Ответы на вопросы к государственному экзамену

1. Многослойные нейронные сети. Алгоритм обратного распространения ошибки

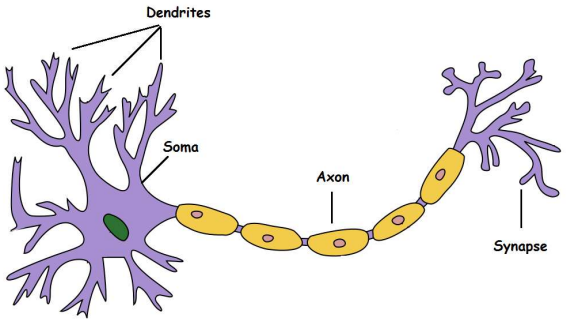
2. Алгоритмическая парадигма плоского заметания. Алгоритм вычисления конечного множества отрезков на плоскости.

3. Классическая процедура динамического программирования для минимизации парно-сепарабельных целевых функций. Функции Беллмана.

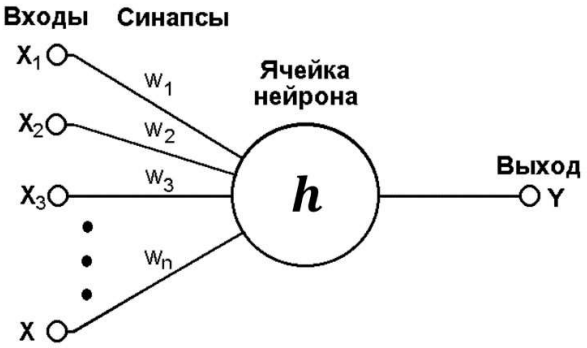
Вопрос №1

Многослойные нейронные сети. Алгоритм обратного распространения ошибки

Многослойные нейронные сети



Формальный нейрон



$$h = \sum_i x_i * w_i \qquad y = f(h)$$

Реализация логических функций искусственным нейроном

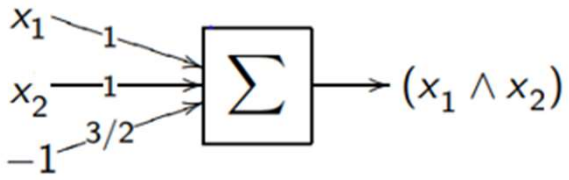
A	B	A&B
0	0	0
0	1	0
1	0	0
1	1	1

$$x_1 \wedge x_2 = \begin{cases} 1, x_1 + x_2 - 1.5 > 0 \\ 0, x_1 + x_2 - 1.5 < 0 \end{cases}$$

Реализация логических функций искусственным нейроном

A	B	A&B
0	0	0
0	1	0
1	0	0
1	1	1

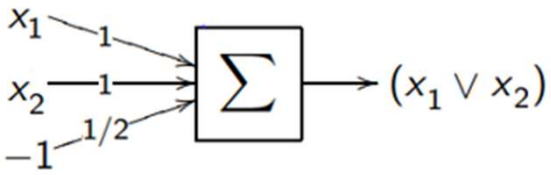
$$x_1 \wedge x_2 = \begin{cases} 1, x_1 + x_2 - 1.5 > 0 \\ 0, x_1 + x_2 - 1.5 < 0 \end{cases}$$



Реализация логических функций искусственным нейроном

A	B	A∨B
0	0	0
0	1	1
1	0	1
1	1	1

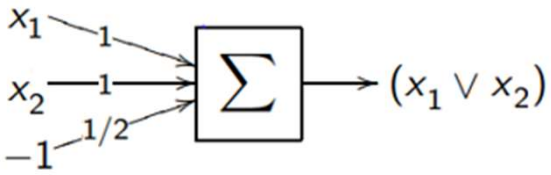
$$x_1 \vee x_2 = \begin{cases} 1, x_1 + x_2 - 0.5 > 0 \\ 0, x_1 + x_2 - 0.5 < 0 \end{cases}$$



Реализация логических функций искусственным нейроном

A	B	A∨B
0	0	0
0	1	1
1	0	1
1	1	1

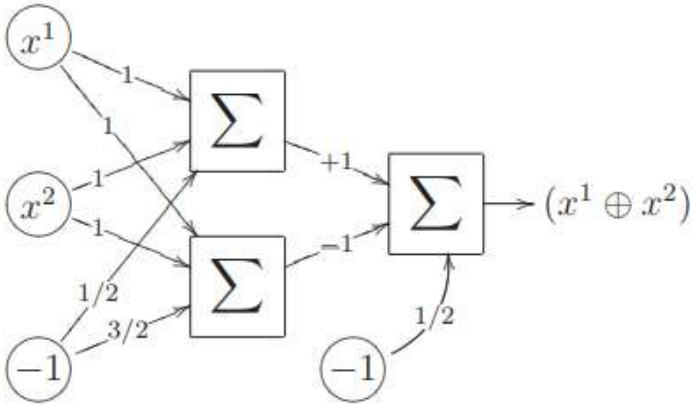
$$x_1 \vee x_2 = \begin{cases} 1, x_1 + x_2 - 0.5 > 0 \\ 0, x_1 + x_2 - 0.5 < 0 \end{cases}$$



