Нечеткие множества. Операции над нечеткими множествами

Нечеткая логика и нейронные сети

Термин "*нечеткая логика*"

- В узком смысле,
- нечеткая логика — это логическое исчисление, являющееся расширением многозначной логики.
- В широком смысле
- *нечеткая логика* равнозначна теории нечетких множеств.

Основатель

- Впервые термин нечеткая логика
- (fuzzy logic) был введен амерканским профессором Лотфи Заде в 1965 году в работе “Нечеткие множества” в журнале “Информатика и управление”.

Родился в БакуРодился в Баку, Азербайджан как **Лотфи Алескерзаде** (или **Аскер Заде**) от русской матери и отца азербайджанца иранского происхождения; с 1932) от русской матери и отца азербайджанца иранского происхождения; с 1932 года жил в Иране) от русской матери и отца азербайджанца иранского происхождения; с 1932 года жил в Иране, учился Тегеранском университете; с 1944 в Соединенных Штатах; работает в Калифорнийском университете (Беркли).

Пример

- В феврале 1991 года была сконструирована первая <интеллектуальная> стиральная машина, в системе управления которой сочетались нечеткая логика.

- Автоматически определяя нечеткие входные факторы :
- объем и качество белья,
- уровень загрязненности,
- тип порошка и т.д.),
- стиральная машина выбирала оптимальный режим стирки из 3800 возможных.

Примеры применения нечеткой логики:

- Распознавание рукописных символов в **карманных компьютерах (записных книжках)**
  (Sony)
- Однокнопочное управление **стиральными машинами**
  (Matsushita, Hitatchi)
- **Распознавание** рукописных текстов, объектов, голоса
  (CSK, Hitachi, Hosai Univ., Ricoh)
- Управление **метрополитенами** для повышения удобства вождения, точности остановки и экономии энергии (*Hitachi)*
- Оптимизация потребления бензина в **автомобилях**
  (NOK, Nippon Denki Tools)
- Повышение чувствительности и эффективности **управления лифтами**
  (Fujitec, Hitachi, Toshiba)

Примеры применения нечеткой логики:

- Автоматическое управление воротами плотины на **гидроэлектростанциях**
- Упрощенное управление **роботами**
- **Наведение телекамер** при трансляции спортивных событий
  Эффективное и стабильное управление **автомобильными двигателями**
  Управление экономичной скоростью **автомобилей**
  (Nissan, Subaru)
- Оптимизированное планирование **автобусных расписаний** *(Toshiba)*
- Системы архивации **документов**
  (Mitsubishi Elec.)
- **Системы прогнозирования** землетрясений(*Japan)*
- Диагностика рака
  (Kawasaki Medical School)

Нечеткое множество

*Основы нечеткой логики были заложены в конце **60**-х лет в работах известного американского математика*

Латфи Заде

- Пусть *E* – универсальное множество, *x* – элемент *E*, а *R* – определенное свойство.

Тогда нечеткое подмножество *A* универсального множества *E* определяется как множество упорядоченной пары

$$A = \{\mu_A(x) / x\},$$

где $\mu_A(x)$ – характеристическая функция принадлежности (или просто функция принадлежности), принимающая значение в некотором упорядоченном множестве M (например, M = [0,1]).

Функция принадлежности указывает степень (или уровень) принадлежности элемента *x* к подмножеству *A*.

Примеры записи нечеткого множества

- Пусть E={x1, x2, x3, x4, x5}, M=[0,1]; A – элемент множества, для которого

$\mu_A(x_1) = 0,2$ $\mu_A(x_2) = 0$ $\mu_A(x_3) = 0,4$ $\mu_A(x_4) = 1$ $\mu_A(x_5) = 0,7$

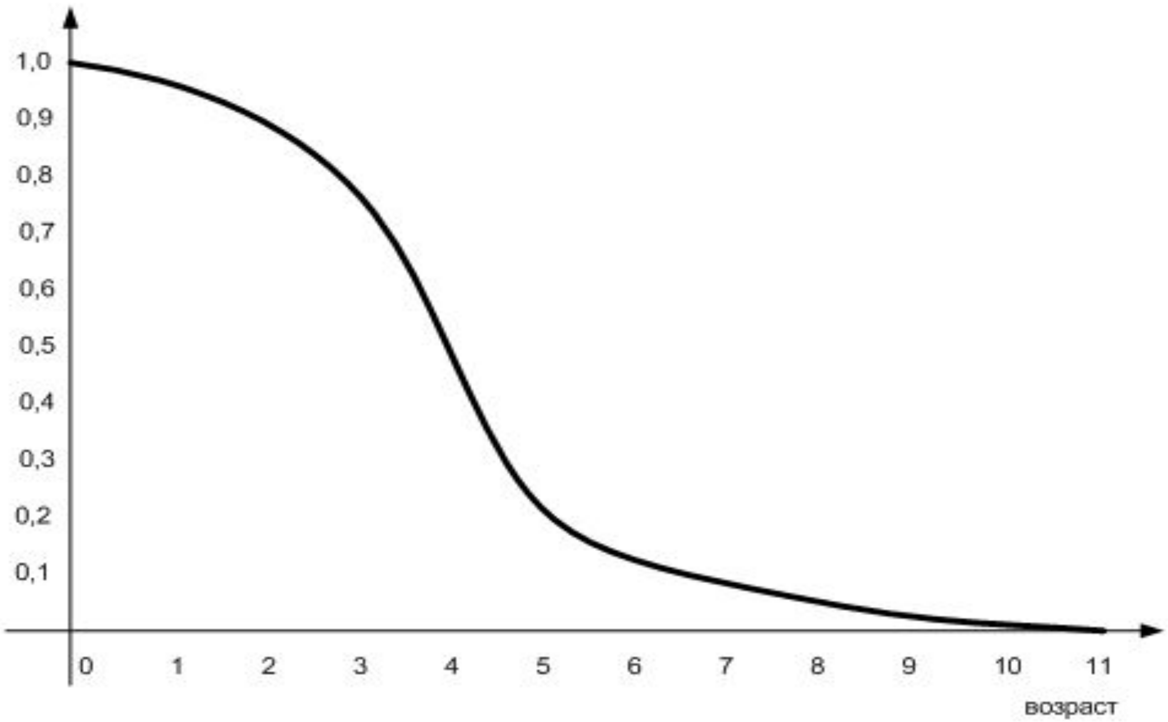
- Тогда A можно представить в виде:

1) A={0,2/x1;0/x2;0,4/x3;1/x4;0.7/x5},
2) A={0,2/x1+0/x2+0,4/x3+1/x4+0,7/x5},
3) A=

x1	x2	x3	x4	x5
0,2	0	0,4	1	0,7

Пример нечеткого множества

$$"\text{младенческий}" = \left\{ \frac{0.5}{1} + \frac{1}{0.9} + \frac{2}{0.8} + \frac{3}{0.7} + \frac{4}{0.6} + \frac{5}{0.3} + \frac{11}{0.1} \right\}$$



Основные характеристики нечётких множеств

Пусть *M*=[0,1] и *A* – нечеткое множество с элементами из универсального множества *E* и множеством принадлежностей *M*.

Высота: $\sup\limits_{x \in E} \mu_A(x)$.

Если $\sup\limits_{x \in E} \mu_A(x) = 1$, то нечёткое множество *A* нормально.

Если $\sup\limits_{x \in E} \mu_A(x) < 1$, то нечёткое множество *A* субнормально.

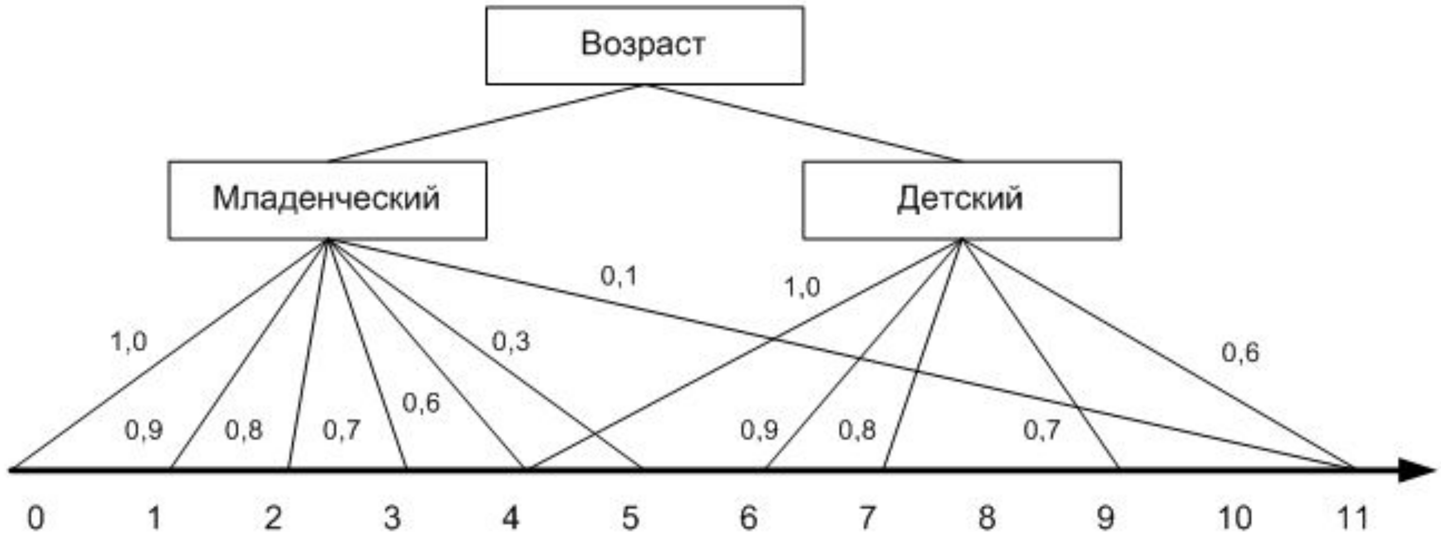
- Нечеткое множество пусто, если $\forall x \in E \quad \mu_A(x) = 0$.
- Непустое субнормальное множество можно нормализовать по формуле: $\mu_A(x) := \dfrac{\mu_A(x)}{\sup\limits_{x \in E} \mu_A(x)}$
- Нечеткое множество унимодально, если $\mu_A(x) = 1$ только в одном *x* из *E*.
- Носителем нечеткого множества А является обычное подмножество со свойством $\mu_A(x) > 0$, т.е. $A = \{x / x \in E, \mu_A(x) > 0\}$
- Элементы $x \in E$, для которых $\mu_A(x) = 0,5$, называются точками перехода множества А.
- α -уровневое подмножество из А это множество в котором $\mu_A(x) \geq \alpha$
- Пример: «Несколько»=0,5/3+0,8/4+1/5+1/6+0,8/7+0,5/8; его характеристики: высота=1, носитель ={3,4,5,6,7,8}, точки перехода – {3,8}.

Лингвистическая переменная «Возраст»

Пусть перед нами стоит задача интерпретации значений ЛП «*возраст»,* таких как «молодой» возраст, «преклонный» возраст или «переходный» возраст. Определим «возраст» как ЛП. Тогда «молодой», «преклонный», «переходный» будут значениями этой лингвистической переменной. Более полный базовый набор значений ЛП «возраст» следующий:

B={младенческий, детский, юный, молодой, зрелый, преклонный, старческий}.

Для ЛП «возраст» базовая шкала — это числовая шкала от 0 до 120, обозначающая количество прожитых лет, а функция принадлежности определяет, насколько мы уверены в том, что данное количество лет можно отнести к данной категории возраста.



Характеристики нечетких множеств

- Величина $\sup\limits_{x\in E}\mu_A(x)$ называется *высотой* нечеткого множества A. Нечеткое множество A *нормально*, если его высота равна 1, т.е. верхняя граница его функции принадлежности равна 1 ($\sup\limits_{x\in E}\mu_A(x) = 1$). При $\sup\limits_{x\in E}\mu_A(x) < 1$ нечеткое множество называется *субнормальным*.
- Нечеткое множество *пусто*, если $\forall x \in E \;\; \mu_A(x) = 0$. Непустое субнормальное множество можно нормализовать по формуле

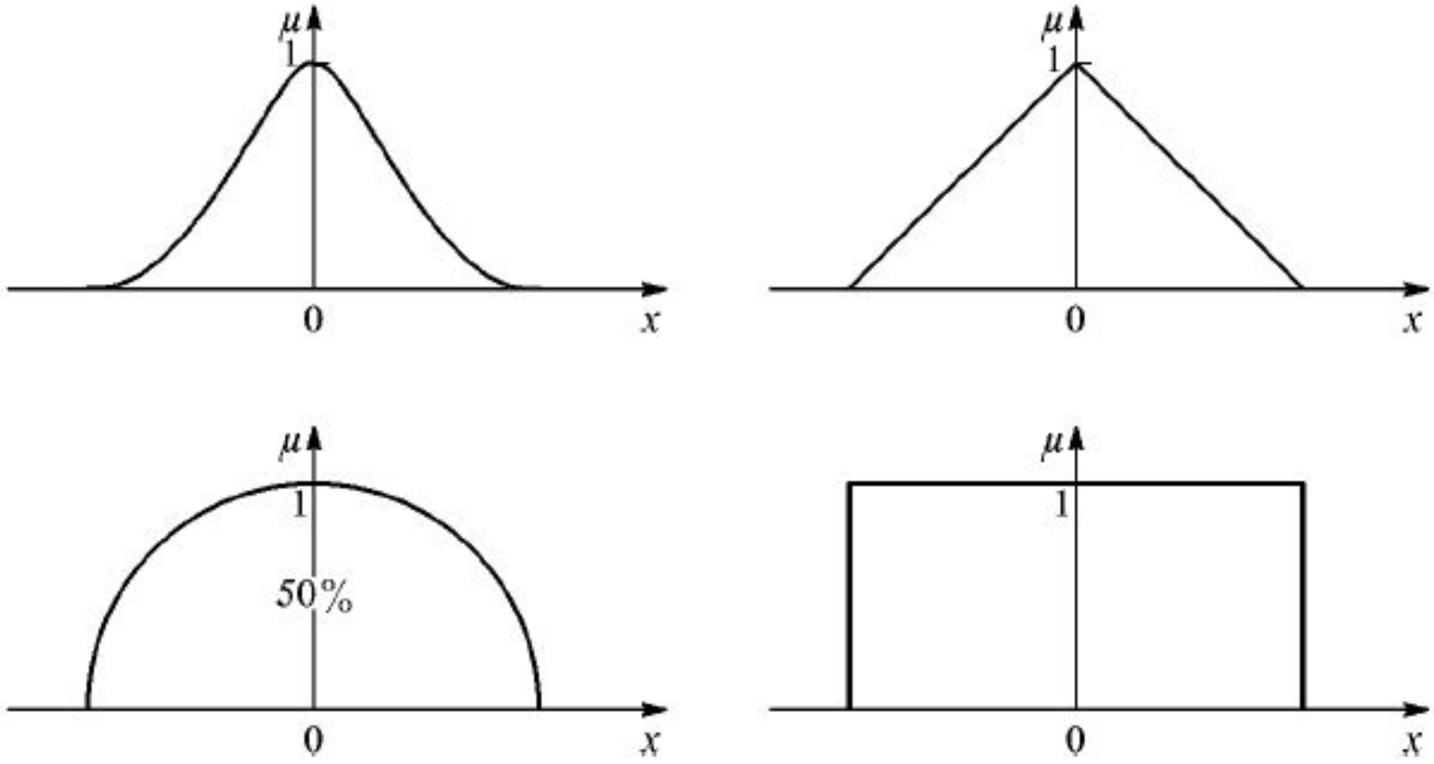
$$\mu_A(x) := \frac{\mu_A(x)}{\sup\limits_{x\in E}\mu_A(x)}.$$

- Нечеткое множество *унимодально*, если $\mu_A(x) = 1$ только на одном x из E.
- *Носителем* нечеткого множества A является обычное подмножество со свойством $\mu_A(x) > 0$, т.е. *носитель* $A = \{x/x \in E, \mu_A(x) > 0\}$.
- Элементы $x \in E$, для которых $\mu_A(x) = 0{,}5$, называются *точками перехода* множества A.

