Осенняя Школа по информационным технологиям ОИЯИ
16 – 20 октября 2023

Применение искусственных нейронных сетей в линейном программировании

Соколинский Леонид Борисович
доктор физ.-мат. наук, профессор

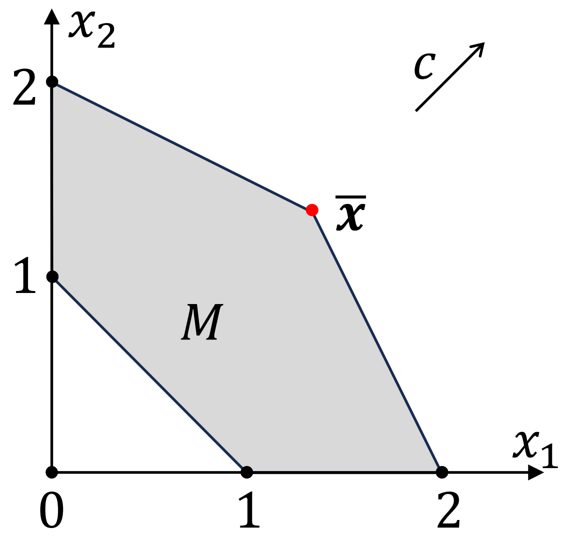
Задача линейного программирования (ЛП)

Пример

$$f(x_1, x_2) = x + x_2 \to max$$

$$\begin{cases} x_1 + 2x_2 \le 2 \\ 2x_1 + x_2 \le 2 \\ x_1 + x_2 \ge 1 \\ x_1 \ge 0 \\ x_2 \ge 0 \end{cases}$$

$$\boldsymbol{c} = (1; 1)$$



Общий вид в стандартной матричной форме

$$\bar{\boldsymbol{x}} = \arg\max\{f(\boldsymbol{x}) | A\boldsymbol{x} \le \boldsymbol{b}\}$$

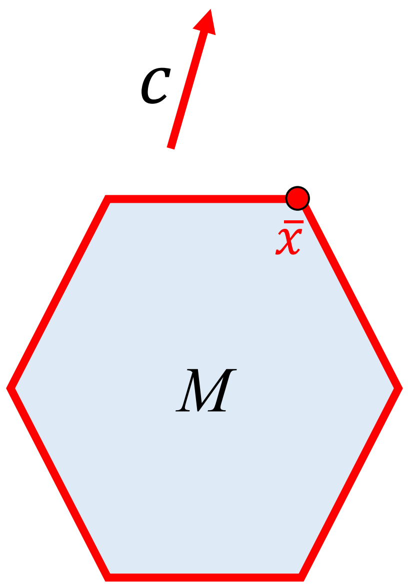
$\boldsymbol{x} \in \mathbb{R}^n$

A – матрица $m \times n$

$\boldsymbol{c}$, $\boldsymbol{b}$ – векторы размерности n

$f(\boldsymbol{x}) = \langle \boldsymbol{c}, \boldsymbol{x} \rangle$ – целевая функция

$\langle \boldsymbol{c}, \boldsymbol{x} \rangle$ – скалярное произведение



$$M = \{x | Ax \le b\}$$

Смена знака неравенства

$$x_1 + x_2 \geq 1 \quad \Rightarrow \quad -x_1 - x_2 \leq -1$$

Переход от равенства к неравенствам

$$x_1 + x_2 - x_3 = 1 \quad \Rightarrow \quad \begin{aligned} x_1 + x_2 - x_3 \le 1 \\ x_1 + x_2 - x_3 \ge 1 \end{aligned}$$

Переход от неравенства к равенству

$$x_1 + x_2 \geq 1 \quad \Rightarrow \quad x_1 + x_2 - x_3 = 1$$

$$x_3 \geq 0$$

Приведение к стандартной форме

$$f(x_1, x_2) = x_1 + x_2 \to max$$

$$\begin{cases} x_1 + 2x_2 \leq 2 \\ 2x_1 + x_2 \leq 2 \\ x_1 + x_2 \geq 1 \\ x_1 \geq 0 \\ x_2 \geq 0 \end{cases}$$

$$c = (1; 1)$$

$\Rightarrow$

$$f(x_1, y) = x_1 + x_2 \to max$$

$$\begin{cases} x_1 + 2x_2 \leq 2 \\ 2x_1 + x_2 \leq 2 \\ -x_1 - x_2 \leq -1 \\ -x_1 \leq 0 \\ -x_2 \leq 0 \end{cases}$$

$$c = (1; 1)$$

Приведение к канонической форме

$$f(x_1, x_2) = x_1 + x_2 \to max$$

$$\begin{cases} x_1 + 2x_2 \le 2 \\ 2x_1 + x_2 \le 2 \\ x_1 + x_2 \ge 1 \\ x_1 \ge 0 \\ x_2 \ge 0 \end{cases}$$

$$c = (1; 1)$$

$$f(x, y) = x_1 + x_2 \to max$$

$$\begin{cases} x_1 + 2x_2 + x_3 = 2 \\ 2x_1 + x_2 + x_4 = 2 \\ x_1 + x_2 - x_5 = 1 \\ x_1 \ge 0 \\ x_2 \ge 0 \\ x_3 \ge 0 \\ x_4 \ge 0 \\ x_5 \ge 0 \end{cases}$$

$$c = (1; 1; 0; 0; 0)$$

Большие задачи ЛП

- Современные задачи ЛП, возникающие в различных областях, могут содержать миллионы переменных и миллионы ограничений
- Для решения таких задач необходимы суперкомпьютеры

Нестационарные задачи ЛП

- Целевая функция и/или ограничения изменяются в течение вычислительного процесса
- Примеры
  - выбор оптимальных стратегий в роботрейдинге
  - оптимальное управление летательными аппаратами
  - оптимальное управление технологическими процессами
  - логистические и транспортные задачи
  - оперативное планирование и управление производством продукции

Линейная оптимизация
в режиме реального времени

- Управление химическим производством
- Управление системой многоточечного впрыска топлива в ДВС
- Управление сотовыми сетями
- Автопилотирование
- Системы самонаведения ракет

