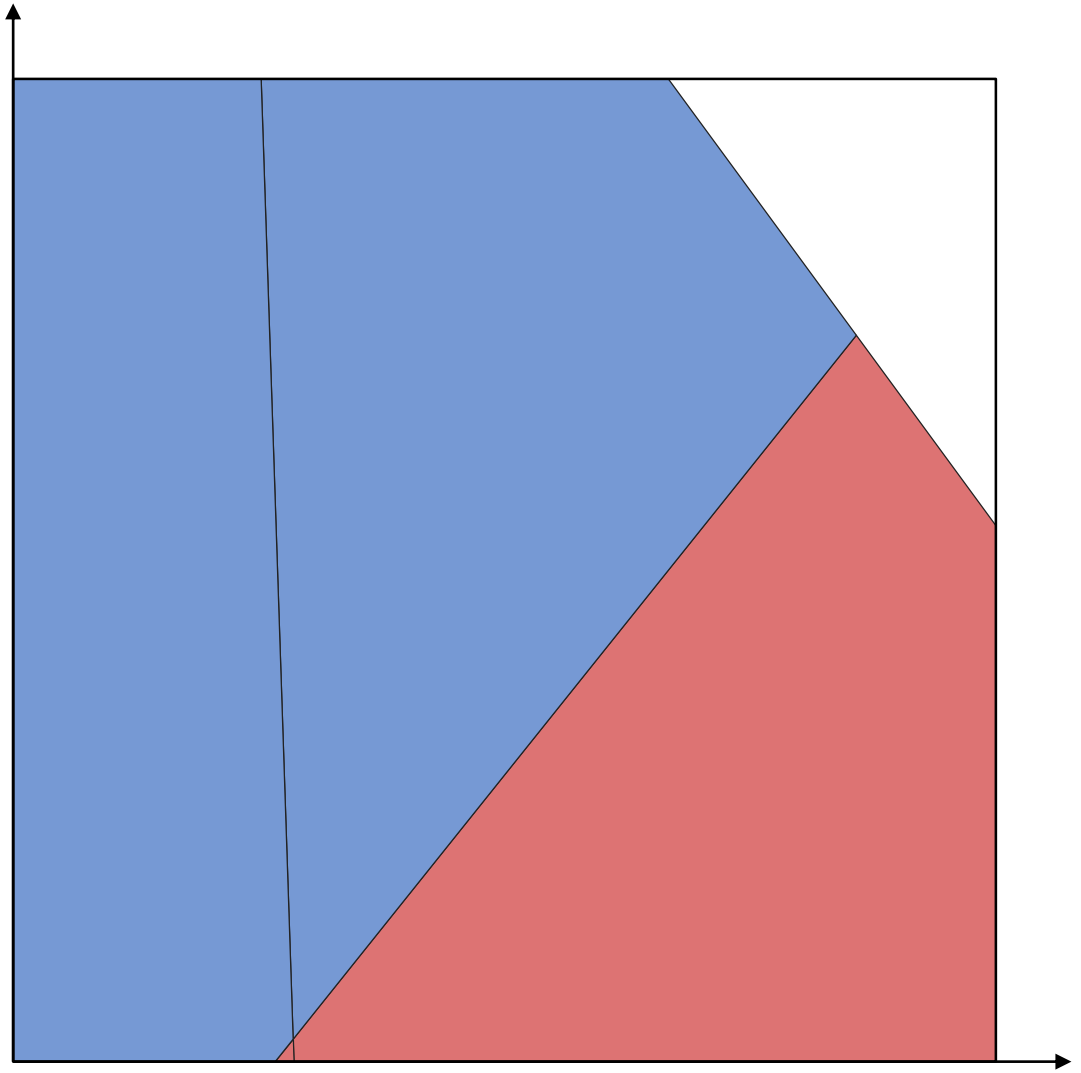




eXBTree: ячейки первого скрытого слоя

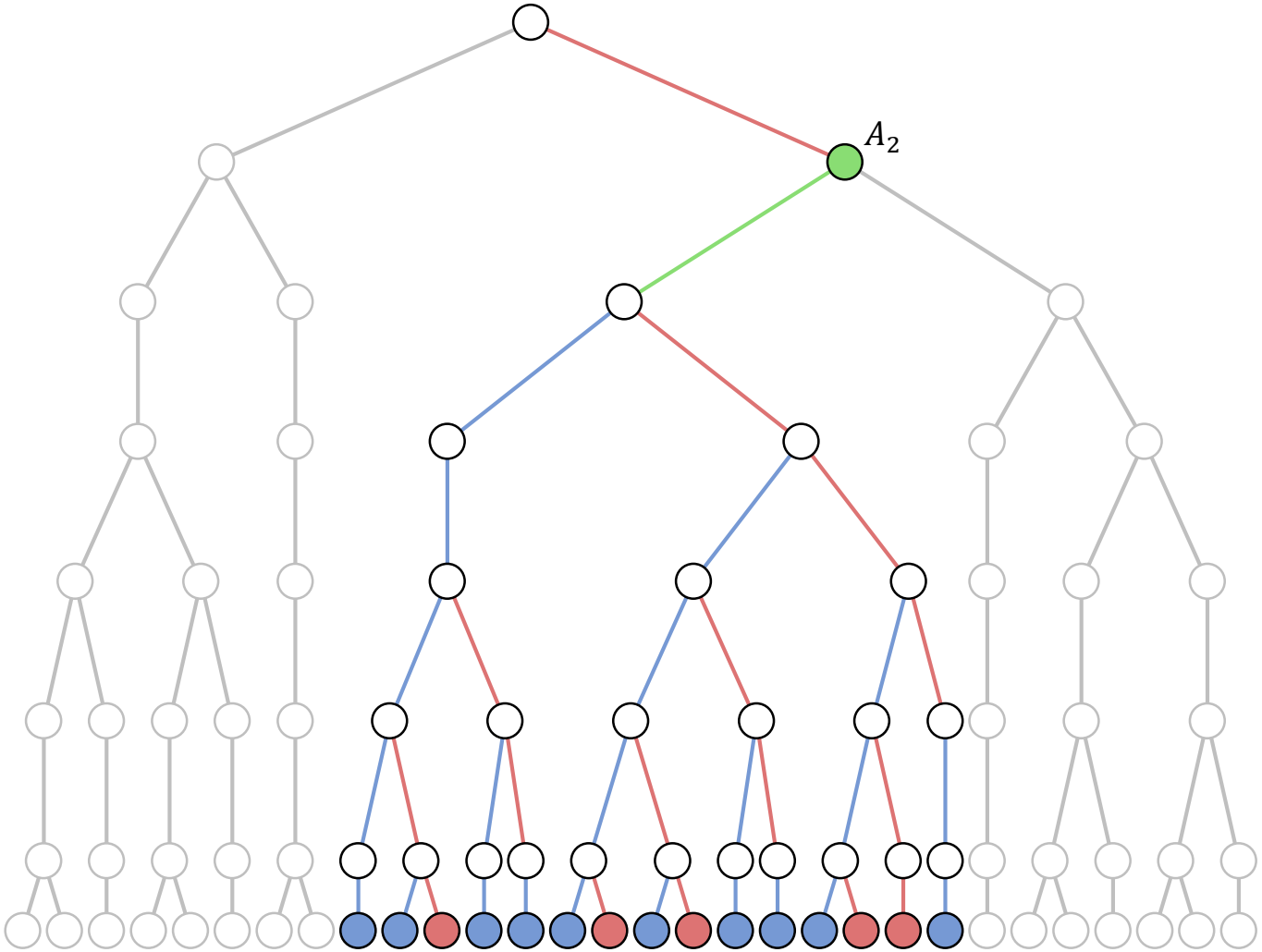
Проследим путь ($+\ -\ +\ -\ -\ +\ -$) по дереву: нейрон A_2

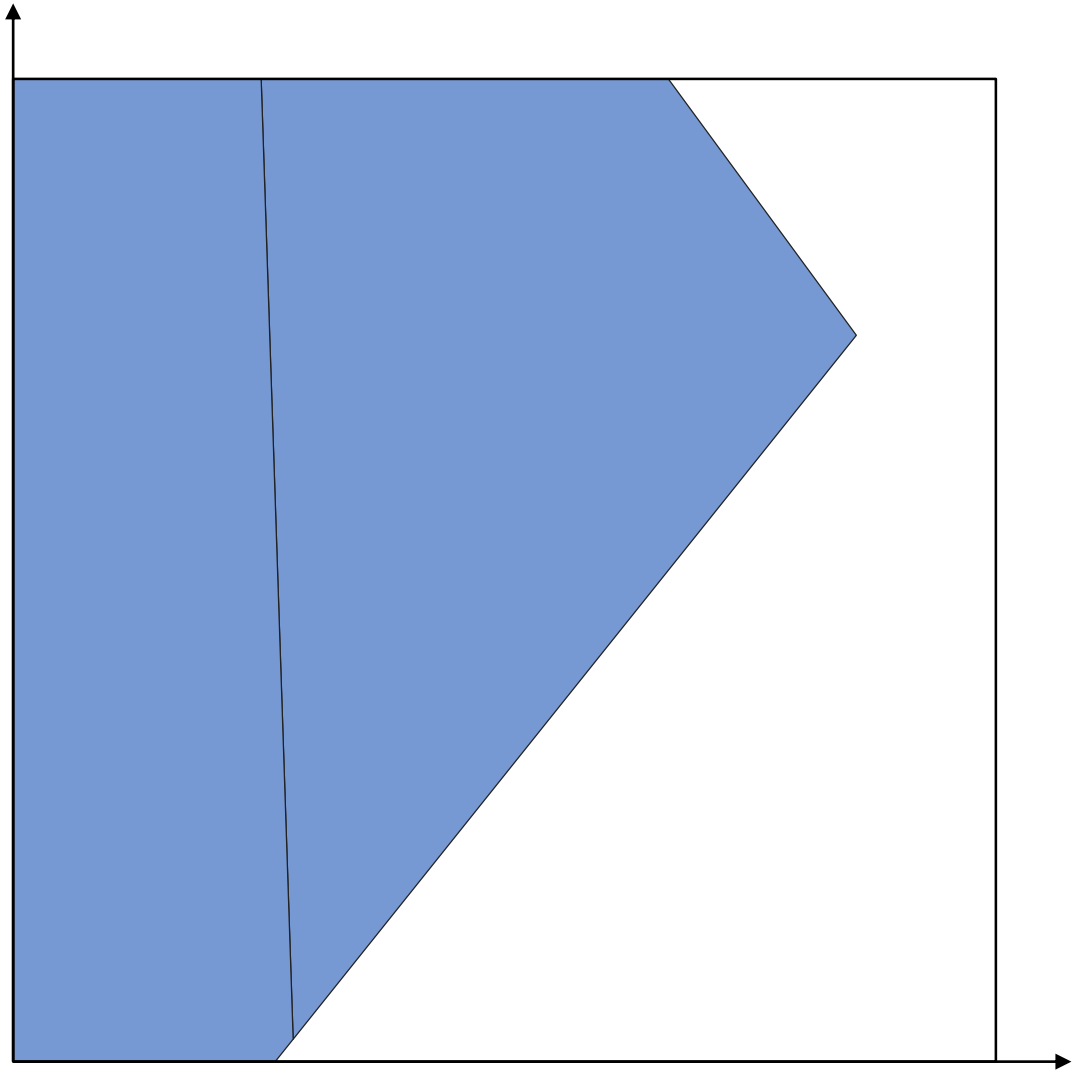




eXBTree: ячейки первого скрытого слоя

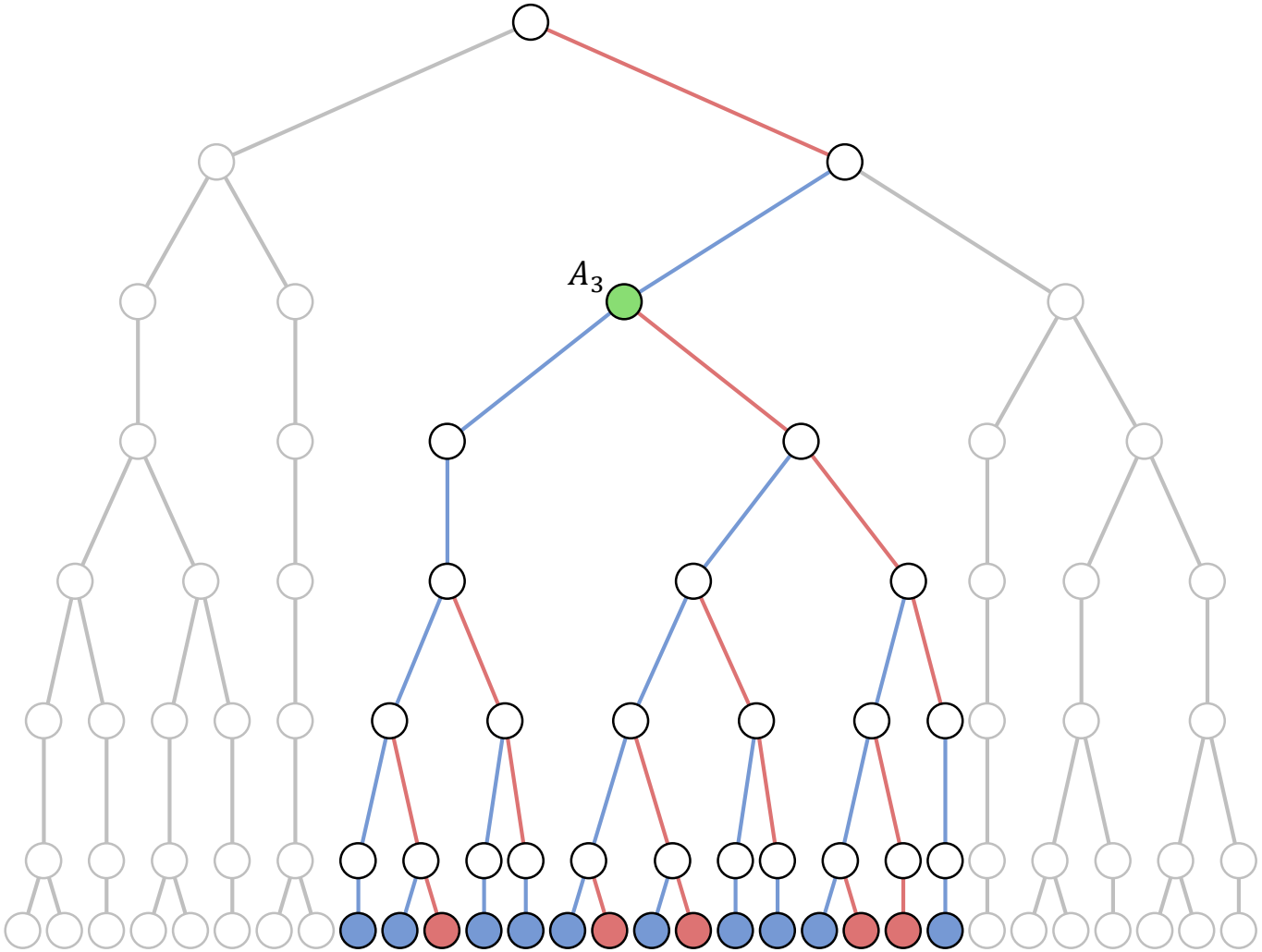
Проследим путь ($+\ -\ +\ -\ -\ +\ -$) по дереву: нейрон $A_2 < 0$