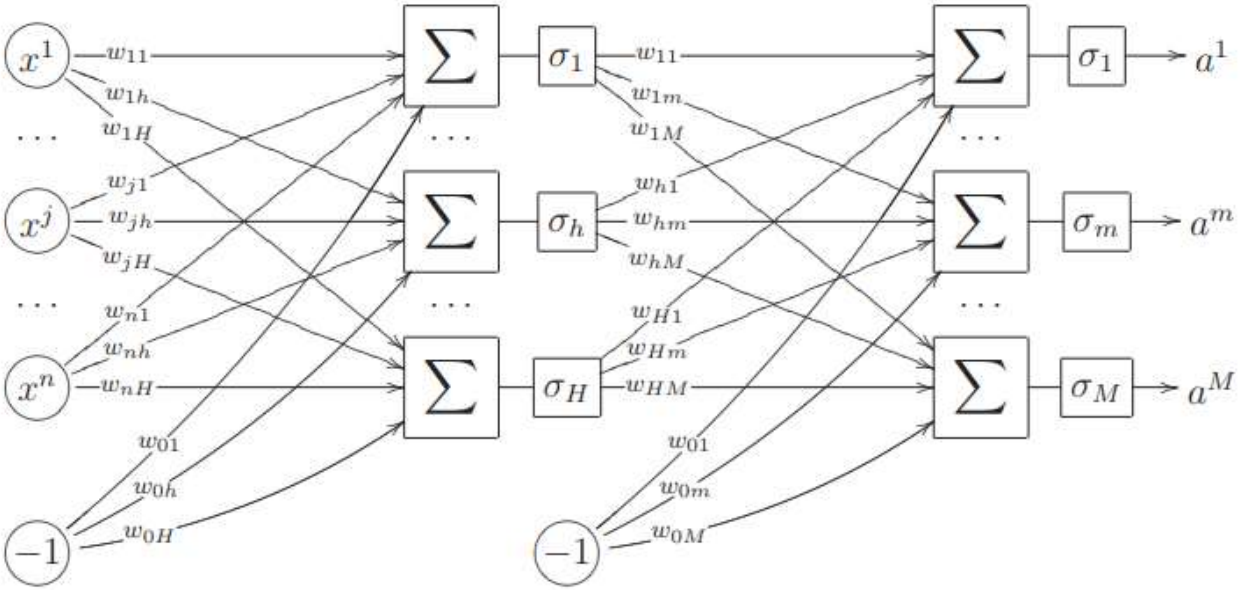
Реализация логических функций искусственным нейроном

A	B	A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

$$x_1 \oplus x_2 = (x_1 \wedge \neg x_2) \vee (x_2 \wedge \neg x_1)$$

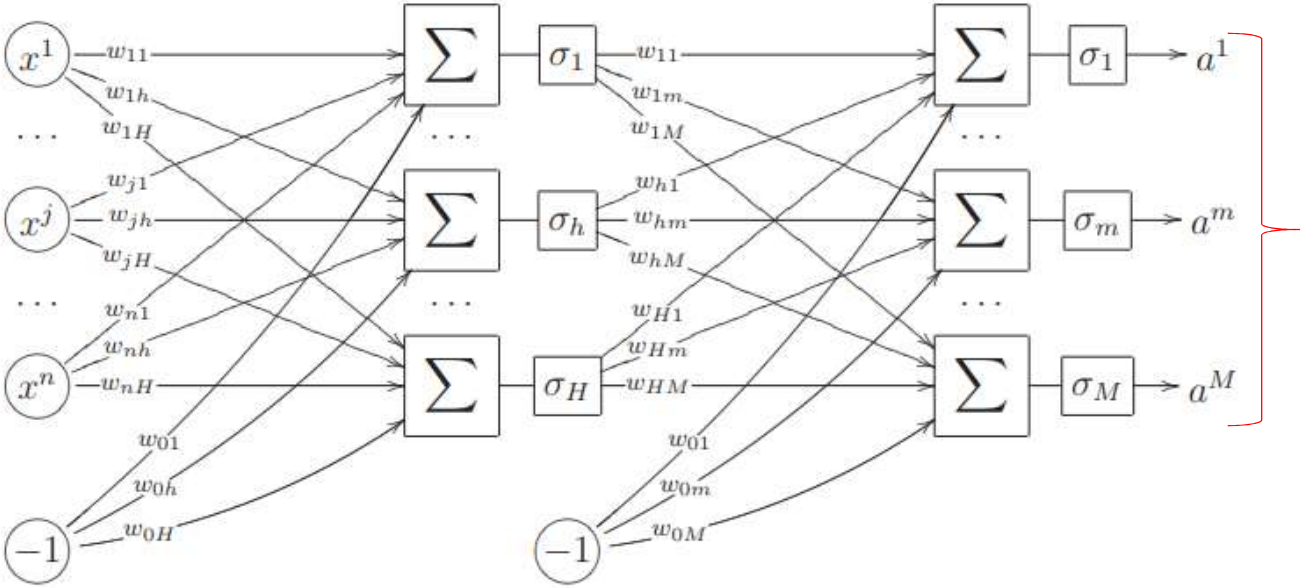


Многослойная нейронная сеть



Метод обратного распространения ошибок

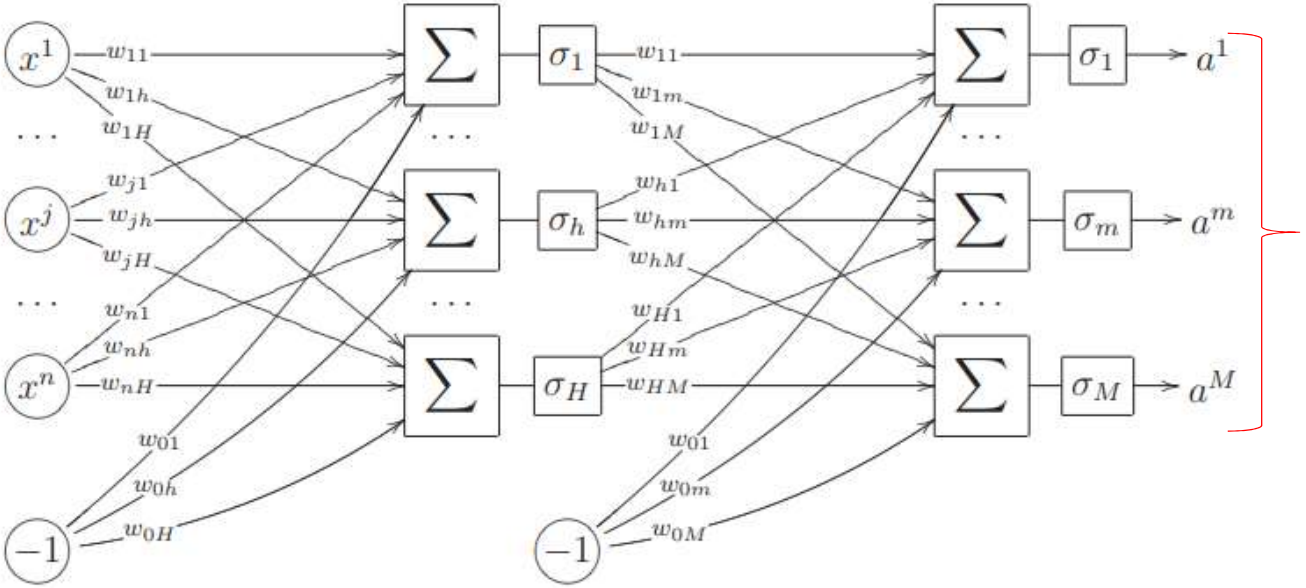
Обозначения:



1) Входной слой состоит из M нейронов

Метод обратного распространения ошибок

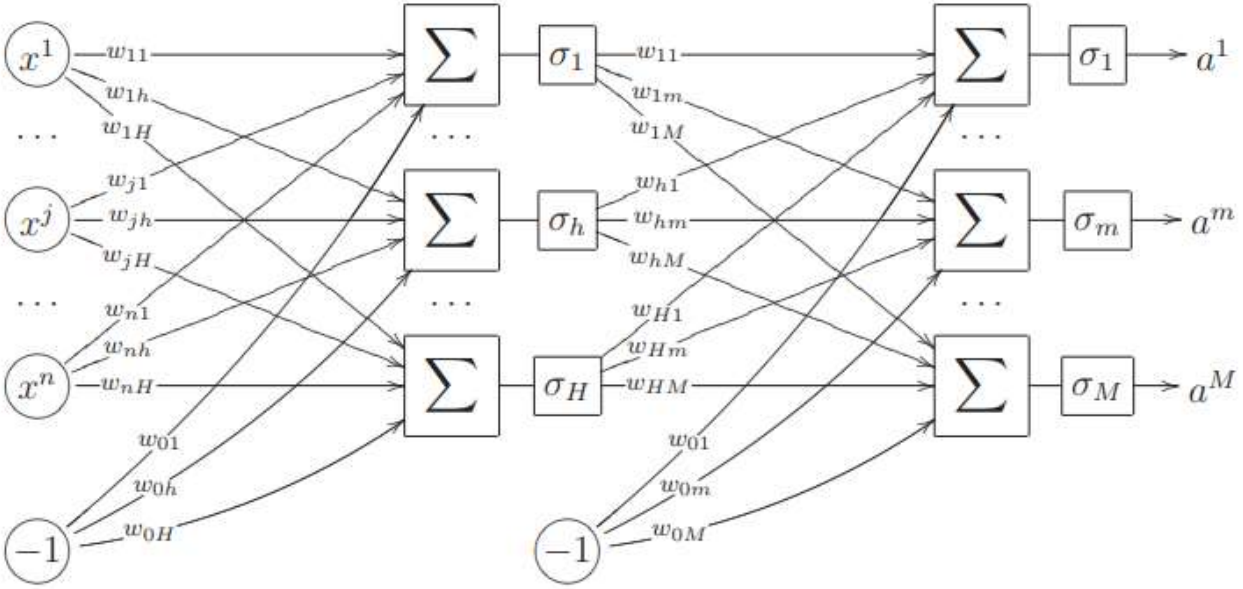
Обозначения:



1) Входной слой состоит из M нейронов
2) Функция активации $\sigma_m(x)$ с выходами a^m

Метод обратного распространения ошибок

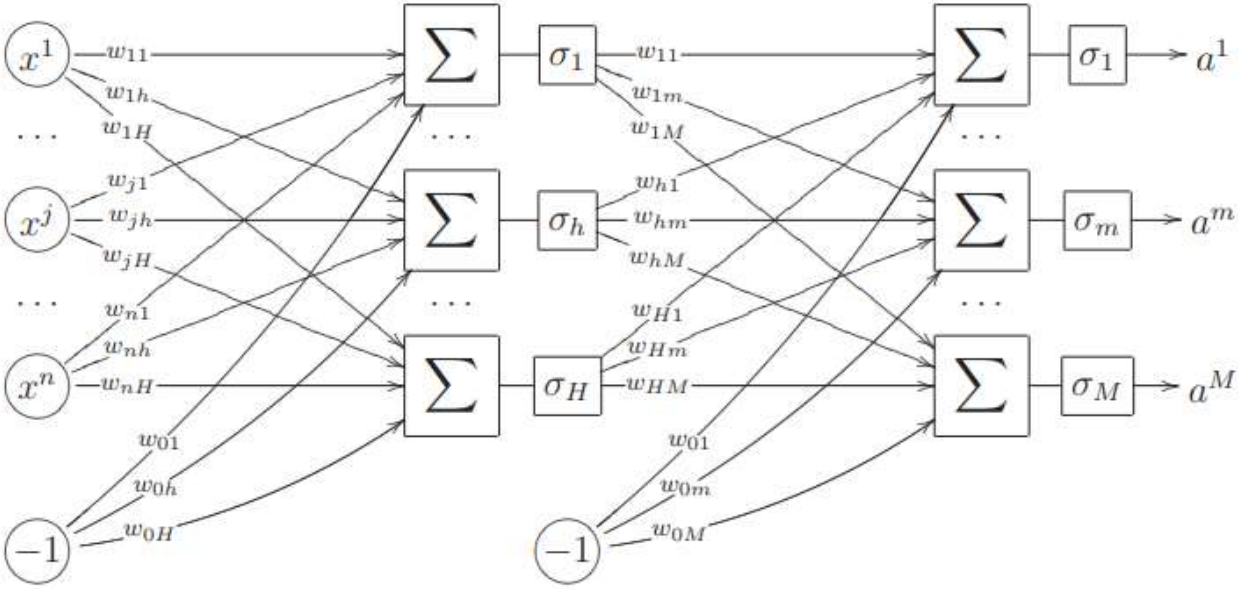
Обозначения:



1) Входной слой состоит из M нейронов
2) Функция активации $\sigma_m(x)$ с выходами a^m
3) Скрытый слой из H нейронов

Метод обратного распространения ошибок

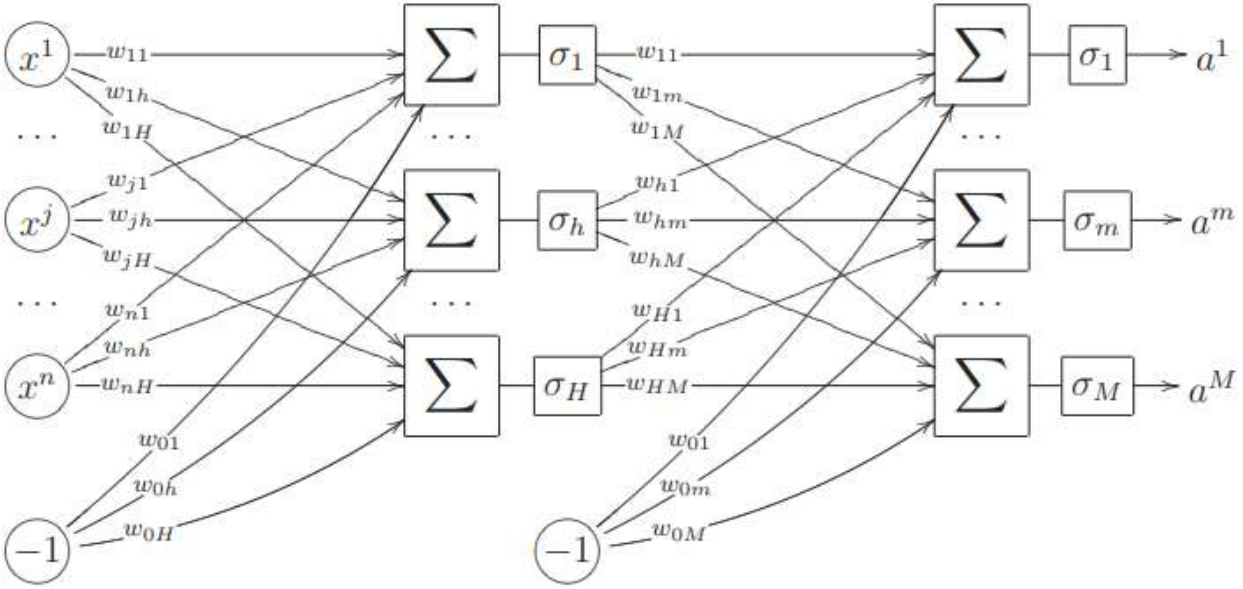
Обозначения:



1) Входной слой состоит из M нейронов
2) Функция активации $\sigma_m(x)$ с выходами a^m
3) Скрытый слой из H нейронов
4) Функции активации $\sigma_h(x)$ с выходами u^h

Метод обратного распространения ошибок

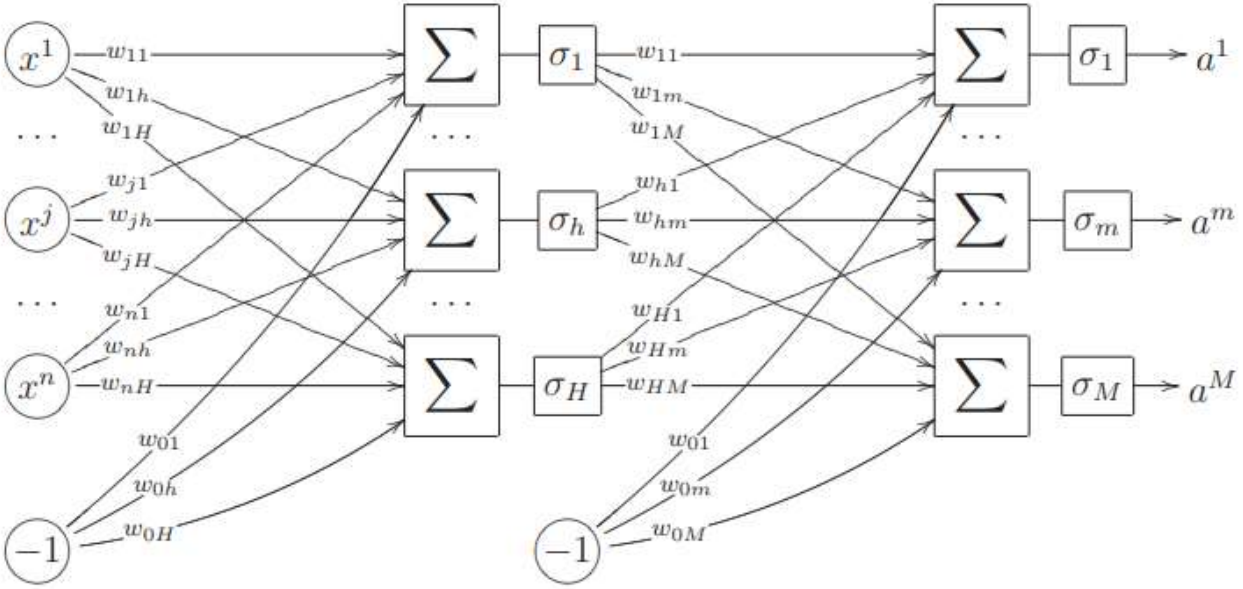
Обозначения:



1) Входной слой состоит из M нейронов
2) Функция активации $\sigma_m(x)$ с выходами a^m
3) Скрытый слой из H нейронов
4) Функции активации $\sigma_h(x)$ с выходами u^h
5) Веса w_{hm}, w_{jh}

Метод обратного распространения ошибок

Обозначения:



Выход на 3 слое нейросети

$$a^m(x_i) = \sigma_m\left(\sum_{h=0}^{H} w_{hm}u^h(x_i)\right)$$

Выход на скрытом слое нейросети

$$u^h(x_i) = \sigma_h\left(\sum_{j=0}^{J} w_{jh}v^j(x_i)\right).$$

Метод обратного распространения ошибок

Функционал ошибки:

$$Q(w) = \frac{1}{2}\sum_{i=1}^{M}\left(a^m(x_i) - y_i^m\right)^2.$$

$$\frac{\partial Q(w)}{\partial a^m} = a^m(x_i) - y_i^m = \varepsilon_i^m$$

Метод обратного распространения ошибок

Функционал ошибки:

$$Q(w) = \frac{1}{2}\sum_{m=1}^{M}\left(a^m(x_i) - y_i^m\right)^2.$$

$$\frac{\partial Q(w)}{\partial a^m} = a^m(x_i) - y_i^m = \varepsilon_i^m$$

Частные производные по выходам скрытого слоя:

$$\frac{\partial Q(w)}{\partial u^h} = \sum_{m=1}^{M}\left(a^m(x_i) - y_i^m\right)\sigma'_m w_{hm} = \sum_{m=1}^{M}\varepsilon_i^m \sigma'_m w_{hm} = \varepsilon_i^h.$$

Метод обратного распространения ошибок

Частные производные по весам:

$$\frac{\partial Q(w)}{\partial w_{hm}} = \frac{\partial Q(w)}{\partial a^m}\frac{\partial a^m}{\partial w_{hm}} = \varepsilon_i^m \sigma'_m u^h(x_i), \qquad m = 1, \ldots, M, \quad h = 0, \ldots, H;$$

$$\frac{\partial Q(w)}{\partial w_{jh}} = \frac{\partial Q(w)}{\partial u^h}\frac{\partial u^h}{\partial w_{jh}} = \varepsilon_i^h \sigma'_h v^j(x_i), \qquad h = 1, \ldots, H, \quad j = 0, \ldots, J;$$

Метод обратного распространения ошибок

Частные производные по весам:

$$\frac{\partial Q(w)}{\partial w_{hm}} = \frac{\partial Q(w)}{\partial a^m}\frac{\partial a^m}{\partial w_{hm}} = \varepsilon_i^m \sigma'_m u^h(x_i), \qquad m = 1, \ldots, M, \quad h = 0, \ldots, H;$$

$$\frac{\partial Q(w)}{\partial w_{jh}} = \frac{\partial Q(w)}{\partial u^h}\frac{\partial u^h}{\partial w_{jh}} = \varepsilon_i^h \sigma'_h v^j(x_i), \qquad h = 1, \ldots, H, \quad j = 0, \ldots, J;$$

Метод обратного распространения ошибок

Достоинства метода:

1) Высокая эффективность
2) Простота реализации на вычислительных устройствах
3) Масштабируемость

Недостатки:

1) Медленная сходимость
2) Застревание в локальных экстремумах

Вопрос №2

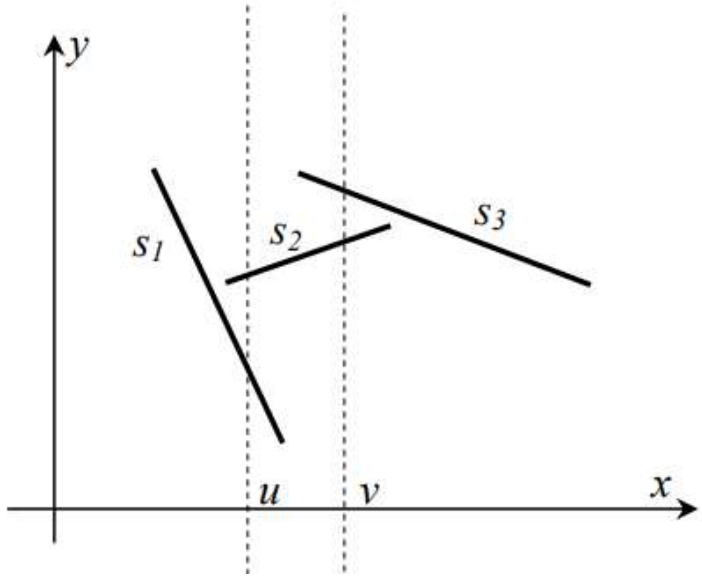
Алгоритмическая парадигма плоского заметания. Алгоритм вычисления пересечений конечного множества отрезков на плоскости.

Задача о пересечении геометрических объектов

Примеры задач:

Пересечение отрезков

N прямолинейных отрезков на плоскости.
Требуется определить факт пересечения хотя бы двух из них

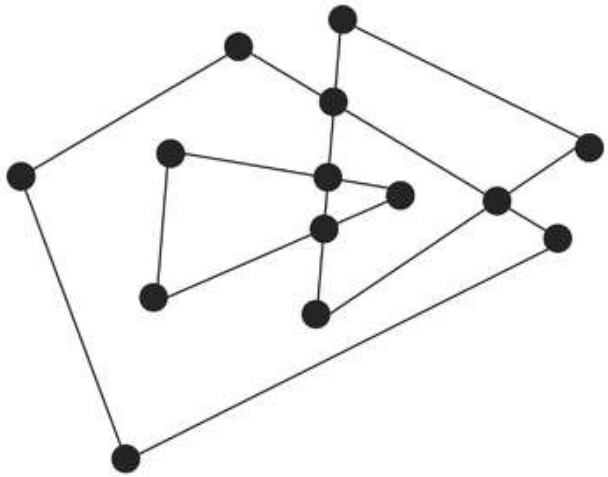


Задача о пересечении геометрических объектов

Примеры задач:

Пересечение простых многоугольников

Даны два простых многоугольника.
Требуется определить, пересекаются ли они.

