



Нейронные сети. Многоагентные системы.

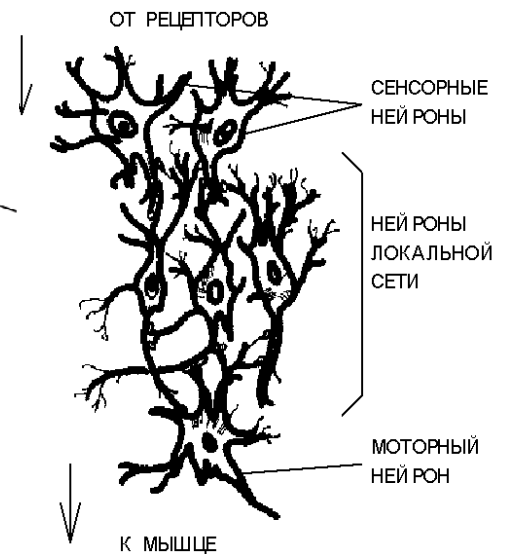
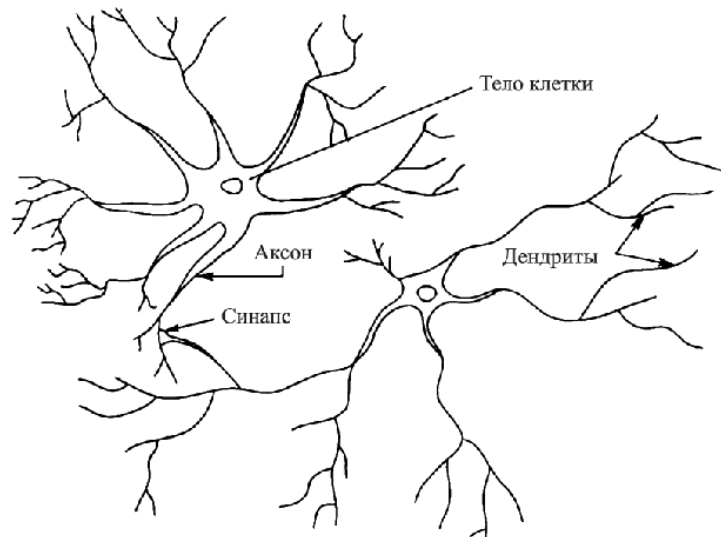
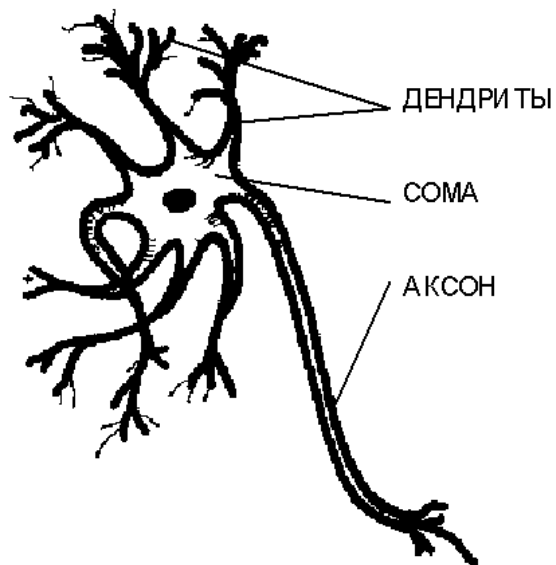
⌘ Гужва А.Г.

Нейрон

⌘ МОЗГ: 10^{10} нейронов, $6 \cdot 10^{13}$ связей

⌘ Время реакции 10^{-3} с

⌘ Размеры 100 микрон

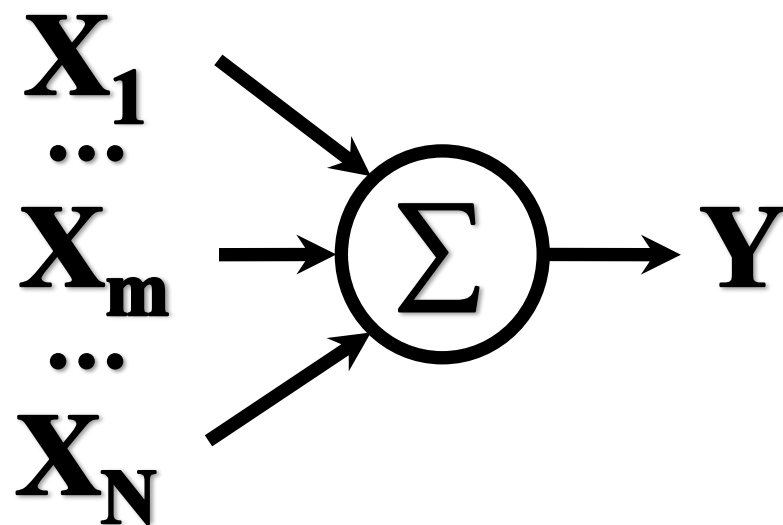


Модель нейрона

⌘ N ВХОДОВ

⌘ 1 ВЫХОД

⌘ N+1 вес ($w_0..w_N$)