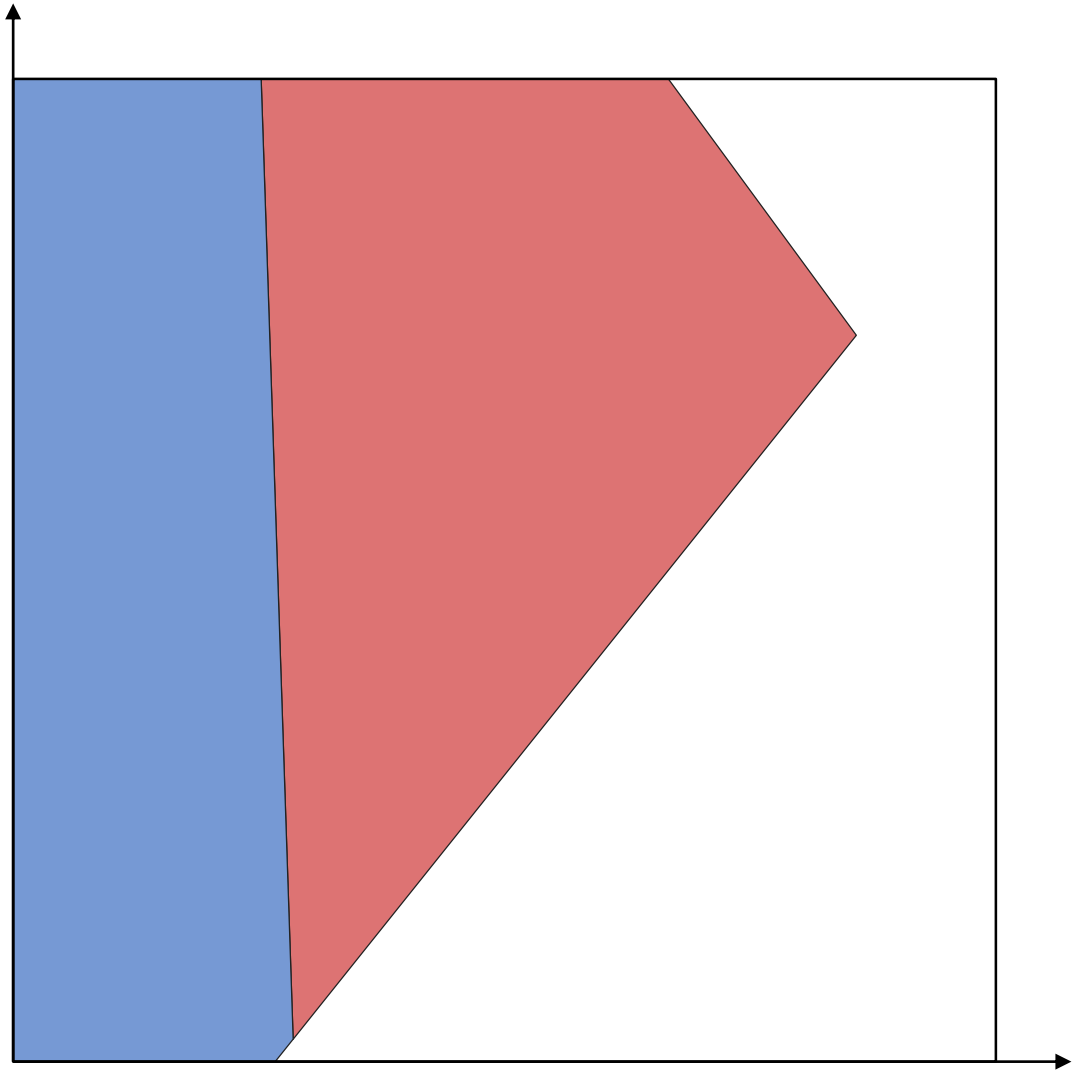




eXBTree: ячейки первого скрытого слоя

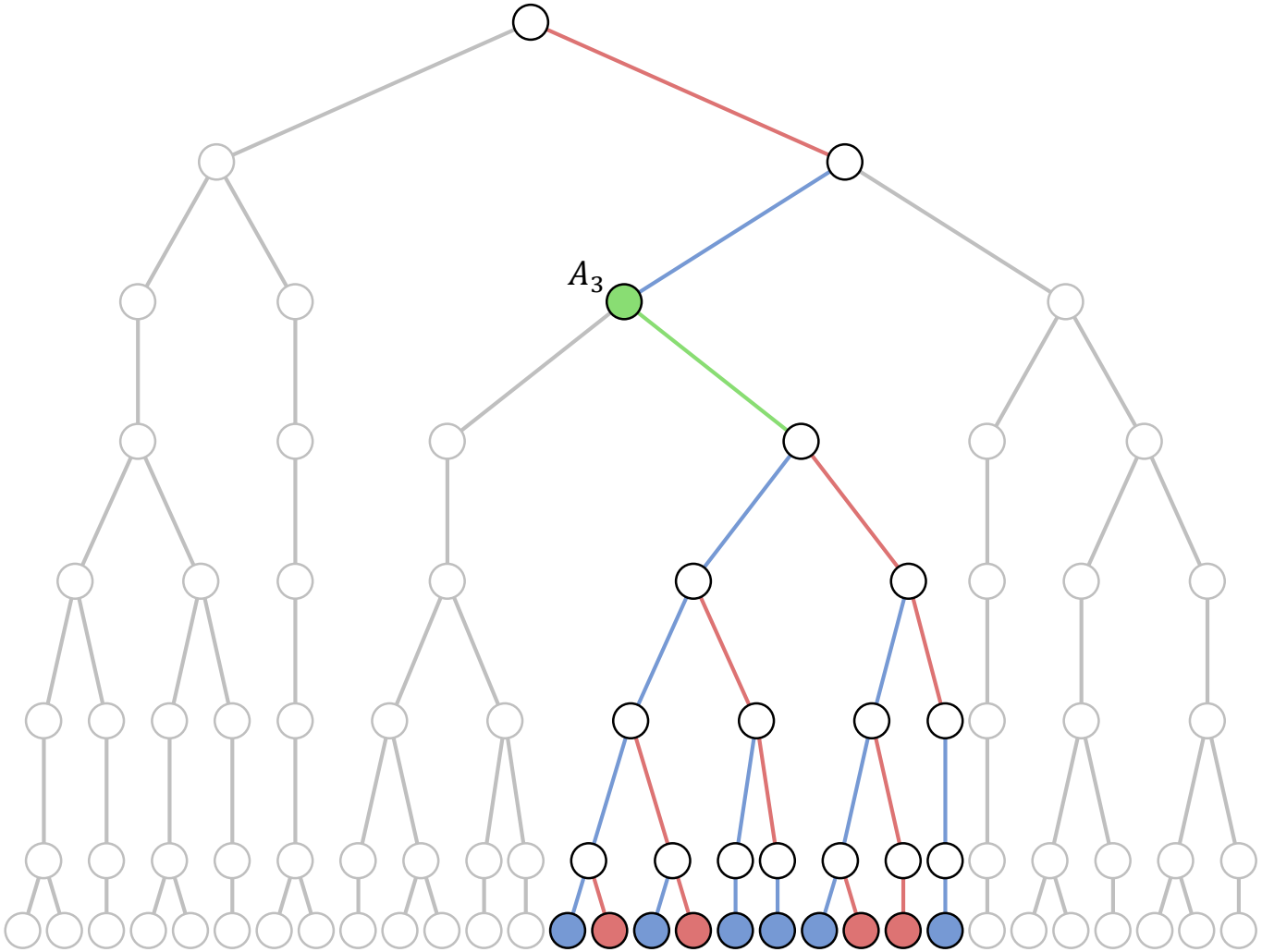
Проследим путь ($+\ -\ +\ -\ -\ +\ -$) по дереву: нейрон A_3

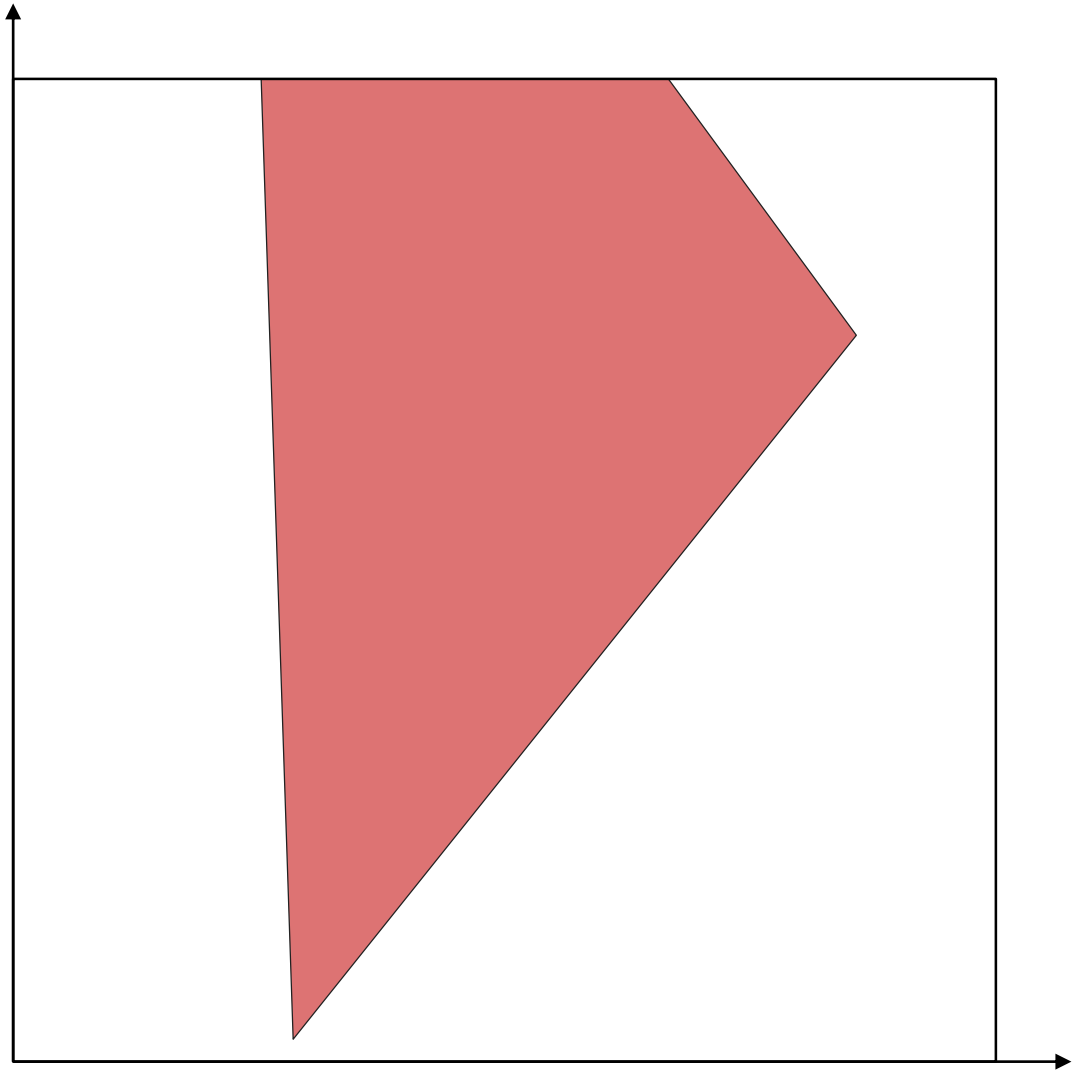




eXBTree: ячейки первого скрытого слоя

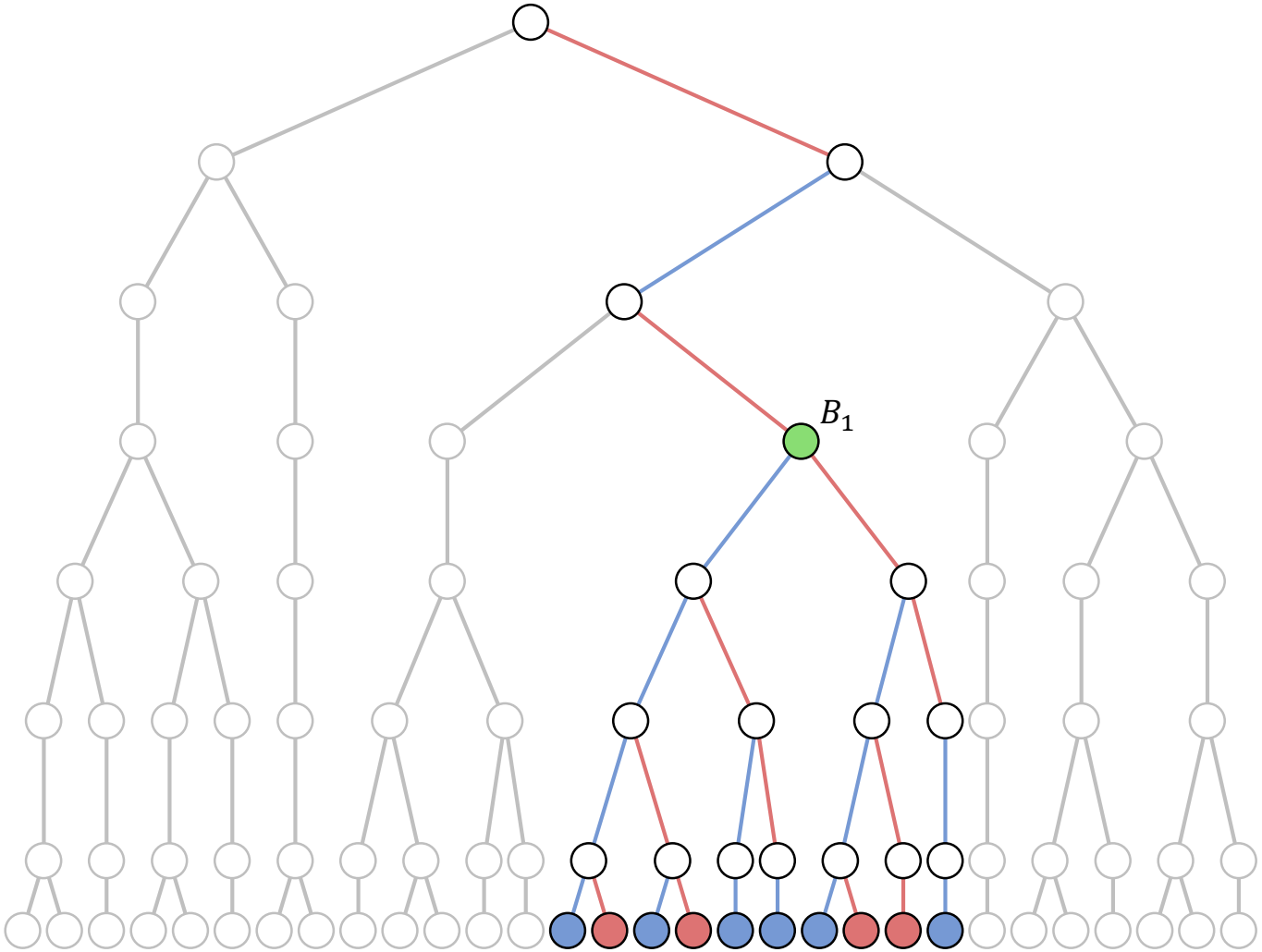
Проследим путь ($+ \ - \ + \ - \ - \ + \ -$) по дереву: нейрон $A_3 > 0$

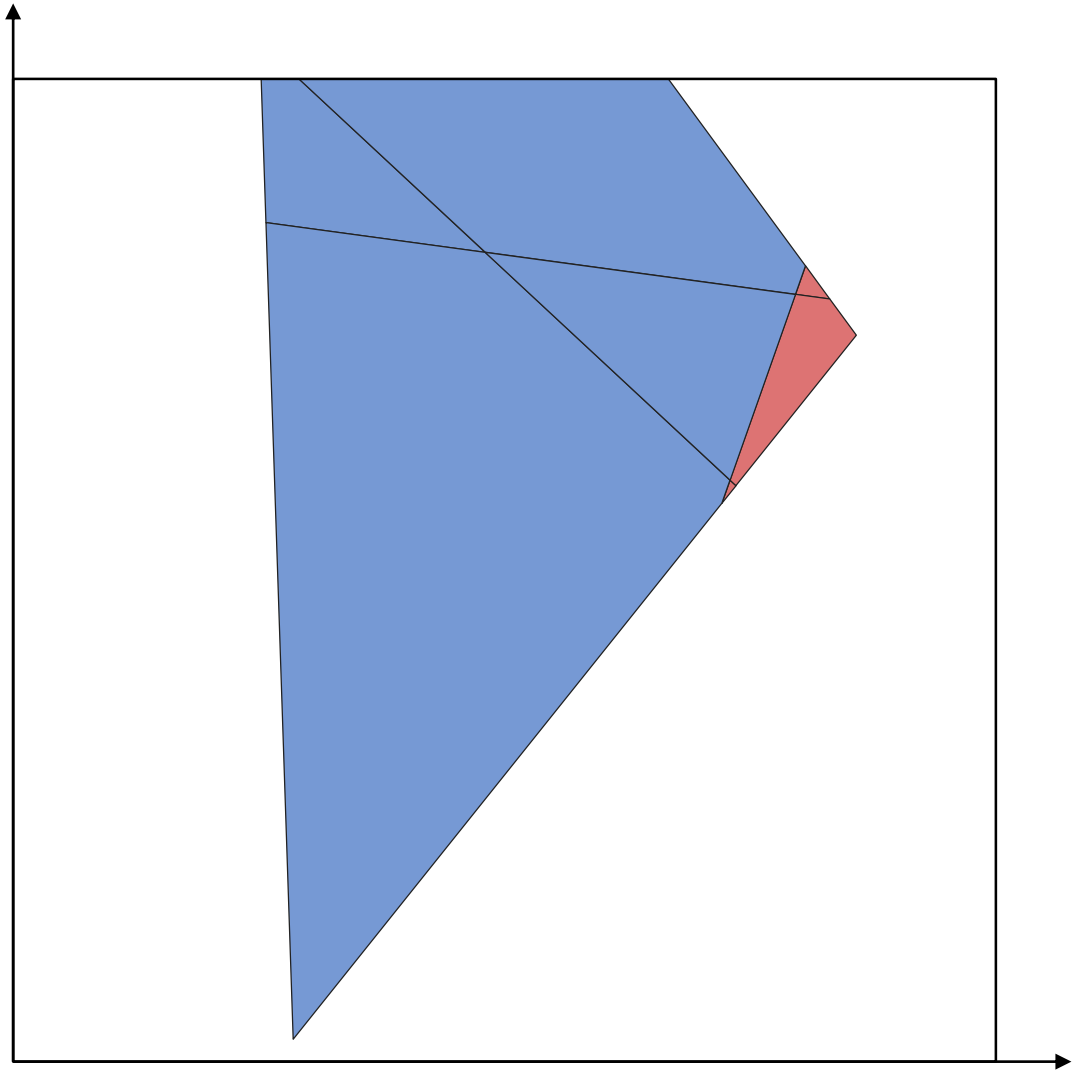




eXBTree: ячейки второго скрытого слоя

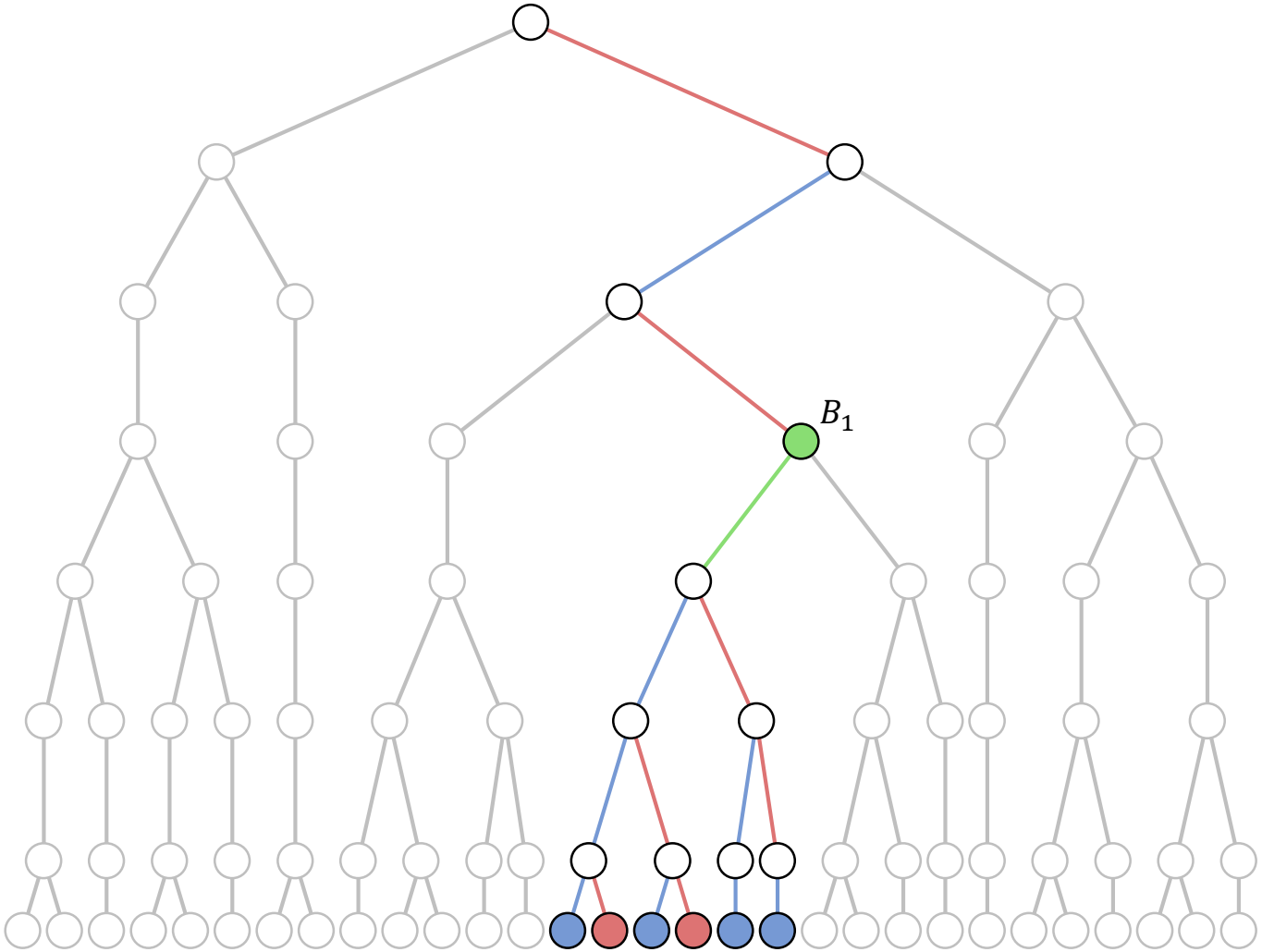
Проследим путь ($+ \; - \; + \; - \; - \; + \; -$) по дереву: нейрон B_1