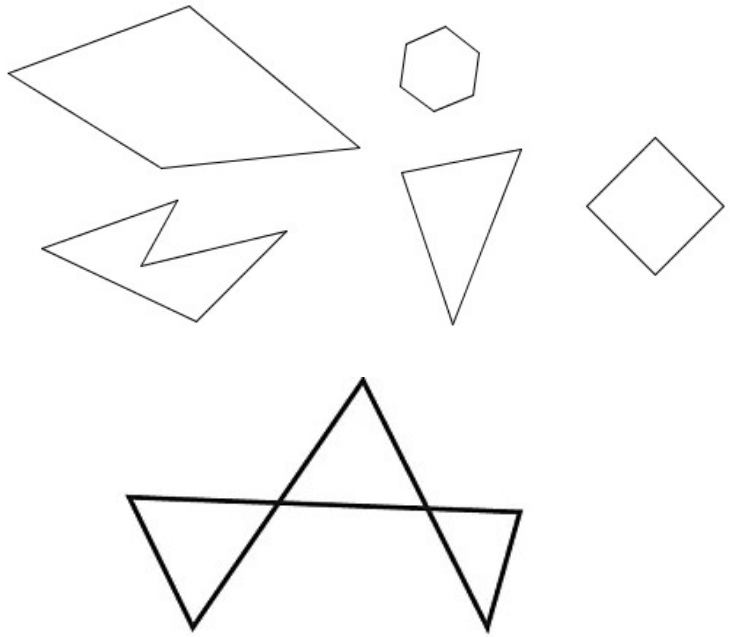


Задача о пересечении геометрических объектов

Примеры задач:

Проверка простоты прямоугольников

Дан многоугольник на плоскости. Проверить, прост ли он.

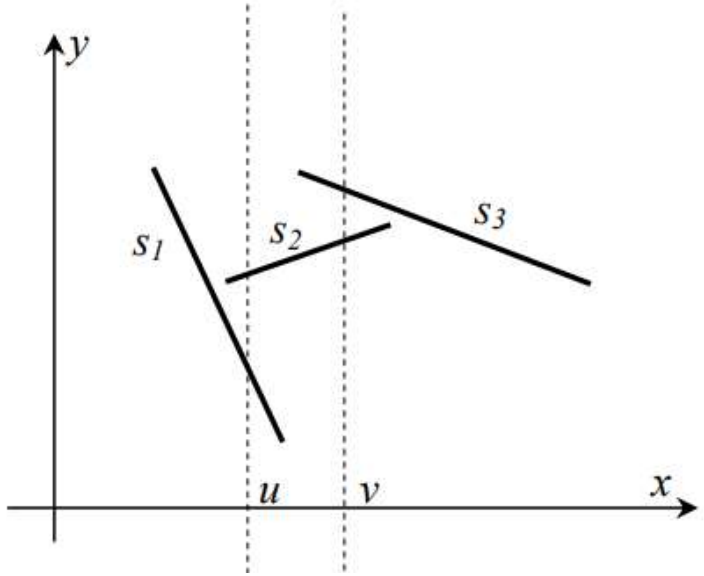


Алгебраическая парадигма плоского заметания

Множество отрезков $\{S_1, S_2, \ldots, S_n\}$

Отношение порядка $>_x$

В данном случае $S_2 >_u S_1, S_3 >_v S_2$



Необходимое условие пересечения пары отрезков:

Если отрезки S_1 и S_2 пересекаются, то существует абсцисса x такая, что S_1 и S_2 являются смежными в упорядочении $>_x$.

Алгебраическая парадигма плоского заметания

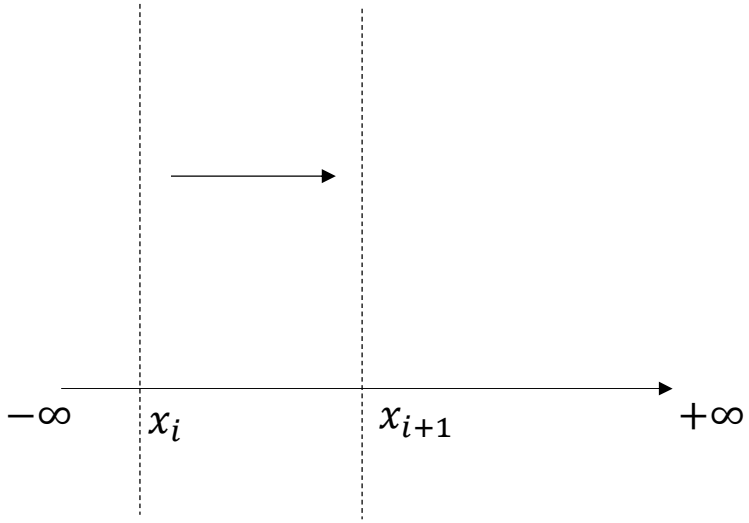
L – статус заметающей прямой.