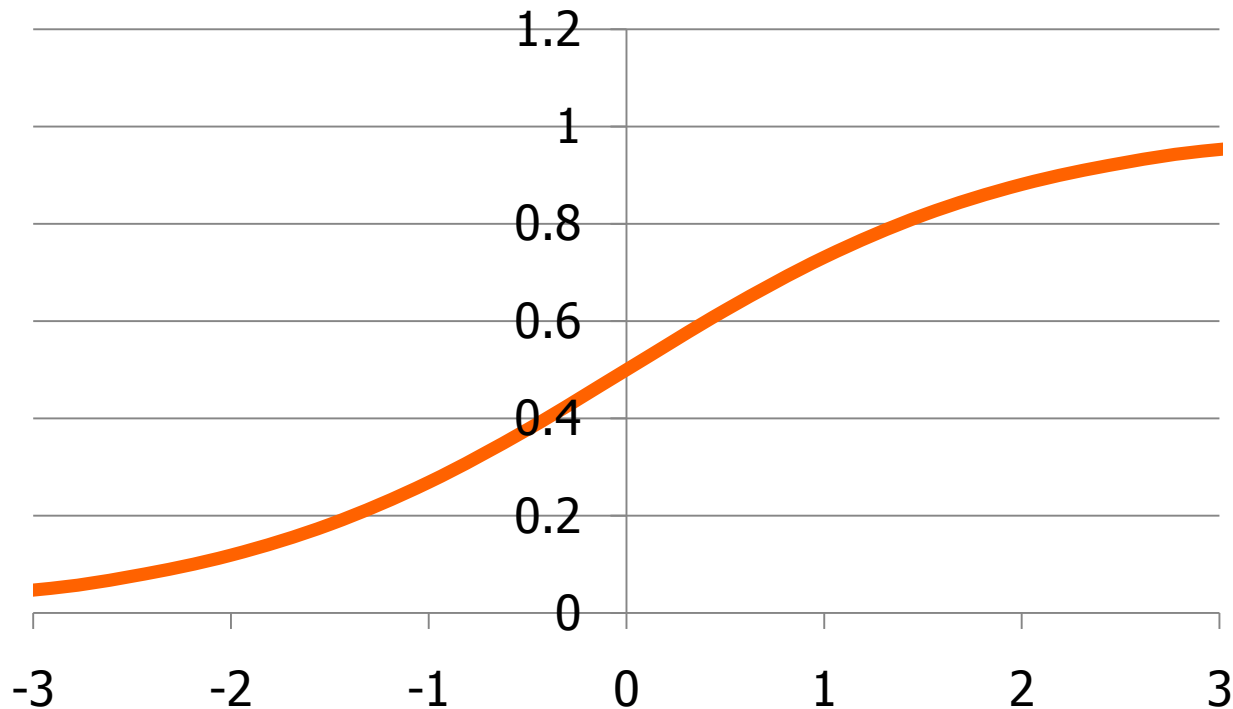
$$net = \sum_{i=1}^N X_i w_i + w_0$$

$$Y = F(net)$$

Модель нейрона.

Передающая функция.

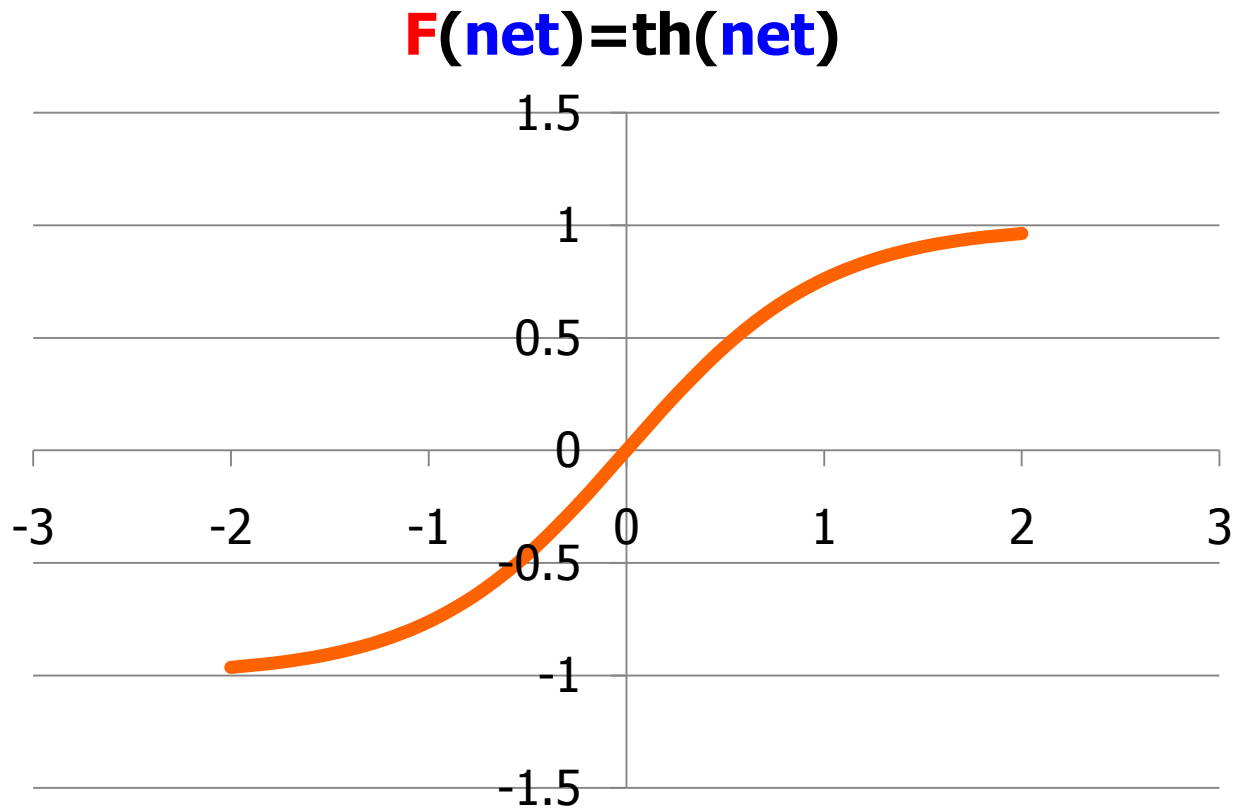
$$F(\text{net}) = 1 / (1 + \exp(-\text{net}))$$



$$F(+\infty) = 1, \quad F(-\infty) = 0, \quad F'(x) = F(x) * (1 - F(x))$$

Модель нейрона.