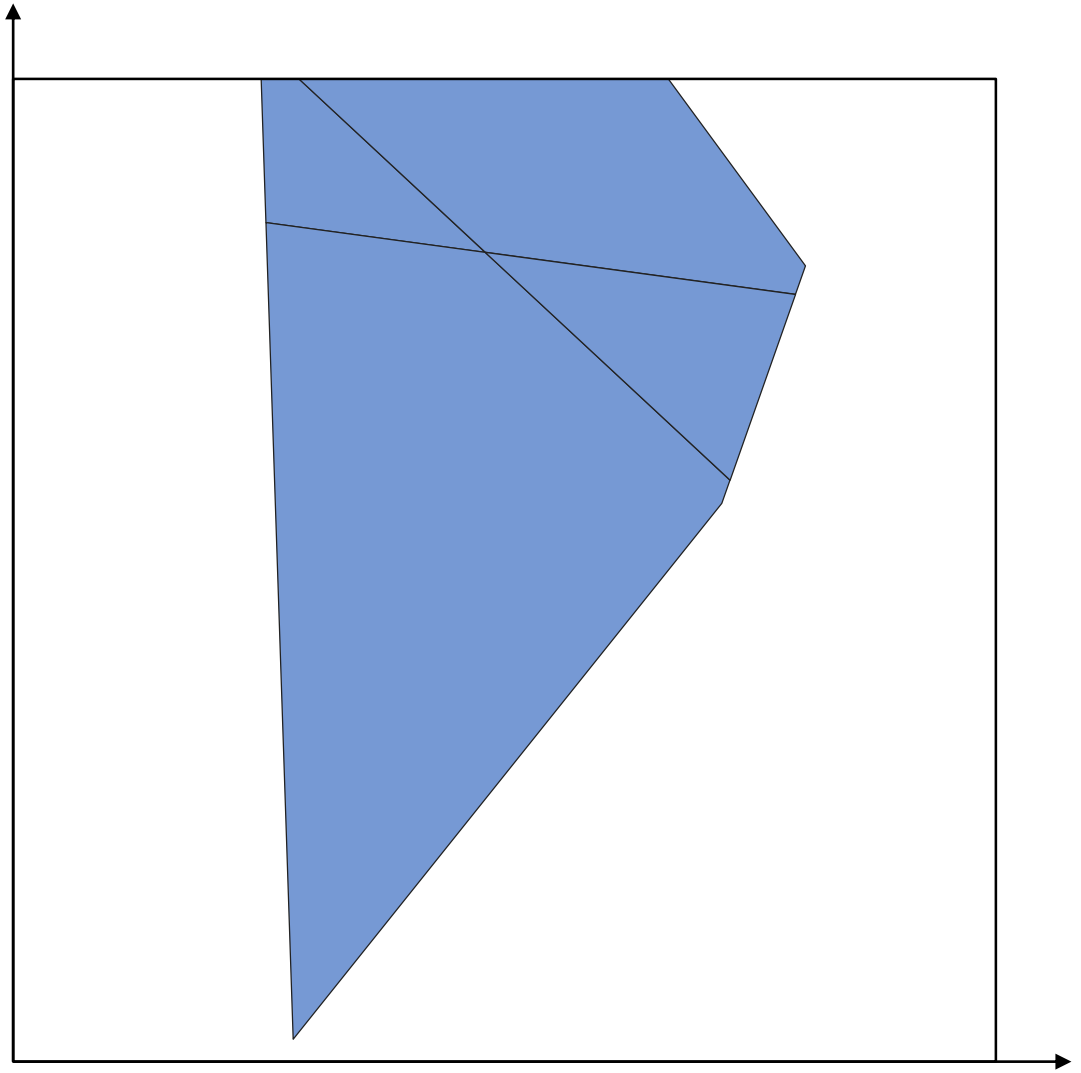




eXBTree: ячейки второго скрытого слоя

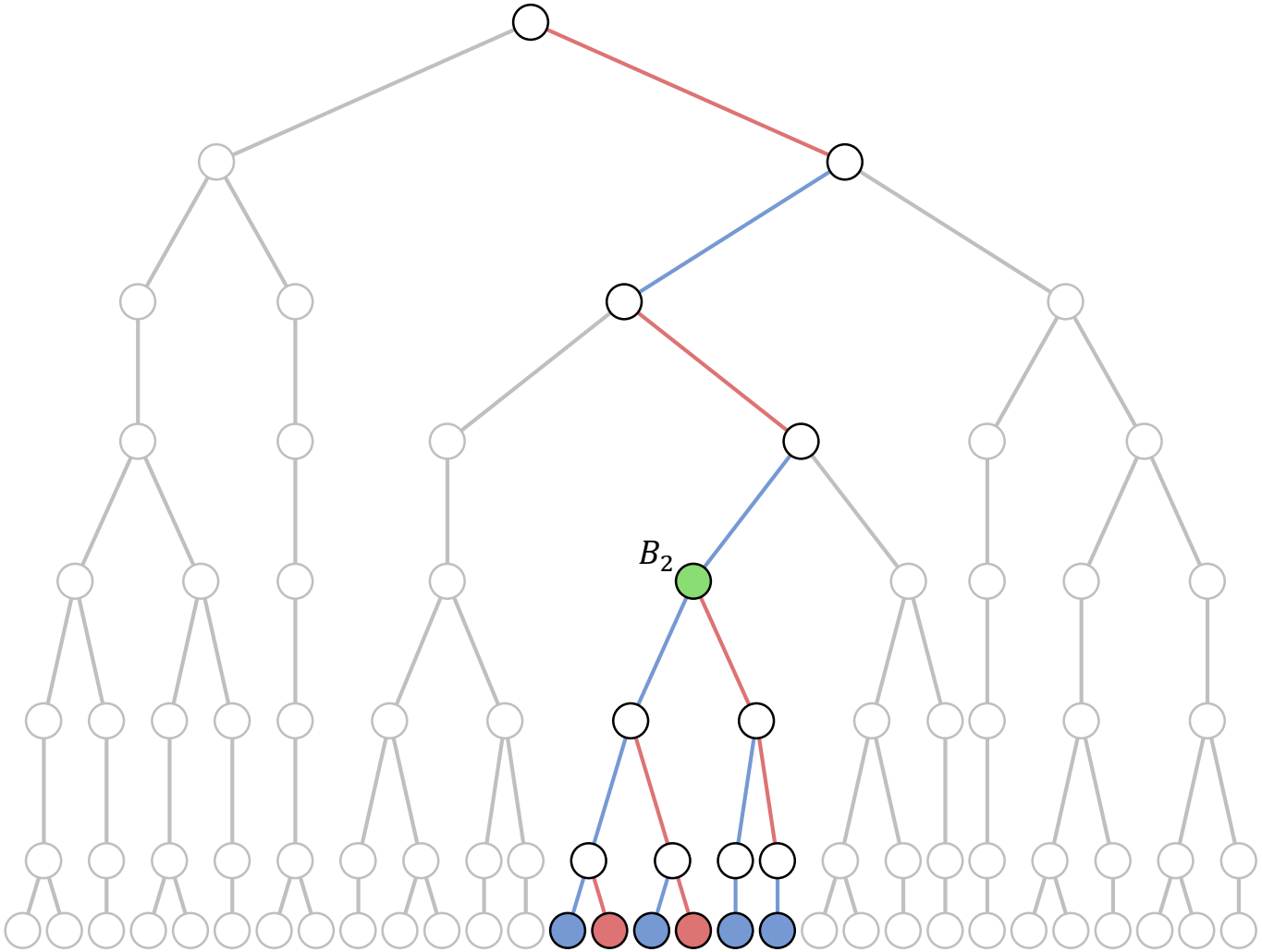
Проследим путь ($+\ -\ +\ -\ -\ +\ -$) по дереву: нейрон $B_1 < 0$

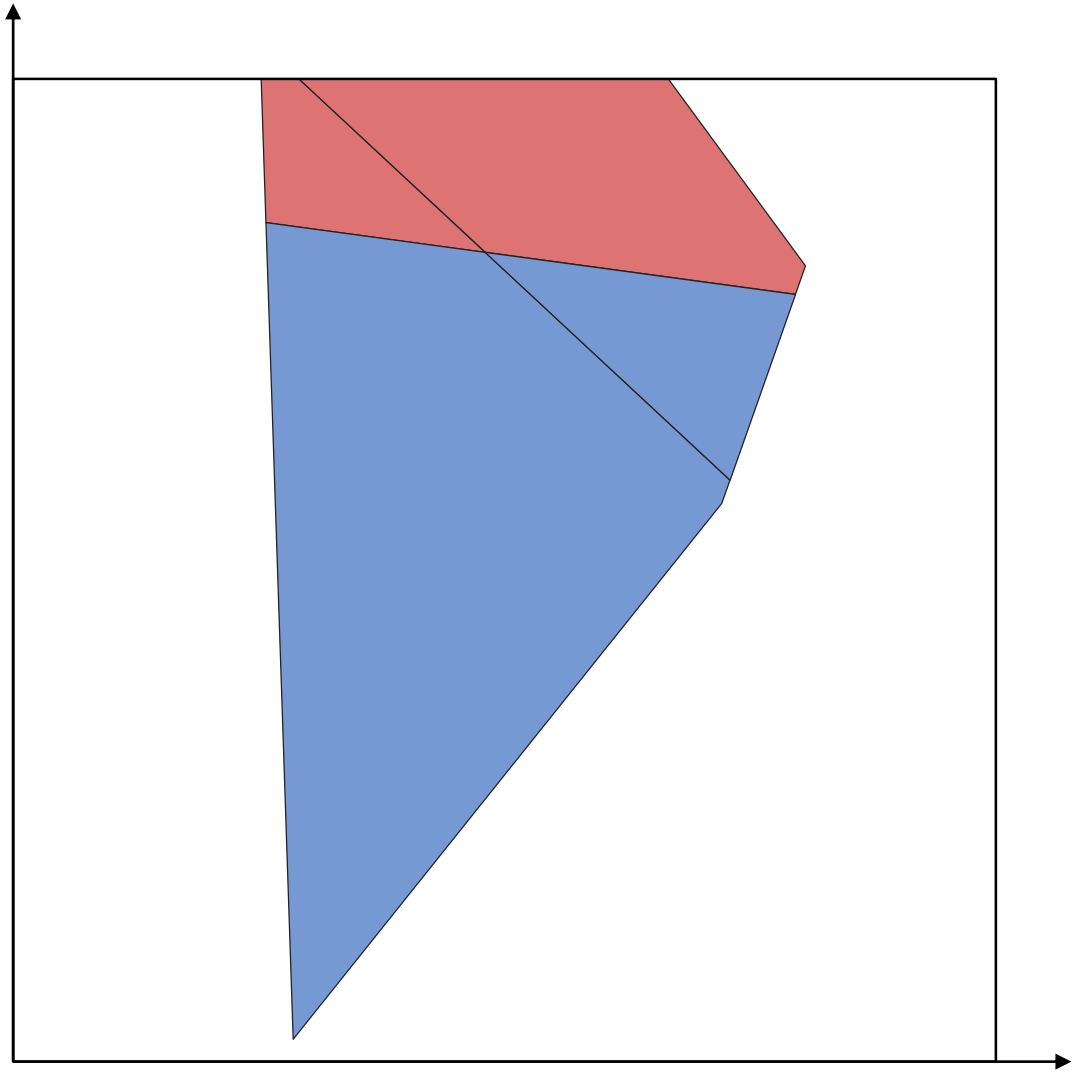




eXBTree: ячейки второго скрытого слоя

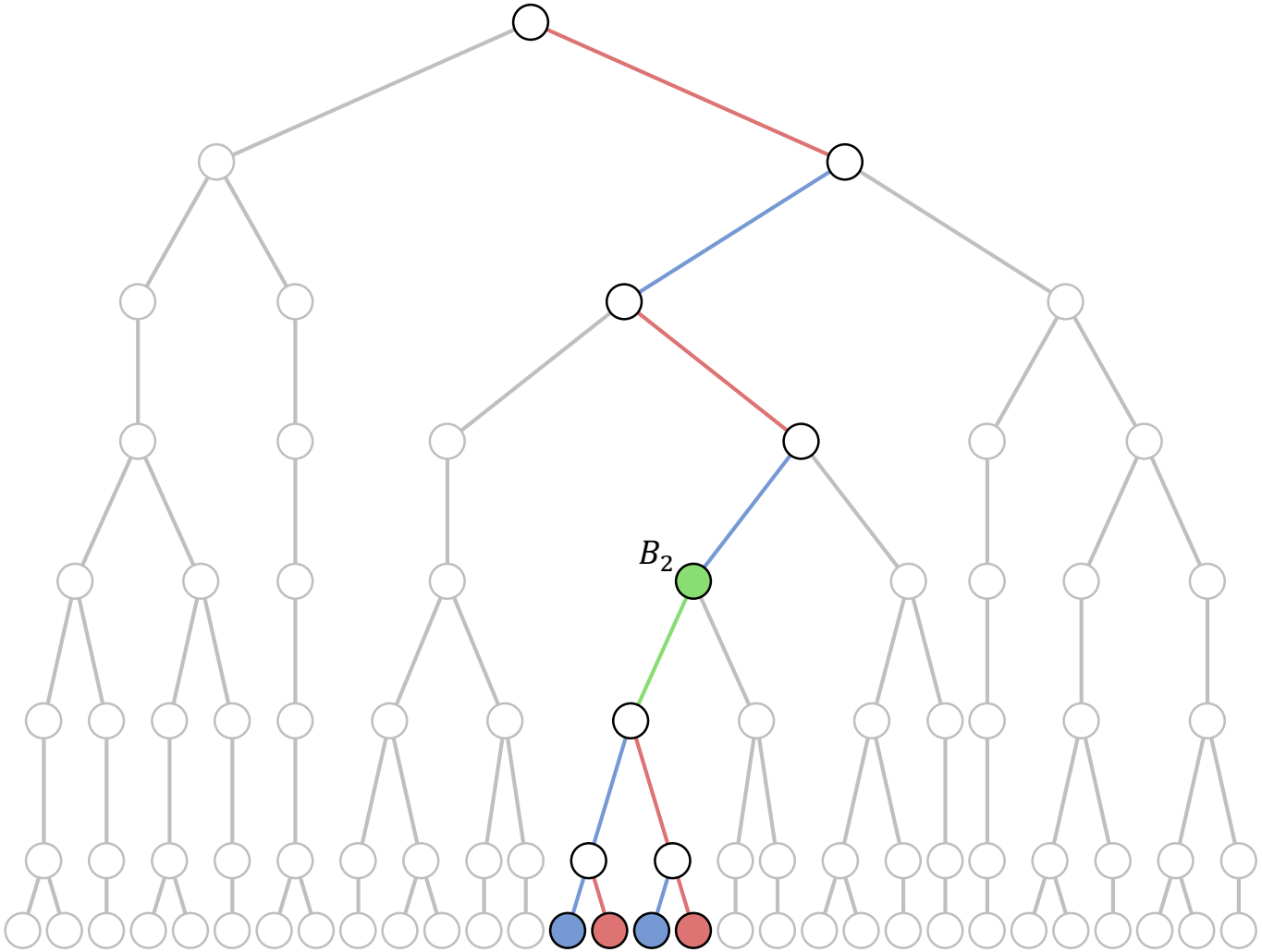
Проследим путь ($+\ -\ +\ -\ -\ +\ -$) по дереву: нейрон B_2