Режим реального времени

- Решение задачи ЛП должно выполняться за определенное время
- Методы решения
  - Рассматривать каждое изменение как появление новой задачи оптимизации, которую необходимо решать с нуля
  - Непрерывно адаптировать решение к изменяющейся среде, повторно используя информацию, полученную в прошлом
    - Подход применим, если алгоритм достаточно быстро отслеживает траекторию движения оптимальной точки
    - В случае больших задач ЛП требует разработки масштабируемых методов и параллельных алгоритмов ЛП

Решатели

- Симплекс-метод
- Метод внутренних точек

Симплекс-метод

- Плохо распараллеливается на распределенной памяти (не более 16-32 процессорных узлов)
- При решении задач больших размерностей наблюдается потеря точности
- При изменении исходных данных необходимо начинать вычислительный процесс заново

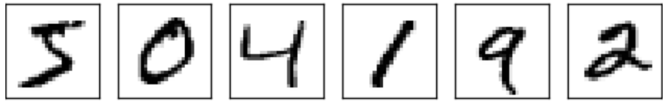
Метод внутренних точек

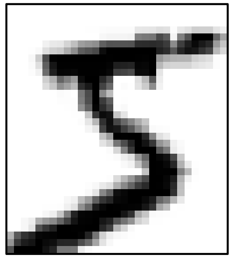
- Адаптируется к динамическим изменениям исходных данных задачи
- Отсутствуют эффективные параллельные реализации для систем с распределенной памятью
- Во многих случаях необходимо находить начальную внутреннюю точку

Возможное решение

Разработать новый метод решения нестационарных задач ЛП в реальном времени с использование суперкомпьютерных и нейросетевых технологий

Глубокая нейронная сеть (на примере распознавания рукописных цифр)



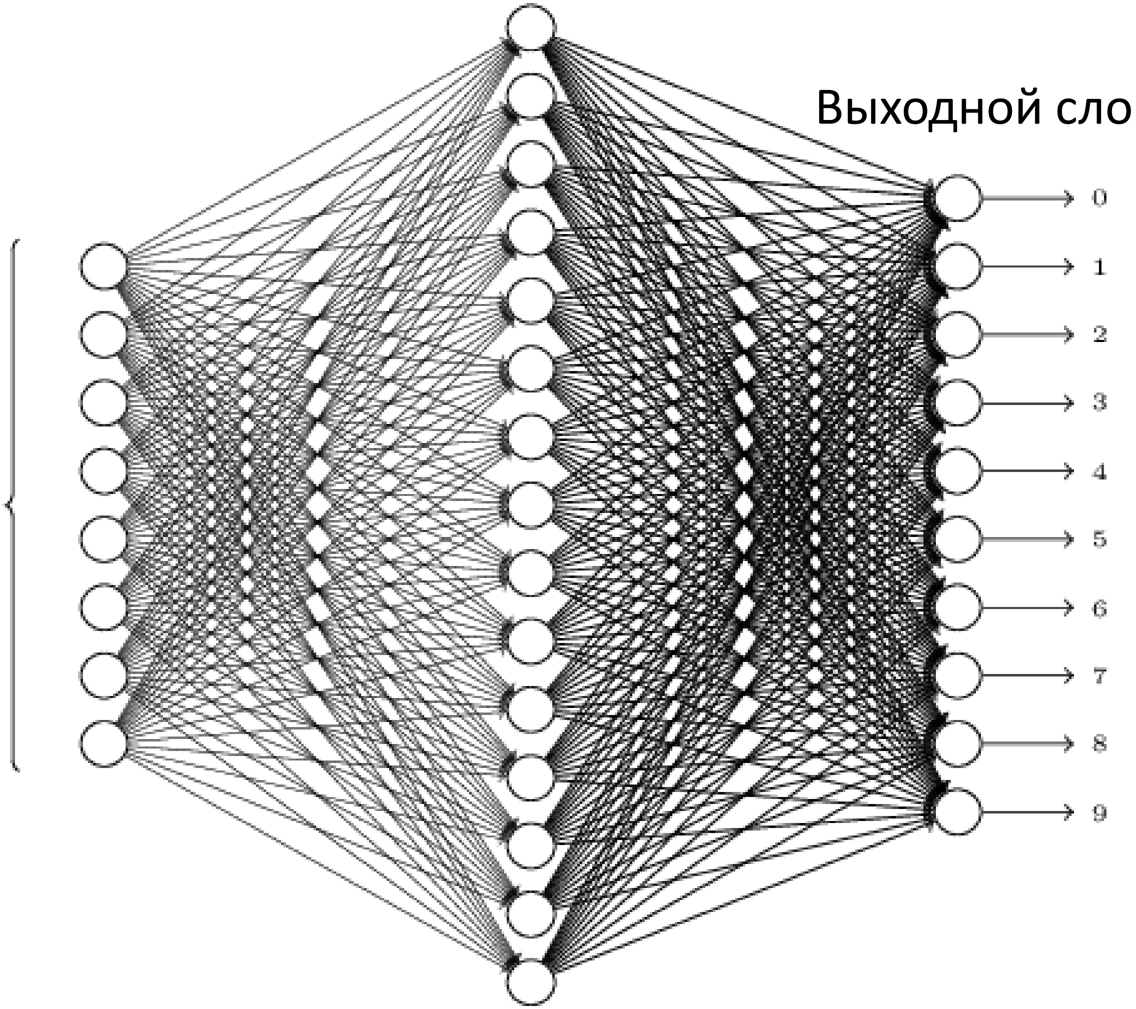


28×28

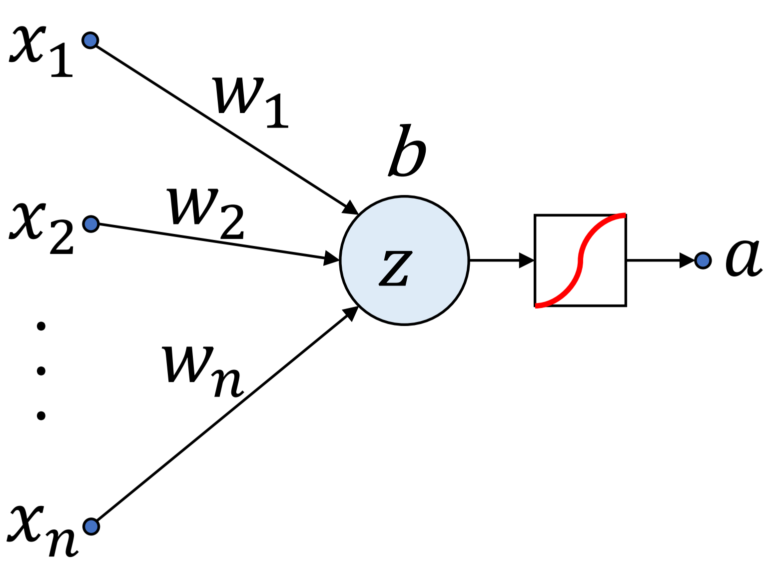
Входной слой (784 нейрона)