Передаточная функция.



$$F(+\infty)=1, F(-\infty)=-1, F'(x)=1-(F(x))^2$$

Модель нейрона.

Передающая функция.

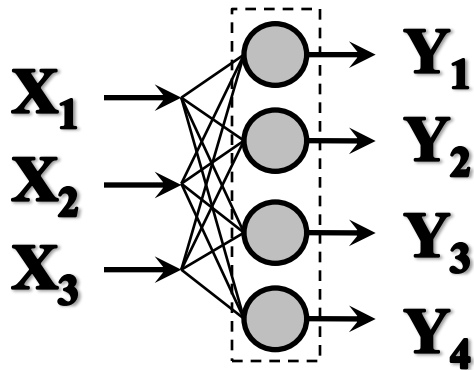
F(net) : Линейная функция



F(+∞)=1, F(-∞)=-1, F'(x)=1 или 0

Блоки / слои нейронов

- ✂ Состоит из нейронов
- ✂ Нейроны объединяются в блоки / слои
- ✂ Входной сигнал одинаков



$$Y_1 = F(w_{11} * X_1 + w_{12} * X_2 + w_{13} * X_3 + w_{10} * 1)$$

$$Y_2 = F(w_{21} * X_1 + w_{22} * X_2 + w_{23} * X_3 + w_{20} * 1)$$

$$Y_3 = F(w_{31} * X_1 + w_{32} * X_2 + w_{33} * X_3 + w_{30} * 1)$$

$$Y_4 = F(w_{41} * X_1 + w_{42} * X_2 + w_{43} * X_3 + w_{40} * 1)$$

$Y = F(W * X)$. Сколько компонент в W ?