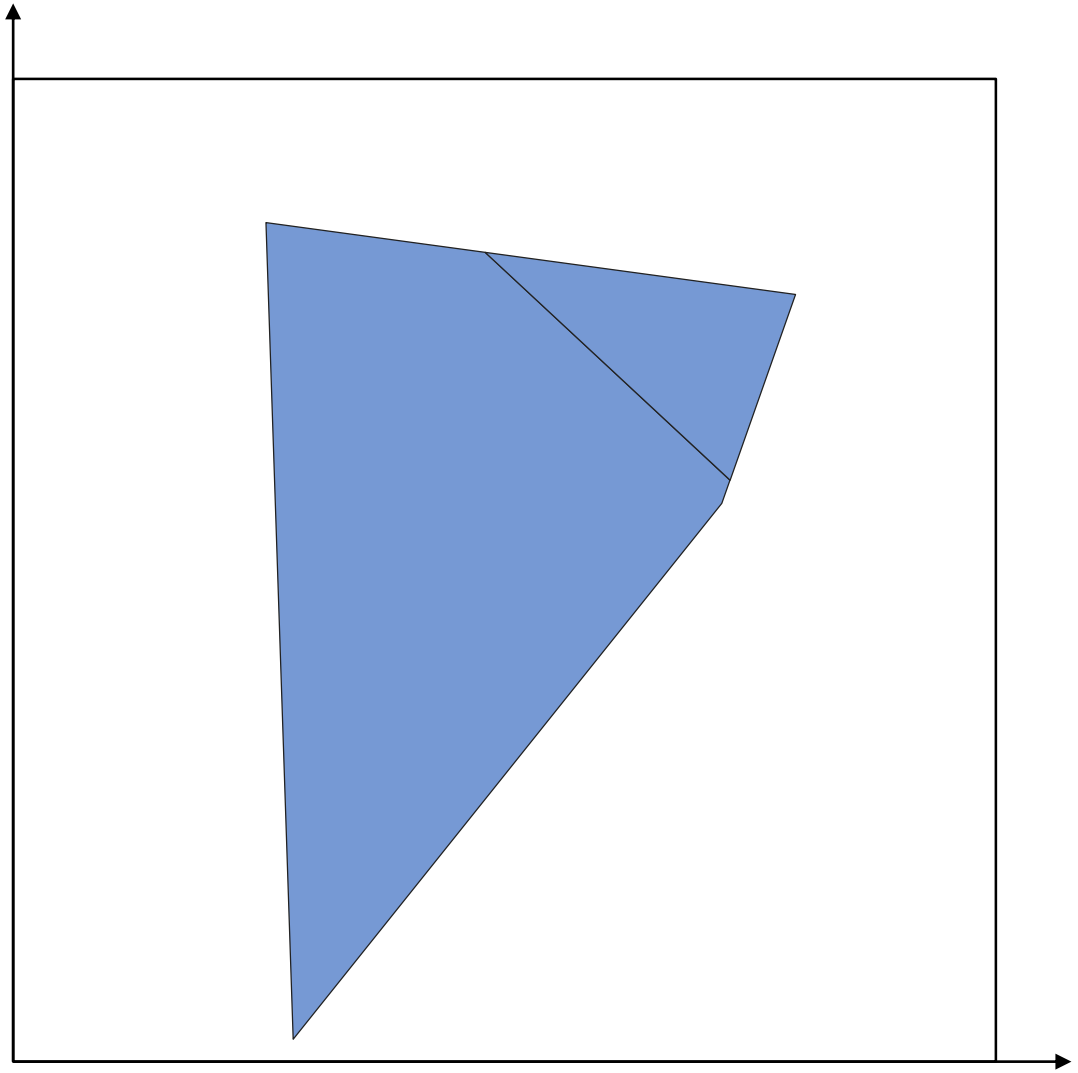




eXBTree: ячейки второго скрытого слоя

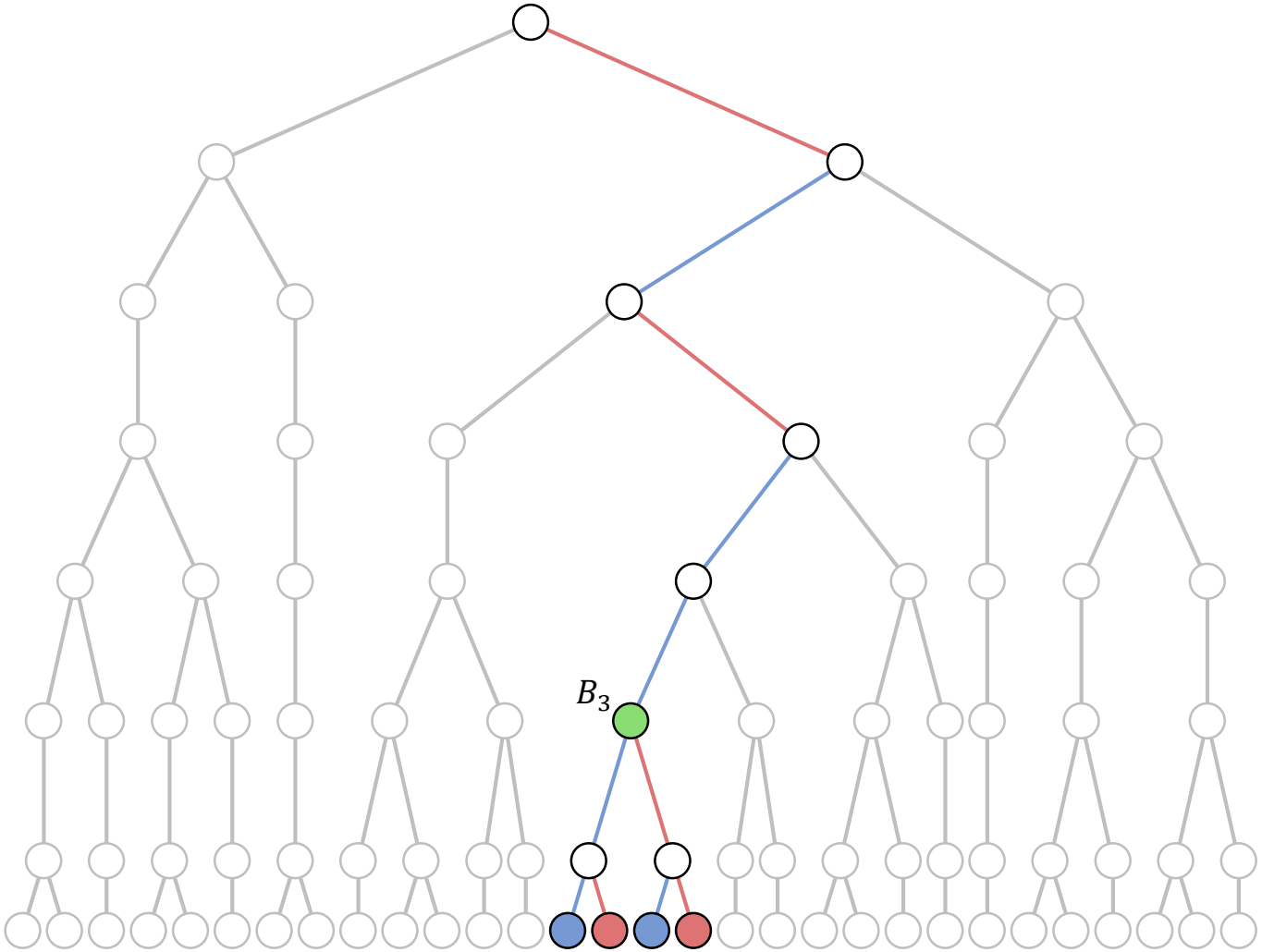
Проследим путь ($+\ -\ +\ -\ -\ +\ -$) по дереву: нейрон $B_2 < 0$

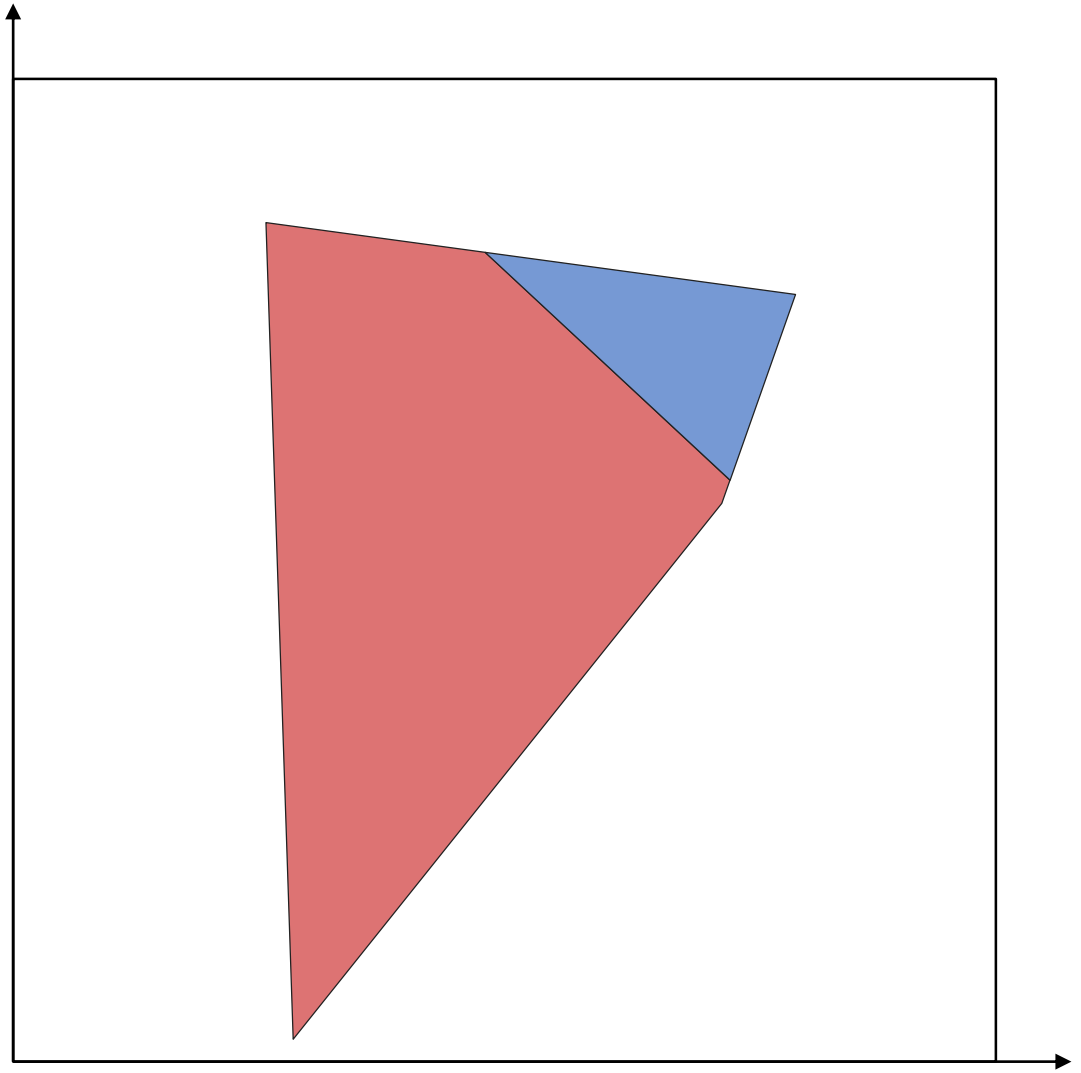




eXBTree: ячейки второго скрытого слоя

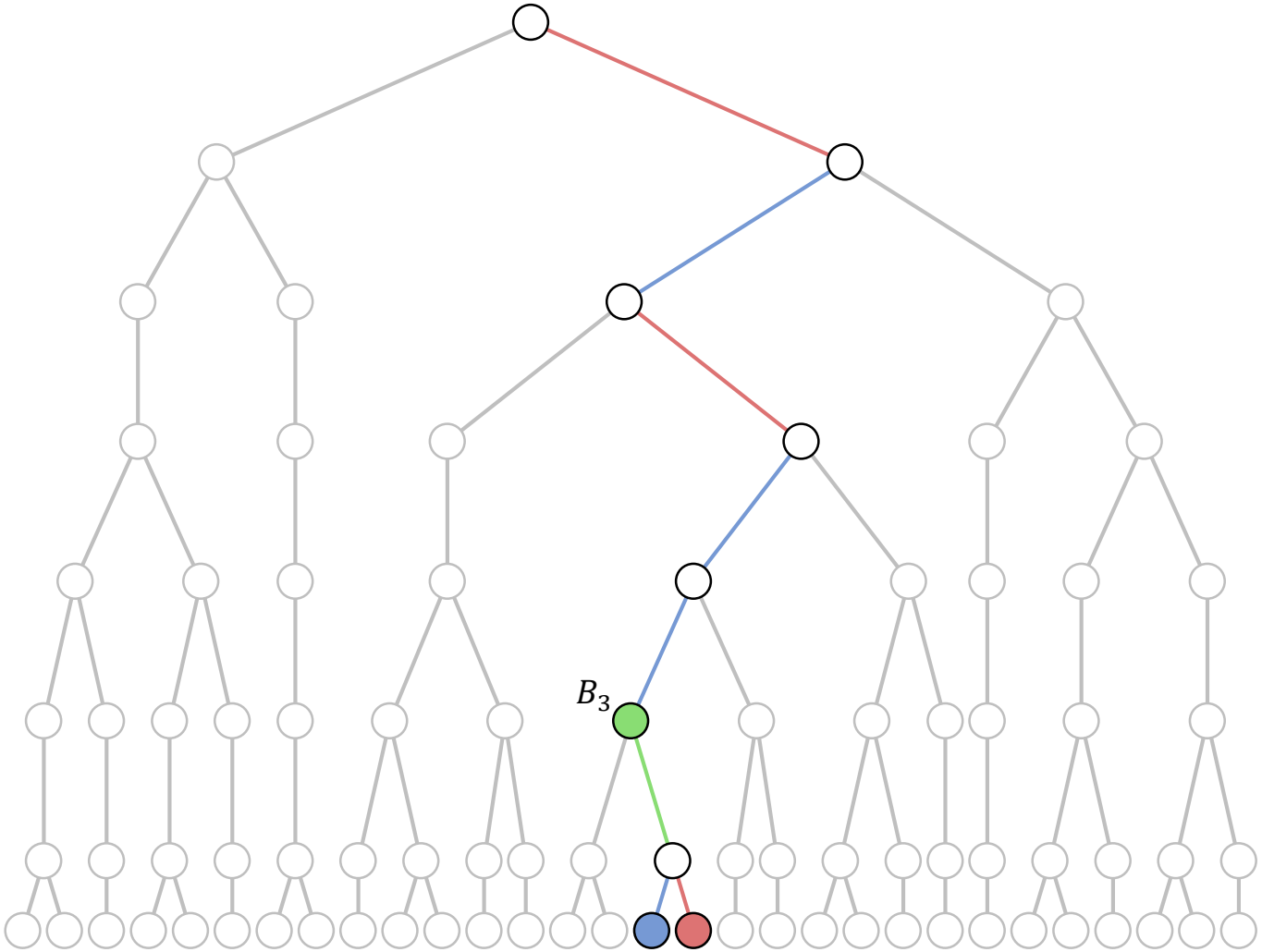
Проследим путь ($+\ -\ +\ -\ -\ +\ -$) по дереву: нейрон B_3

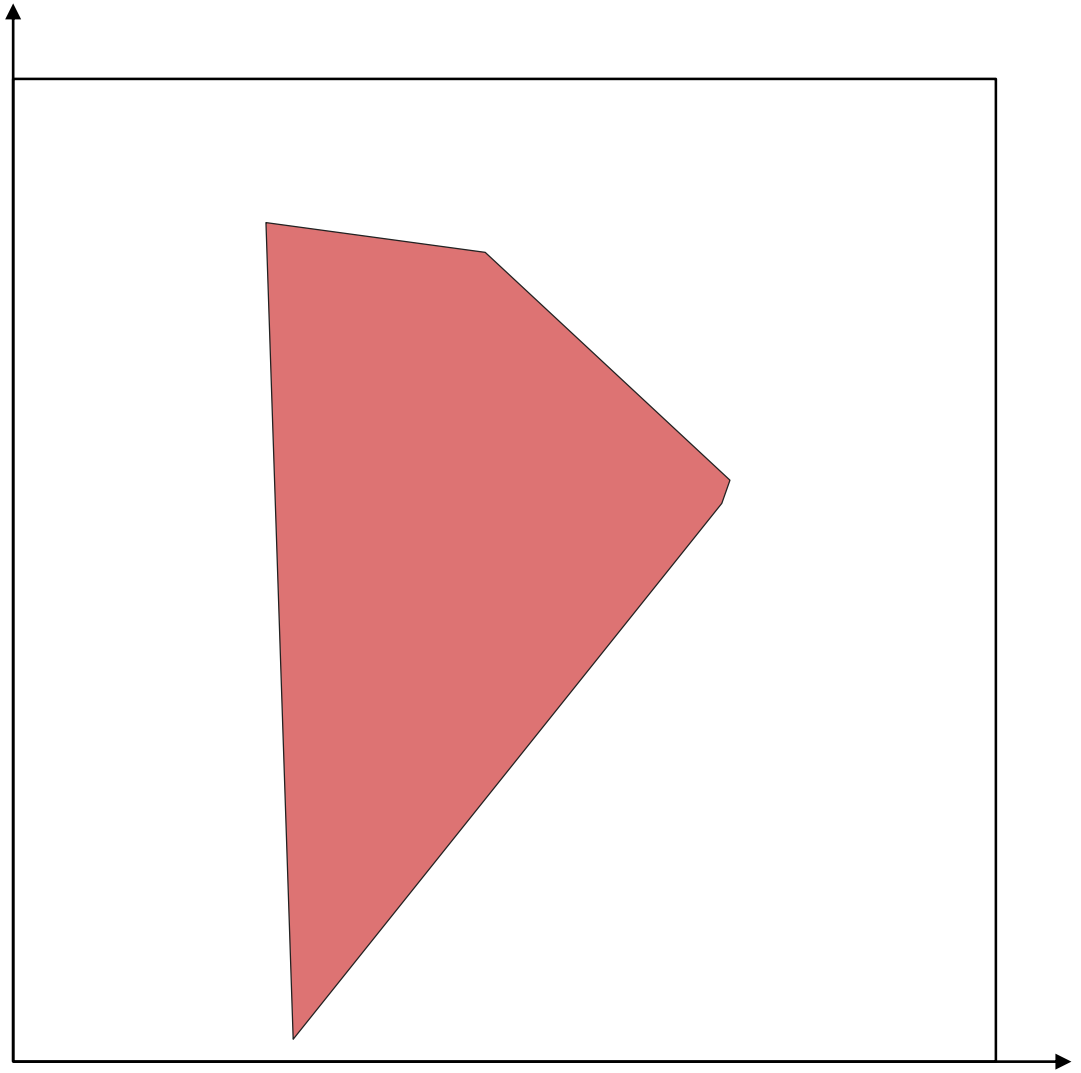




eXBTree: ячейки второго скрытого слоя

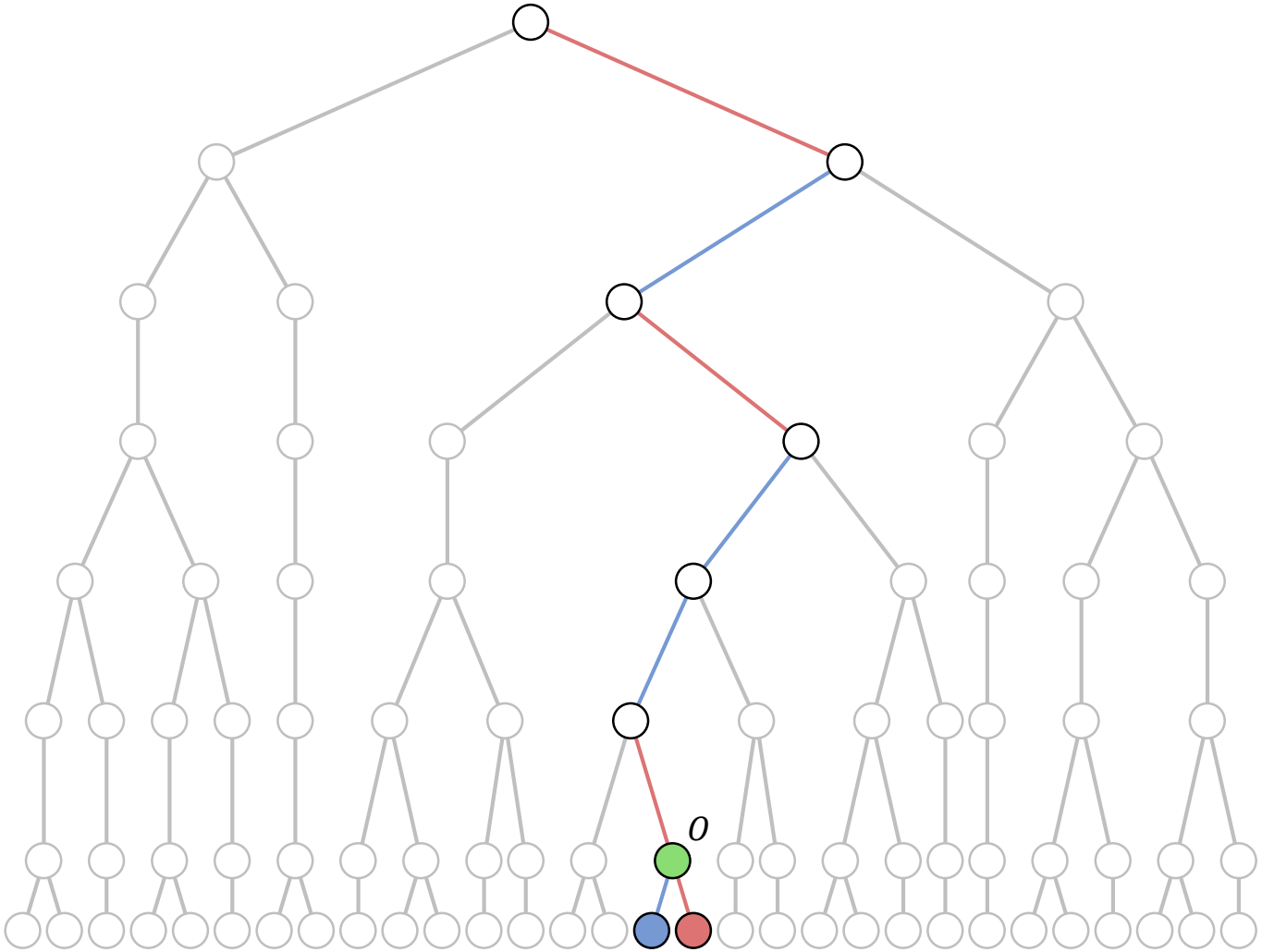
Проследим путь ($+\ -\ +\ -\ -\ +\ -$) по дереву: нейрон $B_3 > 0$