Скрытый слой (15 нейронов)

Выходной слой



Сигмоидальный нейрон (сигмоид)



$$z = \sum_{i=1}^{n} w_i x_i$$

$$a = \frac{1}{1 + e^{-(z+b)}}$$

- $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ – входные сигналы
- $\boldsymbol{w} = (w_1, w_2, \ldots, w_n)$ – синаптические веса $(w_i \in \mathbb{R})$
- b – смещение $(b \in \mathbb{R})$
- z – активационный потенциал
- a – выходной сигнал

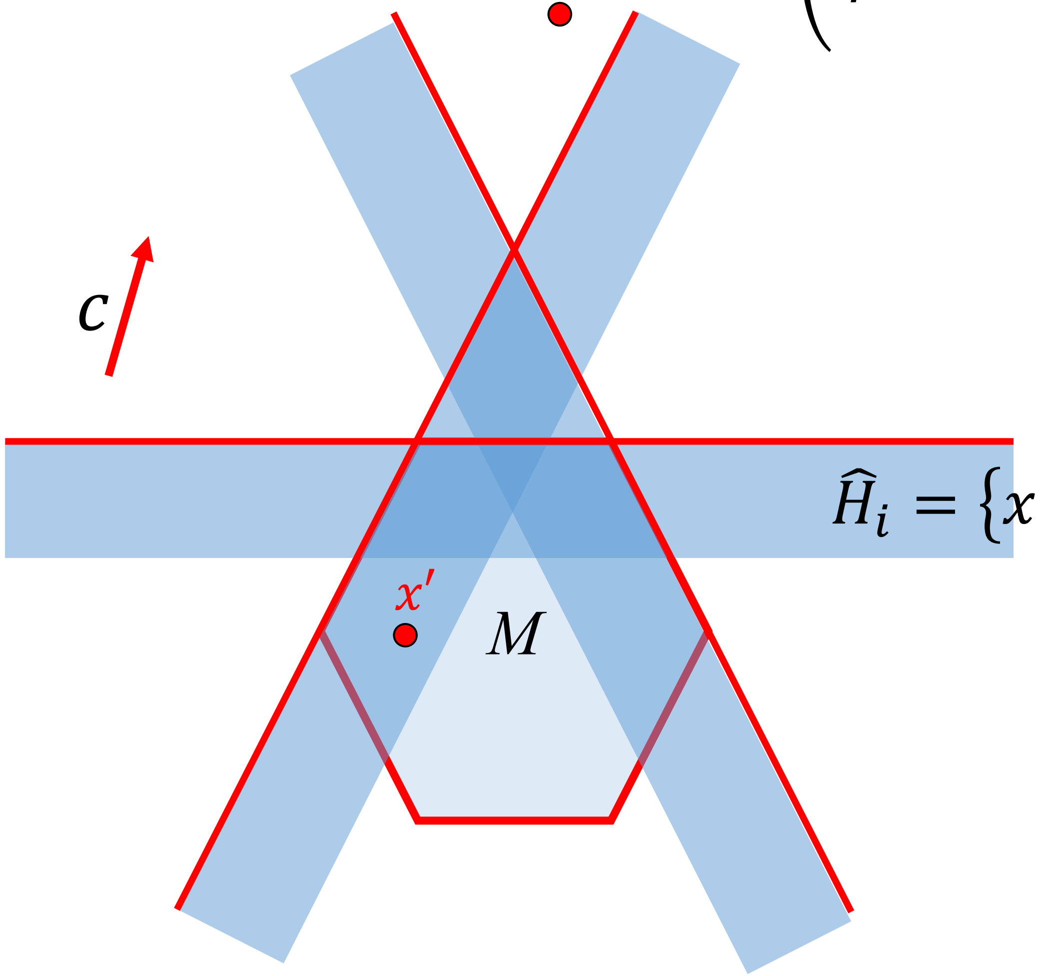
Метод поверхностного движения

1. Выбрать начальную точку u на поверхности многогранника
2. Вычислить на суперкомпьютере локальный образ допустимого многогранника в точке u
3. С помощью нейронной сети вычислить на поверхности многогранника направление d максимального увеличения $f(x)$
4. Двигаться по поверхности допустимого многогранника по направлению d так далеко, насколько это возможно
5. Полученную точку принять за новое приближение u и перейти на шаг 2
6. Закончить работу при $d = 0$

Выбор начальной точки: вычисление точки апекса

$$z = x' + \left(\eta + \max\left\{\frac{b_i - \langle a_i, x'\rangle}{\langle a_i, e_c\rangle}\,\middle|\, i \in \mathcal{J}\right\}\right) e_c$$