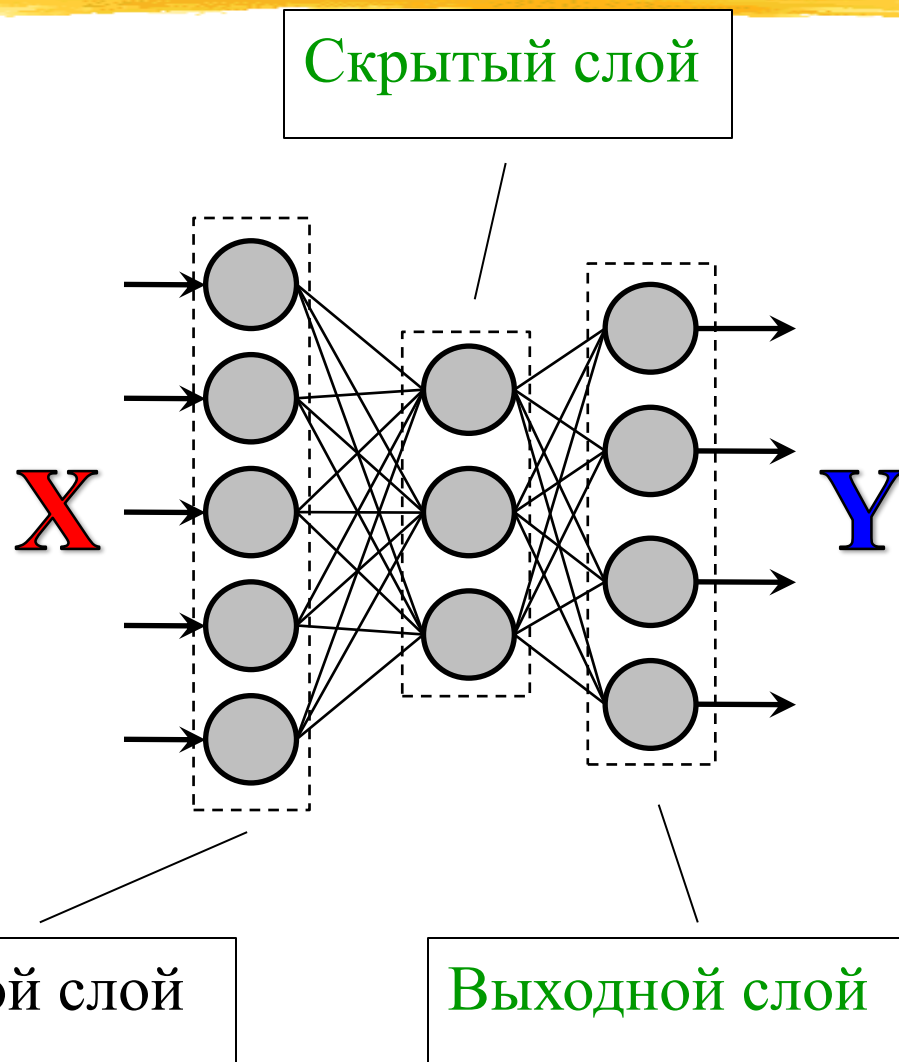
Многослойный персептрон

⌘ $\mathbf{Y} = F(\mathbf{W}_o * F(\mathbf{W}_h * \mathbf{X}))$

⌘ \mathbf{W}_o - матрица весов
ВЫХОДНОГО СЛОЯ

⌘ \mathbf{W}_h - матрица весов
СКРЫТОГО СЛОЯ

⌘ $\mathbf{W} = (\mathbf{W}_h, \mathbf{W}_o)$ –
«матрица» весов сети



Обучение персептрона

- ⌘ Пары **сигнал** - **желаемый отклик** $\{(\mathbf{X}, \mathbf{Y})\}$
- ⌘ Подаем на вход **сигнал** \mathbf{X}_k , получаем **ответ сети** $\mathbf{Z}_k = \mathbf{Z}_k(\mathbf{W}, \mathbf{X}_k)$
- ⌘ Требуем: **ответ сети** = **желаемый отклик**
- ⌘ Подстраиваем \mathbf{W}

Области использования нейросетей



⌘ Классификация

⌘ Кластеризация

⌘ Регрессия и прогнозирование

Классификация



- ⌘ Отнесение неизвестного объекта к одному из известных классов
- ⌘ Распознавание лиц (пример про БОМЖей)
- ⌘ Медицина (диагноз)
- ⌘ Финансовые приложения (кредит)
- ⌘ Сетевые атаки

Классификация







⌘ Объект может к каждому классу в разной степени

⌘ «Не знаю» и «Ни одно из известных»

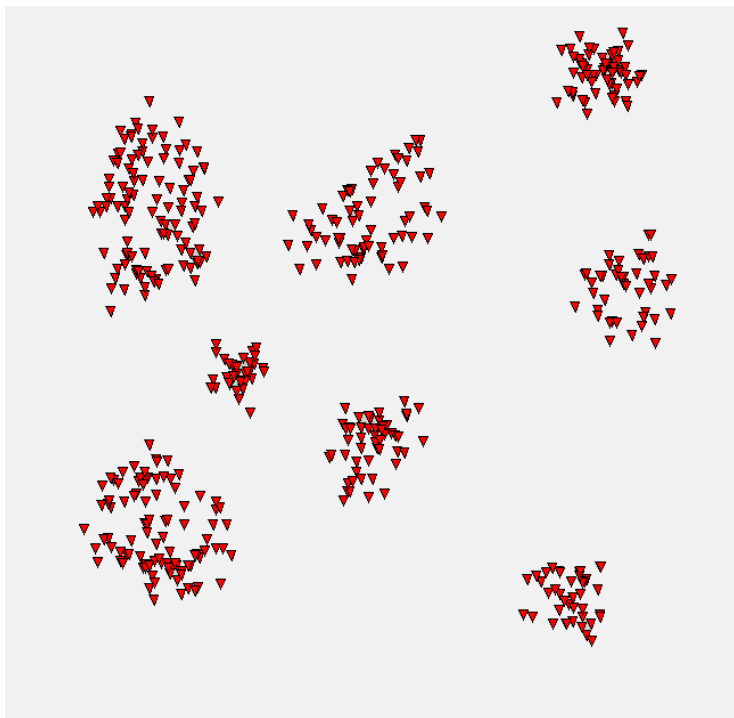
Многослойный персептрон

⌘ Пример (XOR)

- ☒ Сигнал $(-1, -1)$, желаемый ответ $(1, 0)$ 
- ☒ Сигнал $(+1, -1)$, желаемый ответ $(0, 1)$ 
- ☒ Сигнал $(+1, +1)$, желаемый ответ $(1, 0)$ 
- ☒ Сигнал $(-1, +1)$, желаемый ответ $(0, 1)$ 

Кластеризация

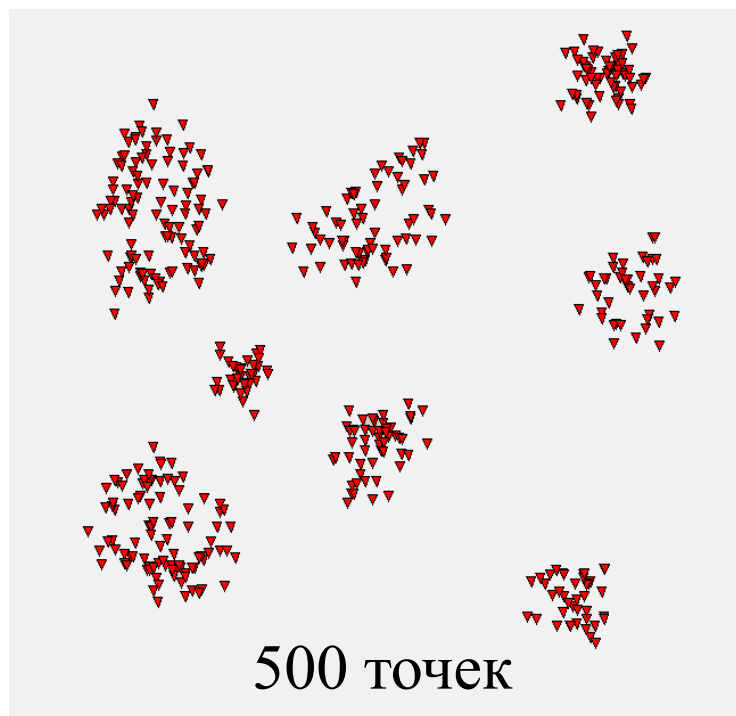
⌘ Разбиение объектов на кластеры



⌘ Что изображено на рисунке для человека?