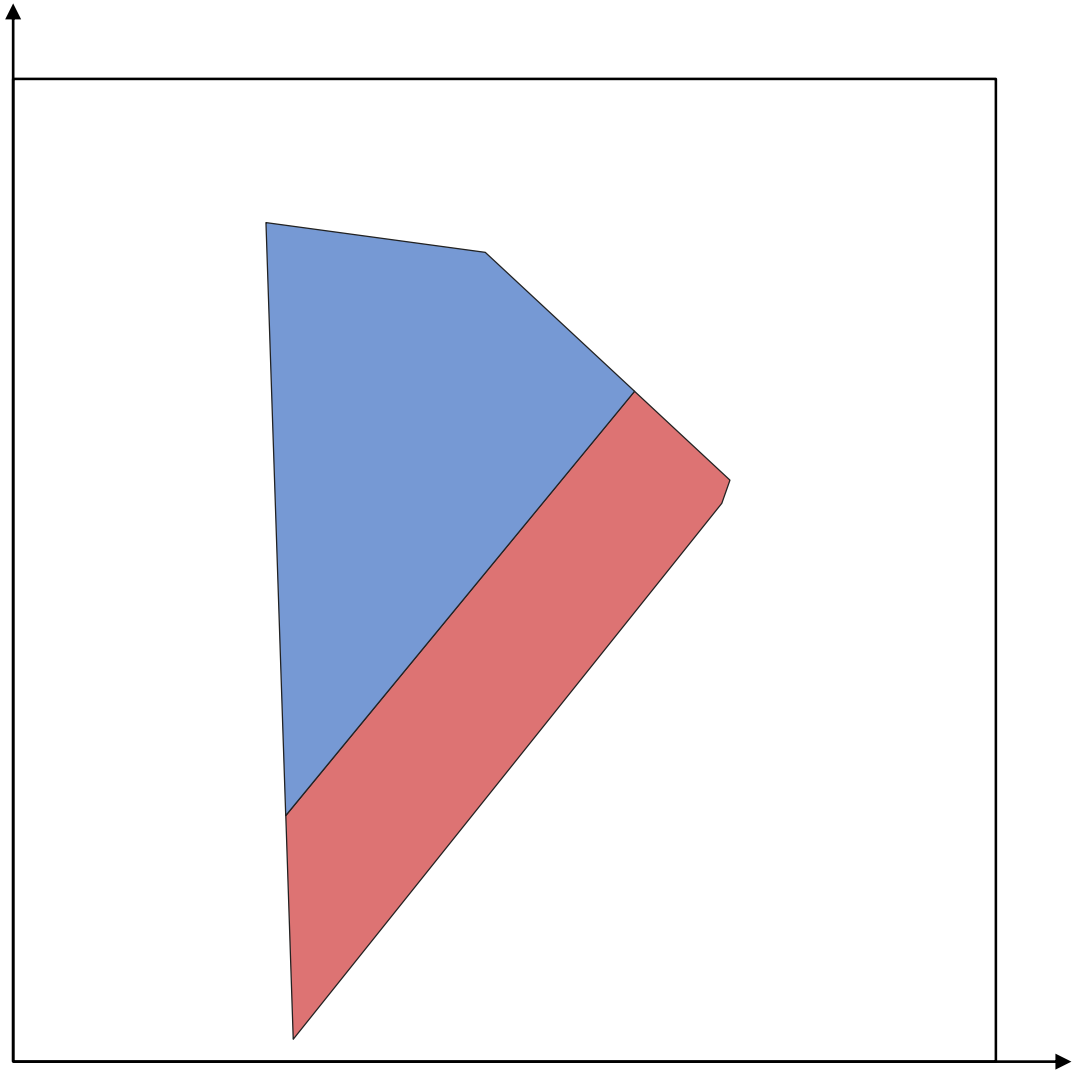




eXBTree: ячейки выходного слоя

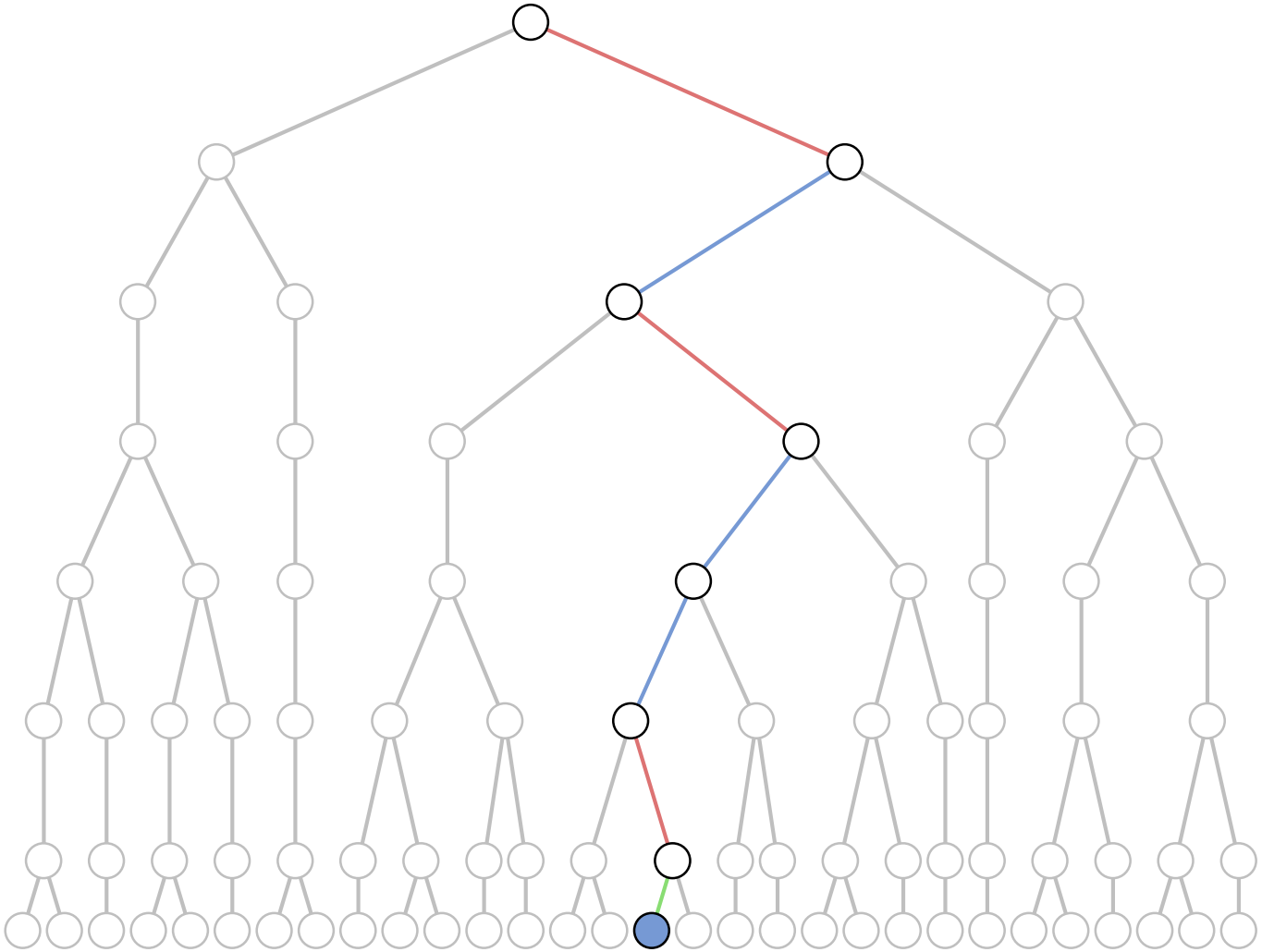
Проследим путь (+ − + − − + −) по дереву: выходной нейрон

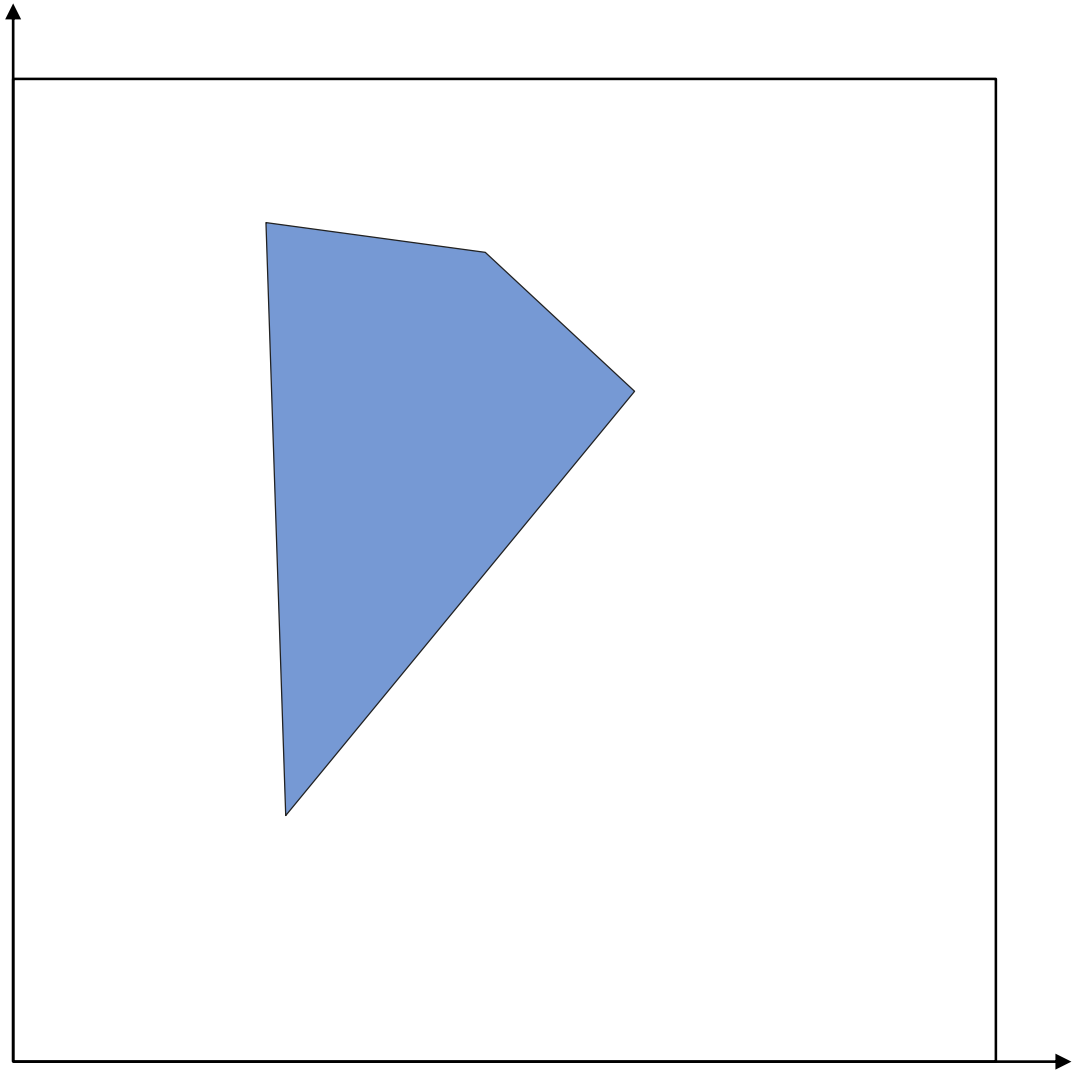




eXBTree: лист дерева

Проследим путь (+ − + − − + −) по дереву: итоговая ячейка





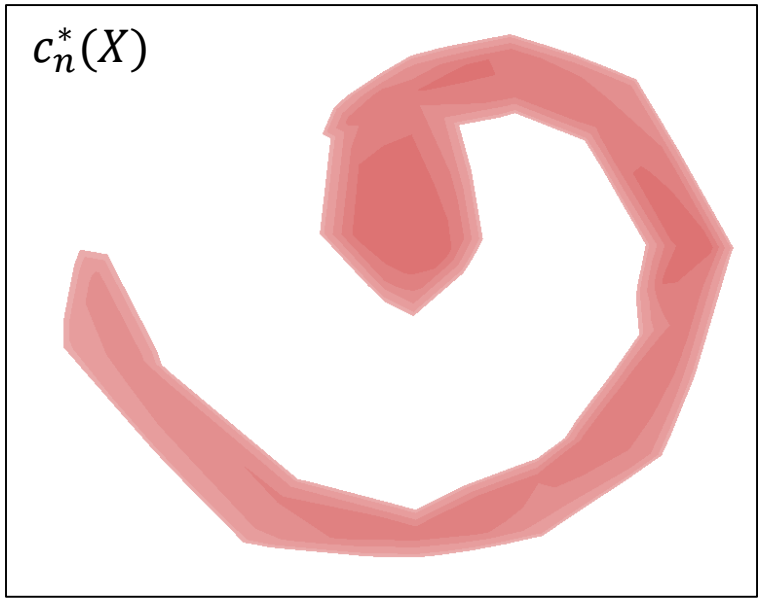
Связь нейросетевой и гистограммной регрессий

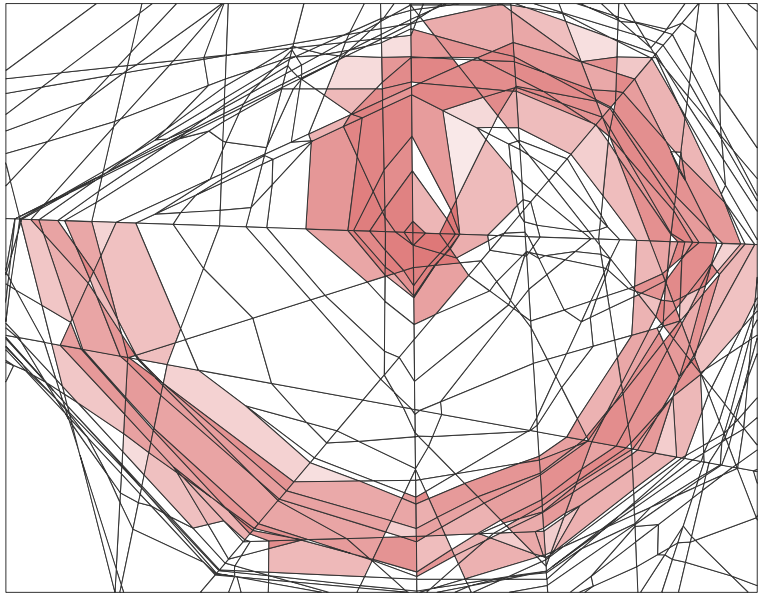
Асимптотические свойства:

При $n \to \infty$ и $kL \to \infty$:

$$\mathbb{E}\{h_n^*(X) - c_n^*(X)\} \to 0$$

Возможна замена гистограммной регрессии на нейросетевую, и при этом MLP вычислительно **эффективнее** и **не хранит** обучающую выборку.





$h_n^*(X)$

Практическое применение:

- Принятие решения с **порогом доверия** β:
$$|c_n^*(X)| > \beta.$$
- Контроль уверенности классификатора.
- Отказ от ответа при низкой апостериорной вероятности.

Состоятельность классификатора

$c_n(x)$ – выборочная оценка апостериорной вероятности $\mathbb{P}(Y = 1|X = x)$.

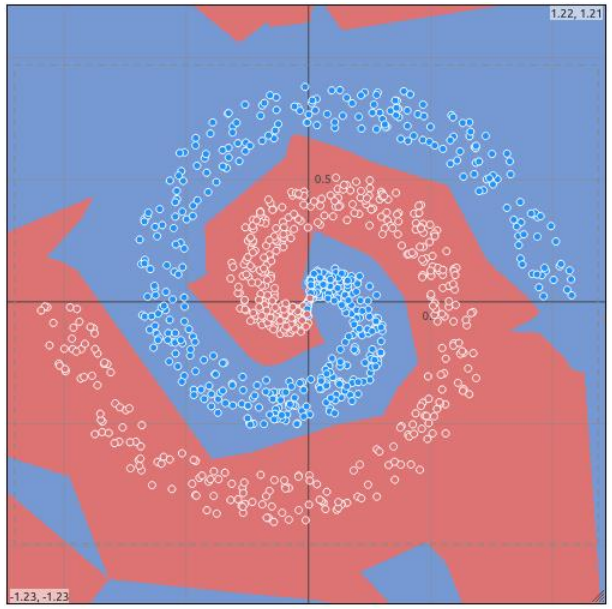
$g(x)$ – истинная байесовская апостериорная вероятность.