$$L = \{S_1, S_2, \ldots, S_k\}$$

E — перечень событий

$E = \{x_1, x_2, \ldots, x_m\}$

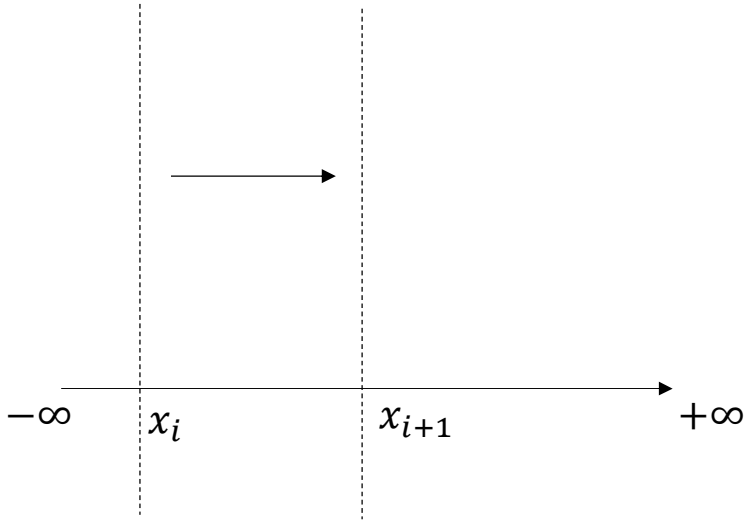
x_i - точка события



Алгебраическая парадигма плоского заметания

Действия:

1) Редактирование перечня событий *E*
2) Редактирование статуса заметающей прямой *L*



Алгебраическая парадигма плоского заметания

Структура, описывающая статус заметающей прямой ***L***

1) Insert(s,L) – установка отрезка в статус
2) Delete(s,L) – удаление отрезка из статуса
3) Upper(s,L) – найти отрезок в статусе, смежный с s и лежащий выше его
4) Under(s,L) - найти отрезок в статусе, смежный с s и лежащий ниже его

Структура, описывающая перечень событий ***E***

1) Min(E) – определить минимальный элемент в E и удалить его
2) Insert(x,E) – вставить элемент x в упорядоченное множество точек событий
3) member(x,E) – является ли элемент x членом E

Алгебраическая парадигма плоского заметания

Алгоритм:

Вход: $\mathbf{S} = \{S_1, S_2, \ldots, S_k\}$ – множество отрезков

Выход: пары пересекающихся отрезков $\{S_1, S_2, \ldots, S_m\}, m \leq k$

1. Sort(**S**)
2. A = ∅
3. Пока E ≠ ∅:

```
p = min(E)
If (p – левый конец отрезка){
        s := отрезок с концом p
        s1 := Upper(s,L)
        s2:= Under(s,L)
        if (s1 пересекается с s) : A.add(s,s1)
        if (s2 пересекается с s) : A.add(s,s2)
}
```

Алгоритм:

```
else If (p – правый конец отрезка){
        s := отрезок с концом p
        s1 := Upper(s,L)
        s2:= Under(s,L)
        if (s1 пересекает s2) : A.add(s1,s2)
        delete(s,L)
}
else If (p – правый конец отрезка){
        s1,s2 := отрезки, пересекающиеся в p
        s3 := Upper(s,L)
        s4:= Under(s,L)
        if (s3 пересекается с s2) : A.add(s3,s2)
        if (s1 пересекается с s4) : A.add(s1,s4)
        replace s1, s2
}
```

Вопрос №3

Классическая процедура динамического программирования для минимизации парно-сепарабельных целевых функций. Функции Беллмана.

Классическая процедура динамического программирования

Массив данных $\boldsymbol{Y} = \{y_t, t \in T\}$

Искомый массив данных: $\boldsymbol{X} = \{x_t, t \in T\}$

Целевая функция $J(X|Y)$ - степень несоответствия *X* и *Y*

Классическая процедура динамического программирования

Результат анализа массива данных:

$$\hat{X}(Y) = \arg\min_{x_t \in \mathsf{X}} J(X \mid Y).$$

Классическая процедура динамического программирования

Предварительная оценка значения отдельно взятой x_t:

$$\psi_t(x) = \psi_t(x \mid Y_t)$$

Однако неизбежные возмущения различной природы не позволяют принять

$$\hat{x}_t(Y_t) = \arg\min_{x_t \in X} \psi(x_t \mid Y_t),\ t \in T$$

Классическая процедура динамического программирования

Предварительная оценка значения отдельно взятой x_t:

$$\psi_t(x) = \psi_t(x \mid Y_t)$$

Однако неизбежные возмущения различной природы не позволяют принять

$$\hat{x}_t(Y_t) = \arg\min_{x_t \in X} \psi(x_t \mid Y_t),\ t \in T$$

Классическая процедура динамического программирования

И в тоже время специфика задачи предполагает наличии некоторых модельных знаний об ожидаемой комбинации значений скрытых переменных.

$\gamma_{t't''}(x_{t'}, x_{t''})$ - функция связи

Классическая процедура динамического программирования

Комбинированный критерий:

$$J(X \mid Y) = \sum_{t \in T} \psi_t(x_t \mid Y_t) + \sum_{(t',t'') \in G} \gamma_{t't''}(x_{t'}, x_{t''})$$

Данная функция – сепарабельна, граф G – граф смежности аргументов сепарабельной целевой функции

Классическая процедура динамического программирования

Комбинированный критерий:

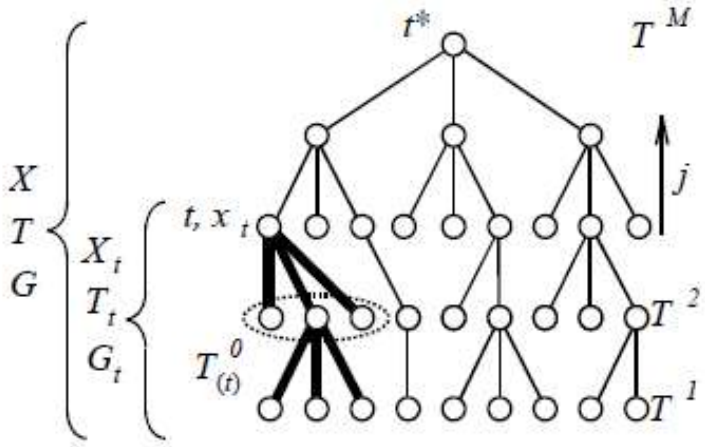
$$J(X \mid Y) = \sum_{t \in T} \psi_t(x_t \mid Y_t) + \sum_{(t',t'') \in G} \gamma_{t't''}(x_{t'}, x_{t''})$$

Проблема оптимизации – вычислительная трудность.