⌘ Что изображено на рисунке для машины?

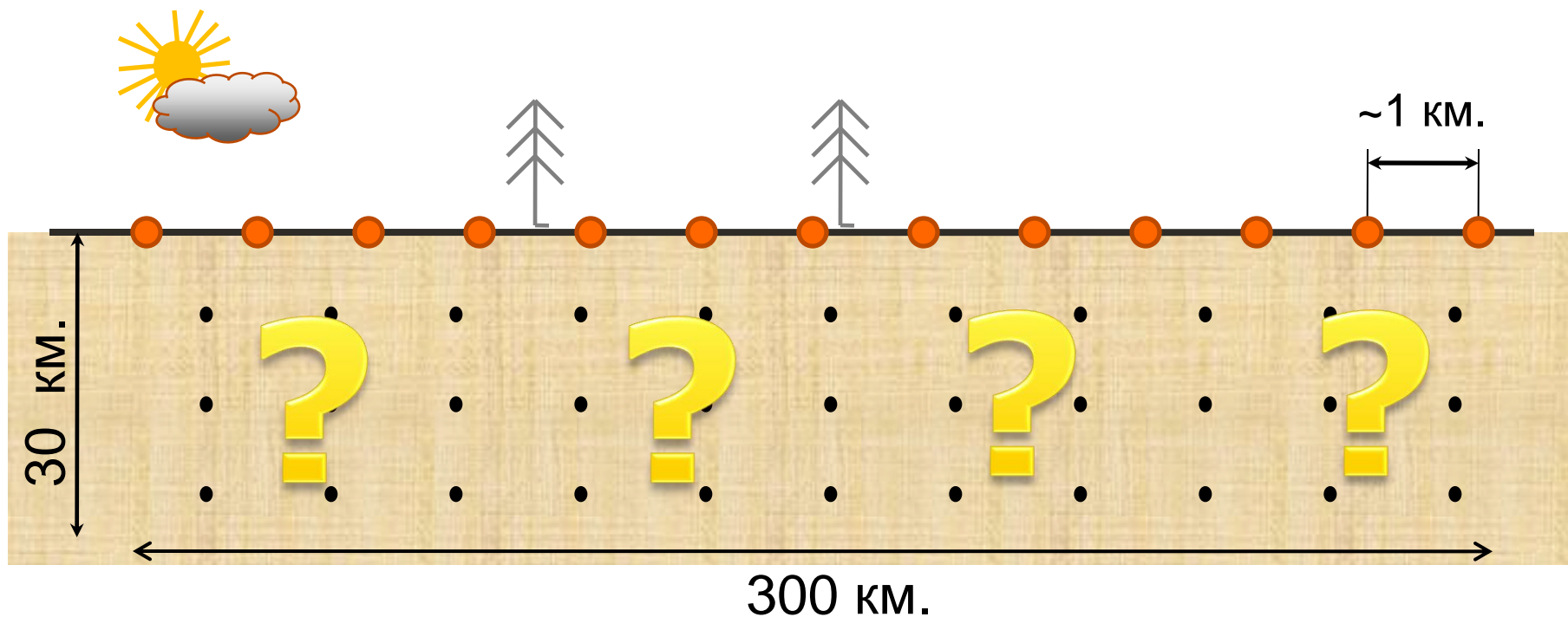
Кластеризация



⌘ Пример про перепись населения

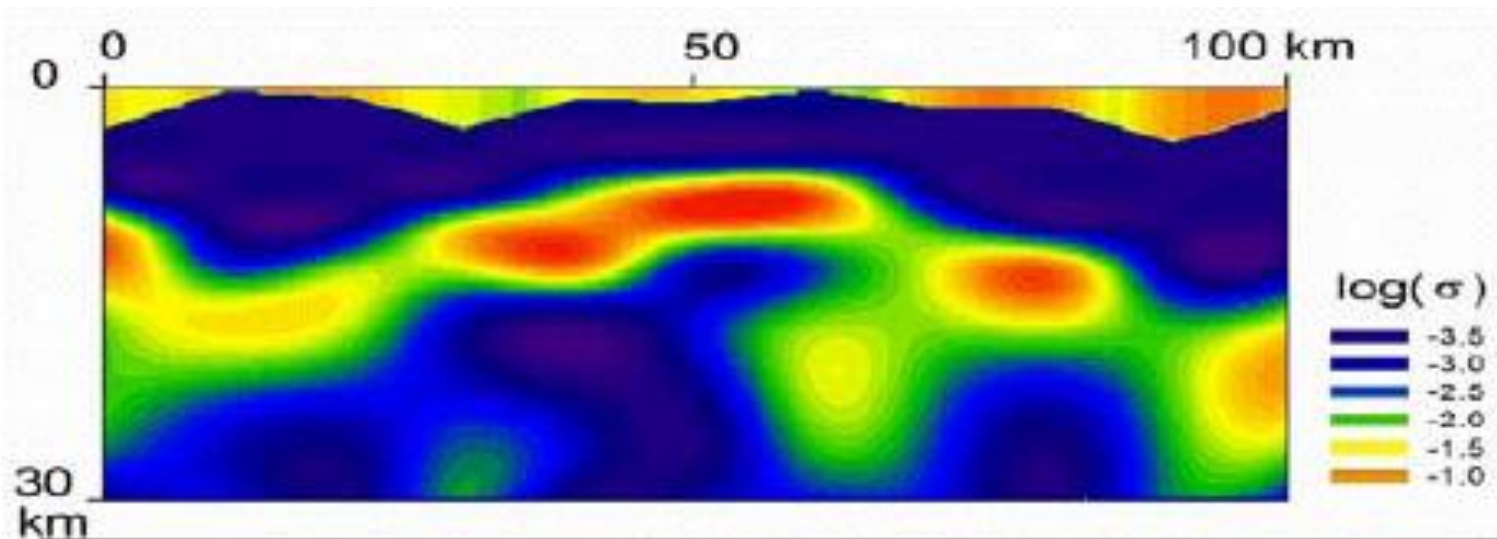
⌘ Пример про депутатов

Обратная задача геофизики. Постановка задачи



⌘ Восстановление распределения
электропроводности

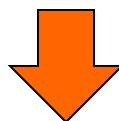
Пример распределения



Математика

⌘ Построить регрессионную модель

- ⌘ Размерность вектора входных данных 6552
- ⌘ Размерность вектора выходных данных 336
- ⌘ Число точек 30000



⌘ 4 комплекта по 1680 многослойных персептронов

- ⌘ МСП как регрессионная модель
- ⌘ Размерность вектора входных данных 1648
- ⌘ Число нейронов в скрытом слое 8
- ⌘ Размерность вектора выходных данных 1
- ⌘ Число точек 30000
- ⌘ Входные данные одинаковы, выходные различаются

Объемы

⌘ 4 * 1680 многослойных персептронов

⌘ Размерность вектора входных данных 1648

⌘ Число нейронов в скрытом слое 8

⌘ Размерность вектора выходных данных 1

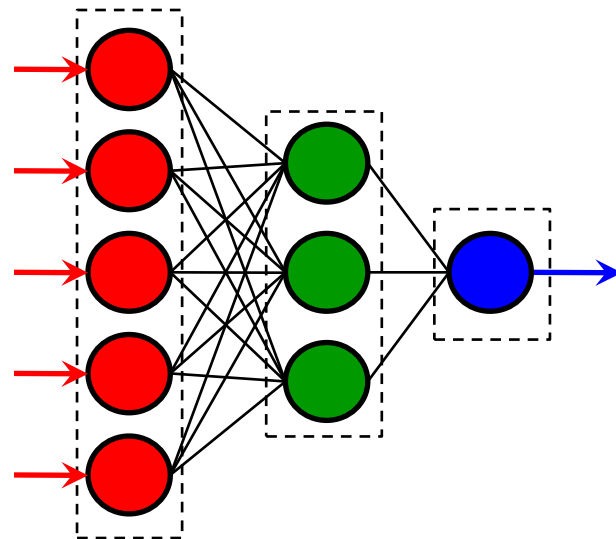
⌘ Число примеров 30000

Матрица весов нейронов скрытого слоя

$$\mathbf{W}_h - (1648 + 1) * 8$$

Матрица весов нейронов выходного слоя

$$\mathbf{W}_o - (8 + 1) * 1$$



Обучение персептрона

- ⌘ Пары **сигнал - желаемый отклик** $\{(\mathbf{X}, \mathbf{Y})\}$