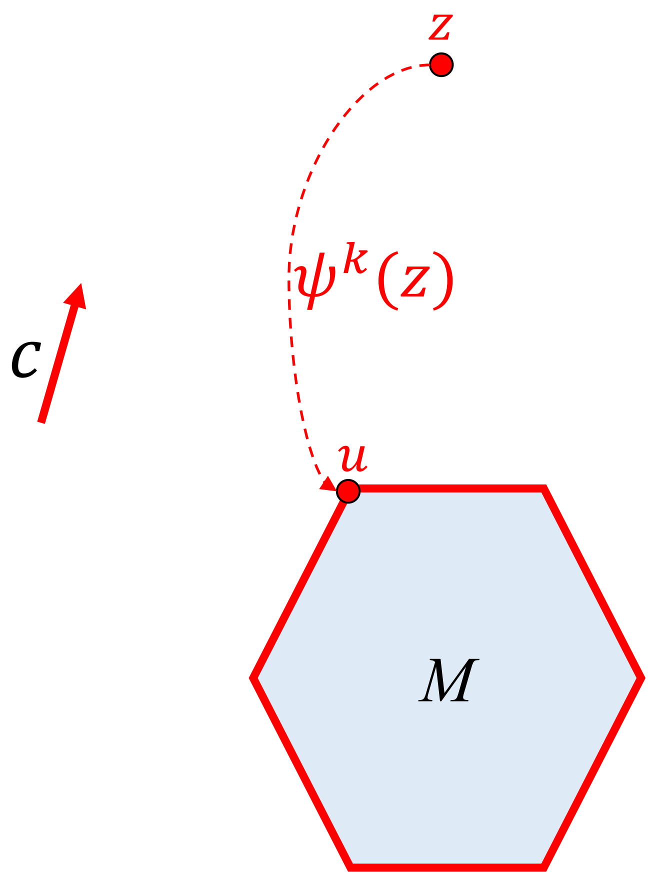
$$\eta > 0$$



$$\widehat{H}_i = \left\{x \in \mathbb{R}^n \,\middle|\, \langle a_i, x\rangle \le b_i\right\}$$

$$\langle a_i, c\rangle > 0$$

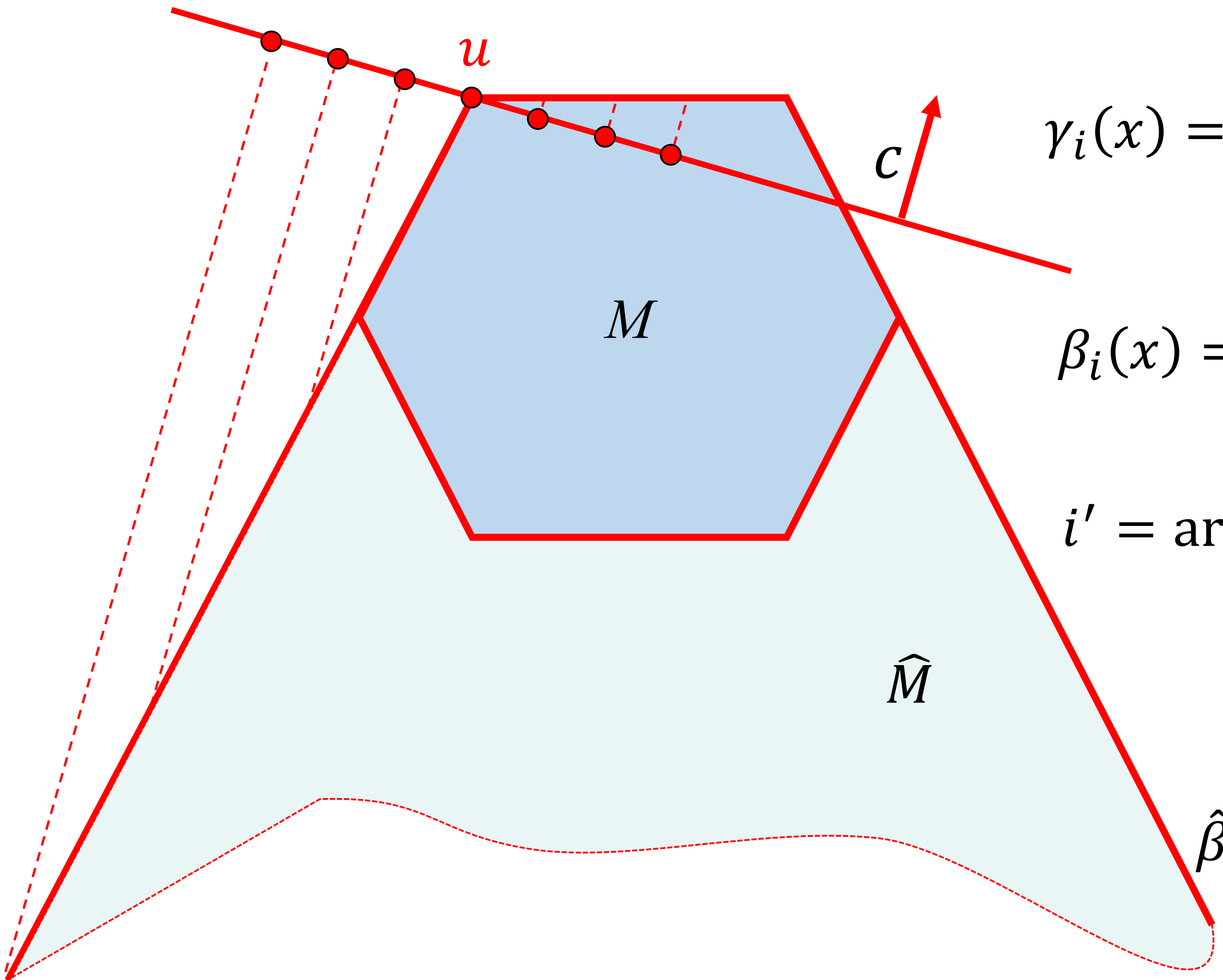
Выбор начальной точки: фейеровское отображение



$$\pi_i(x) = x - \frac{\langle a_i, x \rangle - b_i}{\|a_i\|^2} a_i$$

$$\psi(x) = \begin{cases} x, & \text{если } x \in M \\ \dfrac{1}{|\mathcal{J}_x|} \displaystyle\sum_{i \in \mathcal{J}_x} \pi_i(x), & \text{если } x \notin M \end{cases}$$

Построение локального образа



$$\gamma_i(x) = x - \frac{\langle a_i, x\rangle - b_i}{\langle a_i, c\rangle} c$$