Методы определения функции принадлежности

- Прямые (опросы экспертов)
- Косвенные (парные сравнения)
- L-R - функции

L-R нечеткие числа



Операции над нечёткими множествами

Логические операции

1. *Включение.* Пусть *A* и *B* – нечеткие множества на универсальном множестве E. Тогда *A* содержится в *B*, если $\forall x \in E \quad \mu_A(x) \le \mu_B(x)$
Обозначение: $A \subset B$

2. *Равенство.* А и В равны, если $\forall x \in E \quad \mu_A(x) = \mu_B(x)$ Обозначение: A=B

3. *Дополнение.* Пусть М = [0,1], А и В – нечеткие множества, заданные на Е. А и В дополняют друг друга, если $\forall x \in E \quad \mu_A(x) = 1 - \mu_B(x)$
Обозначение: $B = \bar{A}$

4. *Пересечение* – наибольшее нечеткое подмножество, содержащее одновременно А и В ($A \cap B$): $\mu_{A \cap B}(x) = \min(\mu_A(x),\ \mu_B(x))$

5. *Объединение* – наименьшее нечеткое подмножество, включающее как А, так и В, с функцией принадлежности ($A \cup B$):

$$\mu_{A \cup B}(x) = \max(\mu_A(x),\ \mu_B(x))$$

6. *Разность* – операция с функцией принадлежности ($A - B = A \cap \bar{B}$):

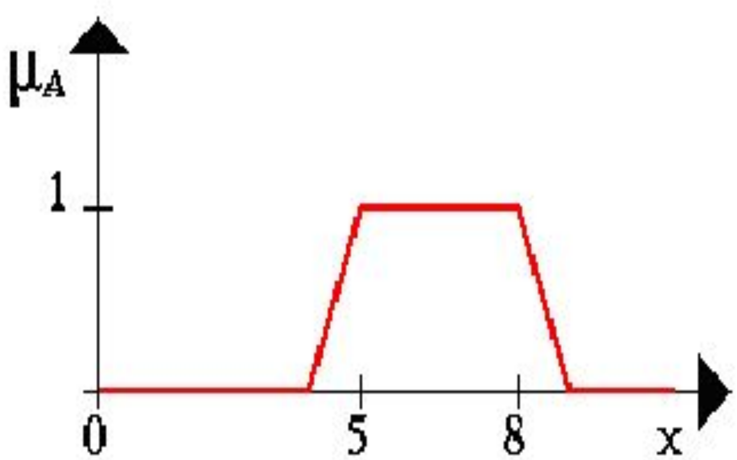
$$\mu_{A-B}(x) = \mu_{A \cap \bar{B}}(x) = \min(\mu_A(x),\ 1 - \mu_B(x))$$

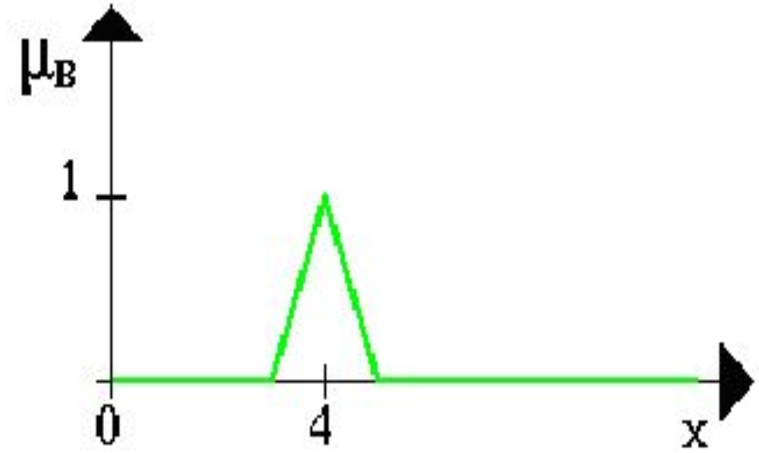
7. *Дизъюнктивная сумма* – логическая операция с функцией принадлежности ($A \oplus B = (A - B) \cup (B - A) = (A \cap \bar{B}) \cup (\bar{A} \cap B)$):

$$\mu_{A-B}(x) = \max(\min(\mu_A(x),\ 1 - \mu_B(x)); \min(1 - \mu_A(x),\ \mu_B(x)))$$

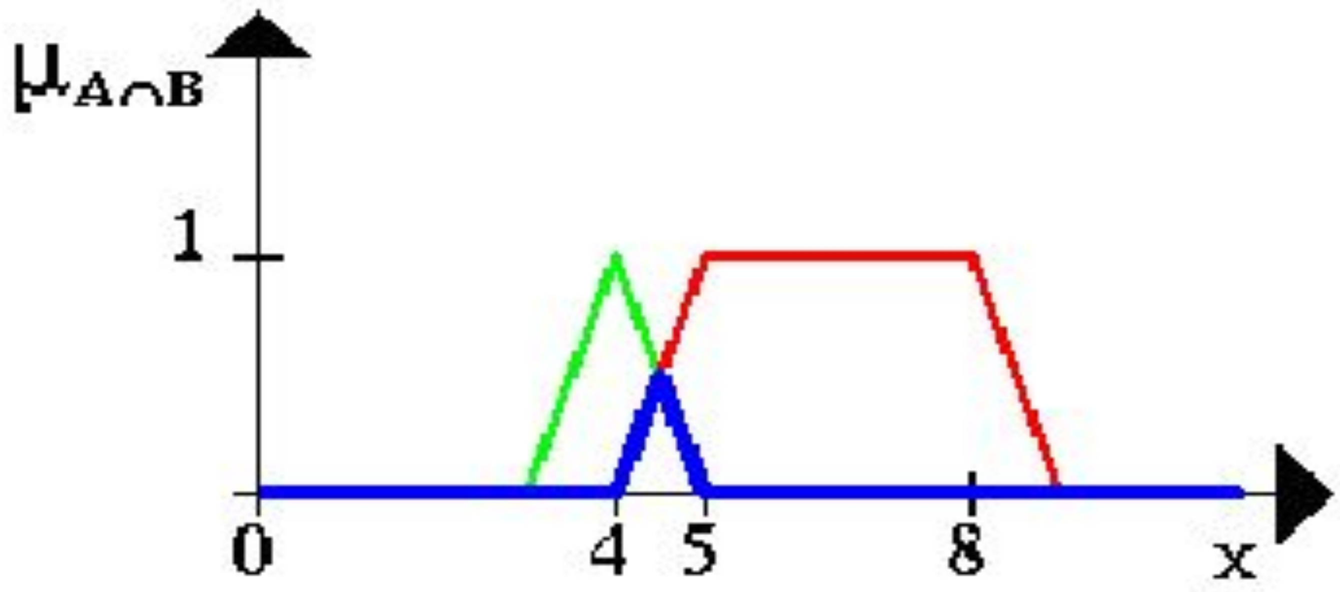
Пример

- Пусть *A* нечеткий интервал от 5 до 8 и *B* нечеткое число *около* 4

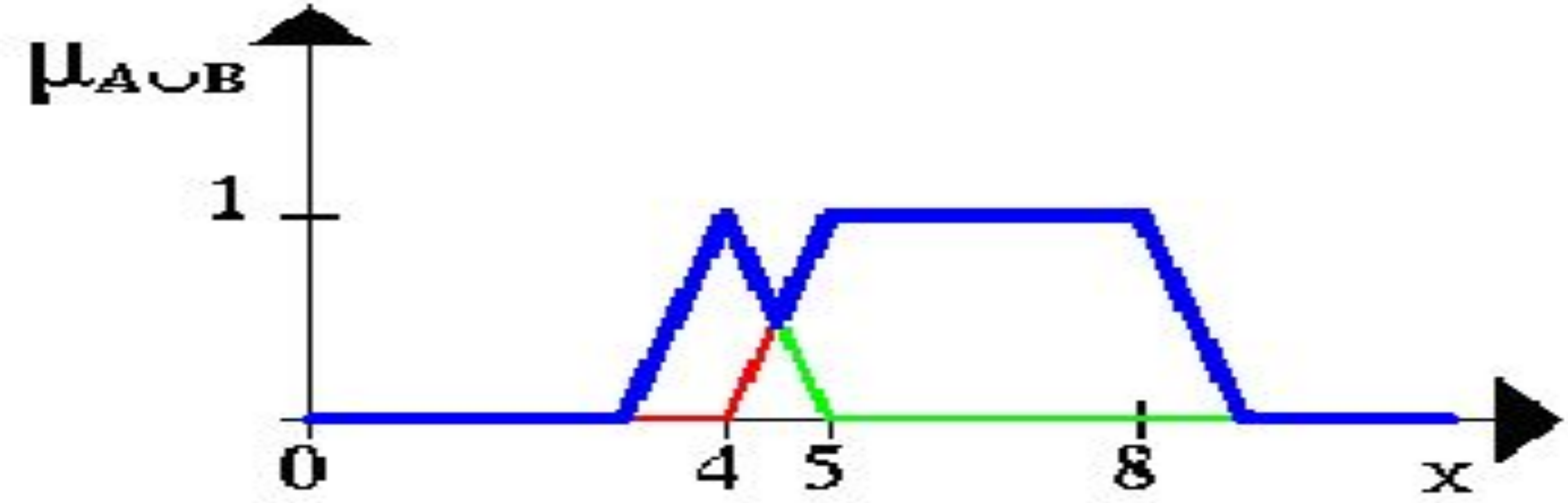




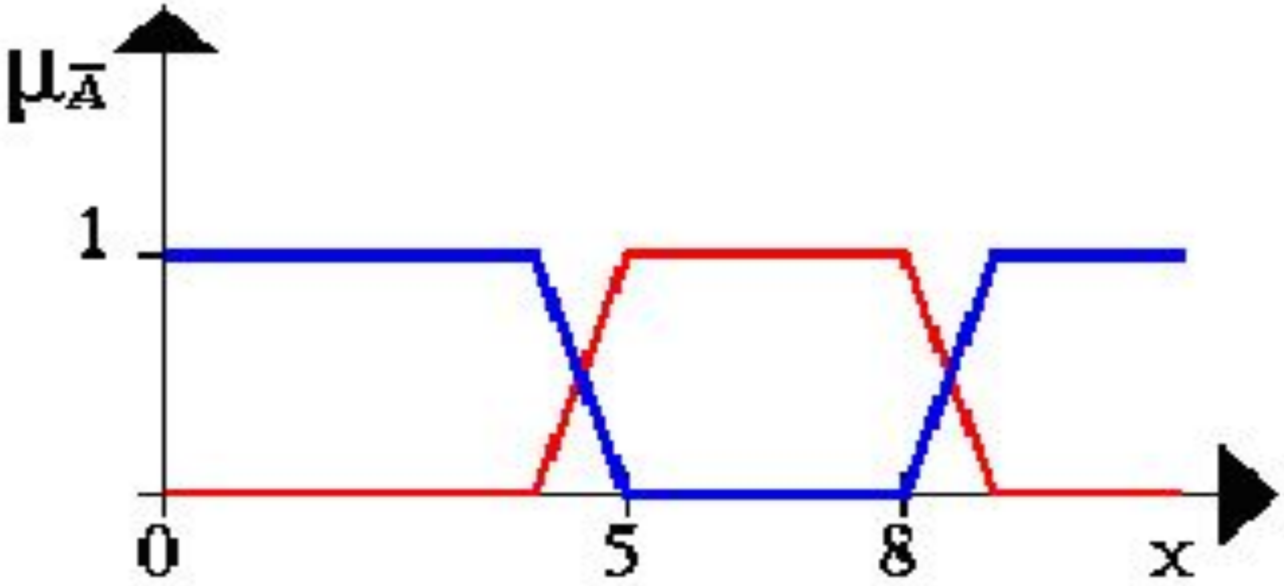
- Пересечение нечеткое множество *между* 5 *и* 8 **И (AND)** *около* 4 (синяя линия).



- Объединение Нечеткое множество *между* 5 *и* 8 **ИЛИ (OR)** *около* 4

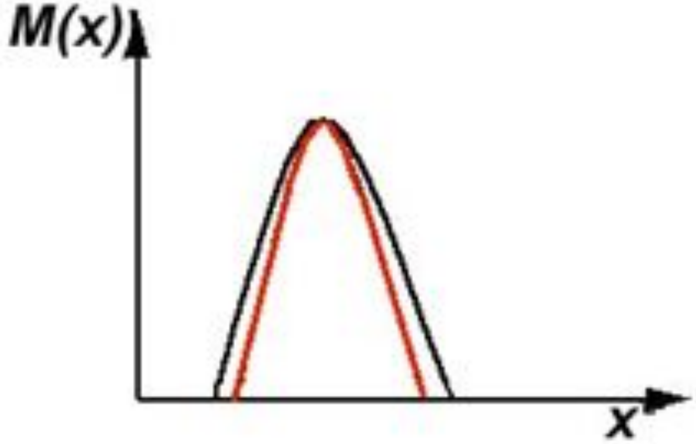


- Дополнение (отрицание) смысл НЕ



- **Концентрация**

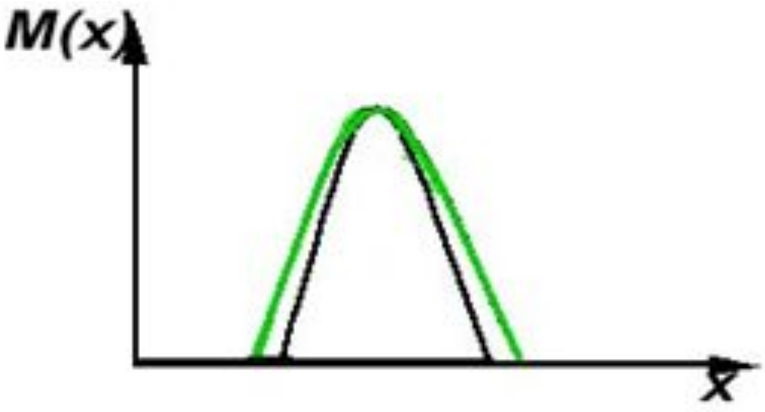
$$kon_n A = \{x, \quad \mu_A(x)^n\}$$



Лингвистический смысл «очень»

- **Размывание (или размытие)**

$$A = \{x, \quad \mu_A(x)^{1/n}\}$$



Лингвистический смысл
«не очень»

Усиление или ослабление лингвистических понятий

Усиление или ослабление лингвистических понятий достигается введением специальных квантификаторов. Например, если понятие «старческий возраст» определяется как

$$\left\{\frac{60}{0.6}+\frac{70}{0.8}+\frac{80}{0.9}+\frac{90}{1}\right\},$$

то понятие «очень старческий возраст» определится как

$$con(A)=A^2=\sum_i \frac{x_i}{\mu_i^2},$$

т. е. НМ для «очень старческий возраст» будет выглядеть так

$$\left\{\frac{60}{0.36}+\frac{70}{0.64}+\frac{80}{0.81}+\frac{90}{1}\right\}.$$

Пример

$A = 0{,}4/\ x_1 + 0{,}2/\ x_2 + 0/\ x_3 + 1/\ x_4;$

$B = 0{,}7/\ x_1 + 0{,}9/\ x_2 + 0{,}1/\ x_3 + 1/\ x_4;$

$C = 0{,}1/\ x_1 + 1/\ x_2 + 0{,}2/\ x_3 + 0{,}9/\ x_4.$

Здесь:

$A \subset B$, т. е. A содержится в B.

$E - A = 0{,}6/\ x_1 + 0{,}8/x_2 + 1/x_3 + 0/x_4$; $E - B = 0{,}3/x_1 + 0{,}1/x_2 + 0{,}9/x_3 + 0/x_4$.

$A \cap B = 0{,}4/x_1 + 0{,}2/x_2 + 0/x_3 + 1/x_4.$

$A \cup B = 0{,}7/x_1 + 0{,}9/x_2 + 0{,}1/x_3 + 1/x_4.$

$A - B = 0{,}3/x_1 + 0{,}1/x_2 + 0/x_3 + 0/x_4$; $B - A = 0{,}6/x_1 + 0{,}8/x_2 + 0{,}1/x_3 + 0/x_4$.

$A \oplus B = 0{,}6/x_1 + 0{,}8/x_2 + 0{,}1/x_3 + 0/x_4.$

Треугольные нормы и конормы

Треугольная норма

$\min(\mu_A, \mu_B)$

произведение $\mu_A \cdot \mu_B$

$\max(0, \mu_A + \mu_B - 1).$

Треугольная конорма

$\max(\mu_A, \mu_B)$

$\mu_A + \mu_B - \mu_A \cdot \mu_B$

$\min(1, \mu_A + \mu_B).$

Алгебраические операции

1. *__Алгебраическое произведение__* А и В обозначается $A \cdot B$ и определяется так:

$$\forall x \in E \quad \mu_{A \cdot B}(x) = \mu_A(x) \cdot \mu_B(x)$$

2. *__Алгебраическая сумма__* этих множеств обозначается $A \hat{+} B$ и определяется так:

$$\forall x \in E \quad \mu_{A \hat{+} B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x)$$

На основе операции алгебраического произведения определяется операция *возведения в степень* α нечеткого множества, где α – положительное число. Нечеткое множество A^{α} определяется функцией принадлежности $\mu_A^{\alpha} = \mu_A^{\alpha}(x)$.