Классификатор на основе $c_n(x)$ **состоятельный**, если при $n \to \infty$ его ошибка стремится к минимально возможной (ошибке Байеса):

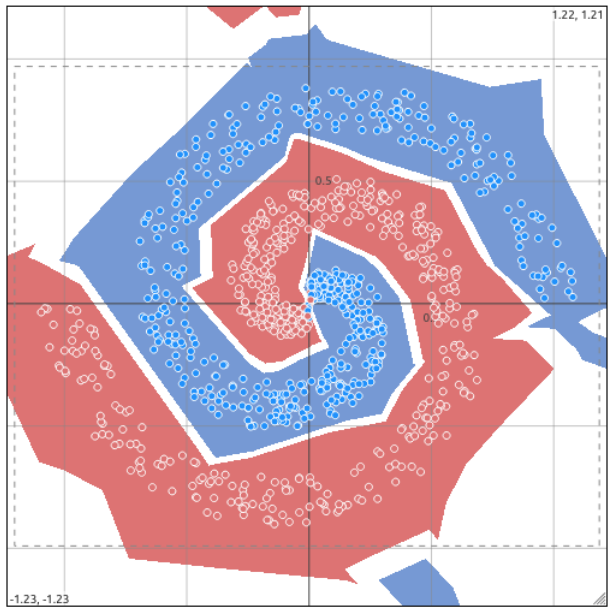
$$c_n(x) \xrightarrow{\mathbb{P}} g(x)$$

Гарантирует, что с ростом данных классификатор приближается к **оптимальному** решению.

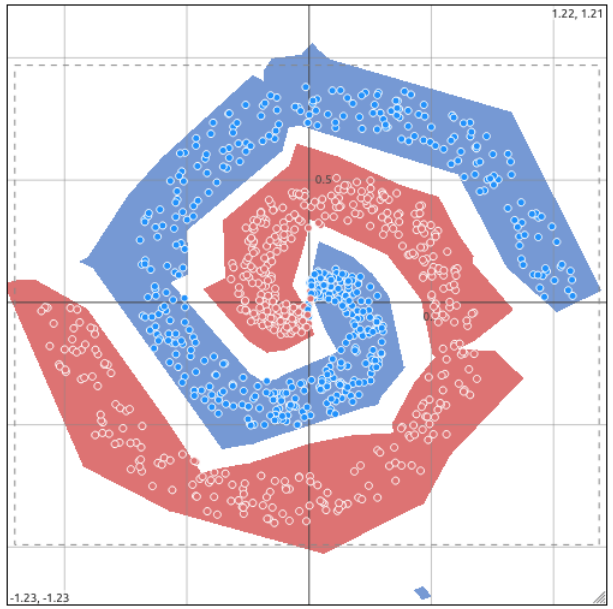
Влияние порога доверия



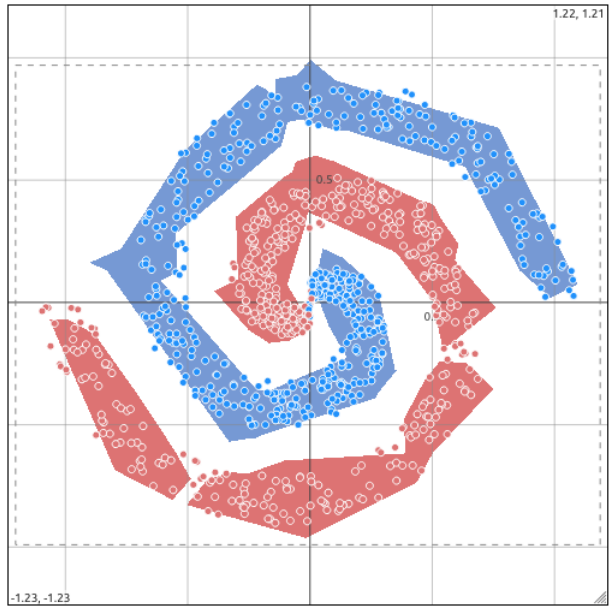
$\beta = 0$



$\beta = 0.1$



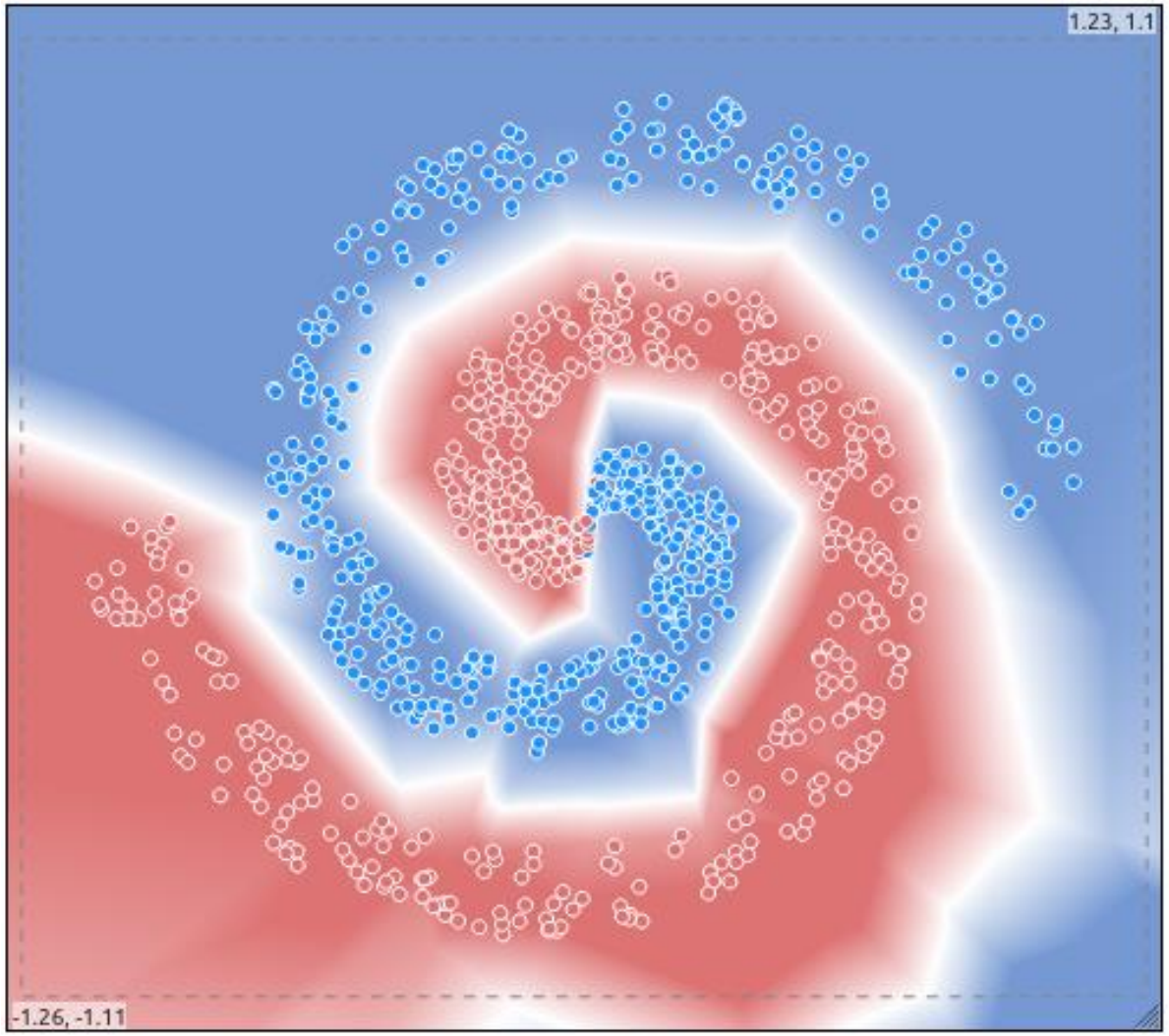
$\beta = 0.3$



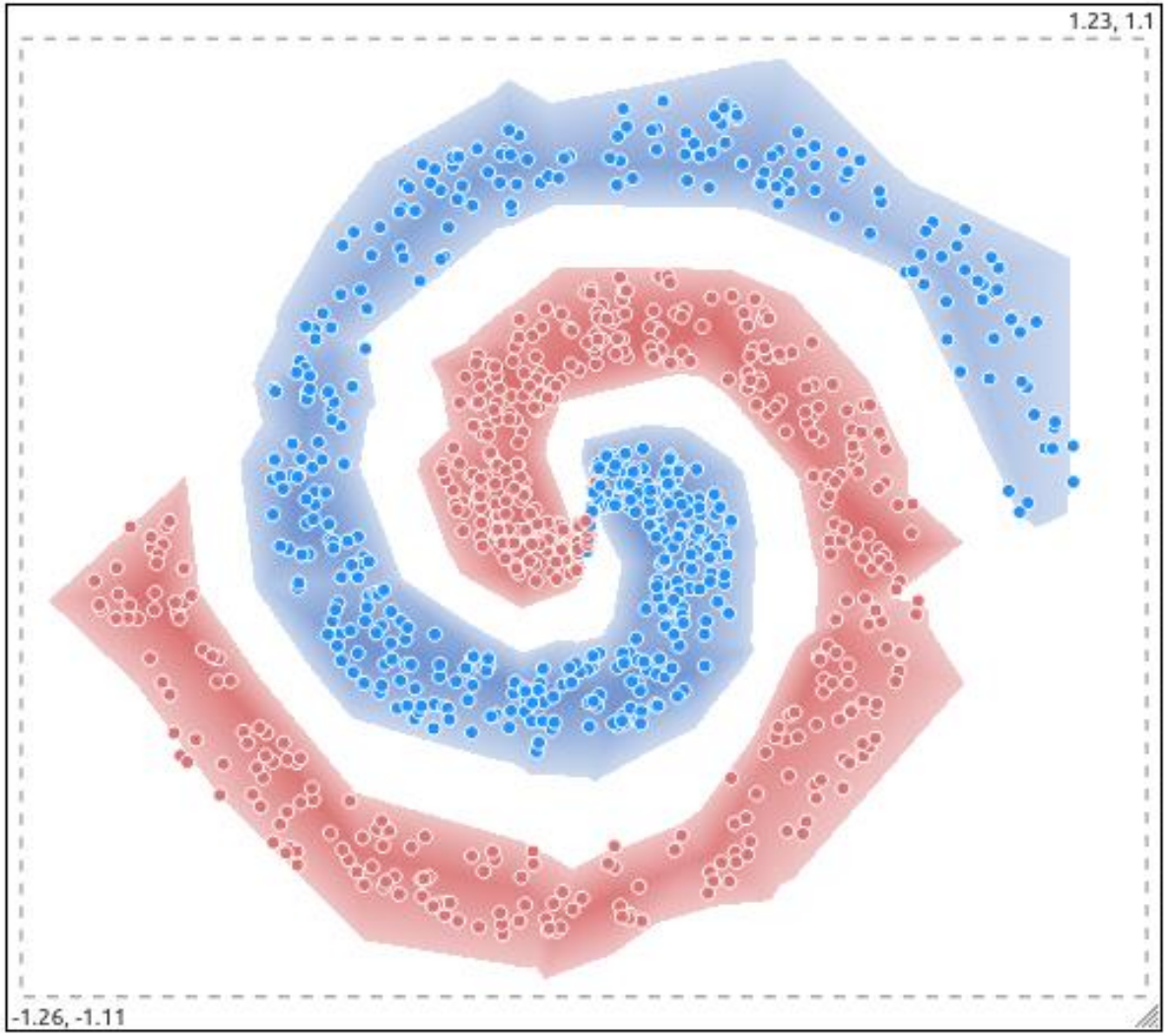
$\beta = 0.5$

При увеличении β область отказов расширяется (обозначена белым цветом), что соответствует желаемому поведению системы в условиях ограниченной уверенности модели.

Поведение вне носителя



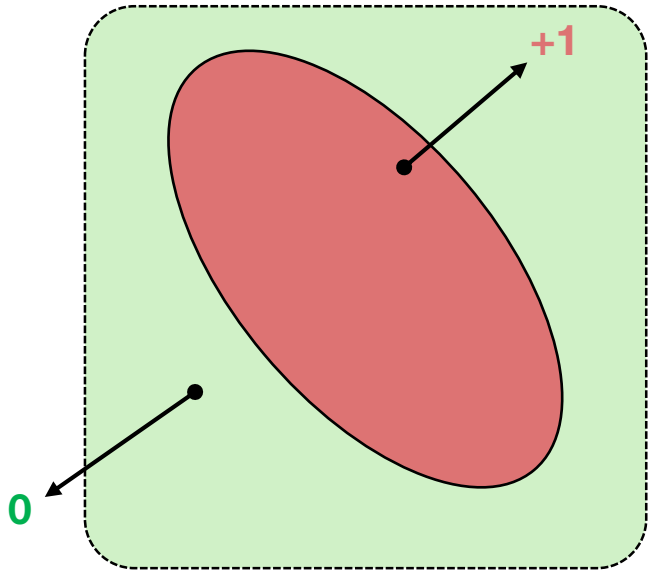
Классический классификатор
(даёт решение в любой точке)



Модифицированный классификатор
(отказывается от классификации)

Метод унарной классификации

1. Оставим только наблюдения класса $+1$ и фоновые.
2. На каждый класс **свой унарный** классификатор $c^{(i)}(X)$.
3. Классификатор является многослойным **персептроном**.
4. Отказ, если предсказание меньше порога β: $c^{(i)}(X) < \beta$.



Преимущества:

01 Устойчивость к дисбалансу

- Обучение на **положительных** объектах класса и фоне в равных пропорциях.
- Исключается **влияние** других классов.
- Устойчивость к **неоднородности** в данных.

02 Векторное предсказание

- Каждый классификатор выдаёт $c^{(i)}(X)$ – оценку принадлежности к классу i.
- Класс с максимальным значением или отказ при

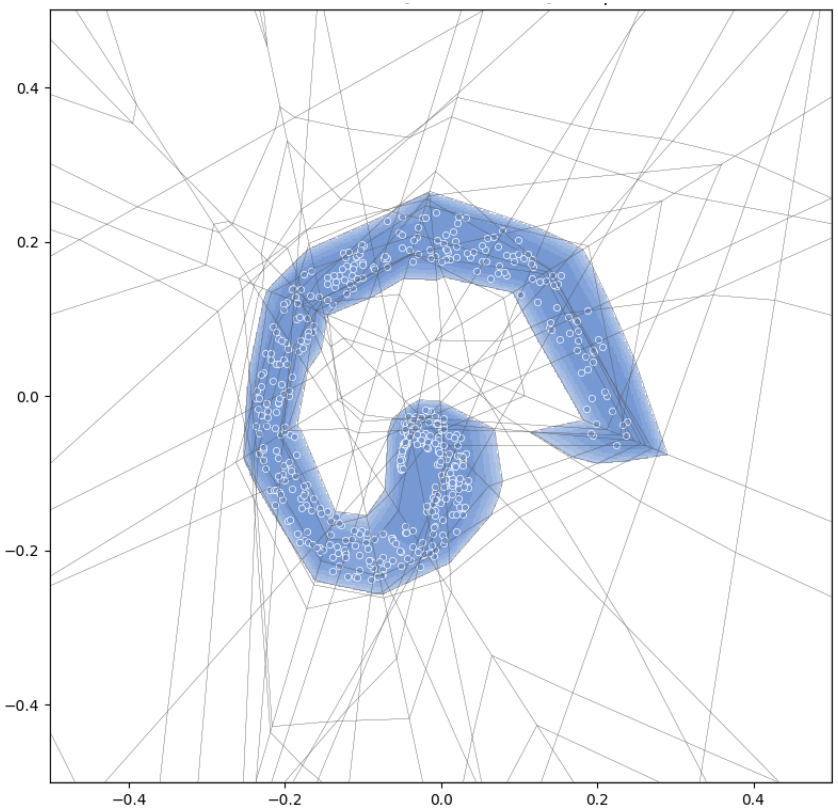
$$\max_{i=1\ldots C} c^{(i)}(x) \leq \beta$$

03 Модульность архитектуры

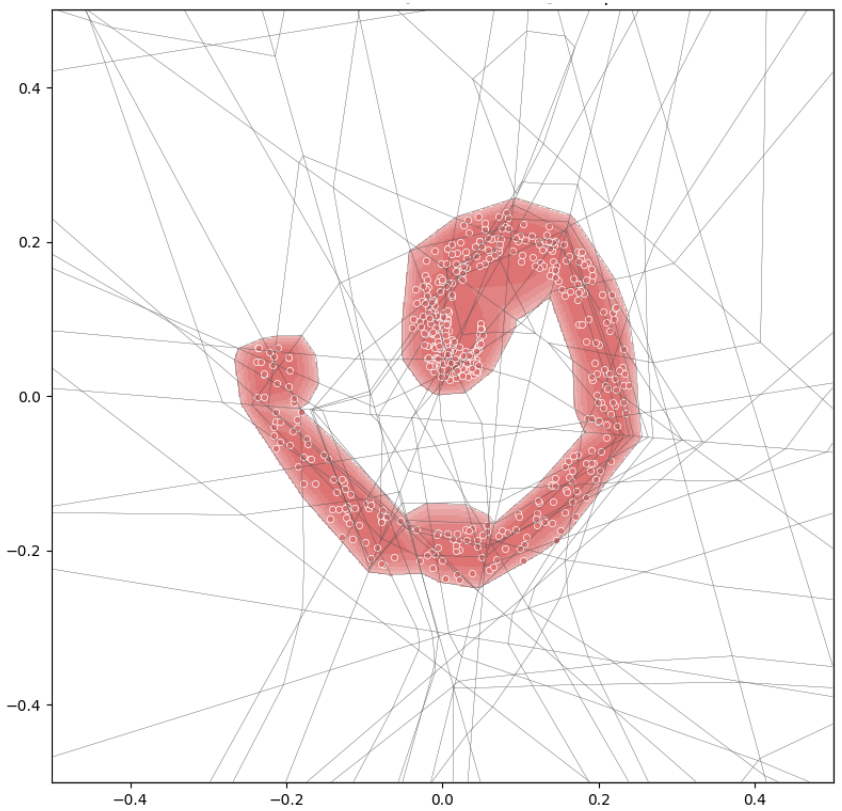
- Классификаторы **независимы**.
- Возможны **разные архитектуры** и глубина для разных классов.
- Адаптация под **особенности** классов.