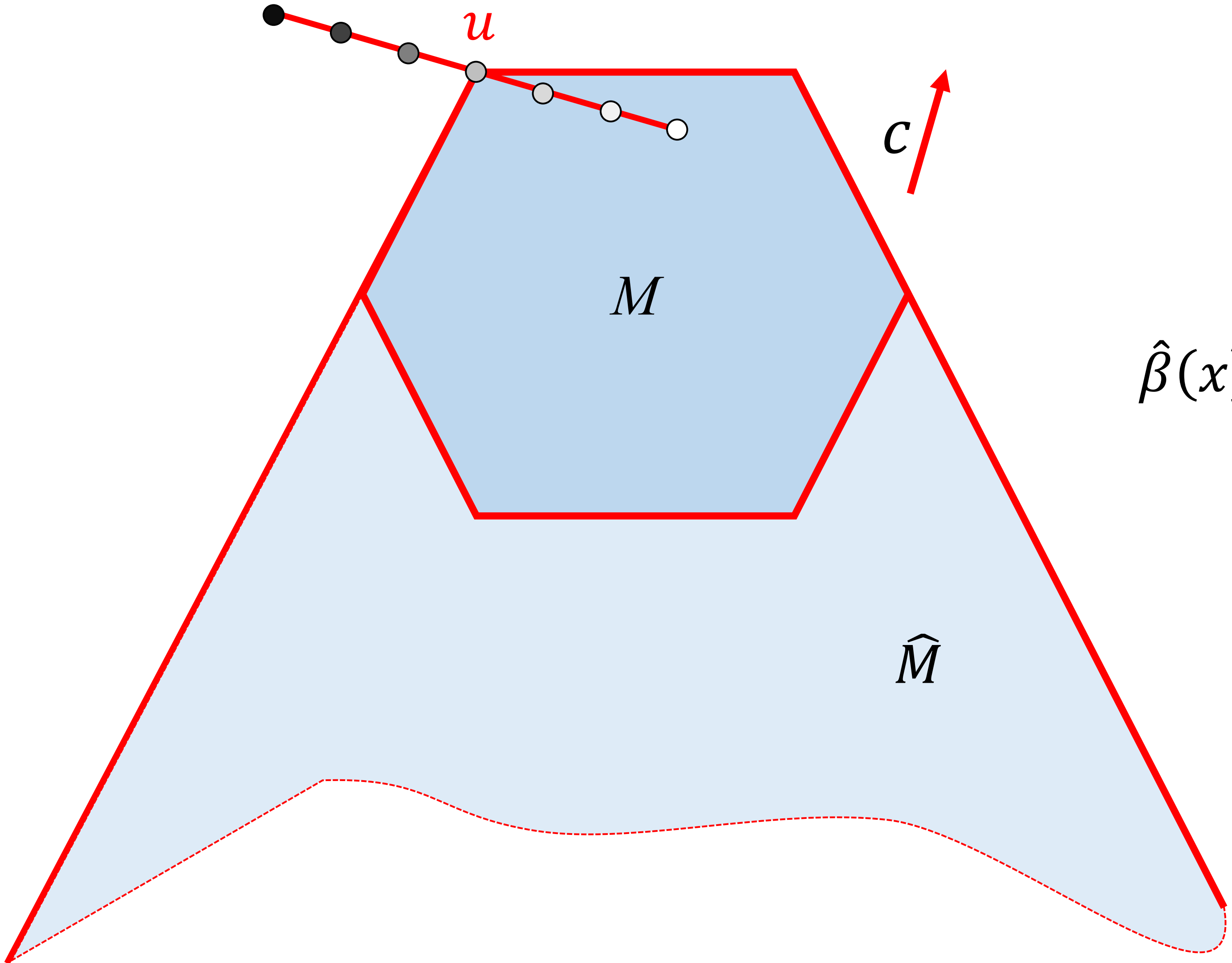
$$\beta_i(x) = -\frac{\langle a_i, x\rangle - b_i}{\langle a_i, c\rangle} \|c\|$$

$$i' = \operatorname{argmin}\{\beta_i(x) | i \in \mathcal{I}\}$$

$$\hat{\gamma}(x) = \gamma_{i'}(x)$$

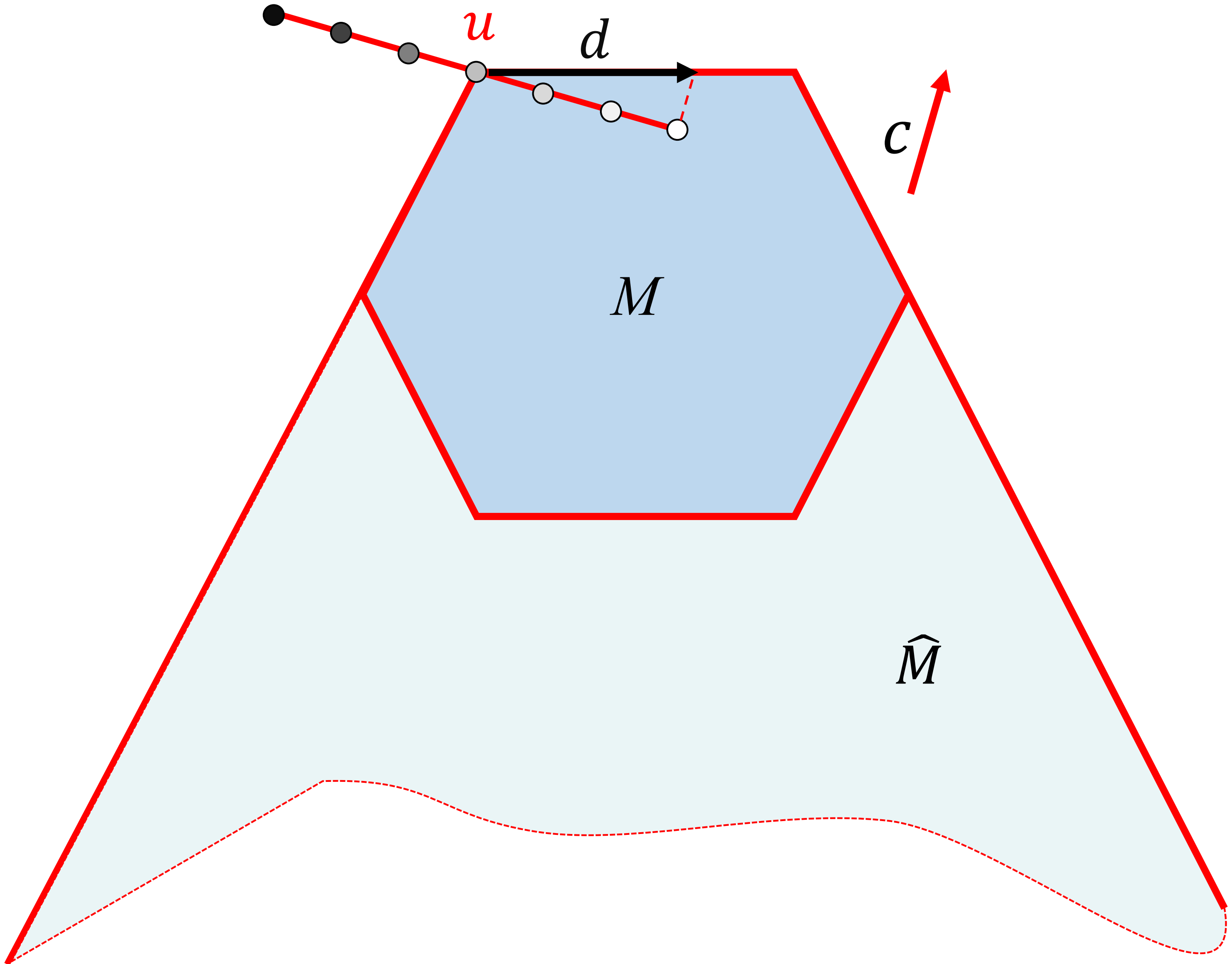
$$\hat{\beta}(x) = \frac{\langle c, \hat{\gamma}(x) - x\rangle}{\|c\|^2}$$