Частным случаем возведения в степень являются следующие.

3. Операция *__концентрирования (уплотнения)__* $CON(A) = A^2$

4. Операция *__растяжения__* $DIL(A) = A^{0.5}$

5. *__Умножение на число__*. Если α – положительное число, такое что $\alpha \cdot \max\limits_{x \in A} \mu_A(x) \leq 1$, то нечеткое множество αA имеет функцию принадлежности:

$$\mu_{\alpha A}(x) = \alpha \mu_A(x)$$

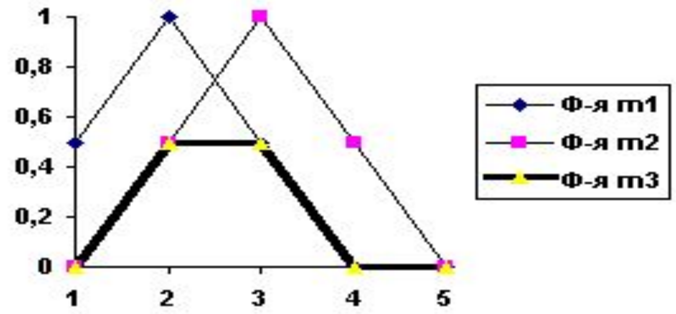
Логические	$\mu_{A\cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}$, $\mu_{A\cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}$
Алгебраические	$\mu_{A\cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x)\cdot\mu_B(x)$, $\mu_{A\cap B}(x) = \mu_A(x)\cdot\mu_B(x)$
Ограниченные	$\mu_{A\cup B}(x) = \min\{1, \mu_A(x) + \mu_B(x)\}$, $\mu_{A\cap B}(x) = \max\{0, \mu_A(x) + \mu_B(x) - 1\}$

	x1	x2	x3	x4
μ_A	0,4	0,2	0	1
μ_B	0,7	0,9	0,1	1
$\mu_{A\cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x)\cdot\mu_B(x)$	0,82	0,92	0,1	1
$\mu_{A\cap B}(x) = \mu_A(x)\cdot\mu_B(x)$	0,28	0,18	0	1
$\mu_{A\cup B}(x) = \min\{1, \mu_A(x) + \mu_B(x)\}$	1	1	0,1	1
$\mu_{A\cap B}(x) = \max\{0, \mu_A(x) + \mu_B(x) - 1\}$	0,1	0,1	0	1

Пример применения треугольных норм и конорм

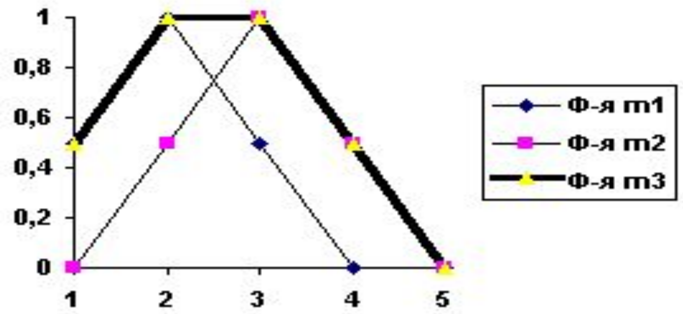
Пересечение II (ограниченное произведение).

$$\mu_3 = (\mu_1 \cap \mu_2)(x) = max\{0, \mu_1(x) + \mu_2(x) - 1\}, \forall x \in X$$



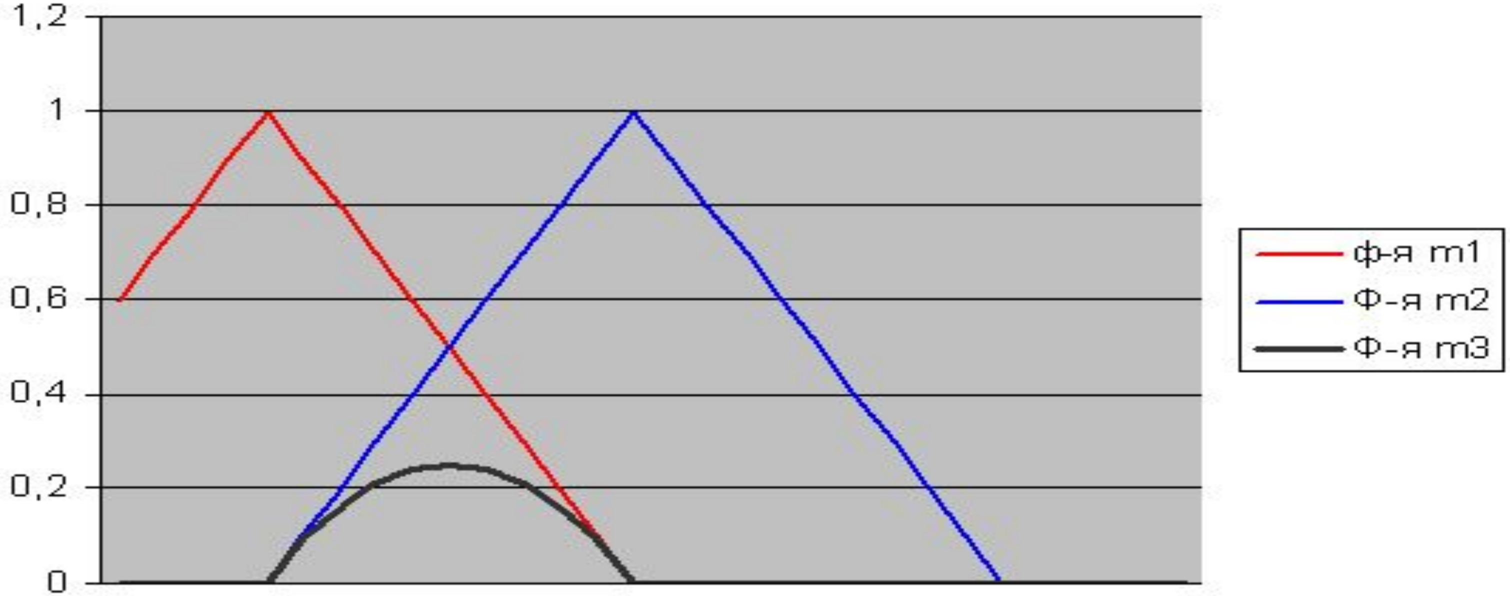
Объединение II (максимум: ограниченная сумма).

$$\mu_3 = (\mu_1 \cup \mu_2)(x) = min\{1, \mu_1(x) + \mu_2(x)\}, \forall x \in X$$



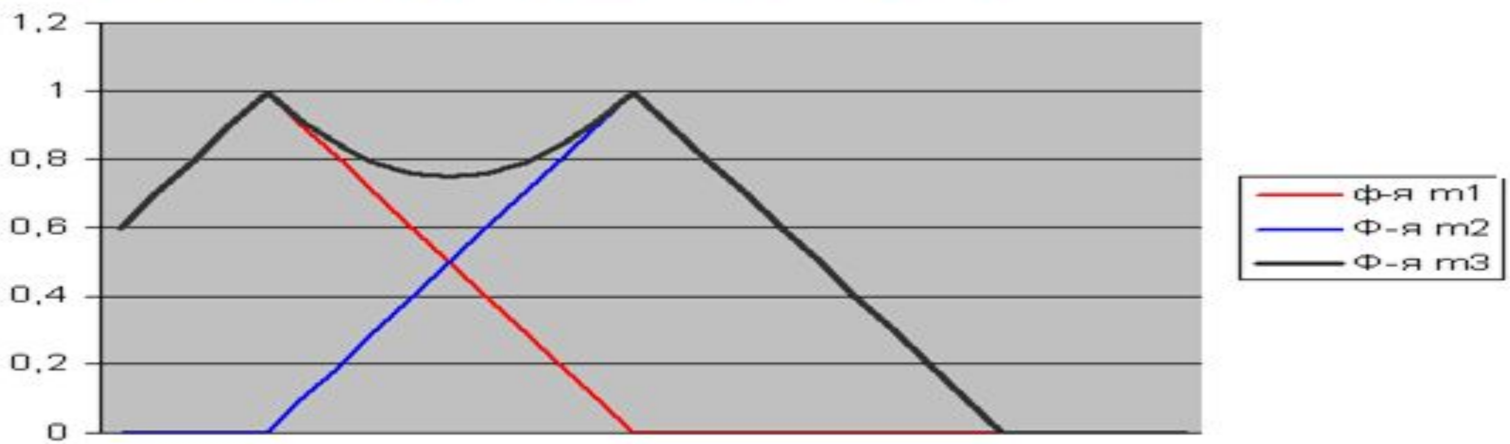
Пересечение III (алгебраическое произведение).

$\mu_3 = (\mu_1 \cap \mu_2)(x) = \mu_1(x) * \mu_2(x), \ \forall x \in X$



Объединение III (алгебраическая сумма).

$\mu_3 = (\mu_1 \cup \mu_2)(x) = \mu_1(x) + \mu_2(x) - \mu_1(x) \bullet \mu_2(x), \ \forall x \in X$



Нечеткие отношения. Операции над нечеткими отношениями

Нечеткая логика и нейронные сети

Пример нечеткого отношения

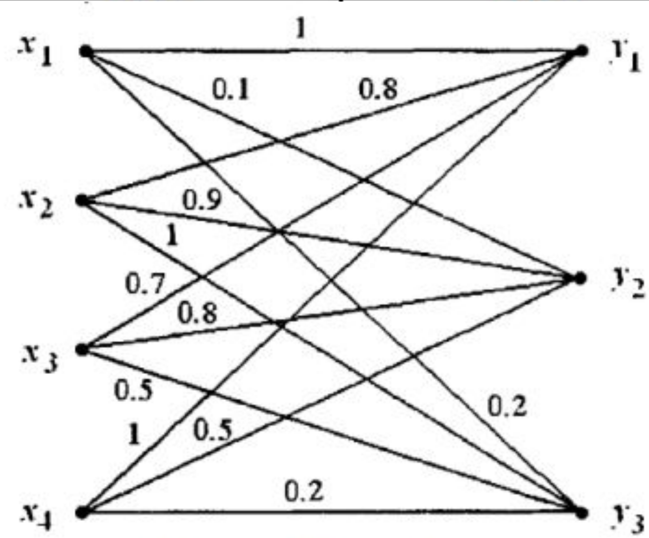
Предположим, необходимо построить нечеткое отношение, которое содержательно описывает упрощенную ситуацию поиска неисправности в автомобиле. С этой целью в качестве первого универсума рассмотрим множество предпосылок или причин неисправности $\boldsymbol{X} = \{x_1, x_2, x_3, x_4\}$, в котором x_1 — "*неисправность аккумулятора*", x_2— "*неисправность карбюратора*", x_3 — "*низкое качество бензина*", x_4 — "*неисправность системы зажигания*". В качестве второго универсума рассмотрим множество заключений или проявлений неисправности $Y = \{y_1, y_2, y_3\}$, где y_1 — "*двигатель не запускается*", y_2 — "*двигатель работает неустойчиво*", y_3 — "*двигатель не развивает полной мощности*". При этом между каждым элементом множества предпосылок и каждым элементом множества следствий существует некоторая причинная взаимосвязь.

Пример представления 1

. Нечеткое отношение диагностики неисправности в автомобиле

	y_1	y_2	y_3
x_1	1	0.1	0.2
x_2	0.8	0.9	1
x_3	0.7	0.8	0.5
x_4	1		

$$M_P = \begin{bmatrix} 1 & 0.1 & 0.2 \\ 0.8 & 0.9 & 1 \\ 0.7 & 0.8 & 0.5 \\ 1 & 0.5 & 0.2 \end{bmatrix}.$$

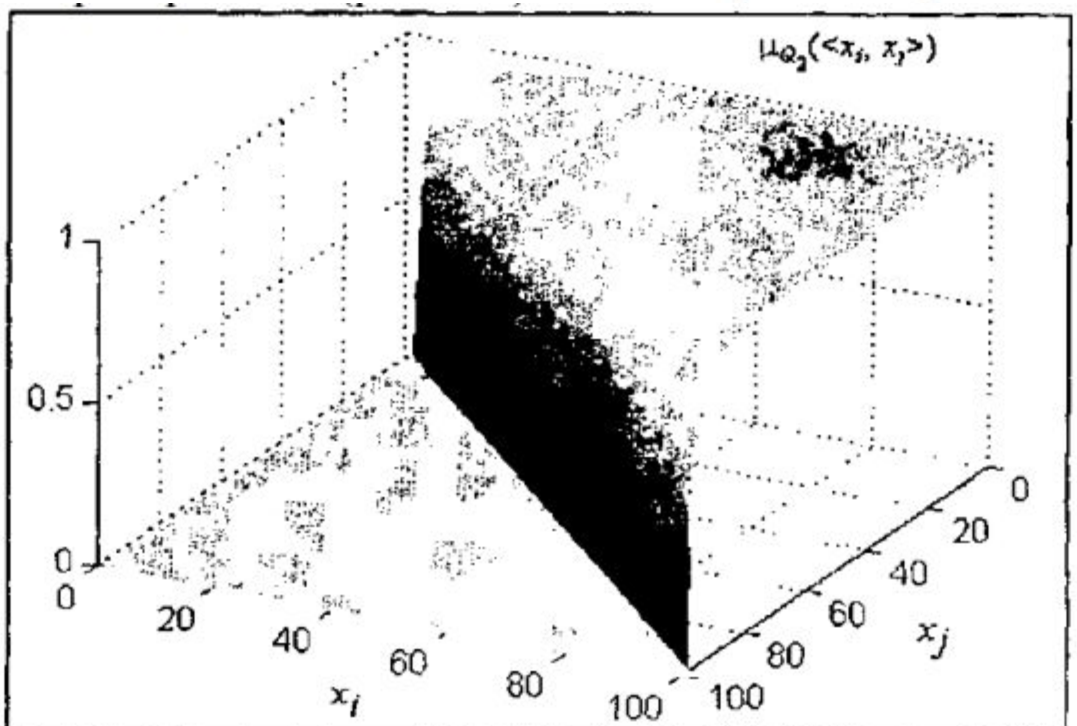


Нечеткий граф отношения P
(стрелки дуг, направленных от вершин x_i к вершинам y_j для удобства не указаны)

Пример представления 2

$$\mu_{\tilde{Q}2}(< x_i, x_j >) = 0 \quad x_i \leq x_j$$

$$\mu_{\tilde{Q}2}(< x_i, x_j >) = 1 - \frac{1}{x_i - x_j} \qquad x_i > x_j \ (\forall x_i, x_j \in \mathbb{R}+).$$



Графическое представление нечеткого отношения $\tilde{Q}_2$
в форме графика его функции принадлежности

Модель «Рынок-Продукция»

Нечеткое отношение модели "Продукция/Рынок"

	Продукция, выпускаемая в настоящее время	Новая продукция, связанная с выпускаемой	Совершенно новая продукция
Имеющийся известный рынок	0.9	0.6	0.3
Новый рынок, связанный с имеющимся	0.6	0.4	0.2
Совершенно новый рынок	0.3	0.2	0.1

Операции над нечеткими отношениями

Пусть R_1 и R_2 – два нечетких отношения такие, что

$$\forall (x, y) \in X \times Y: \mu_{R_1}(x, y) \le \mu_{R_2}(x, y),$$

тогда говорят, что R_2 содержит R_1 или R_1 содержится в R_2.

Обозначение: $R_1 \subseteq R_2$.

Объединение двух отношений R_1 и R_2 обозначается $R_1 \cup R_2$ и определяется выражением $\mu_{R_1 \cup R_2}(x, y) = \max\{\mu_{R_1}(x, y), \mu_{R_2}(x, y)\}$.

Пересечение двух отношений R_1 и R_2. R_1 и R_2 обозначается $R_1 \cap R_2$ и определяется выражением $\mu_{R_1 \cap R_2}(x, y) = \min\{\mu_{R_1}(x, y), \mu_{R_2}(x, y)\}$.

Дополнением отношения R называется нечеткое отношение Q с функцией принадлежности $\mu_Q(x, y) = 1 - \mu_R(x, y)$.

Операции над нечеткими отношениями

Обратное к отношению $\widetilde{R}$ отношение $\widetilde{R}^{-1}$ определяется $\forall x, y \in X$ выражением $\mu_{\widetilde{R}^{-1}}(x, y) = \mu_{\widetilde{R}}(y, x)$.