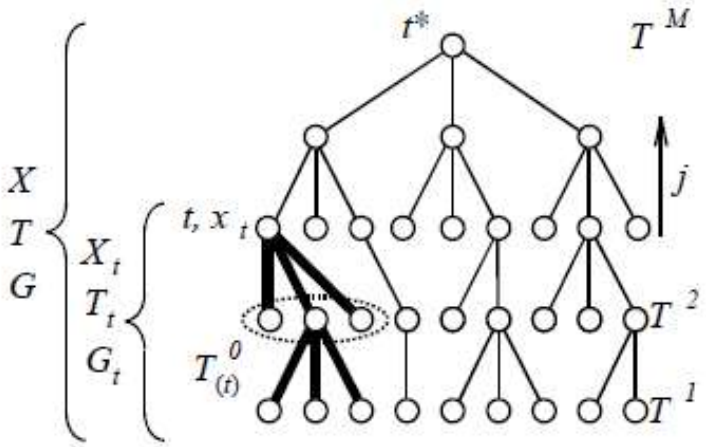
Классическая процедура динамического программирования

Обобщение классической процедуры динамического программирования

$$J(X) = \sum_{t \in T} \psi_t(x_t) + \sum_{(t',t'') \in G} \gamma_{t't''}(x_{t'}, x_{t''}), \; x_t \in X_t$$ - древовидная сепарабельная целевая функция



Классическая процедура динамического программирования



t^* – корень дерева

$T = \bigcup_{j=1}^{M} T^j$, $T^j \bigcap T^k = \varnothing$, $j \neq k$ - разбиение множества узлов

T_t – множество всех узлов в дереве потомков узла t

$$T_{(t)} = T_t \backslash \mathrm{t}$$

$T_{(t)}^0 \subseteq T_{(t)}$ - подмножество непосредственных потомков узла t

$X_t = (x_s, s \in T_t)$ и $X_{(t)} = (x_s, s \in T_{(t)})$ - соответствующие частичные совокупности узловых переменных

Классическая процедура динамического программирования

Частичная целевая функция

$$J_t(X_t) = J_t(x_t, X_{(t)}) = \sum_{s \in T_t} \psi_s(x_s) + \sum_{(s',s'') \in G_t} \gamma_{s's''}(x_{s'}, x_{s''}),$$

откуда получаем

$$J_t(X_t) = J_t(x_t, X_{(t)}) = \psi_t(x_t) + \sum_{s \in T^0_{(t)}} \left\{ \gamma_{ts}(x_t, x_s) + J_s(x_s, X_{(s)}) \right\}.$$

Классическая процедура динамического программирования

Основная идея процедуры оптимизации основана на понятии функции Беллмана

$$\tilde{J}_t(x_t) = \min_{X_{(t)}} J_t(x_t, X_{(t)}) = \psi_t(x_t) + \min_{x_s, X_{(s)}, s \in T^0_{(t)}} \sum_{s \in T^0_{(t)}} \left\{ \gamma_{ts}(x_t, x_s) + J_s(x_s, X_{(s)}) \right\}, \quad x_t \in X_t$$

Фундаментальное свойство $\tilde{J}_t(x_t) = \psi_t(x_t) + \sum_{s \in T^0_{(t)}} \min_{x_s \in X_s} \left\{ \gamma_{ts}(x_t, x_s) + \tilde{J}_s(x_s) \right\}$ примем за прямое рекуррентное соотношение

$$\tilde{x}_s(x_t) = \arg\min_{x_s \in X_s} \left\{ \gamma_{ts}(x_t, x_s) + \tilde{J}_s(x_s) \right\}, \quad s \in T^0_{(t)}$$

примем за обратное рекуррентное соотношение

Классическая процедура динамического программирования

прямое рекуррентное соотношение:

$$\widetilde{J}_t(x_t) = \psi_t(x_t),\ x_t \in X_t,\ t \in T^1.$$ поскольку $$T^0_{(t)} = \varnothing$$

В корне получаем:

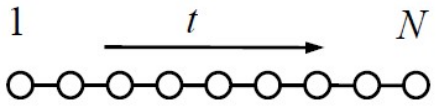
$$\hat{x}_{t^*} = \arg\min_{x_{t^*} \in X_{t^*}} \widetilde{J}_{t^*}(x_{t^*}).$$

примем за обратное рекуррентное соотношение

$$\hat{x}_s = \widetilde{x}_s(\hat{x}_t),\ s \in T^0_{(t)}.$$

Классическая процедура динамического программирования

Рассмотрим частный случай, когда дерево не имеет ветвлений



$$J(X)=\sum_{t=1}^{N}\psi_t(x_t)+\sum_{t=2}^{N}\gamma_t(x_{t-1},x_t).$$ – сепарабельная функция

Классическая процедура динамического программирования

Рассмотрим частный случай, когда дерево не имеет ветвлений

Процедура начинается в $t = 1$ и равна

$$\widetilde{J}_1(x_1) = \psi_1(x_1)$$

$$\widetilde{J}_t(x_t) = \psi_t(x_t) + \min_{x_{t-1} \in X_{t-1}} \{\gamma_t(x_{t-1}, x_t) + \widetilde{J}_{t-1}(x_{t-1})\}$$

Обратные рекуррентные соотношения

$$\widetilde{x}_{t-1}(x_t) = \arg\min_{x_{t-1} \in X_{t-1}} \{\gamma_t(x_{t-1}, x_t) + \widetilde{J}_{t-1}(x_{t-1})\}, \quad x_t \in X$$

Спасибо за внимание!