- ⌘ Подаем на вход **сигнал** \mathbf{X}_k , получаем **ответ сети** $\mathbf{Z}_k = \mathbf{Z}_k(\mathbf{W}, \mathbf{X}_k)$
- ⌘ Требуем: **ответ сети = желаемый отклик**

Обучение персептрона

Что бы такого распараллелить...

⌘ Алгоритм обучения: поиск в пространстве весов НС

⏏ Инициализировать \mathbf{W} числами из $[-0.1 .. 0.1]$

⏏ (*) Выбрать группу из K примеров из $\{(\mathbf{X}, \mathbf{Y})\}$

⏏ Рассчитать ответы сети:

⏏ $\mathbf{z}_1(\mathbf{W}, \mathbf{x}_1) = F(\mathbf{W}_o * F(\mathbf{W}_h * \mathbf{x}_1))$

⏏ ...

⏏ $\mathbf{z}_K(\mathbf{W}, \mathbf{x}_K) = F(\mathbf{W}_o * F(\mathbf{W}_h * \mathbf{x}_K))$

⏏ Рассчитать $E(\mathbf{W}) = \sum(|\mathbf{Y}_k - \mathbf{z}_k(\mathbf{W}, \mathbf{x}_k)|^2)$

⏏ $\mathbf{W} -= r * \sum(\partial E / \partial \mathbf{W}(\mathbf{W}, \mathbf{x}_k, \mathbf{y}_k, \mathbf{z}_k))$

⏏ Проверка на останов, если нет, то goto (*)

} Параллельно

} Параллельно

} Параллельно

⌘ Чем K меньше, тем лучшая сеть получится

Обучение персептрона

Что бы такого распараллелить...

⌘ K=10

⌘ Число сетей n = 256

⌘ Рассчитать ответы сети 1:

⊞ $\mathbf{z}_1(\mathbf{w}_1, \mathbf{x}_1) = F(\mathbf{w}_{1o} * F(\mathbf{w}_{1h} * \mathbf{x}_1))$

⊞ ...

⊞ $\mathbf{z}_K(\mathbf{w}_1, \mathbf{x}_K) = F(\mathbf{w}_{1o} * F(\mathbf{w}_{1h} * \mathbf{x}_K))$

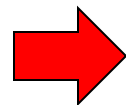
⌘ ...