Композицией отношений (произведением) называется отношение:

К1 – максиминная композиция:

$$\mu_{\widetilde{R}_1 \cdot \widetilde{R}_2}(x, y) = \sup_{z \in X} \min\{\mu_{\widetilde{R}_1}(x, z), \mu_{\widetilde{R}_2}(z, y)\};$$

К2 – минимаксная композиция:

$$\mu_{\widetilde{R}_1 \cdot \widetilde{R}_2}(x, y) = \min_{z \in X} \max\{\mu_{\widetilde{R}_1}(x, z), \mu_{\widetilde{R}_2}(z, y)\};$$

К3 – максмультипликативная композиция:

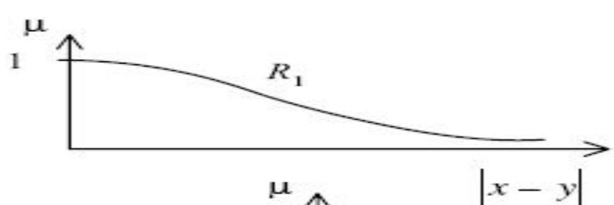
$$\mu_{\widetilde{R}_1 \cdot \widetilde{R}_2}(x, y) = \sup_{z \in X}\{\mu_{\widetilde{R}_1}(x, z)\mu_{\widetilde{R}_2}(z, y)\}.$$

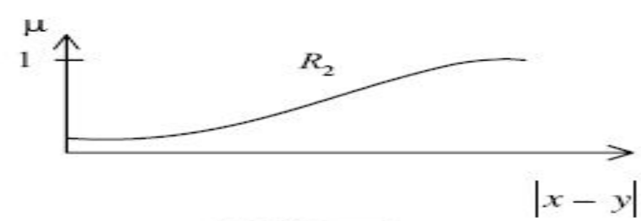
Пример объединения нечетких отношений

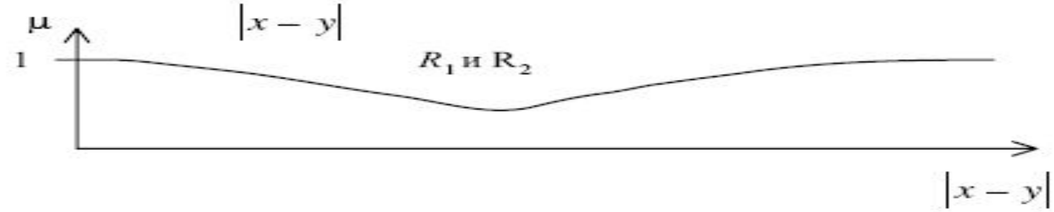
1. Ниже изображены отношения действительных чисел, содержательно означающие: xR_1y – «числа x и y очень близкие», xR_2y – «числа x и y очень различны» и их объединение $xR_1 \cup R_2y$ – «числа x и y очень близкие или очень различные».

Функции принадлежности отношений заданы на $|y-x|$.

$$\mu_{R_1 \cup R_2}(x, y) = \begin{cases} \mu_{R_1}(x, y), & |y-x| \le \alpha, \\ \mu_{R_2}(x, y), & |y-x| > \alpha. \end{cases}$$







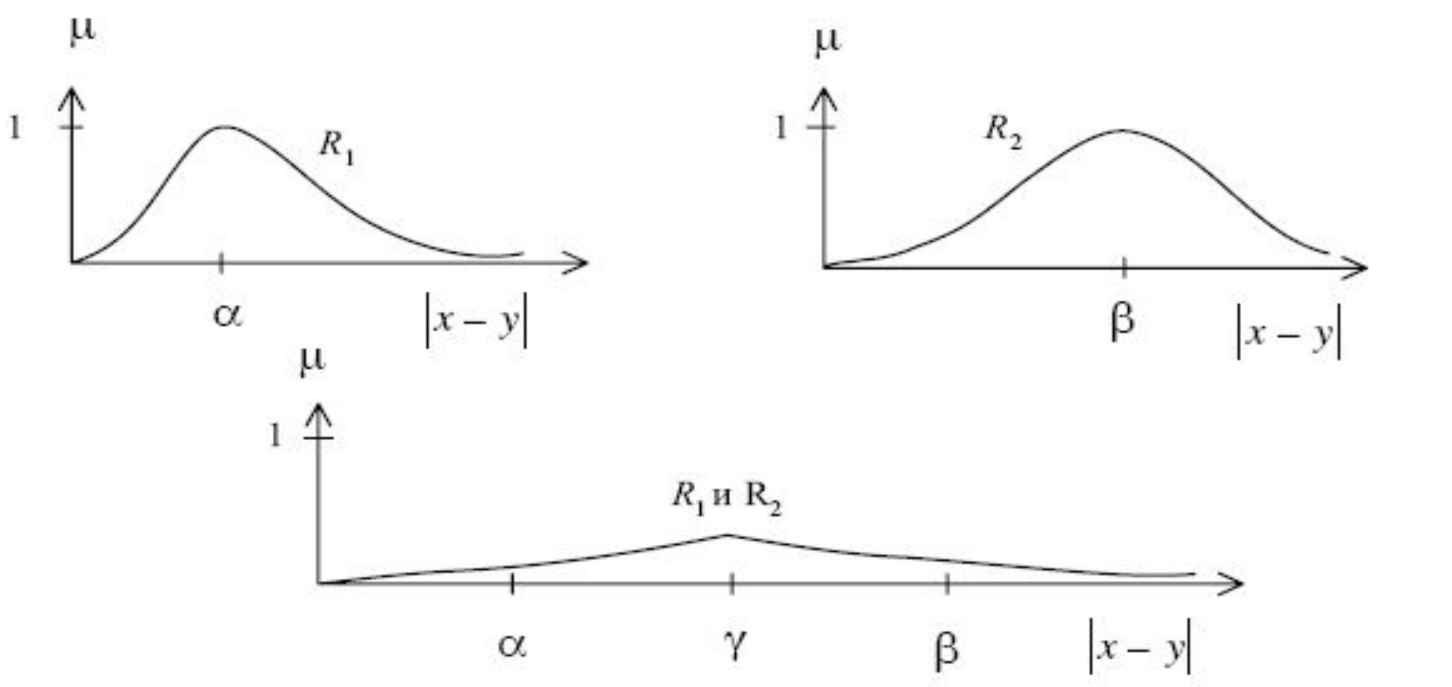
где α – такое $|y-x|$, что $\mu_{R_1}(x, y) = \mu_{R_2}(x, y)$

2.

R_1				R_2				$R_1 \cup R_2$			
	y_1	y_2	y_3		y_1	y_2	y_3		y_1	y_2	y_3
x_1	0,1	0	0,8	x_1	0,7	0,9	1	x_1	0,7	0,9	1
x_2	1	0,7	0	x_2	0,3	0,4	0,5	x_2	1	0,7	0,5

Пример пересечения нечетких отношений

Ниже изображены отношения: R_1, означающее «модуль разности $|y - x|$ близок к α», R_2, означающее «модуль разности $|y - x|$ близок к β», и их пересечение.



Примеры композиций

Пусть заданы два нечетких отношения A и B на U, состоящем из двух элементов $U=\{u_1, u_2\}$, где матрицы нечетких отношений таковы:

$$\mu_A(x,y)=\begin{vmatrix}0.2 & 0.6\\ 0.5 & 0.8\end{vmatrix},\quad \mu_B(y,z)=\begin{vmatrix}0.5 & 0.7\\ 0.3 & 1\end{vmatrix}$$

Тогда композиция (произведение) нечетких отношений определяется так:

а) максиминная $R_1^2 = A\cdot B$

$$\mu_{A\cdot B}(x,z)=\begin{vmatrix}0.3 & 0.6\\ 0.5 & 0.8\end{vmatrix}$$

б) минимаксная $R_2^2 = A\circ B$

$$\mu_{A\cdot B}(x,z)=\begin{vmatrix}0.5 & 0.7\\ 0.5 & 0.7\end{vmatrix}$$

в) максимультиплекативная $R_3^2 = A * B$

$$\mu_{A\cdot B}(x,z)=\begin{vmatrix}0.18 & 0.6\\ 0.25 & 0.8\end{vmatrix}$$

Примеры композиций

Пусть заданы два нечетких отношения A и B на U, состоящем из двух элементов $U=\{u_1, u_2\}$, где матрицы нечетких отношений таковы:

$$\mu_A(x,y) = \begin{vmatrix} 0.2 & 0.6 \\ 0.5 & 0.8 \end{vmatrix}, \quad \mu_B(y,z) = \begin{vmatrix} 0.5 & 0.7 \\ 0.3 & 1 \end{vmatrix}$$

Тогда композиция (произведение) нечетких отношений определяется так:

а) максиминная $R_1^2 = A \cdot B$

$$\mu_{A \cdot B}(x,z) = \begin{vmatrix} 0.3 & 0.6 \\ 0.5 & 0.8 \end{vmatrix}$$

б) минимаксная $R_2^2 = A \circ B$

$$\mu_{A \cdot B}(x,z) = \begin{vmatrix} 0.5 & 0.7 \\ 0.5 & 0.7 \end{vmatrix}$$

в) максимультиплекативная $R_3^2 = A * B$

$$\mu_{A \cdot B}(x,z) = \begin{vmatrix} 0.18 & 0.6 \\ 0.25 & 0.8 \end{vmatrix}$$

Композиция двух нечёткых отношений

R_1			
	y_1	y_2	y_3
x_1	0,1	0,7	0,4
x_2	1	0,5	0

•

R_2				
	z_1	z_2	z_3	z_4
y_1	0,9	0	1	0,2
y_2	0,3	0,6	0	0,9
y_3	0,1	1	0	0,5

=

$R_1 \cdot R_2$				
	z_1	z_2	z_3	z_4
x_1	0,3	0,6	0,1	0,7
x_2	0,9	0,5	1	0,5

Выбор кандидатов на обучение

Нечеткое отношение $\tilde{S}$ профилирования специальностей обучения

	Быстрота и гибкость мышления	Умение быстро принимать решения	Устойчивость и концентрация внимания	Зрительная память	Быстрота реакции
Менеджер	0.9	0.9	0.8	0.4	0.5
Программист	0.8	0.5	0.9	0.3	0.1
Водитель	0.3	0.9	0.6	0.5	0.9
Секретарь	0.5	0.4	0.5	0.5	0.2
Переводчик	0.7	0.8	0.8	0.2	0.6
	Двигательная память	Физическая выносливость	Координация движений	Эмоционально-волевая устойчивость	Ответственность
Менеджер	0.3	0.6	0.2	0.9	0.8
Программист	0.2	0.2	0.2	0.5	0.5
Водитель	0.8	0.9	0.8	0.6	0.3
Секретарь	0.2	0.3	0.3	0.9	0.8
Переводчик	0.2	0.2	0.3	0.3	0.2

Нечеткое отношение $\tilde{T}$ профилирования кандидатов на обучение

	Петров	Иванов	Сидоров	Васильева	Григорьева
Быстрота и гибкость мышления	0.9	0.8	0.7	0.9	1
Умение быстро принимать решения	0.6	0.4	0.8	0.5	0.6
Устойчивость и концентрация внимания	0.5	0.2	0.3	0.8	0.7
Зрительная память	0.5	0.9	0.5	0.8	0.4
Быстрота реакции	1	0.6	0.5	0.7	0.4
Двигательная память	0.4	0.5	1	0.7	0.8
Физическая выносливость	0.5	0.8	0.9	0.5	0.4
Координация движений	0.5	0.6	0.7	0.6	0.5
Эмоционально-волевая устойчивость	0.8	1	0.2	0.5	0.6
Ответственность	0.3	0.5	0.9	0.6	0.8

Выбор кандидатов на обучение

Нечеткое отношение $\tilde{S}$ профилирования специальностей обучения

	Быстрота и гибкость мышления	Умение быстро принимать решения	Устойчивость и концентрация внимания	Быстрота реакции
Менеджер	0.9	0.9	0.8	0.5
Программист	0.8	0.5	0.9	0.1
Переводчик	0.7	0.8	0.8	0.9

Нечеткое отношение $\tilde{T}$ профилирования кандидатов на обучение

	Петров	Иванов
Быстрота и гибкость мышления	0.9	0.8
Умение быстро принимать решения	0.6	0.4
Устойчивость и концентрация внимания	0.5	0.2
Быстрота реакции	0.2	0.9

Нечеткая и лингвистическая переменные. Нечеткие числа

Нечеткая логика и нейронные сети

Определение нечеткой переменой

Нечеткой переменной называется совокупность (кортеж) вида

$<X, U, \tilde{X}>$, где

X – наименование нечеткой переменной.

$U = \{u\}$ область ее определения (универсальное множество);

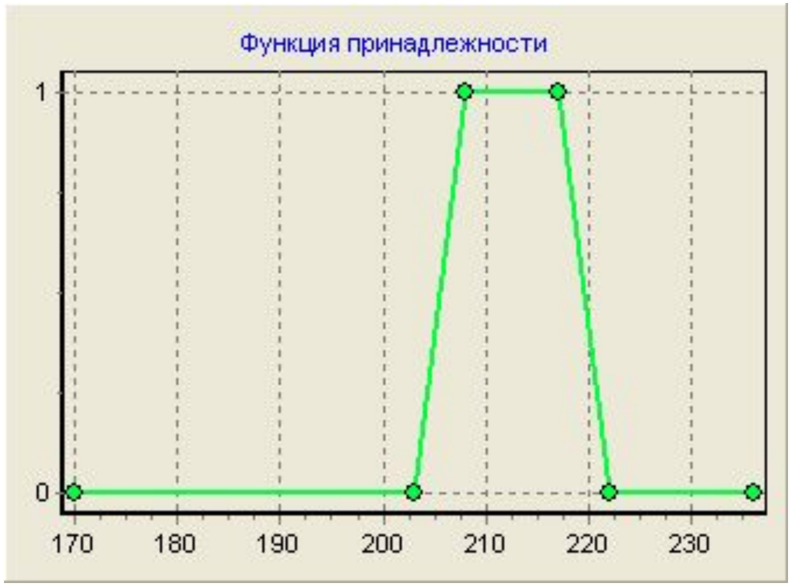
$\tilde{X} = \bigcup_{u \in U} \mu_{\tilde{X}}(u)/u$ - нечеткое множество на U, описывающее ограничения

(т.е. $\mu_{\tilde{X}}(u)$) на значения нечеткой переменной X.

Пример : нечеткая переменная *«высокий рост»*

X - «*высокий рост*» (наименование переменной),

- U = [130,240],
- $\mu_{\tilde{X}}(u)$– функция принадлежности элементов из универса X данной нечеткой переменной.



<u>Пояснение:</u> Нечеткая переменная – именованное <u>нечеткое множество</u>

Определение лингвистической переменной

Лингвистической переменной (ЛП) называется кортеж вида <β, T, U, G, M>, где

β – наименование лингвистической переменной

T – множество ее значений (терм-множество), представляющих собой наименование нечетких переменных, областью определения каждой из которых является множество U. Множество T называется базовым терм-множеством лингвистической переменной

G – синтаксическая процедура, описывающая процесс образования из элементов множества Т новых, осмысленных для данной задачи значений лингвистической переменной (терм).

M – семантическая процедура, позволяющая превратить каждое новое значение ЛП, образуемое процедурой G, в нечеткую переменную, т.е. сформировать соответствующее нечеткое множество.

Пример: ЛП *«температура в комнате»*

β = «температура в комнате» - имя лингвистической переменной;

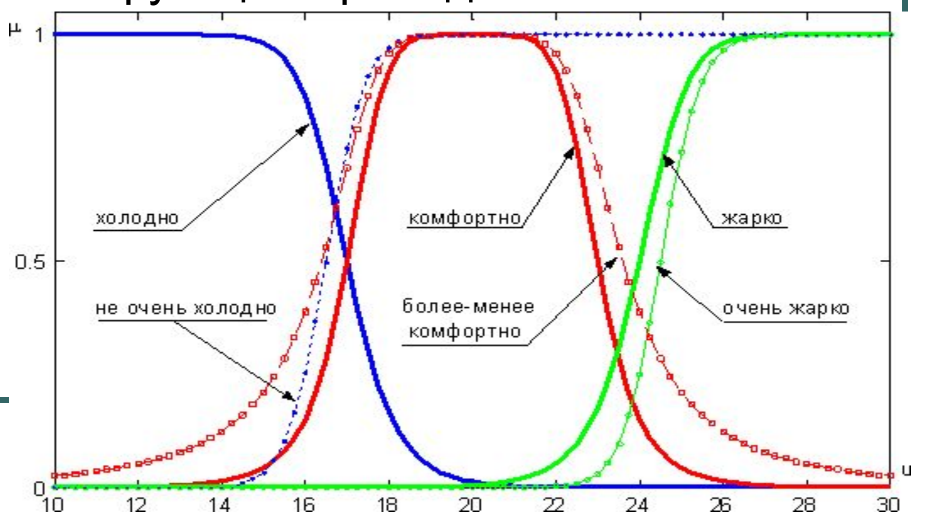
U = [5,35] – универс определения;

T = {"холодно", "комфортно", "жарко"} - базовое терм-множество;

G - синтаксические правила, порождающее новые термы с использованием квантификаторов "и","или", "не", "очень", "более-менее";

M - процедура, ставящая каждому новому терму в соответствие функцию принадлежности (т.е. задавая нечеткое множество) по правилам: если термы A и B имели функции принадлежности μ**a**(*x*) и μ**B**(*x*) соответственно, то новые термы будут иметь функции принадлежности:

Квантификатор	Функция принадлежности:
не t	$1-\mu_t(u)$
очень t	$(\mu_t(u))^2$
более-менее t	$\sqrt{\mu_t(u)}$
A и B	max($\mu_A(x)$, $\mu_B(x)$)
A или B	min($\mu_A(x)$, $\mu_B(x)$)



Пример : ЛП «дисциплина»

- *β* – дисциплина;
- *T* – {«*Сложная дисциплина*», «*Интересная дисциплина*», «*Пригодится в будущей работе*»};
- *U* = [«*Программирование*», «*Базы данных*», «*Нечеткая логика*», «*История*»] – множество дисциплин, изучаемых студентами направления «Бизнес-информатика»;
- *G* – процедура перебора элементов базового терм-множества;
- *M* – процедура экспертного опроса.