Ограничения: размерность пространства признаков невелика ($d \leq 30$)

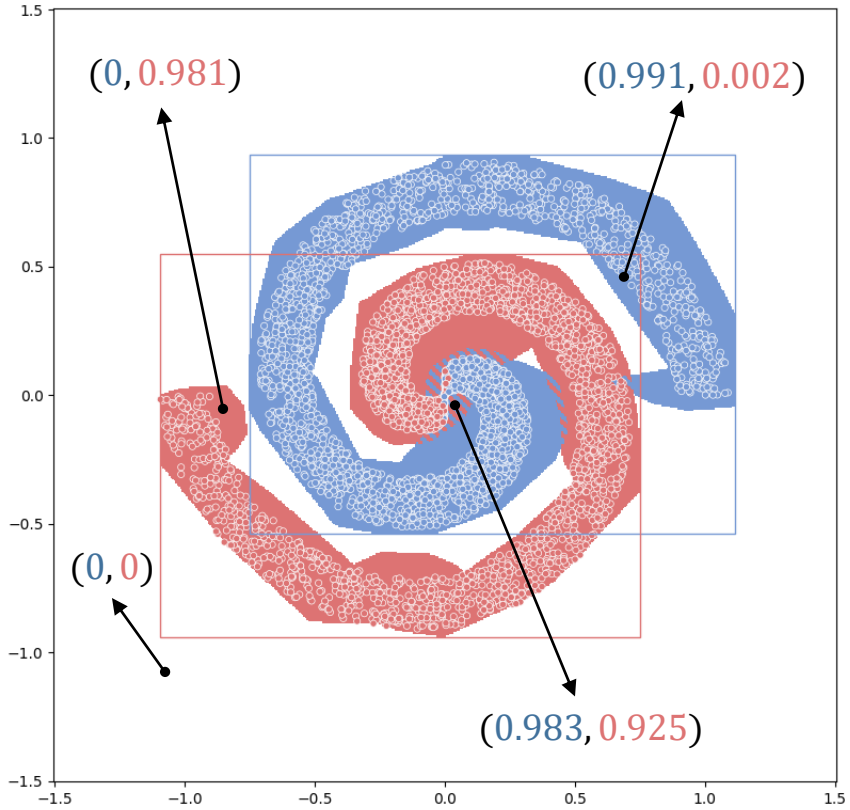
Унарная бинарная классификация



модель для **первого** класса $c^{(1)}(x)$



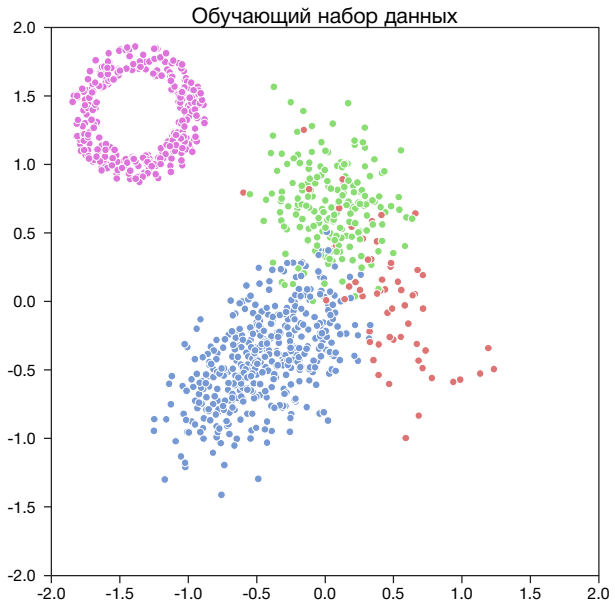
модель для **второго** класса $c^{(2)}(x)$

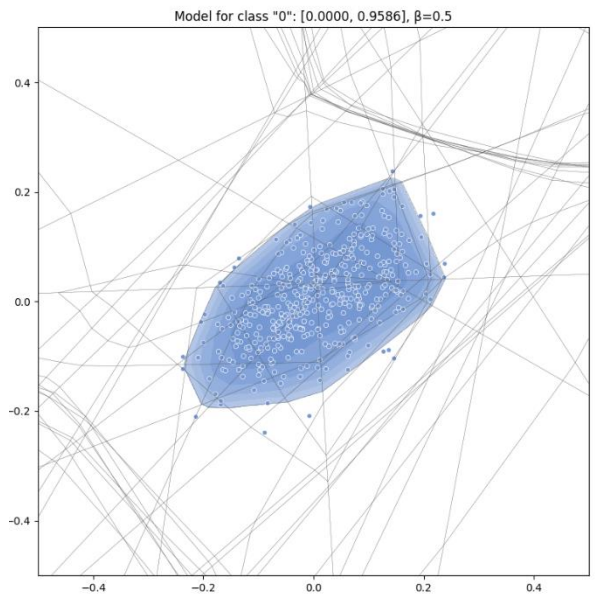


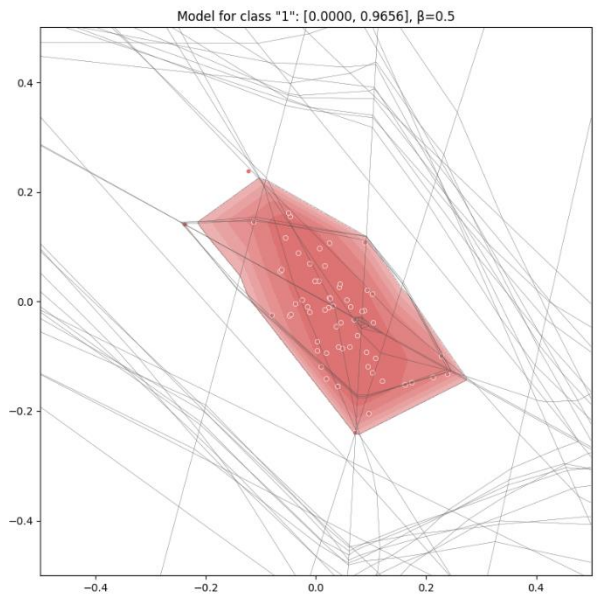
итоговый классификатор $c(x)$

1. Для каждого класса **своя область** принятия решений и **модель**.
2. **Вне области** происходит **отказ** от распознавания.
3. В смеси распределений требуется **дополнительный анализ**.

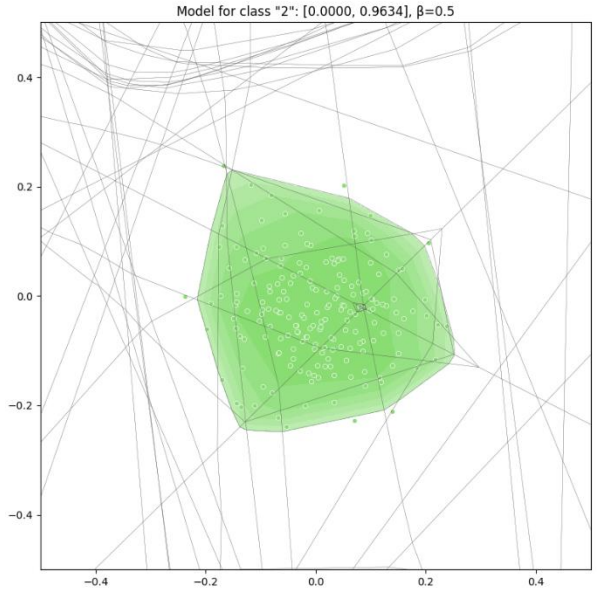
Унарная классификация: устойчивость к дисбалансу

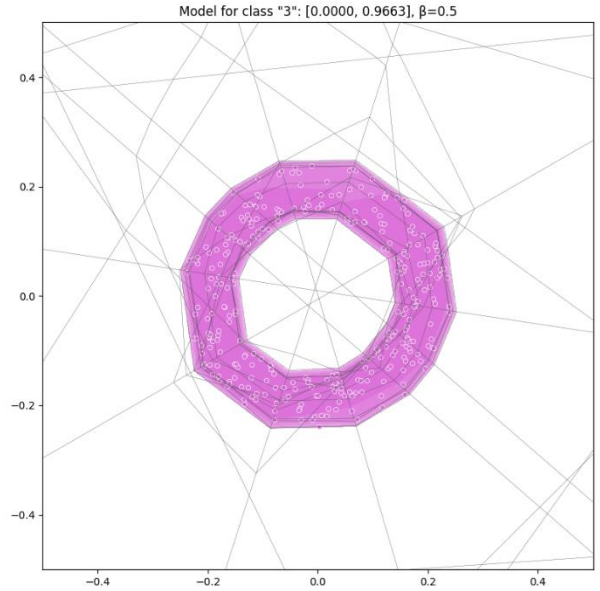


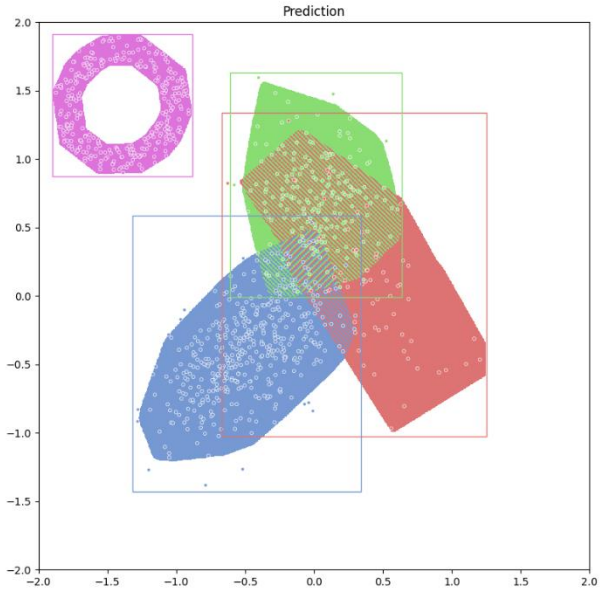




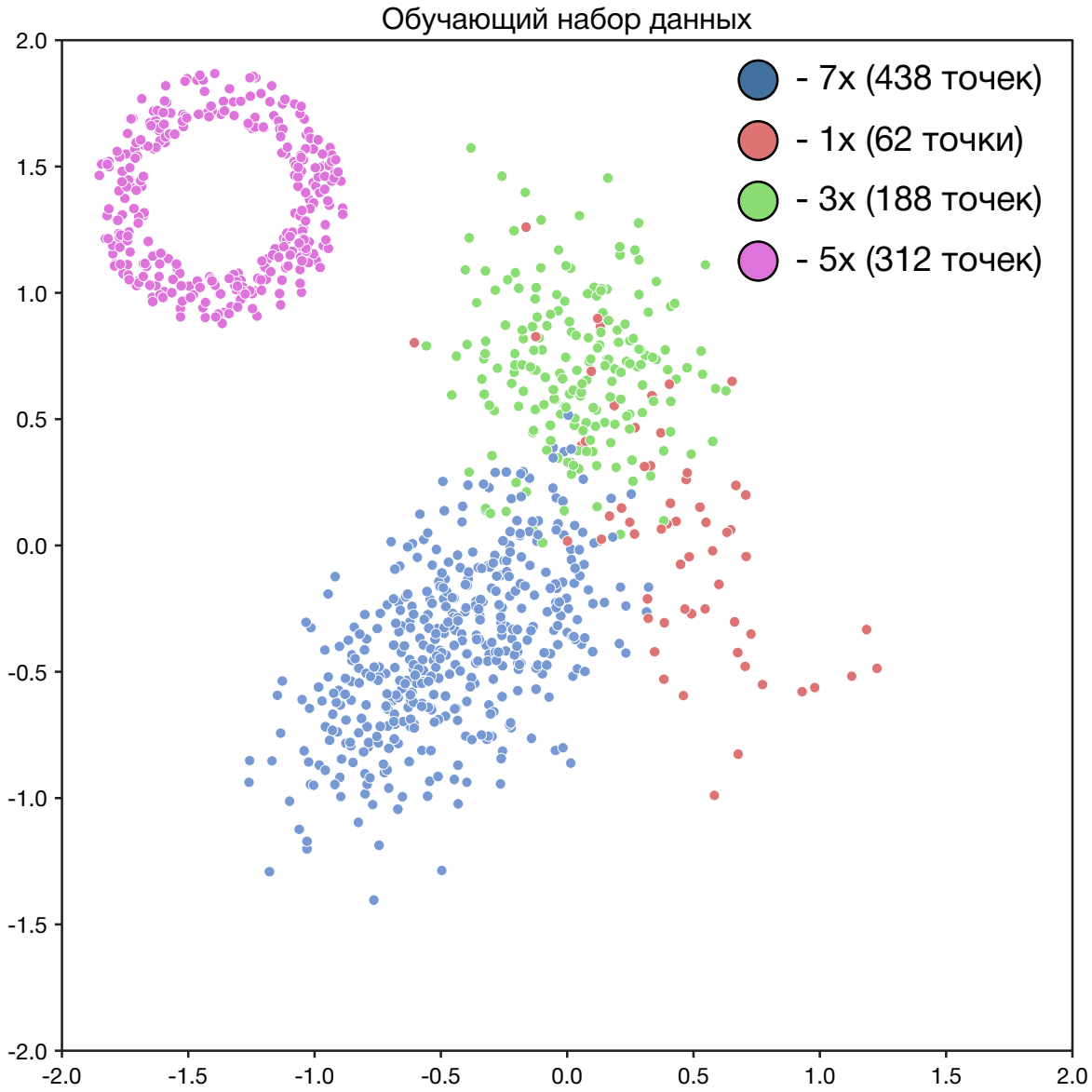
- 7x (438 точек)
- 1x (62 точки)
- 3x (188 точек)
- 5x (312 точек)

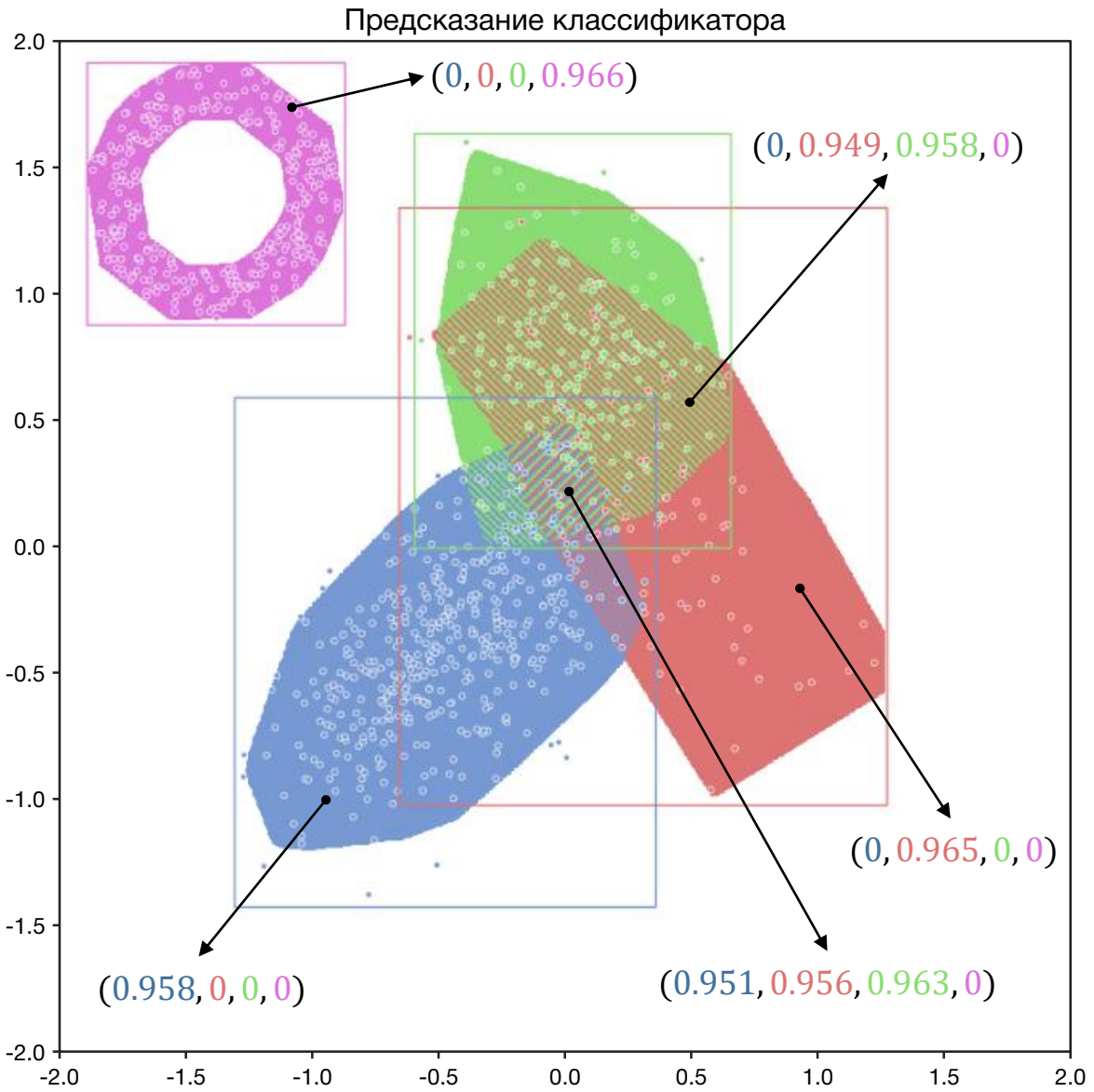






Унарная классификация: устойчивость к дисбалансу





Унарная классификация: проблемы

Проклятие размерности ограничивает применимость для больших размерностей пространства.

При увеличении размерности пространства число ячеек N увеличивается существенно быстрее, чем растёт объём обучающей выборки n, доминируют «одноэлементные» и «пустые» ячейки, понятие **частоты** просто **исчезает**.

При $d = 10$ необходимы **миллионы точек фона**, чтобы в достаточной мере заполнить компакт.

Можно ли как-то побороть эту проблему?