⌘ Рассчитать ответы сети p:

⊞ $\mathbf{z}_1(\mathbf{w}_n, \mathbf{x}_1) = F(\mathbf{w}_{no} * F(\mathbf{w}_{nh} * \mathbf{x}_1))$

⊞ ...

⊞ $\mathbf{z}_K(\mathbf{w}_n, \mathbf{x}_K) = F(\mathbf{w}_{no} * F(\mathbf{w}_{nh} * \mathbf{x}_K))$



$\mathbf{w}_{all} * \mathbf{x}_{all}$, где

$\mathbf{w}_{all} - 2048 * 1648$

$\mathbf{x}_{all} - 1648 * 10$

Обучение персептрона

Вычисление градиента

$$\text{⌘ } \partial E / \partial \mathbf{W}(\mathbf{W}, \mathbf{X}_k, \mathbf{Y}_k, \mathbf{Z}_k) \Rightarrow \partial E / \partial \mathbf{W}_h(\mathbf{W}_h, \mathbf{X}_k, \mathbf{Y}_k, \mathbf{Z}_k)$$

$$\frac{\partial E}{\partial W_h[net, a, b]} = \sum_{p=0}^9 \alpha_{p, net} \cdot u_{a, net} \cdot f_{a, p, net} \cdot \begin{cases} X_{b-1, p}, b \neq 0 \\ 1, b = 0 \end{cases}$$

$$\text{⌘ } a=0..7, b=0..1648, net=0..255$$

$$\text{⌘ } \text{Число потоков } net * b = 420K$$

$$\text{⌘ } \text{Kernel} \lll net * x, b / x \ggg$$

Результаты

| N | CPU / GPU | Программа, система | Число сетей | Эпох обучения 1 сети за минуту |
|-----|-----------|--------------------|-------------|--------------------------------|
| ➔ 1 | GPU | CUDA, (C) | 256 | 2580 |
| 2 | GPU | CUDA, (B) | 256 | 1818 |
| 3 | GPU | CUDA, (A) | 64 | 144 |
| ➔ 4 | CPU | MLP библиотека (D) | 1 | 35 |
| 5 | CPU | Neuroshell 2 (D) | 1 | 21 |
| 6 | CPU | Matlab 2008a (D) | 1 | 7 |

A. GeForce 8600M GT

B. GeForce GTX 260

C. GeForce GTX 285

D. AMD Athlon 64 x2 Dual 6000+ 3.0 GHz

Итог



⌘ CUDA:

- ☑ **2580** эпох обучения (на 1 сеть за 1 минуту)
- ☑ **13 часов** на все вычисления на **GTX 285**

⌘ CPU:

- ☑ **35** эпох обучения (на 1 сеть за 1 минуту на 1 ядро)
- ☑ Примерно **2 месяца** на **11 ядрах класса AMD 64 x2 3.0 ГГц**

Задачи оптимизации

⌘ Общая постановка задачи:

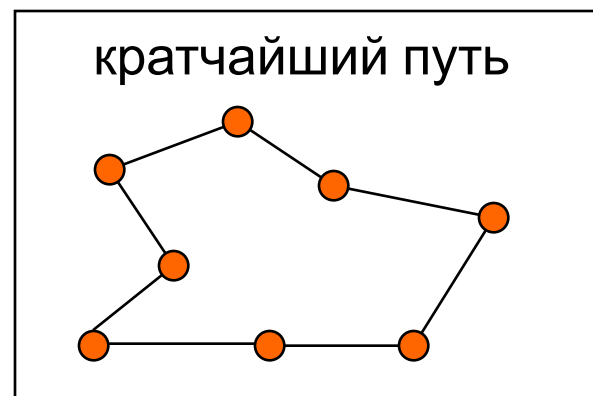
⌘ Найти $\min F(\mathbf{X})$, $\mathbf{X} \in \mathbb{R}^N$, $\mathbf{X} \in U$, где U – множество допустимых значений

⌘ Сложности:

⌘ Большая размерность \mathbf{X}

⌘ «Нехорошая» $F(\mathbf{X})$

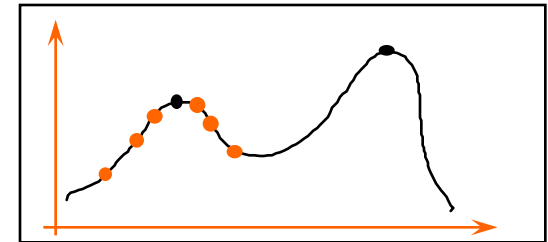
⌘ Комбинаторные задачи



Задачи оптимизации

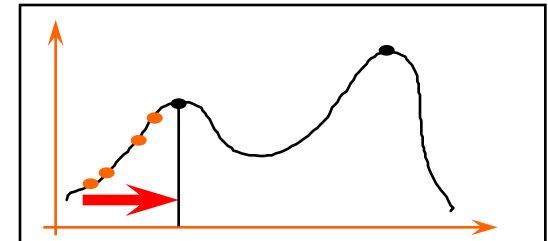
⌘ Локально-оценочные методы

☒ Метод золотого сечения



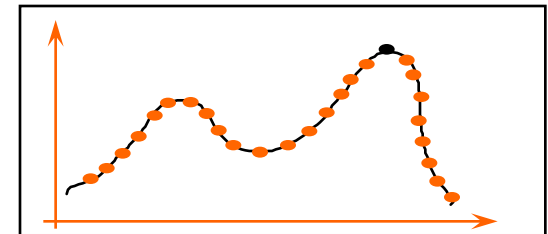
⌘ Локально-градиентные методы

☒ Градиентный спуск



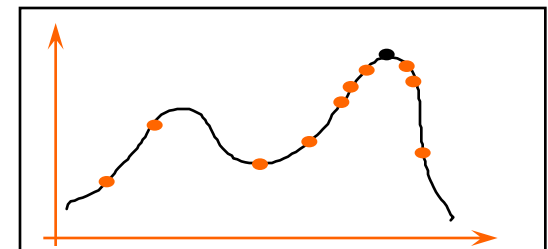
⌘ Переборные методы

☒ Полный перебор



⌘ Многоагентные системы

☒ Генетические алгоритмы



Многоагентные системы



- ⌘ Swarm (рой)