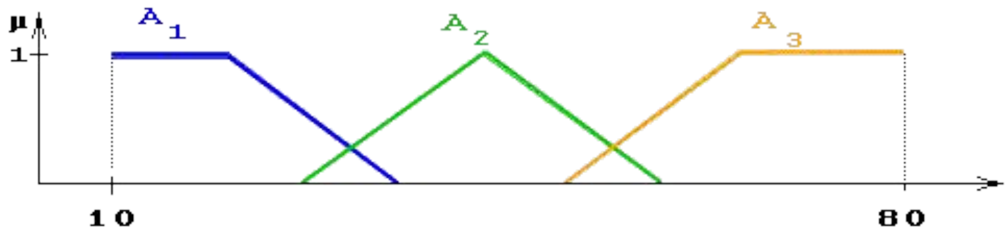
Пример: толщина детали

Пусть эксперт определяет толщину выпускаемого изделия с помощью понятий «*малая толщина*», «*средняя толщина*» и «*большая толщина*», при этом минимальная толщина равна 10 мм, а максимальная – 80 мм.

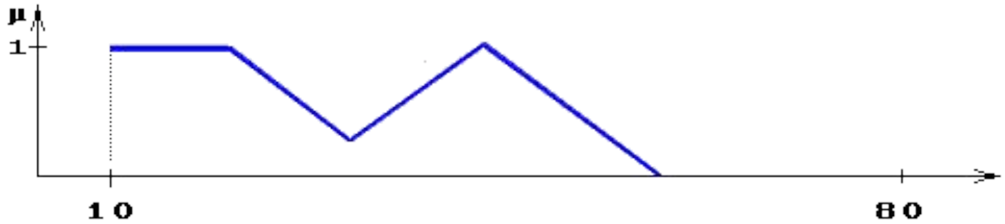
Формализация такого описания может быть проведена с помощью следующей лингвистической переменной < β, T, X, G, M>, где

- β – толщина изделия;
- T – {«*малая толщина*», «*средняя толщина*», «*большая толщина*»};
- U = [10, 80];
- G – процедура образования новых термов с помощью связок *и*, *или* и модификаторов типа *очень*, *не*, *слегка* и др. Например: «*малая или средняя толщина*» (рис. 24), *«очень малая толщина*» и др.;
- M – процедура задания на X = [10, 80] нечетких подмножеств A1 = «*малая толщина*», A2 = «*средняя толщина*», A3 = «*большая толщина*», а также нечетких множеств для термов из G(T) в соответствии с правилами трансляции нечетких связок и модификаторов *и*, *или*, *не*, *очень*, *слегка* и др.

Пример: толщина детали

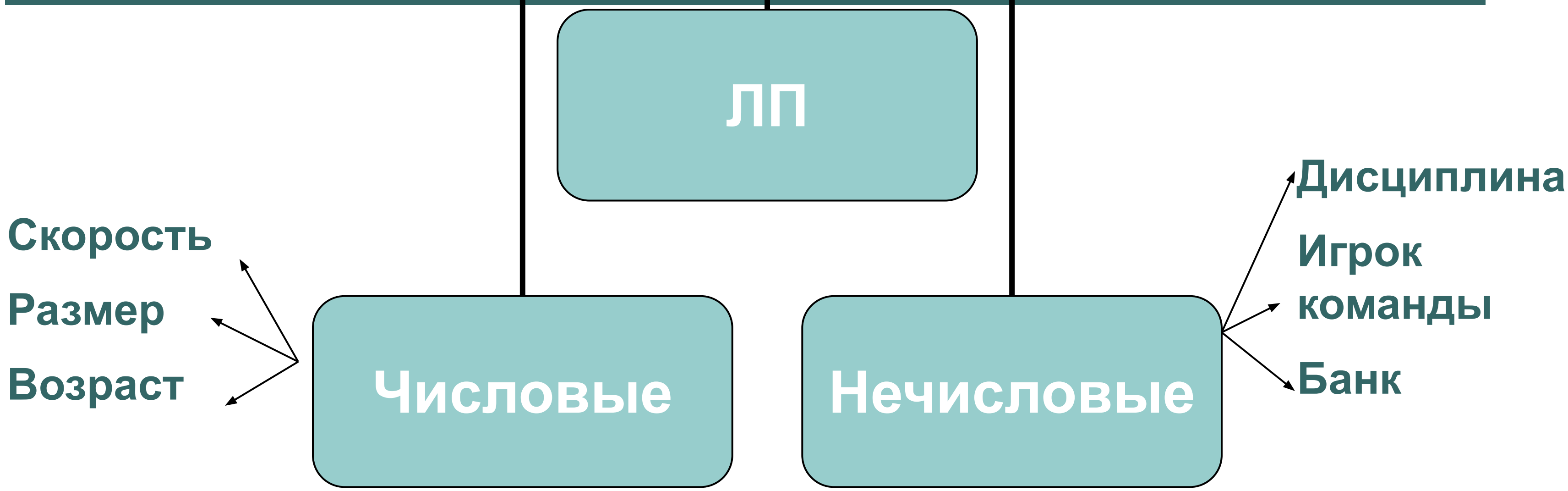


Функции принадлежности нечетких множеств:
«малая толщина» = A1, «*средняя толщина*» = A2, «*большая толщина*» = A3



Функция принадлежности
нечеткого множества «*малая или средняя толщина*» = A1 U A1

Виды ЛП



Числовой называют лингвистическую переменную, у которой $U \subset R^1$, $R^1=(-\infty, \infty)$, и которая имеет измеримую базовую переменную.

Нечеткие числа

Нечеткие числа – нечеткие переменные, определенные на числовой оси, т.е. нечеткое число определяется как нечеткое множество *A* на множестве *R* с функцией принадлежности $\mu_A(u) \in [0, 1], u \in R.$

Нечеткое число — это нечеткое подмножество универсального множества действительных чисел, имеющее нормальную и выпуклую функцию принадлежности, то есть такую, что:

a) существует значение носителя, в котором функция принадлежности равна единице, а также

b) при отступлении от своего максимума влево или вправо функция принадлежности не возрастает.

<u>Пример</u>:

«*Толщина*» (Т = {«*малая толщина*», «*средняя толщина*», «*большая толщина*»})

Возможны значения, зависящие от области определения U: в данном случае значения лингвистической переменной «толщина изделия» могут быть определены как «*около 20 мм*», «*около 50 мм*», «*около 70 мм*», то есть в виде ***нечетких чисел*.**

Операции над нечеткими числами

$$C = A \tilde{+} B \quad \Leftrightarrow \quad \sup_{z=x+y} \left(\mu_A(x) \wedge \mu_B(y)\right).$$

$$C = A \tilde{-} B \quad \Leftrightarrow \quad \sup_{z=x-y} \left(\mu_A(x) \wedge \mu_B(y)\right).$$

$$C = A \tilde{\cdot} B \quad \Leftrightarrow \quad \sup_{z=xy} \left(\mu_A(x) \wedge \mu_B(y)\right).$$

$$C = A \tilde{\div} B \quad \Leftrightarrow \quad \sup_{z=x/y} \left(\mu_A(x) \wedge \mu_B(y)\right).$$

П р и м е р ы

Выполнение арифметических операций над дискретными нечеткими числами $A = \{0{,}1/5,\ 0{,}8/6,\ 0{,}4/7\}$, $B = \{0{,}2/4,\ 0{,}9/5,\ 0{,}3/6\}$:

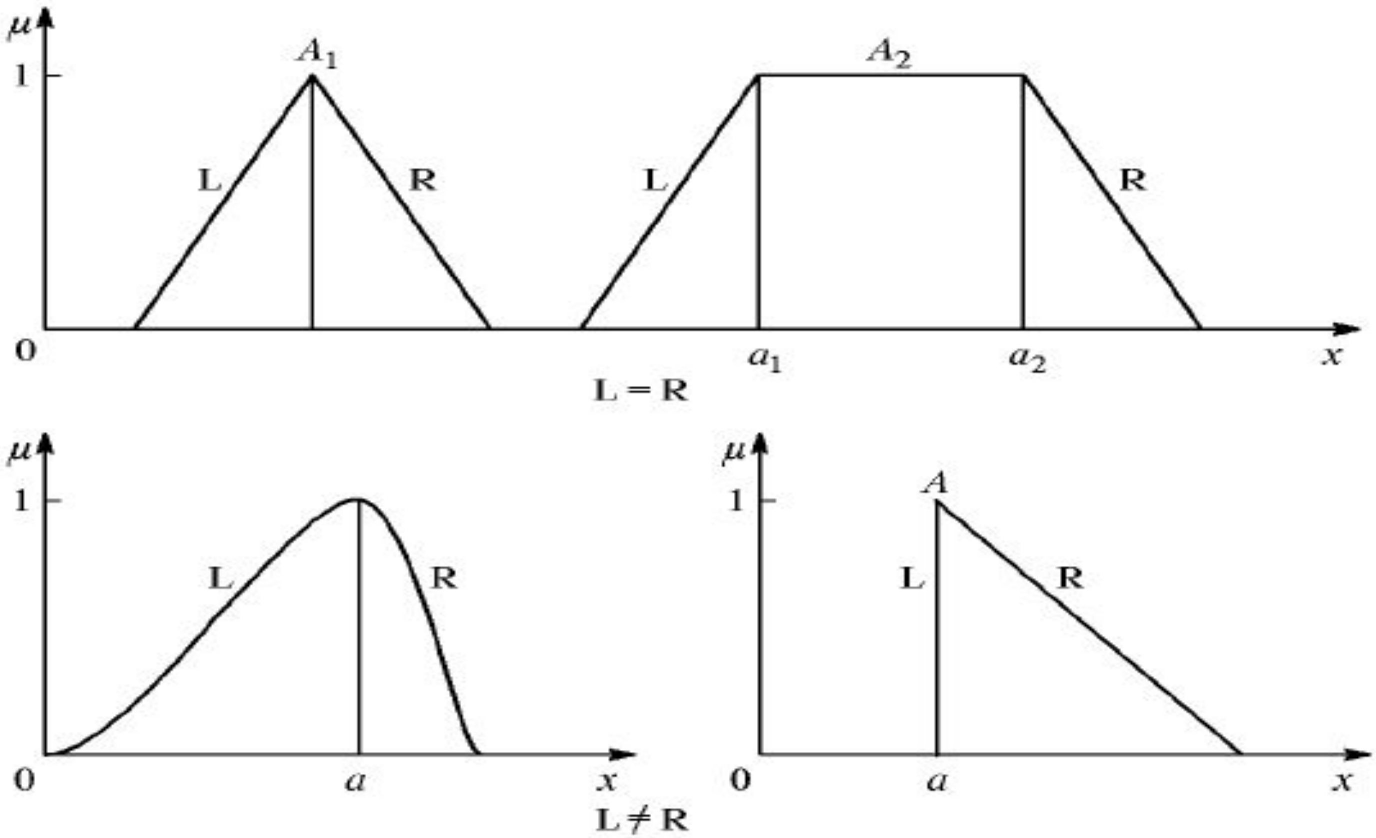
$A + B = \{0{,}1/9,\ 0{,}2/10,\ 0{,}8/11,\ 0{,}4/12,\ 0{,}3/13\}$;

$A - B = \{0{,}1/{-1},\ 0{,}3/0,\ 0{,}8/1,\ 0{,}4/2,\ 0{,}2/3\}$;

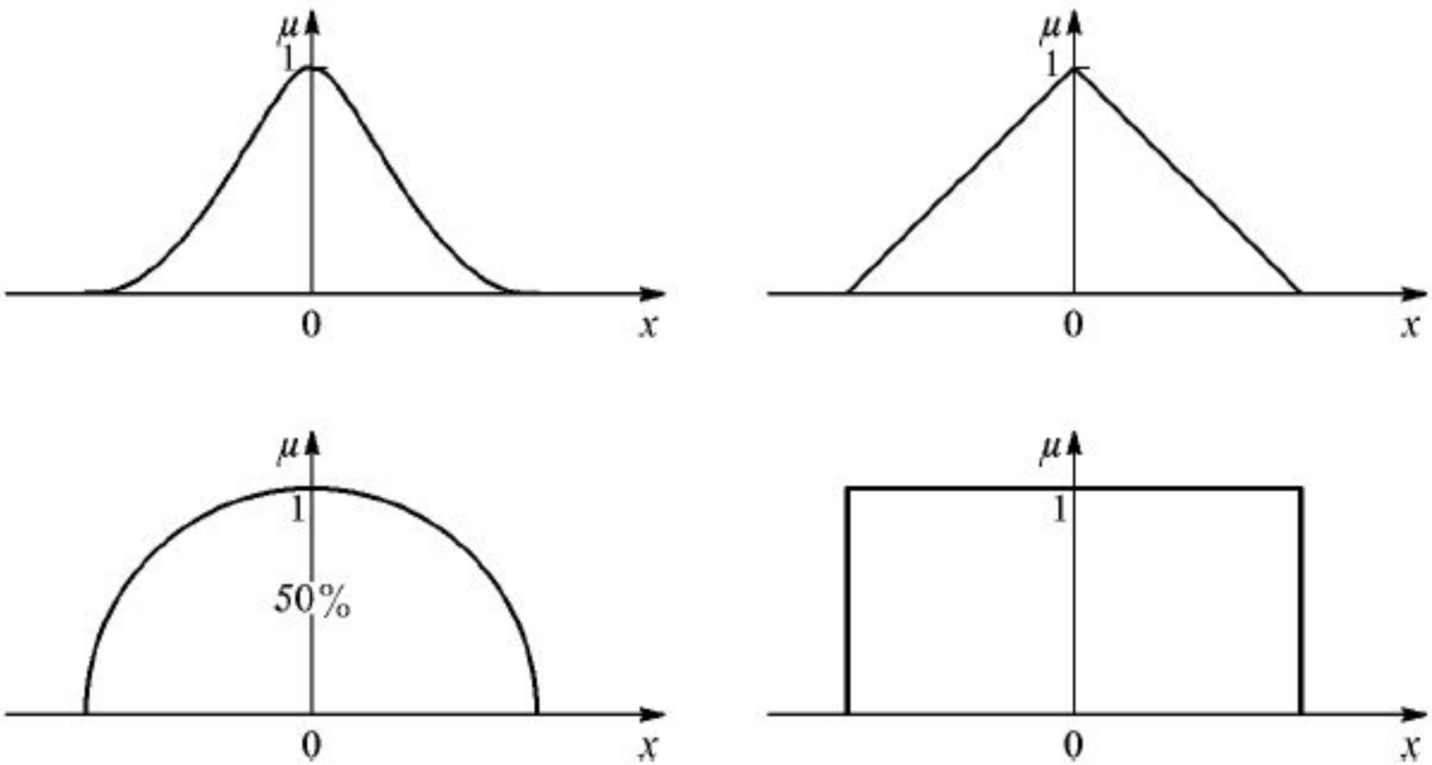
$A \times B = \{0{,}1/20,\ 0{,}2/24,\ 0{,}1/25,\ 0{,}2/28,\ 0{,}8/30,\ 0{,}4/35,\ 0{,}3/42\}$;

$A : B = \{0{,}1/0{,}83,\ 0{,}3/1{,}0,\ 0{,}3/1{,}17,\ 0{,}8/1{,}22,\ 0{,}1/1{,}25,\ 0{,}4/1{,}4,\ 0{,}2/1{,}5,\ 0{,}2/1{,}75\}$.

L-R нечеткие числа



L-R нечеткие числа



L-R нечеткие числа

- **Толерантные** нечеткие числа (L-R)-типа **называют** трапезоидными числами.