Локальный образ

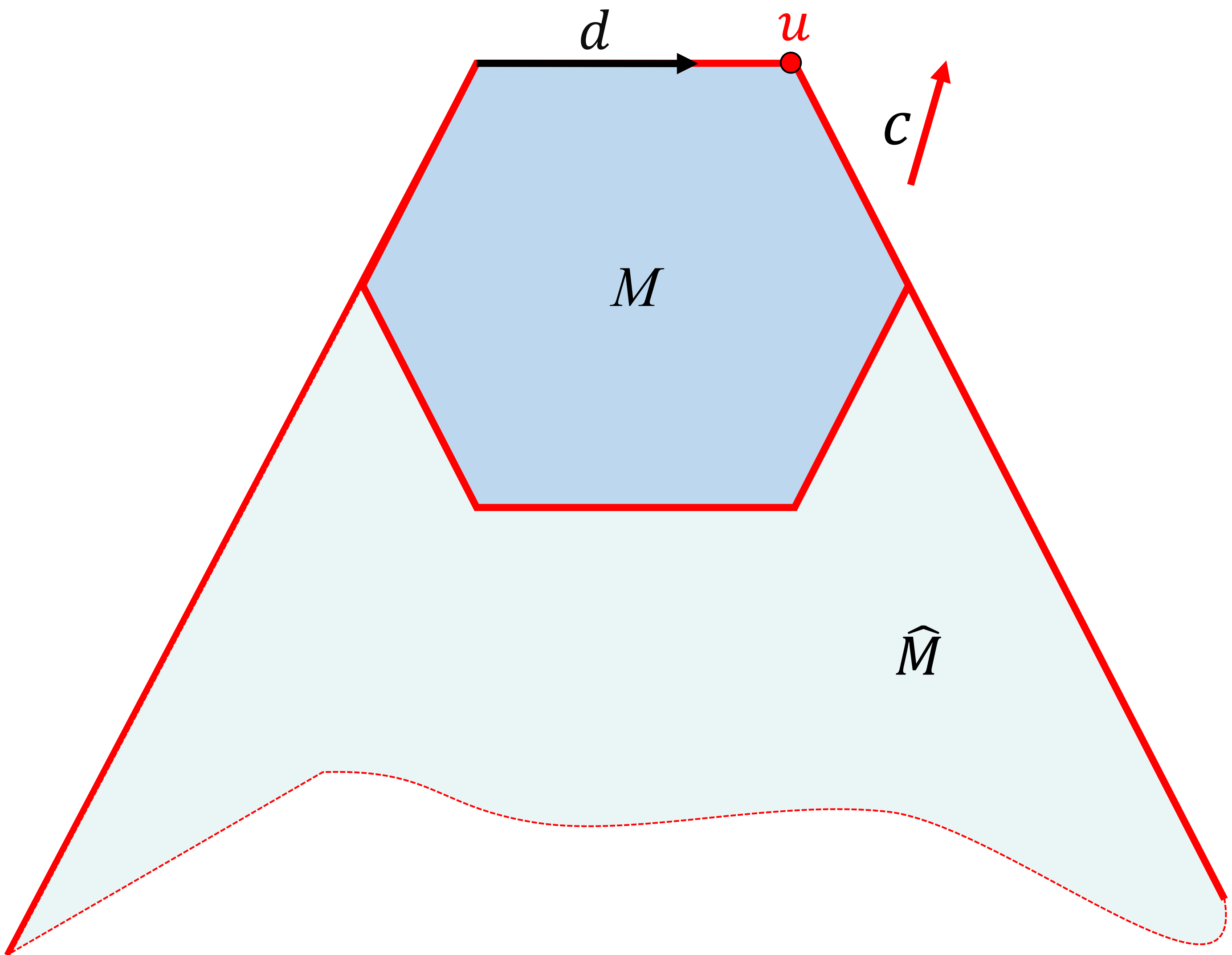


$$\hat{\beta}(x) = \frac{\langle c, \hat{\gamma}(x) - x\rangle}{\|c\|^2}$$

Нейронная сеть вычисляет направление движения



Получаем следующее приближение

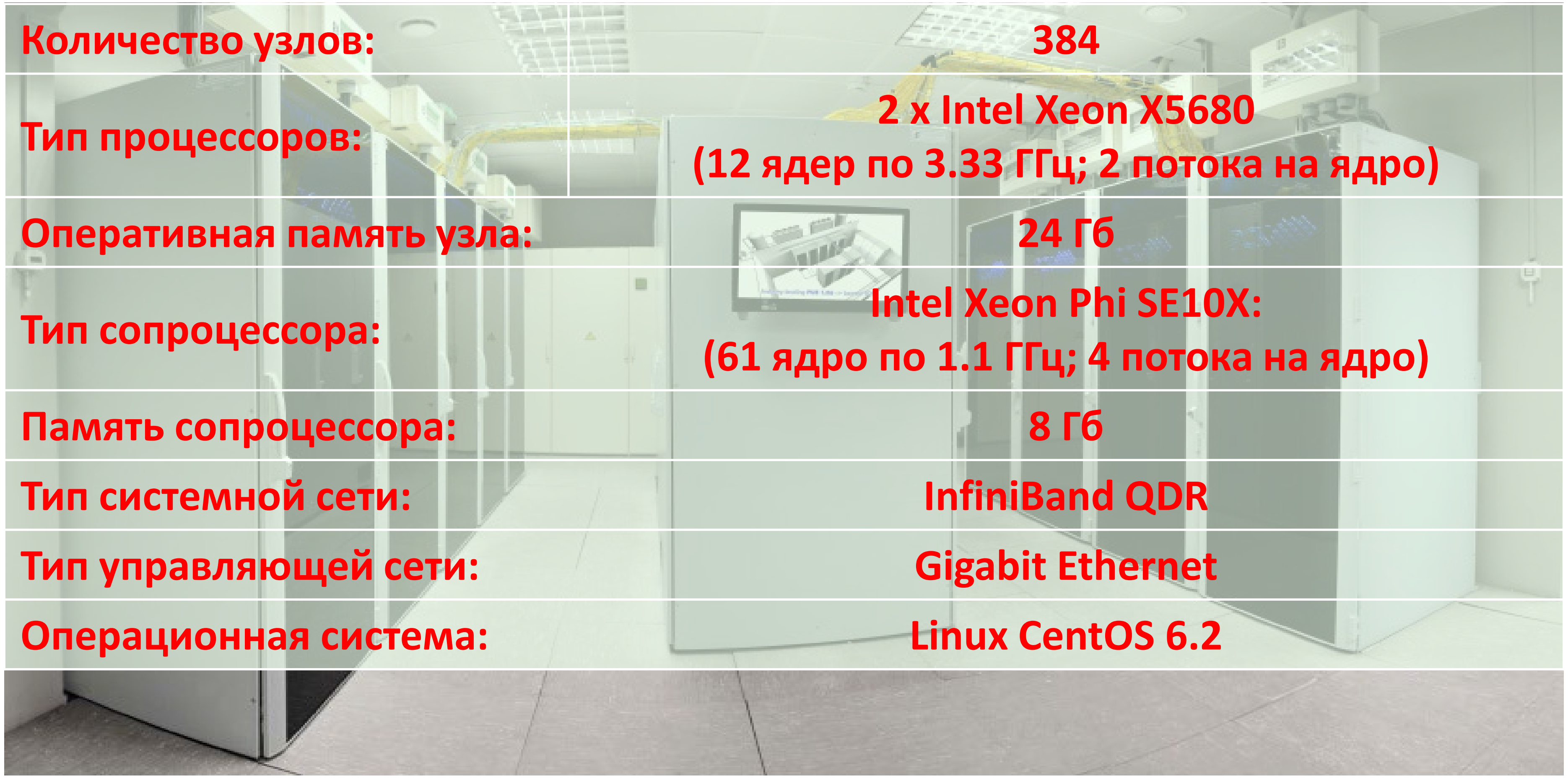


Обучающее множество

- Обучающее множество может быть построено с помощью апекс-метода