- ⌘ Генетические алгоритмы

 - ☐ Эволюция

 - ☐ Генетическое программирование

- ⌘ Ant Colony Optimization

 - ☐ Колония муравьев

- ⌘ Particle Swarm Optimization

 - ☐ Стаи птиц, косяки рыб

Particle Swarm Optimization

⌘ $F(\mathbf{X})$, $\mathbf{X} \in \mathbb{R}^N$, найти минимум F

⌘ Рой частиц

- ⌘ Состоит из большого числа частиц
- ⌘ Координата лучшего решения роя $g[s]$

⌘ k -ая частиц:

- ⌘ Координата $x_k[s]$, $s=1..N$
- ⌘ Скорость $v_k[s]$
- ⌘ Координата лучшего решения для частицы $p_k[s]$

Particle Swarm Optimization

⌘ Для всех k :

⏏ Инициализировать $x_k, v_k; p_k = x_k$

⌘ Пусть $g = p_m: F(p_m) = \min$ среди всех p_k

⌘ (*) Для всех k

⏏ Выбрать векторы r_p, r_g как $U^N(0,1)$

⏏ Для всех s :

⏏ $vm[s] = a_p * r_p[s] * (p_k[s] - x_k[s])$ – индивидуальное пов-е

⏏ $vs[s] = a_g * r_g[s] * (g[s] - x_k[s])$ – социальное поведение

⏏ $v_k[s] = w * v_k[s] + vm[s] + vs[s]$

⏏ $x_k[s] = v_k[s]$

⌘ Для всех k :

⏏ Если $F(x_k) < F(p_k)$, то $p_k = x_k$

⏏ Если $F(x_k) < F(g)$, то $g = x_k$

⌘ Остановиться или goto (*)

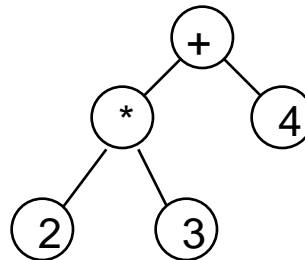
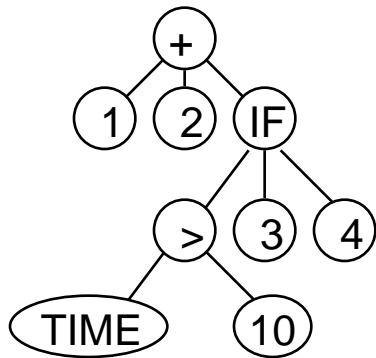
CUDA



- ⌘ Многократное вычисление значения целевой функции F
- ⌘ Большое число роев и частиц

Генетическое программирование

- ⌘ Эволюция программ
- ⌘ Каждая программа оценивается
- ⌘ Программы обмениваются кусками кода
- ⌘ Выделение «хороших» кусков кода



Генетическое программирование

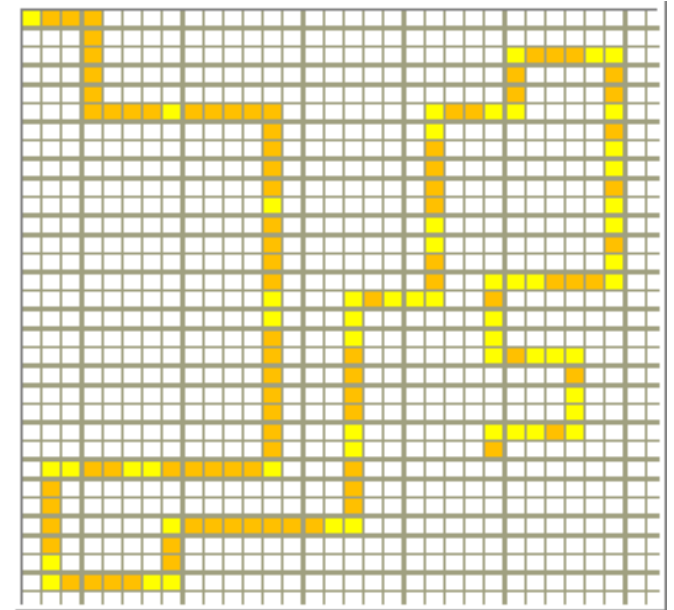
⌘ Пример:

⏏ Построение аналитической формулы

⌘ Пример:

⏏ Искусственный муравей

⊗ LEFT, RIGHT, P2, P3, MOVE
EAT, IF_FOOD_AHEAD



⌘ (IF_FOOD_AHEAD (MOVE) (P3

⌘ (LEFT)

⌘ (P2 (IF-FOOD-AHEAD (MOVE) (RIGHT)) (P2 (RIGHT) (P2 (LEFT) (RIGHT))))

⌘ (P2 (IF-FOOD-AHEAD (MOVE) (LEFT)) (MOVE))))

Вопросы