Если мы оцениваем параметр качественно, например, говоря: "Это значение параметра является **средним**", необходимо ввести уточняющее высказывание типа " **Среднее** значение — это **примерно** от *a* до *b*", которое есть предмет экспертной оценки (нечеткой классификации), и тогда можно использовать для моделирования нечетких классификаций трапезоидные числа.

!!! это самый естественный способ неуверенной классификации.

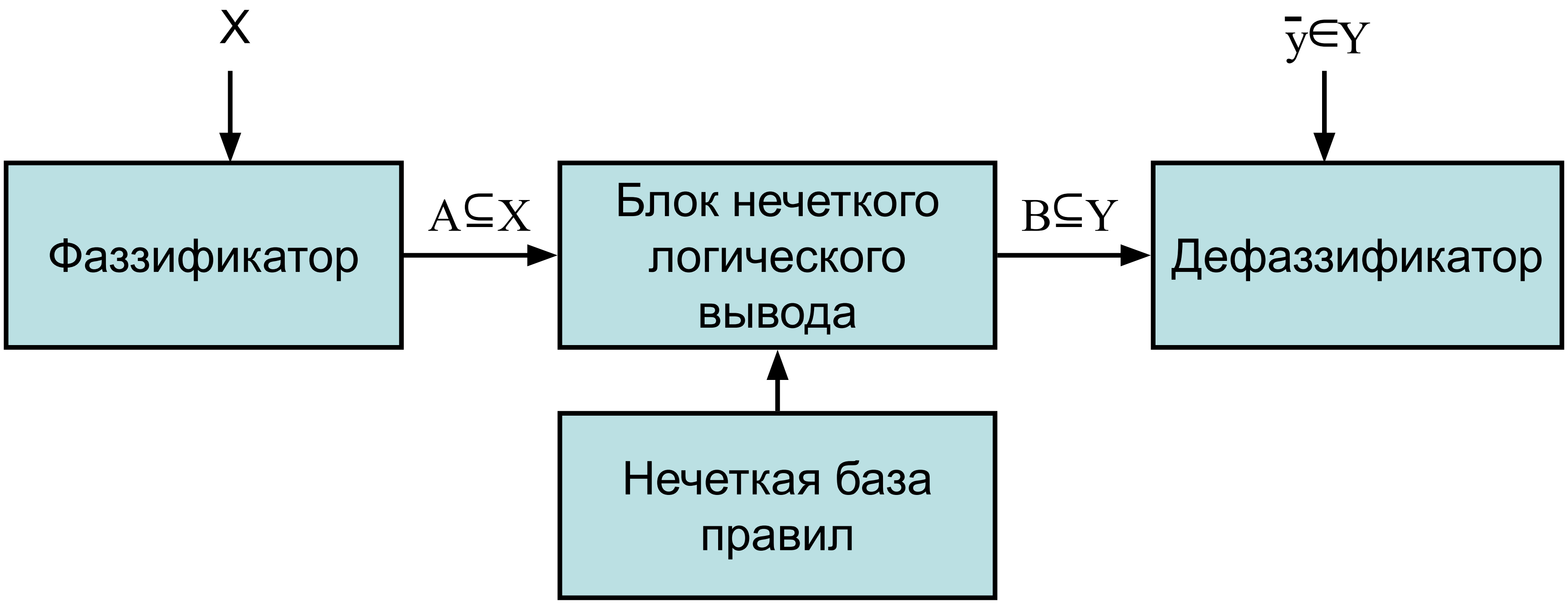
- **Унимодальные** нечеткие числа (L-R)-типа **называют** треугольными числами.

Треугольные числа формализуют высказывания типа **"приблизительно равно** α**"**. Ясно, что $\alpha+\sigma\approx\alpha$, причем по мере убывания σ до нуля степень уверенности в оценке растет до единицы.

Нечеткий вывод

Нечеткая логика и нейронные сети

Нечеткое (логико-лингвистическое) моделирование



Логико-лингвистическое описание систем, нечеткие модели

Логико-лингвистические методы описания систем основаны на том, что поведение исследуемой системы описывается в естественном (или близком к естественному) языке в терминах лингвистических переменных.

L1 : Если $<a_{11}>$ и/или … и/или $<\mathrm{a}_{1m}>$ то $<b_{11}>$ и/или… и/или $<\mathrm{b}_{1n}>$

L2 : Если $<a_{21}>$ и/или … и/или $<\mathrm{a}_{2m}>$ то $<b_{21}>$ и/или… и/или $<\mathrm{b}_{2n}>$

………………..

L*k* : Если $<a_{k1}>$ и/или … и/или $<\mathrm{a}_{km}>$ то $<b_{k1}>$ и/или… и/или $<\mathrm{b}_{kn}>$

Нечёткие высказывания типов 1 и 2

Логико-лингвистическое описание систем, нечеткие модели

L1 : если <A1 > то <B1 >
L2 : если <A2 > то <B2 >

....

L*k* : если <A*k* > то <B*k* >

Нечёткие высказывания типа 3

Совокупность импликаций **{L1, L2, ..., L*k*}** отражает функциональную взаимосвязь входных и выходных переменных и является основой построения нечеткого отношения **XRY,** заданного на произведении X x Y универсальных множеств входных и выходных переменных. Отношение **R** строится как $\bigcup_{i} L_i$.

Система “Набор баскетболистов”

Лингвистические переменные

- **Рост баскетболиста**

  Множество определения – [170,236]

  Множество термов **- {**очень высокий, высокий, средний, низкий**}**

- **Техника игры баскетболиста**

  Множество определения – [0,100]

  Множество термов **- {**отличная, очень хорошая, хорошая, средняя, плохая**}**

- **Уверенность принятия в команду**

  Множество определения – [0,100]

  Множество термов **- {**полная, средняя, малая, не берём**}**

Рост баскетболиста

Множество определения – [170,236]

Очень высокий

высокий

средний

низкий

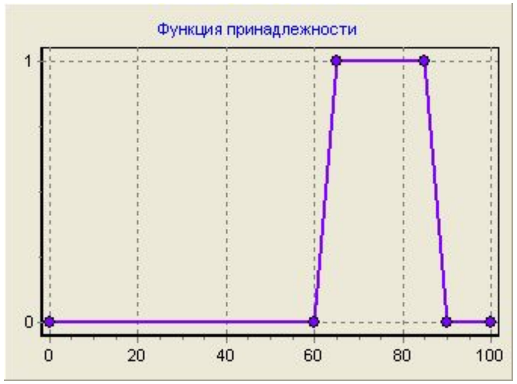
Система "Набор баскетболистов"

Техника игры баскетболиста

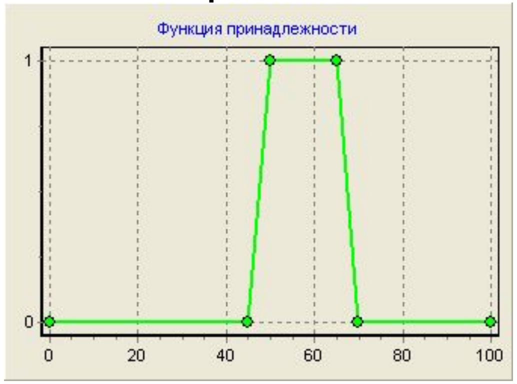
Множество определения – [0,100]

очень хорошая

отличная

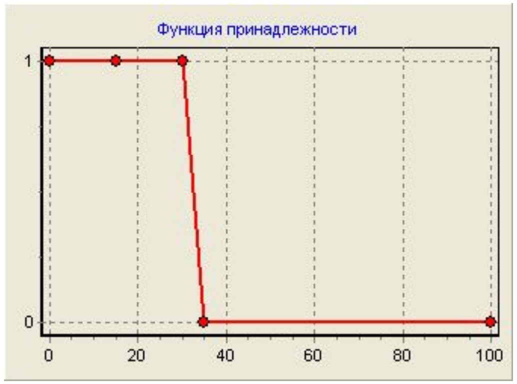


хорошая



средняя

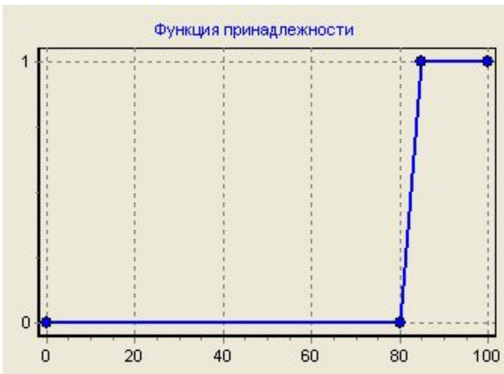
плохая



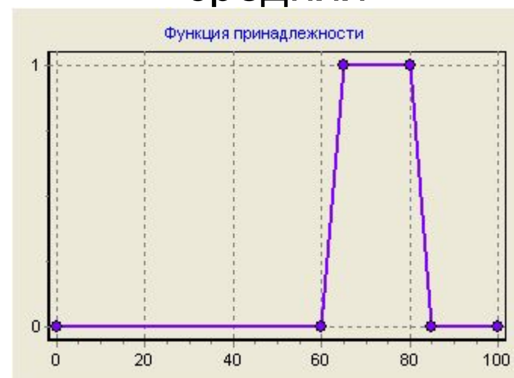
Уверенность принятия в команду

Множество определения – [0,100]

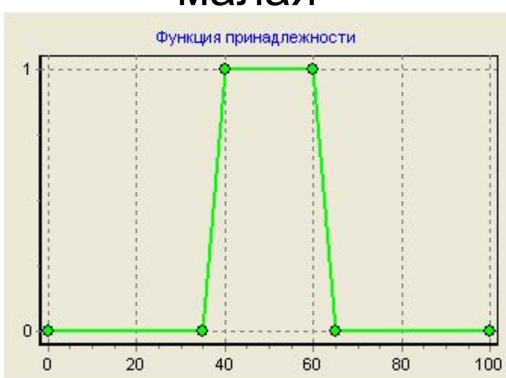
полная



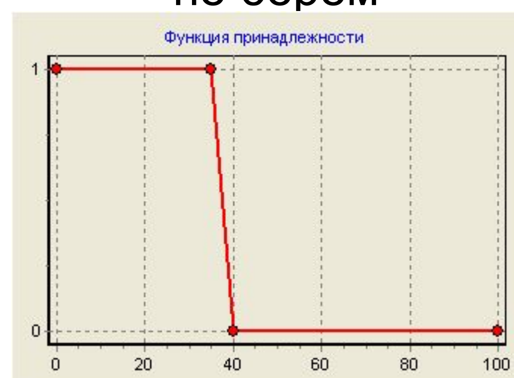
средняя



малая



не берём



Система "Набор баскетболистов"- Правила

Входные лингвистические переменные		Выходная ЛП
Техника игры	Рост игрока	Уверенность отбора
Отлично	Очень высокий	Полная
Отлично	Высокий	Полная
Отлично	Не очень высокий	Средняя
Отлично	Низкий	Средняя
Очень хорошо	Очень высокий	Полная
Очень хорошо	Высокий	Полная
Очень хорошо	Не очень высокий	Средняя
Очень хорошо	Низкий	Средняя
Хорошо	Очень высокий	Полная
Хорошо	Высокий	Полная
Хорошо	Не очень высокий	Средняя
Хорошо	Низкий	Малая
Не очень хорошо	Очень высокий	Средняя
Не очень хорошо	Высокий	Средняя
Не очень хорошо	Не очень высокий	Малая
Не очень хорошо	Низкий	Не берём
Плохо	Очень высокий	Малая
Плохо	Высокий	Малая
Плохо	Не очень высокий	Малая
Плохо	Низкий	Не берём

Схемы нечеткого вывода

- Схема 1: Алгоритм Мамдани (Mamdani). Импликация моделируется минимумом, а агрегация – максимумом.
- Схема 2: Алгоритм Цукамото (Tsukamoto). Исходные посылки – как у предыдущего алгоритма, но предполагается, что функции принадлежности являются монотонными.
- Схема 3. Алгоритм Суджено (Sugeno). Алгоритм предполагает, что правые части правил вывода представлены в виде линейных функций.
- Схема 4. Алгоритм Ларсена (Larsen). В алгоритме Ларсена нечеткая импликация моделируется с использованием операции умножения.
- Схема 5. Упрощенный алгоритм нечеткого вывода. Исходные правила в данном случае задаются в виде:

Если X есть Ai и Y есть Bi , то z=Zi, где *Zi* – четкое значение.

Алгоритм Мамдани

- Пусть некоторая система описывается следующими нечёткими правилами:

$П_1$: если *x* есть *A*, тогда *w* есть *D*,

$П_2$: если *y* есть *B*, тогда *w* есть *E*,

$П_3$: если *z* есть *C*, тогда *w* есть *F*,

где *x, y, z* – имена входных переменных, *w* – имя переменной вывода, а *A, B, C, D, E, F* – заданные функции принадлежности (треугольной формы).

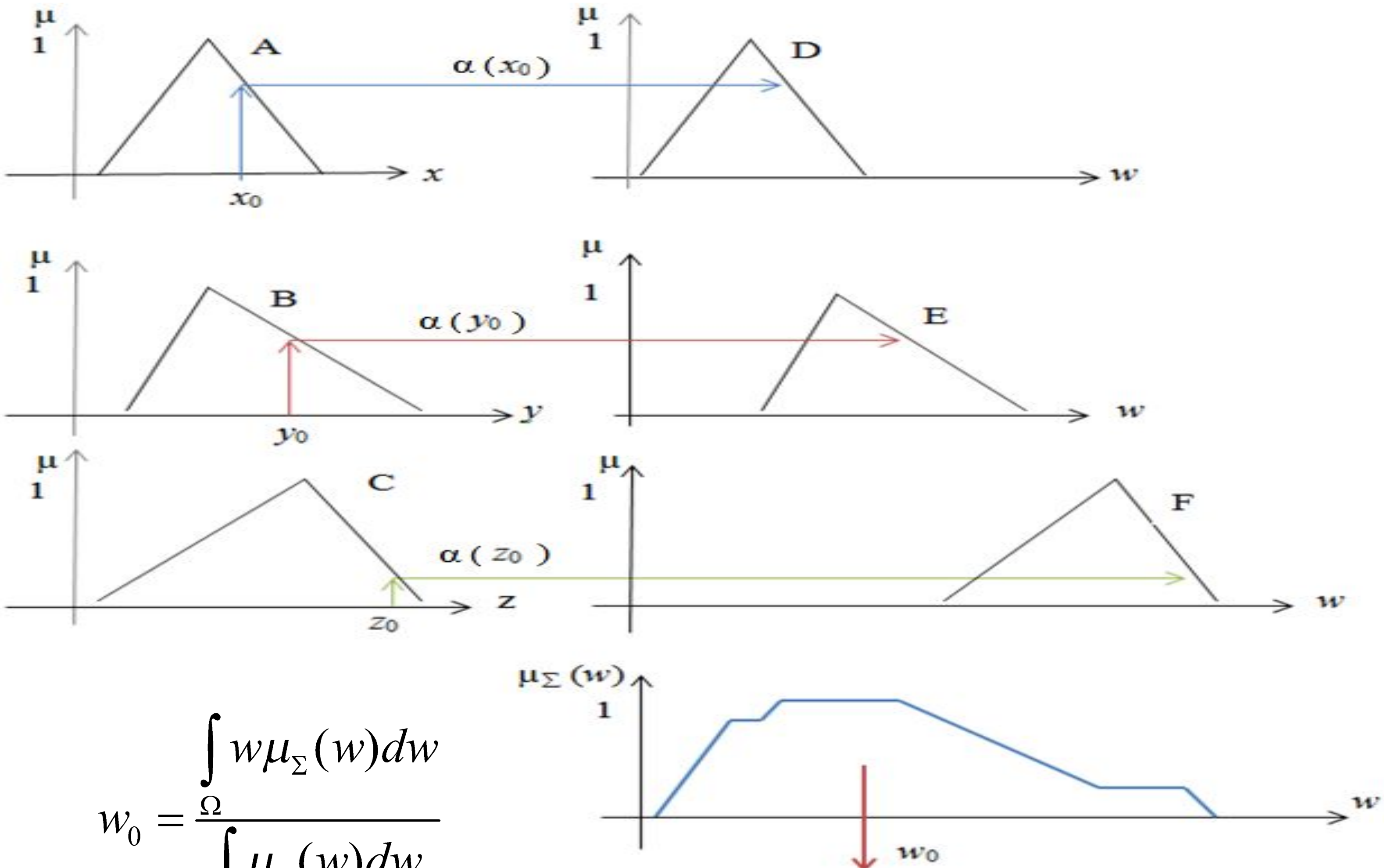
Предполагается, что входные переменные приняли некоторые конкретные (чёткие) значения – x_0, y_0, z_0.