SLAP – Simple Linear Attack for Perceptron

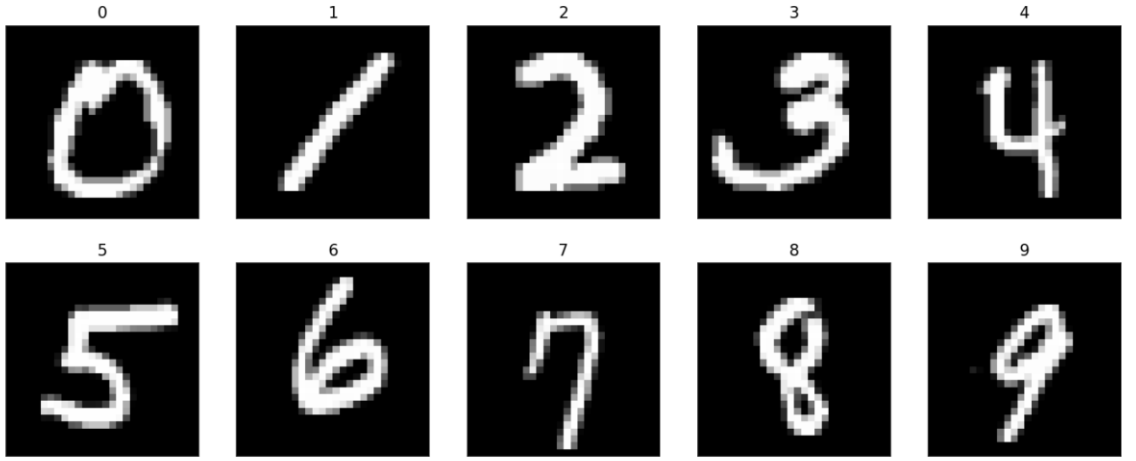
Простая линейная атака на персептрон

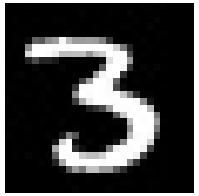
- Cat – 2%
- Dog – 98%

Постановка задачи

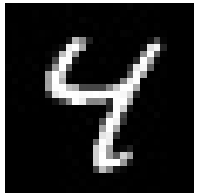
Что имеем?

- полносвязная сеть для классификации изображений с **кусочно-линейными** функциями активации ($ReLU$, Abs, …);
- пара изображений – целевое ($\boldsymbol{x_t}$) и атакуемое ($\boldsymbol{x_a}$) (в виде векторов);
- магия линейной алгебры.





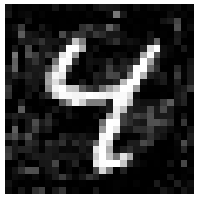
Целевое изображение – $\boldsymbol{x_t}$
Выход модели – $\boldsymbol{y_t}$: $x_1, x_2, x_3, \ldots$



Атакуемое изображение – $\boldsymbol{x_a}$
Выход модели – $\boldsymbol{y_a}$: $y_1, y_2, y_3, \ldots$

Что хотим получить?

- изображение ($\boldsymbol{x}$), имеющее тот же выход сети (или тот же класс) как и целевое, но похожее на атакуемое.



Атакующее изображение – $\boldsymbol{x}$
Выход модели – $\boldsymbol{y}$: $x_1, x_2, x_3, \ldots$

Атака: один слой

Рассматриваем первый слой сети, количество выходов которого меньше, чем входов: $\boldsymbol{y = Wx + b}$

$$
\begin{aligned}
w_{1,1}x_1 + w_{1,2}x_2 + \cdots + w_{1,d}x_d + b_1 &= y_1 \\
w_{2,1}x_1 + w_{2,2}x_2 + \cdots + w_{2,d}x_d + b_2 &= y_2 \\
&\cdots \\
w_{C,1}x_1 + w_{C,2}x_2 + \cdots + w_{C,d}x_d + b_C &= y_C
\end{aligned}
$$

- $\boldsymbol{d}$ – размер изображения, $\boldsymbol{C}$ – количество классов (и размер слоя)
- $\boldsymbol{W}$ – матрица весов ($C \times d$), $\boldsymbol{b}$ – вектор смещений ($C \times 1$)
- $\boldsymbol{x_t}$ – целевое изображение ($d \times 1$)
- $\boldsymbol{x_a}$ – атакуемое изображение ($d \times 1$)
- $\boldsymbol{x}$ – атакующее изображение ($d \times 1$)
- $\boldsymbol{y_t}$ – выход слоя ($C \times 1$)

Атака: один слой

Решаем **СЛАУ** относительно коэффициентов первого слоя и ищем решения, удовлетворяющие следующим условиям:

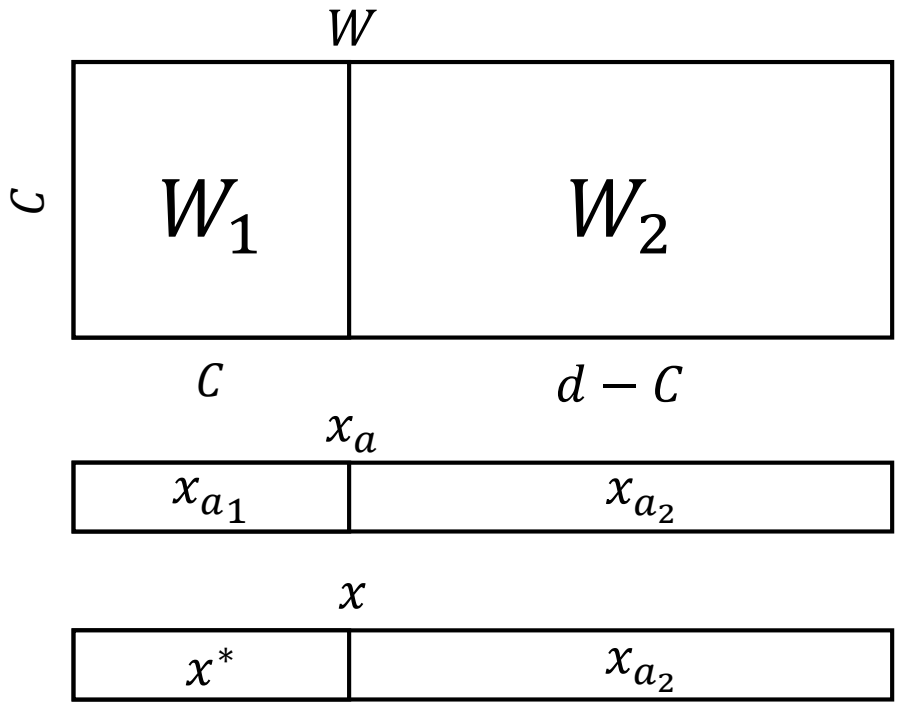
- $Wx + b = y_t$ (равенство выходов сети)
- $\|x_a - x\| \to \min$ (максимальная похожесть на атакуемое)
- $\|x_t - x\| > 0$ (**в идеале** не получить целевое)
- $0 \le x_i \le 1$ (**в идеале** остаться в диапазоне значений)

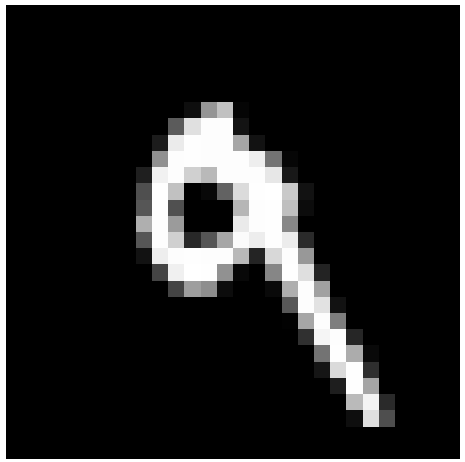
Альтернативная цель: равенство классов

- $argmax(Wx + b) = argmax(y_t)$

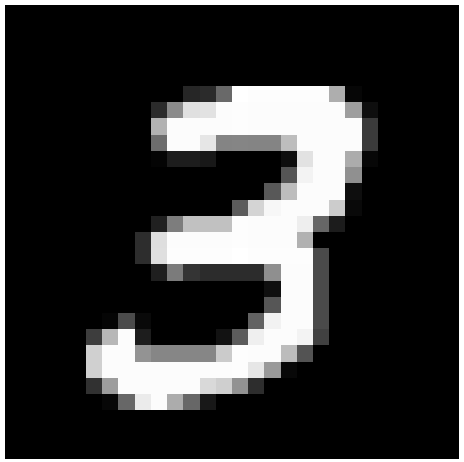
Атака: один слой, без учёта входного диапазона

- выберем подматрицу W_1 размера $C \times C$ (первые или случайные столбцы);
- оставшуюся подматрицу размера $C \times (d - C)$ будем называть W_2;
- аналогично разделим x_a на x_{a_1} и x_{a_2};
- первые C значений атаки получим как $x^* = W_1^{-1} \cdot \left(b^T - W_2 \cdot x_{a_2}\right)$;
- атакующее изображение получается **конкатенацией** x^* и x_{a_2}.





$\boldsymbol{x_t} \in [0, 1]$



$\boldsymbol{x_a} \in [0, 1]$

$\boldsymbol{x} \in [-68, 82]$

Атака: один слой, без учёта входного диапазона

Достоинства:

- Всегда работает (вероятность необратимости W_1 крайне мала);
- Минимально заметна (работает исключительно в пределах тензора).

Недостатки:

- Применима только при доступе к тензорам;
- Не учитывается диапазон входных значений;
- Не воспроизводима при сохранении x (может быть **достоинством**, поскольку визуально не будет видно заметных дефектов).

$\boldsymbol{x_t} \in [0, 1]$

$\boldsymbol{x_a} \in [0, 1]$

$\boldsymbol{x} \in [-938, 919]$

Атака: один слой – QP решатель

QP task

$$\begin{cases} \frac{1}{2}x^T Px + q^T x \to min \\ Ax = b \\ Gx \leq h \\ l_b \leq x_i \leq u_b \end{cases}$$

$$\begin{gathered} P = E, q = -x_a \\ A = W, b = y_t - b \\ G = 0, h = 0 \\ l_b = 0, u_b = 1 \end{gathered}$$

QP attack

$$\begin{cases} \frac{1}{2}x^T x - x_a^T x \to min \\ Wx = y - b \\ 0 \leq x_i \leq 1 \end{cases}$$

Достоинства:

- Сохранение «не убивает» атаку;
- Учитывается входной диапазон.

Недостатки:

- Не всегда применима;
- Требуется вносить больше искажений.

$\boldsymbol{x_t} \in [0,1]$ $\boldsymbol{x_a} \in [0,1]$ $\boldsymbol{x} \in [0,1]$

Атака: много слоёв

Пусть модель – трёхслойный персептрон с $\boldsymbol{Abs}$ активацией:

$$y = \sigma(W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 x + b_1) + b_2) + b_3)$$

Проблема: присутствуют нелинейности, **не позволяющие** использовать QP решатель.

Атака: много слоёв

Пусть модель – трёхслойный персептрон с $\boldsymbol{Abs}$ активацией:

$$y = \sigma(W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 x + b_1) + b_2) + b_3)$$

Проблема: присутствуют нелинейности, **не позволяющие** использовать QP решатель.

Решение: используем **кусочную линейность** функций активации: фиксируем знаки и раскрываем нелинейности как в eXBTree.

Атака: много слоёв – раскрываем первый слой

Пусть все значения первого слоя **неотрицательны**:

$$\begin{cases} W_1 x + b_1 \geq 0 \\ y = f(W_3 \cdot f(W_2 W_1 x + W_2 b_1 + b_2) + b_3) \end{cases}$$

Атака: много слоёв – раскрываем второй слой

Пусть все значения второго слоя **неположительны**:

$$\begin{cases} W_1 x + b_1 \geq 0 \\ W_2 W_1 x + W_2 b_1 + b_2 \leq 0 \\ y = f(-(W_3 W_2 W_1 x + W_3 W_2 b_1 + W_3 b_2) + b_3) \end{cases}$$

Атака: много слоёв – упрощаем уравнения

Введём новые переменные:

- $W_{21} = W_2 W_1$
- $b_{21} = W_2 b_1 + b_2$
- $W_{321} = -W_3 W_{21}$
- $b_{321} = b_3 - W_3 b_{21}$

Имеем **однослойный случай** с системой неравенств:

$$\begin{cases} W_1 x + b_1 \geq 0 \\ W_{21} x + b_{21} \leq 0 \\ y = f(W_{321} x + b_{321}) \end{cases}$$

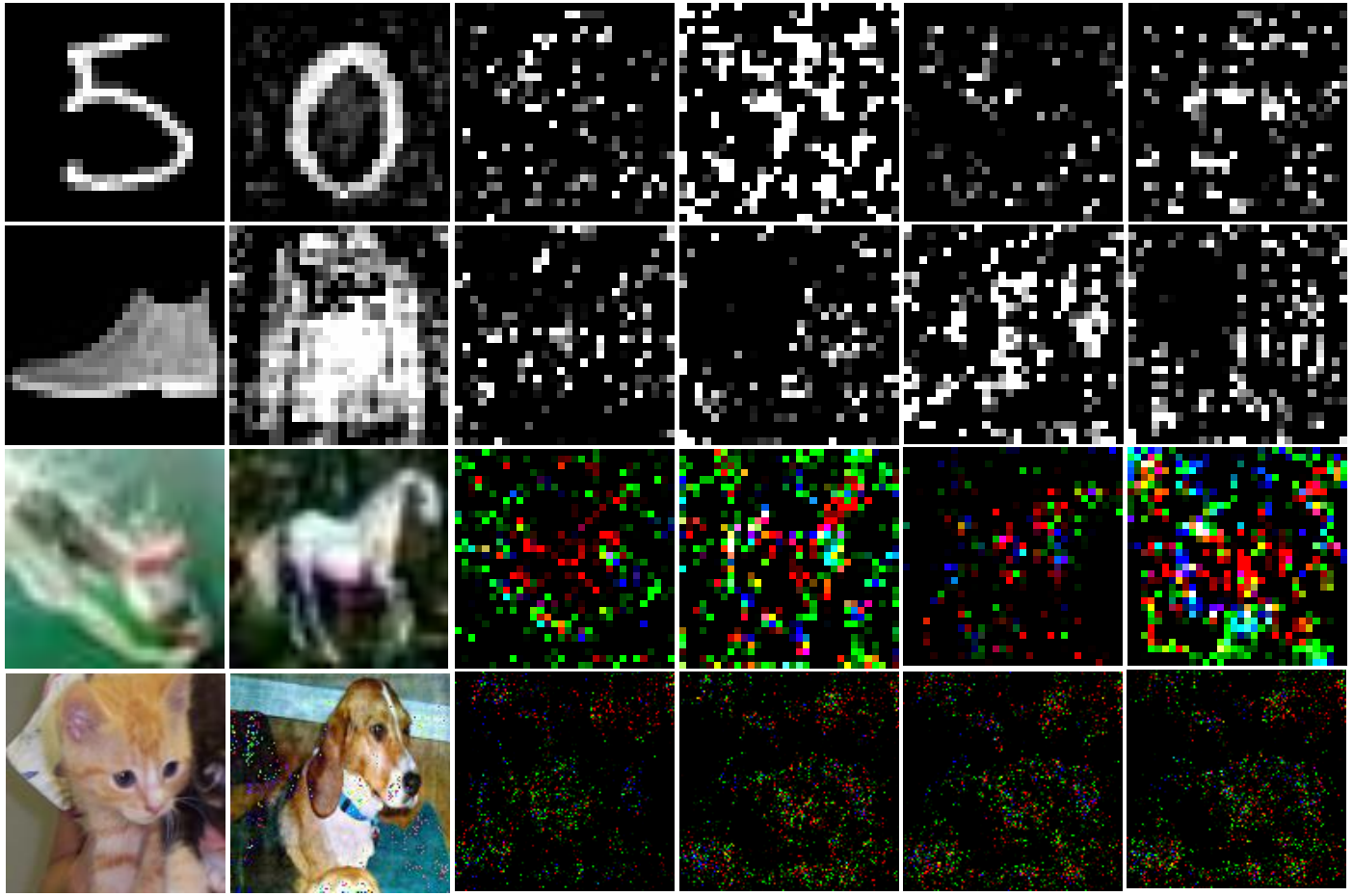
$\boldsymbol{x_t} \in [0,1]$

$\boldsymbol{x_a} \in [0,1]$

$\boldsymbol{x} \in [0,1]$

Бонус: произвольная атака с выходом, равным y_t

- Заменим P **случайной диагональной** матрицей;
- Заменим q **случайным** вектором;
- Запустим QP решатель.



все эти изображения имеют **тот же выход модели**, что и x_t

SLAP атака: выводы

- Персептрон – такая **простая** модель, но такая **уязвимая** перед атаками.
- **Однослойные** модели **более уязвимы**, чем многослойные.
- **Одному** и тому же выходу сети почти всегда соответствует **множество** входов.
- Предлагаемая атака позволяет получить **все возможные** входы (с применением аналитических решателей).

Что на счёт свёрточного слоя?

Свёрточный слой **сводится к полносвязному**, если аккуратно выписать все получаемые уравнения:

$$\begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix} \times \begin{bmatrix} f_1 & f_2 \\ f_3 & f_4 \end{bmatrix} = \begin{bmatrix} f_1 a_1 + f_2 a_2 + f_3 a_4 + f_4 a_5 \\ f_1 a_2 + f_2 a_3 + f_3 a_5 + f_4 a_6 \\ f_1 a_4 + f_2 a_5 + f_3 a_7 + f_4 a_8 \\ f_1 a_5 + f_2 a_6 + f_3 a_8 + f_4 a_9 \end{bmatrix}$$

$$\begin{bmatrix} f_1 & f_2 & 0 & f_3 & f_4 & 0 & 0 & 0 & 0 \\ 0 & f_1 & f_2 & 0 & f_3 & f_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & f_1 & f_2 & 0 & f_3 & f_4 & 0 \\ 0 & 0 & 0 & 0 & f_1 & f_2 & 0 & f_3 & f_4 \end{bmatrix} \times \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \\ a_9 \end{bmatrix}$$

Менее строгая цель

Использование **приблизительного сходства** выходов модели вместо точного соответствия (достигается с помощью матрицы G и вектора h):

$$\begin{cases} Wx + b \geq y_t - \varepsilon \\ Wx + b \leq y_t + \varepsilon \end{cases} \longrightarrow \begin{cases} -Wx \leq b - y_t + \varepsilon \\ Wx \leq y_t - b + \varepsilon \end{cases}$$

$$G = concat(-Wx, Wx)$$
$$h = concat(b - y_t + \varepsilon, y_t - b + \varepsilon)$$

Как следует выбирать знаки выходов слоёв?

- Перебор всех возможных комбинаций (**крайне непрактично**).
- Использование знаков, получаемых во время распространения по сети целевого и атакуемого изображений (или их комбинации с шумом).

Вопросы?