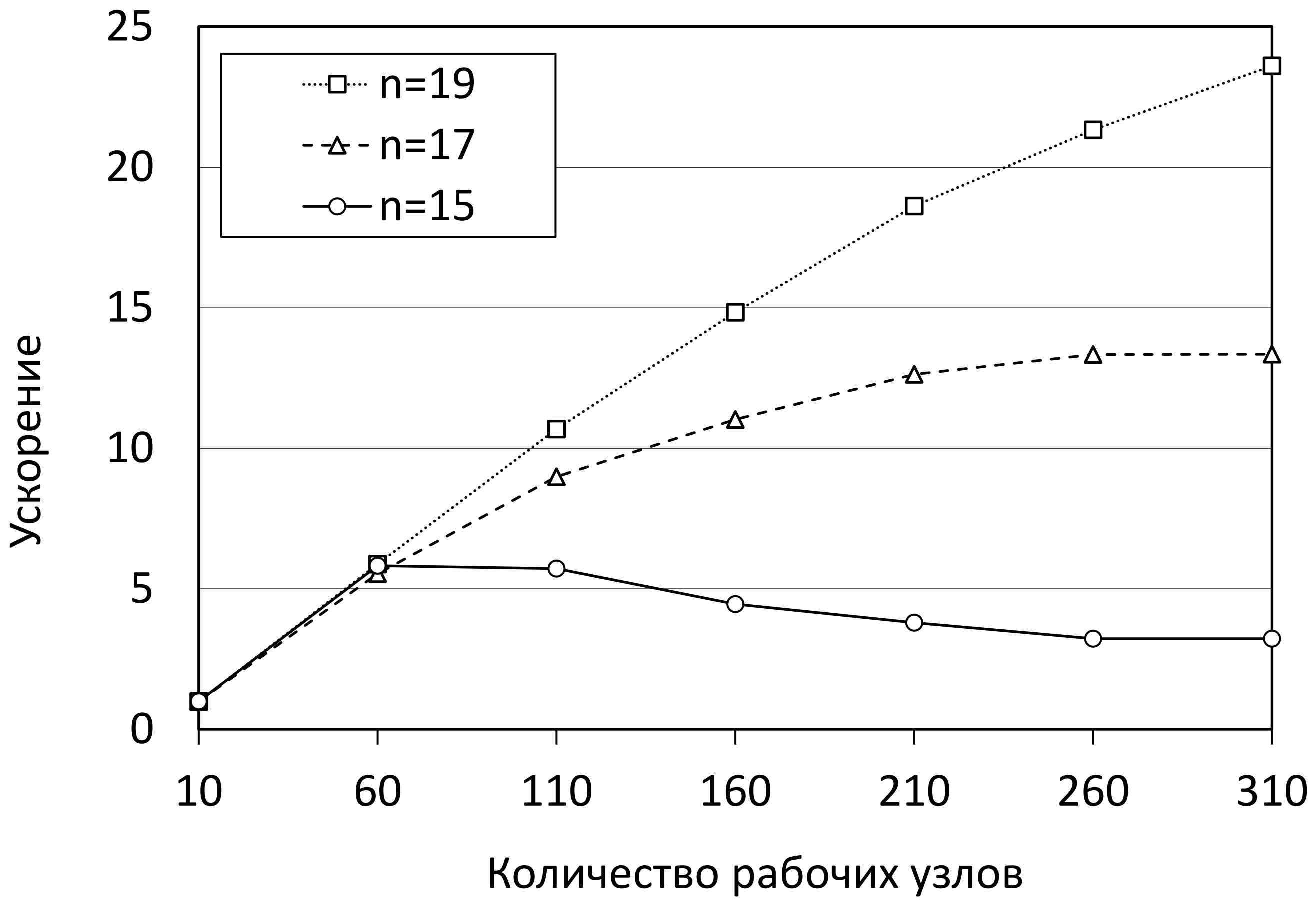
[Соколинский Л.Б., Соколинская И.М. О новой версии апекс-метода для решения задач линейного программирования // Вестник Южно-Уральского государственного университета. Серия: Вычислительная математика и информатика. 2023. Т. 12, № 2. С. 5–46. DOI: https://doi.org/10.14529/cmse230201]