Алгоритм Мамдани

- **Этап 1**. Для данных значений и исходя из функций принадлежности *A*, *B*, *C*, находятся степени истинности $\alpha(x_0)$, $\alpha(y_0)$, $\alpha(z_0)$ для предпосылок каждого из трёх приведённых правил.
- **Этап 2**. Происходит «отсекание» функций принадлежности заключений правил (т.е. *D*, *E*, *F*) на уровнях $\alpha(x_0)$, $\alpha(y_0)$, $\alpha(z_0)$.
- **Этап 3**. Рассматриваются усечённые на втором этапе функции принадлежности и производится их объединение с использованием операции *max*, в результате чего получается комбинированное нечёткое подмножество, описываемое функцией принадлежности $\mu_\Sigma(w)$ и соответствующее логическому выводу для выходной переменной *w*.
- **Этап 4** (при необходимости). Находится чёткое значение выходной переменной, например, с применением центроидного метода: чёткое значение выходной переменной определяется как центр тяжести для кривой $\mu_\Sigma(w)$:

$$w_0 = \frac{\int_\Omega w\mu_\Sigma(w)dw}{\int_\Omega \mu_\Sigma(w)dw}$$

Алгоритм Мамдани



$$w_0 = \frac{\int_{\Omega} w\mu_{\Sigma}(w)dw}{\int_{\Omega} \mu_{\Sigma}(w)dw}$$

Приведение к четкости (скаляризация)

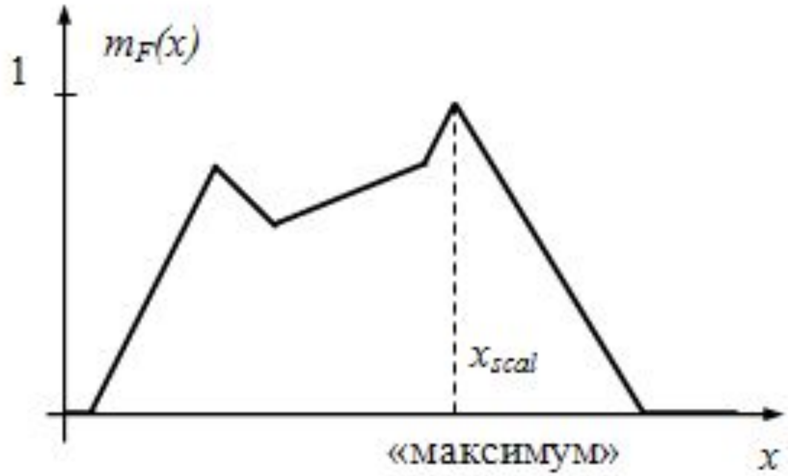
Скаляризация осуществляется различными способами. Чаще всего используется определение «центра тяжести» H функции принадлежности нечеткого подмножества $\int\limits_{[\underline{x},\overline{x}]} \frac{m_F(x)}{x}$ по формуле (см. рис. 5):

$$H = (\int\limits_{\underline{x}}^{\overline{x}} x m_F(x)dx) / (\int\limits_{\underline{x}}^{\overline{x}} m_F(x)dx).$$

Скаляризация методом «центра тяжести».

Приведение к четкости (скаляризация)

При этом используются три разновидности взятия максимума: наибольшего из максимумов (*LOM*), наименьшего из максимумов (*SOM*) и центра максимумов (*MOM*).



Скаляризация методом «максимума».

Алгоритм Ларсена

1. Первый этап такой же, как в алгоритме Мамдани.

2. На втором этапе сначала определяются (как в алгоритме Мамдани) уровни «отсечения» α_1 и α_2:

$$\alpha_1 = A_1(x_0) \wedge B_1(y_0), \quad \alpha_2 = A_2(x_0) \wedge B_2(y_0),$$

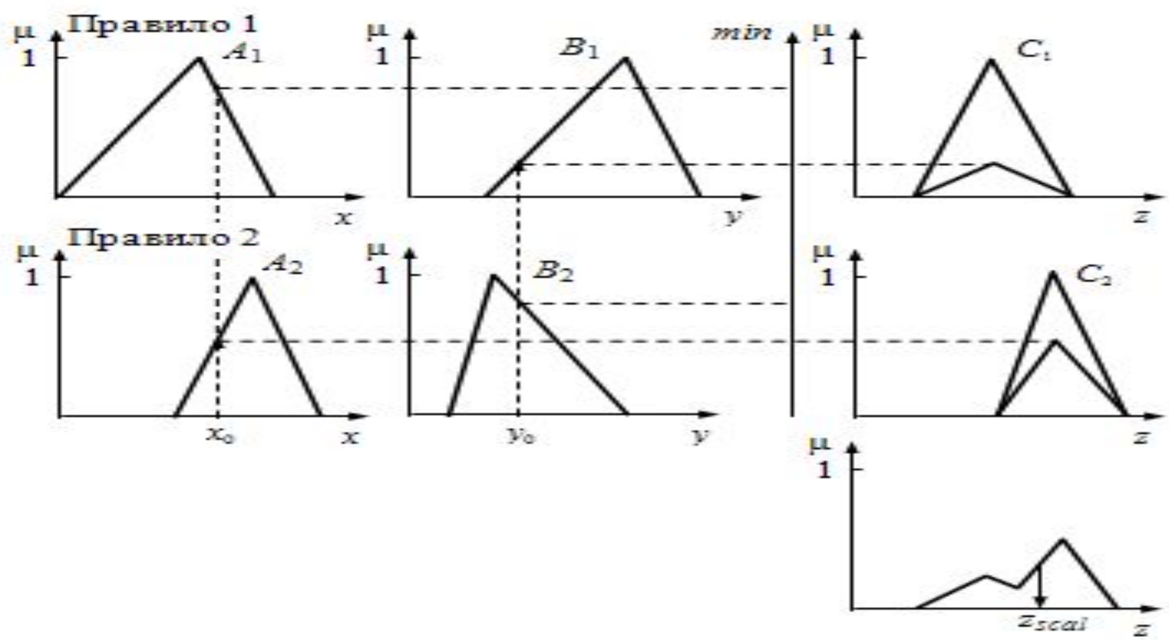
а затем – нечеткие подмножества $\alpha_1 C_1(z)$, $\alpha_2 C_2(z)$.

3. Вычисляется обобщенное нечеткое подмножество с функцией принадлежности $\mu_\Sigma(z) = C(z) = (\alpha_1 C_1(z)) \vee (\alpha_2 C_2(z))$.

В случае n правил $\quad \mu_\Sigma(z) = C(z) = \bigvee_{i=1}^{n} (\alpha_i C_i(z))$.

4. Выполняется приведение к четкости методом центра тяжести.

Алгоритм Ларсена иллюстрирует рис.



Задача об управления кондиционером

Правила:

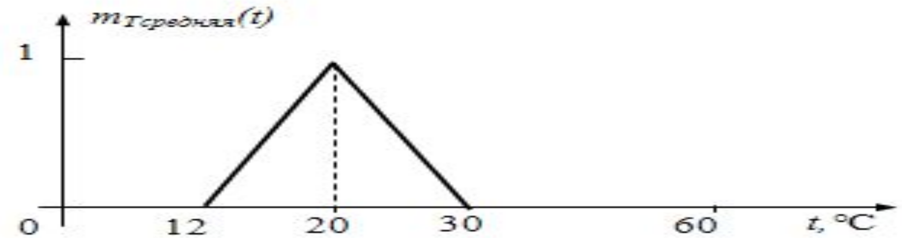
1. Если температура воздуха в комнате высокая, то скорость вращения вентилятора высокая.

2. Если температура воздуха в комнате средняя, то скорость вращения вентилятора средняя.

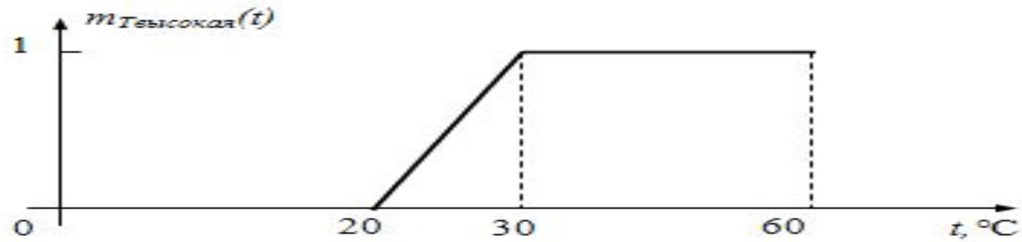
3. Если температура воздуха в комнате низкая, то скорость вращения вентилятора низкая.

Задача об управления кондиционером

Нечеткое подмножество «низкая», определенное на множестве значений температуры.



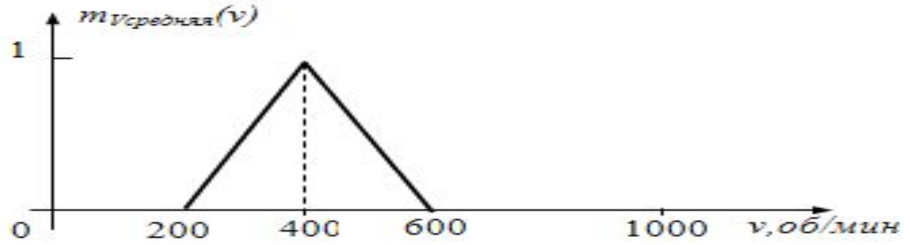
Нечеткое подмножество «средняя», определенное на множестве значений температуры



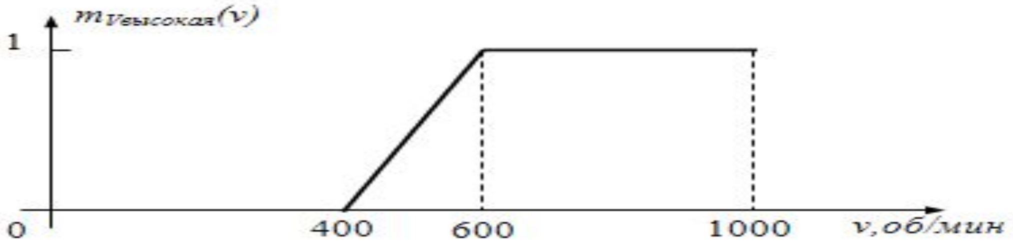
Нечеткое подмножество «высокая», определенное на множестве значений температуры

Задача об управления кондиционером

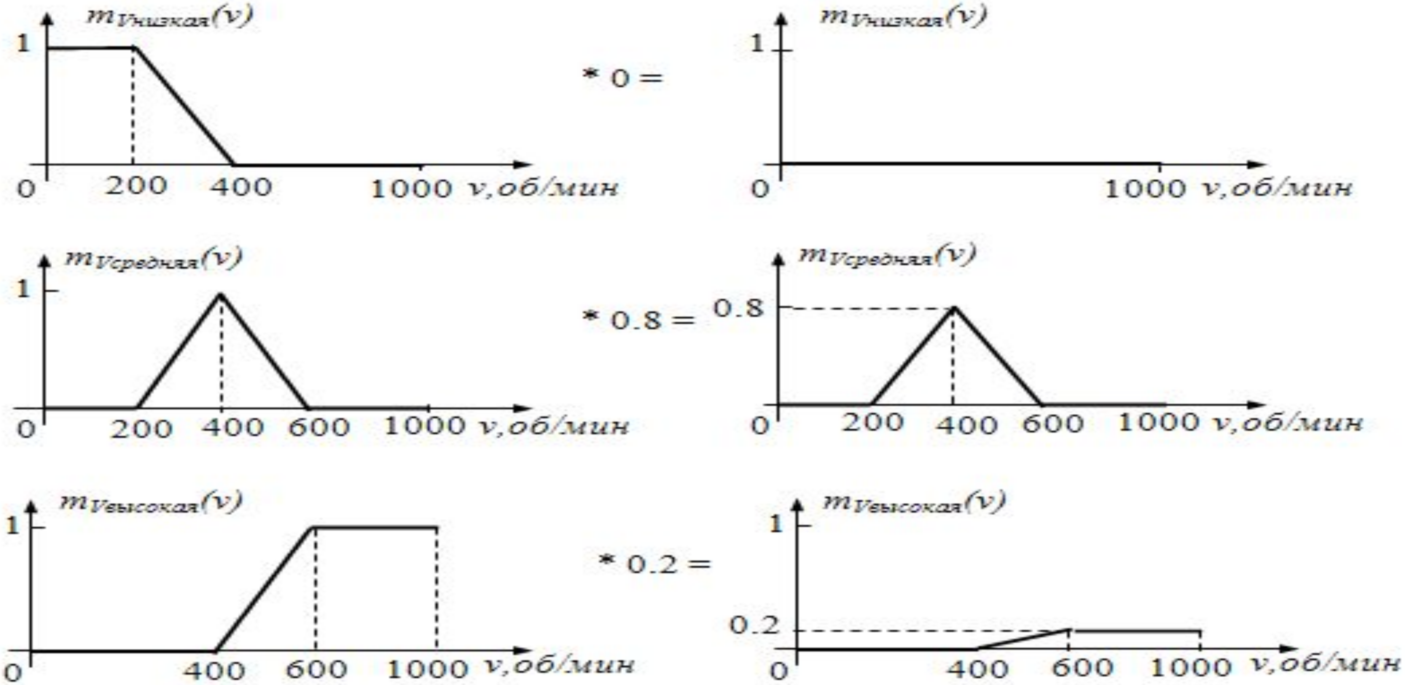
Нечеткое подмножество «низкая», определенное на множестве значений скорости вращения вентилятора



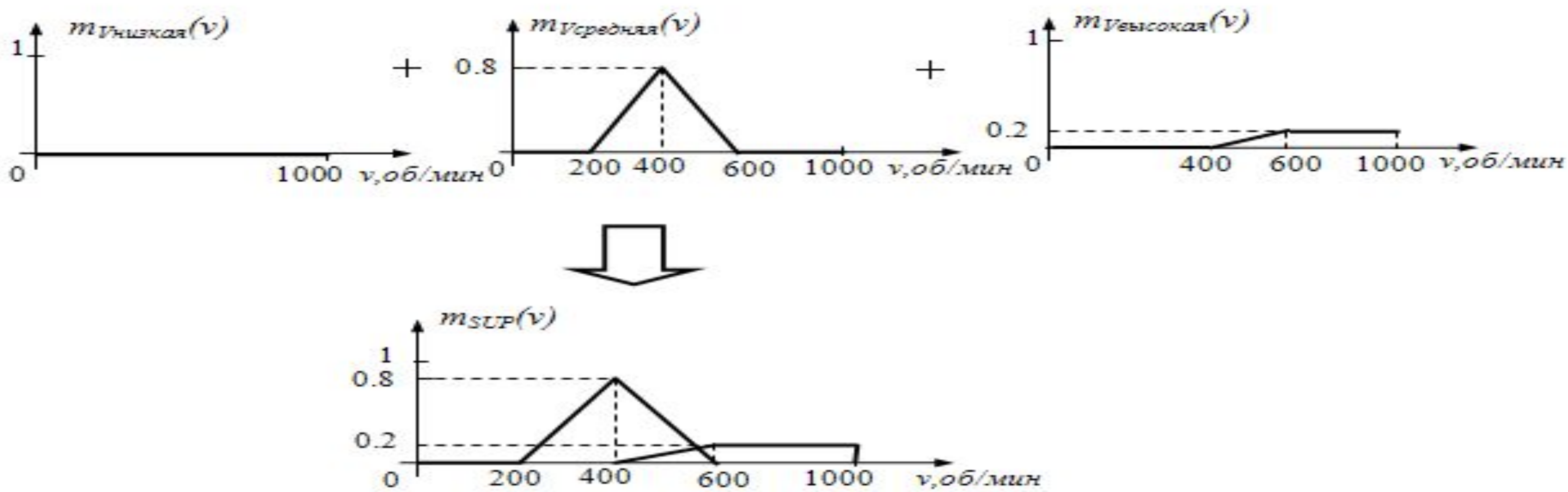
Нечеткое подмножество «средняя», определенное на множестве значений скорости вращения вентилятора



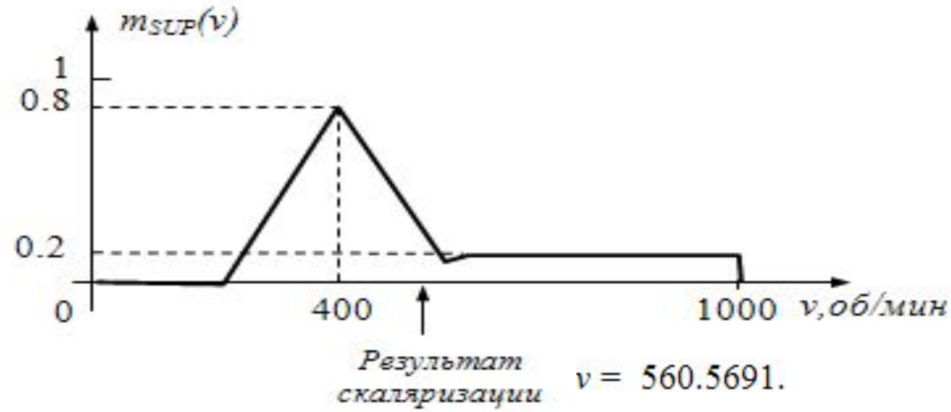
Нечеткое подмножество «высокая», определенное на множестве значений скорости вращения вентилятора



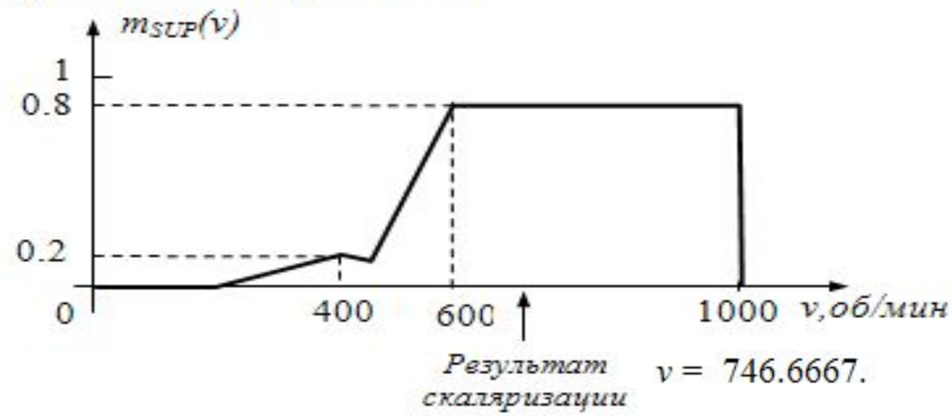
Модификация нечетких подмножеств, определенных на интервале изменения скорости вращения вентилятора



Результат композиции нечетких подмножеств



Получение скалярного значения скорости вращения вентилятора методом «центра тяжести» для $t = 22°C$



Получение скалярного значения скорости вращения вентилятора методом «центра тяжести» для $t = 28°C$

Алгоритм Цукамото

Исходные посылки — как у предыдущего алгоритма, но в данном случае предполагается, что функции $C_1(z)$, $C_2(z)$ являются монотонными.