Суперкомпьютер «Торнадо ЮУрГУ»

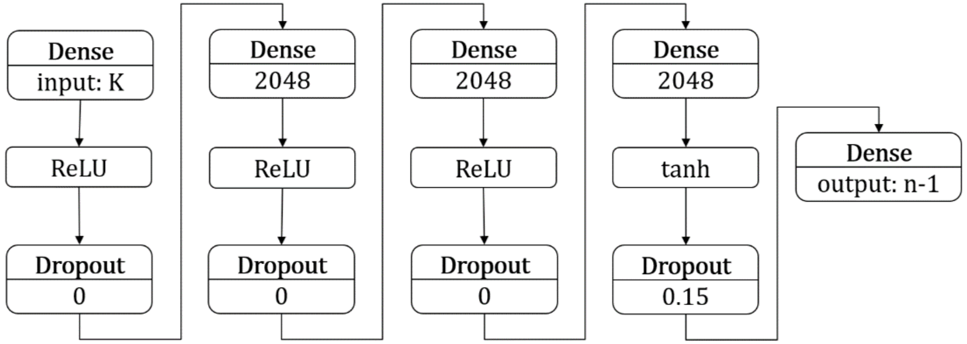


Количество узлов:	384
Тип процессоров:	2 x Intel Xeon X5680 (12 ядер по 3.33 ГГц; 2 потока на ядро)
Оперативная память узла:	24 Гб
Тип сопроцессора:	Intel Xeon Phi SE10X: (61 ядро по 1.1 ГГц; 4 потока на ядро)
Память сопроцессора:	8 Гб
Тип системной сети:	InfiniBand QDR
Тип управляющей сети:	Gigabit Ethernet
Операционная система:	Linux CentOS 6.2

Построение локального образа задачи ЛП

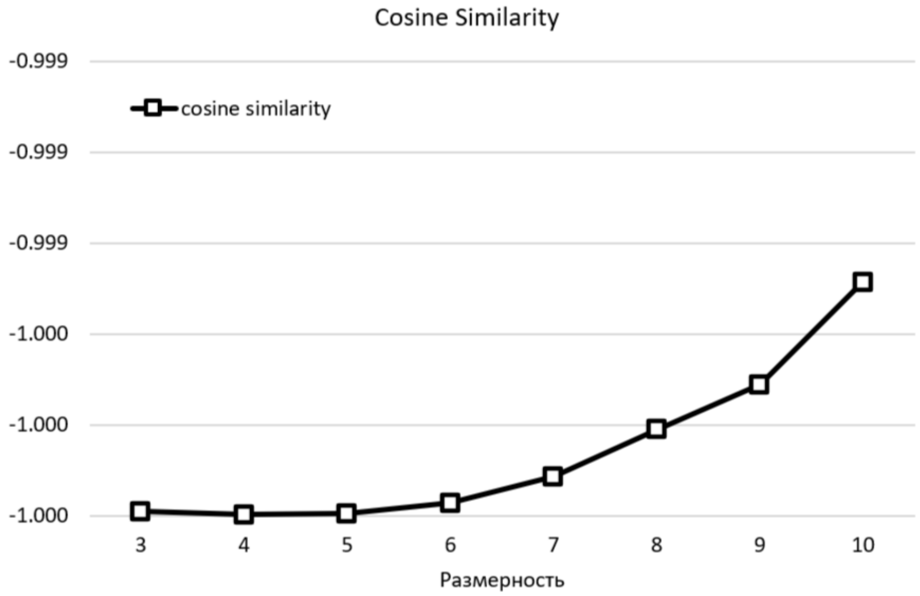


Нейронная сеть



NVIDIA Tesla V100

$$\frac{\sum_{k=1}^{n}(\boldsymbol{\alpha}_k \cdot \boldsymbol{y}_k)}{\sqrt{\sum_{k=1}^{n}\boldsymbol{\alpha}_k^2} \cdot \sqrt{\sum_{k=1}^{n}\boldsymbol{y}_k^2}}$$



Ольховский Н. А. 2023. Исследование структуры рецептивного поля в визуальном методе решения задачи линейного программирования. PREPRINTS.RU. https://doi.org/10.24108/preprints-3112771

Публикации по теме лекции

1. Ольховский Н. А., Соколинский Л. Б. О новом методе линейного программирования с использованием нейронных сетей. PREPRINTS.RU. 2023. https://doi.org/10.24108/preprints-3112762
2. Sokolinsky L.B., Sokolinskaya I.M. Apex Method: A New Scalable Iterative Method for Linear Programming // Mathematics. 2023. Vol. 11, no. 7. Article number 1654. DOI:10.3390/math11071654
3. Olkhovsky N.A., Sokolinsky L.B. Visualizing Multidimensional Linear Programming Problems // Parallel Computational Technologies. PCT 2022. Communications in Computer and Information Science, vol. 1618. Cham: Springer, 2022. P. 172-196. DOI:10.1007/978-3-031-11623-0_13

Спасибо за внимание!

Леонид Борисович Соколинский

Южно-Уральский государственный университет (НИУ)

Научный руководитель НОЦ «Искусственный интеллект и квантовые технологии»

Заведующий кафедрой системного программирования

Россия, 454080, г. Челябинск, проспект им. В. И. Ленина, 76

Рабочий телефон: (351) 272-30-80

E-mail: leonid.sokolinsky@susu.ru

Личная страница: http://sok.susu.ru

Skype: BpZ9wgmtUscK