1. Первый этап — такой же, как в алгоритме Mamdani.

2. На втором этапе сначала находятся (как в алгоритме Mamdani) уровни «отсечения» α_1 и α_2, а затем — посредством решения уравнений

$$\alpha_1 = C_1(z_1), \quad \alpha_2 = C_2(z_2)$$

— четкие значения (z_1 и z_2) для каждого из исходных правил.

3. Определяется четкое значение переменной вывода (как взвешенное среднее z_1 и z_2):

$$z_0 = \frac{\alpha_1 z_1 + \alpha_2 z_2}{\alpha_1 + \alpha_2};$$

в общем случае (дискретный вариант центроидного метода)

$$z_0 = \frac{\sum_{i=1}^{n} \alpha_i z_i}{\sum_{i=1}^{n} \alpha_i}.$$

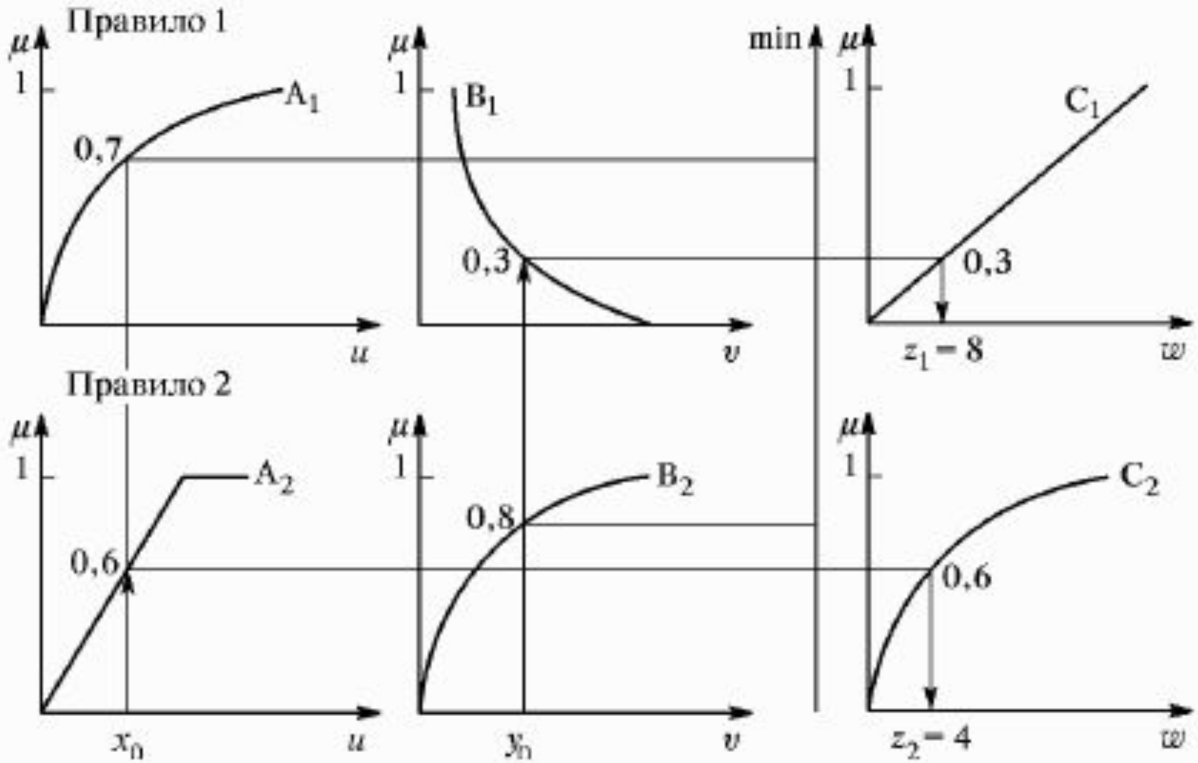
Пусть имеем $A_1(x_0) = 0{,}7$, $A_2(x_0) = 0{,}6$, $B_1(y_0) = 0{,}3$, $B_2(y_0) = 0{,}8$, соответствующие уровни отсечения

$$\alpha_1 = \min\left(A_1(x_0), B_1(y_0)\right) = \min(0{,}7; 0{,}3) = 0{,}3,$$

$$\alpha_2 = \min\left(A_2(x_0), B_2(y_0)\right) = \min(0{,}6; 0{,}8) = 0{,}6$$

и значения $z_1 = 8$ и $z_2 = 4$, найденные в результате решения уравнений

$$C_1(z_1) = 0{,}3, \quad C_2(z_2) = 0{,}6.$$



При этом четкое значение переменной вывода $z_0 = (8 \cdot 0{,}3 + 4 \cdot 0{,}6)/(0{,}3 + 0{,}6) = 6$

Алгоритм Суджено и Такажи

Π_1: если x есть A_1 и y есть B_1, тогда $z_1 = a_1 x + b_1 y$,
Π_2: если x есть A_2 и y есть B_2, тогда $z_2 = a_2 x + b_2 y$.

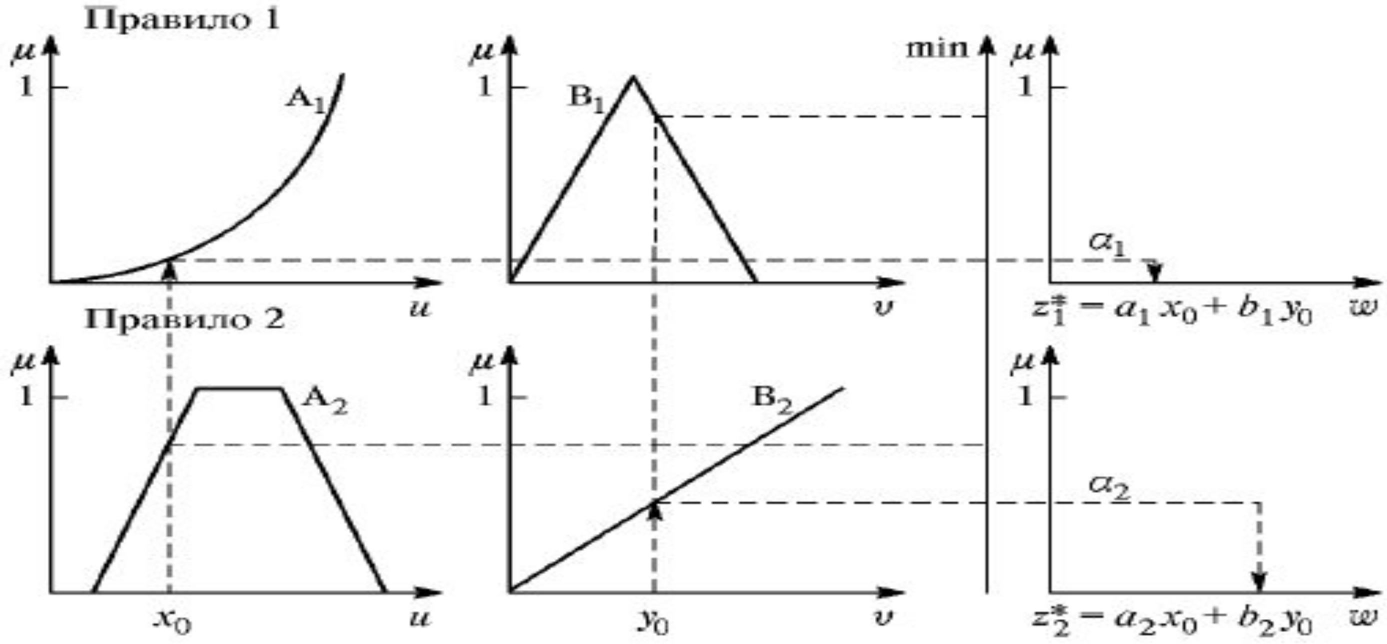
Представление алгоритма

1. Первый этап — как в алгоритме Mamdani.
2. На втором этапе находятся $\alpha_1 = A_1(x_0) \wedge B_1(y_0)$, $\alpha_2 = A_2(x_0) \wedge B_2(y_0)$ и индивидуальные выходы правил:

$$z_1^* = a_1 x_0 + b_1 y_0,$$
$$z_2^* = a_2 x_0 + b_2 y_0.$$

3. На третьем этапе определяется четкое значение переменной вывода:

$$z_0 = \frac{\alpha_1 z_1^* + \alpha_2 z_2^*}{\alpha_1 + \alpha_2}.$$



Алгоритм упрощенного выбора

П$_1$: если x есть A_1 и y есть B_1, тогда $z_1 = c_1$,

П$_2$: если x есть A_2 и y есть B_2, тогда $z_2 = c_2$,

где c_1 и c_2 — некоторые обычные (четкие) числа.

Описание алгоритма

1. Первый этап — как в алгоритме Mamdani.

2. На втором этапе находятся числа $\alpha_1 = A_1(x_0) \wedge B_1(y_0)$, $\alpha_2 = A_2(x_0) \wedge B_2(y_0)$.

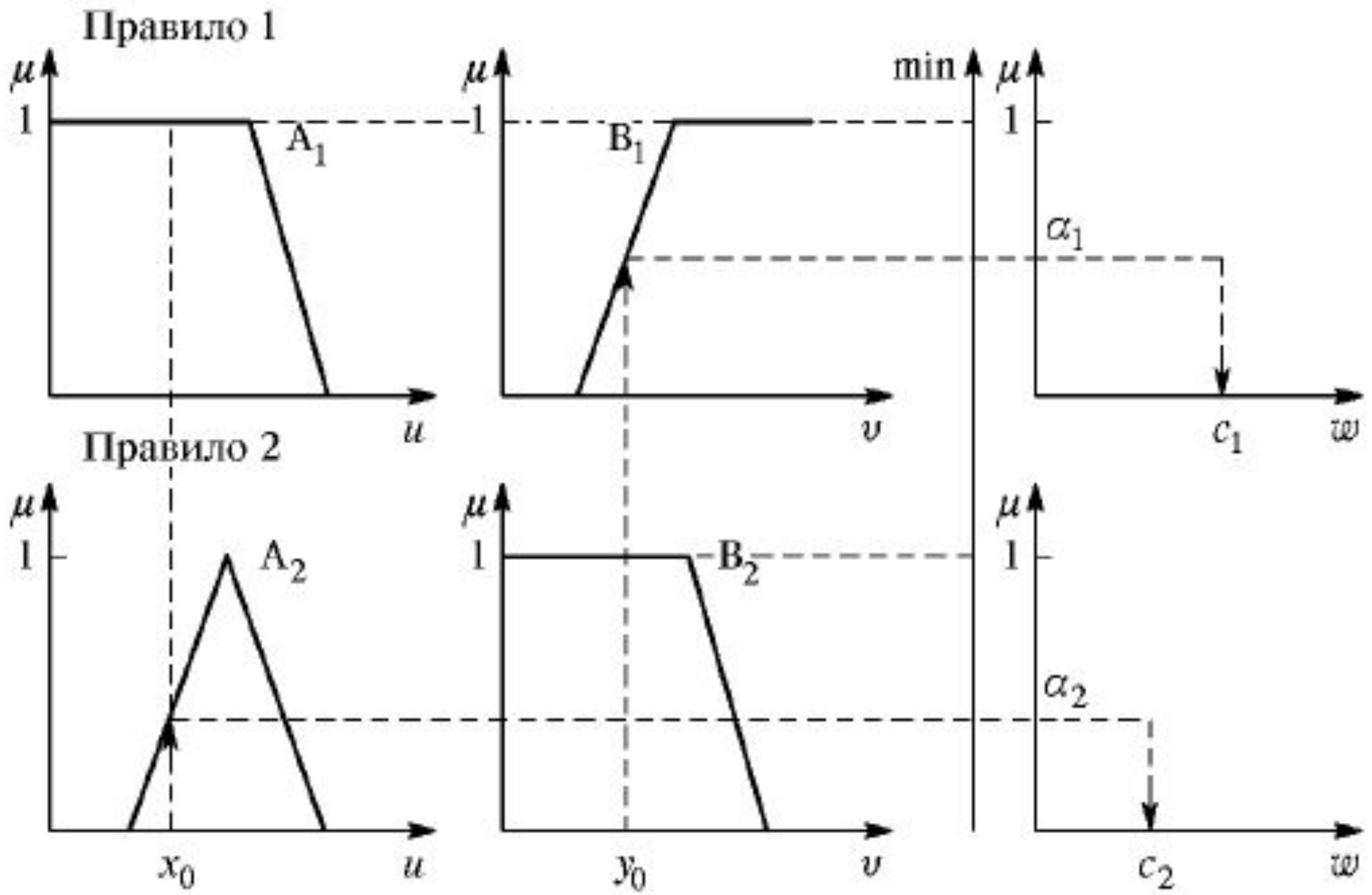
3. На третьем этапе находится четкое значение выходной переменной по формуле

$$z_0 = \frac{\alpha_1 c_1 + \alpha_2 c_2}{\alpha_1 + \alpha_2},$$

или — в общем случае наличия n правил — по формуле

$$z_0 = \frac{\sum_{i=1}^{n} \alpha_i c_i}{\sum_{i=1}^{n} \alpha_i}.$$

Алгоритм упрощенного выбора



- Спасибо за внимание!
- Успехов!!!

Нейроны и нейронные сети

Нечеткая логика и нейронные сети

Нейронные сети...

- раздел искусственного интеллекта, в котором для обработки сигналов используются явления, аналогичные происходящим в нейронах живых существ.

Задачи

Аппроксимация

Ассоциативное управление

Классификация и распознавание образов

Идентификация и оценивание

Прогнозирование

Задачи, успешно решаемые нейросетями

- распознавание зрительных, слуховых образов;
- ассоциативный поиск информации и создание ассоциативных моделей; синтез речи; формирование естественного языка;
- формирование моделей и различных нелинейных и трудно описываемых математически систем, прогнозирование развития этих систем во времени:
- применение на производстве; прогнозирование развития циклонов и других природных процессов, прогнозирование изменений курсов валют и других финансовых процессов;
- системы управления и регулирования с предсказанием; управление роботами, другими сложными устройствами
- разнообразные конечные автоматы: системы массового обслуживания и коммутации, телекоммуникационные системы;
- принятие решений и диагностика, исключающие логический вывод; особенно в областях, где отсутствуют четкие математические модели: в медицине, криминалистике, финансовой сфере.

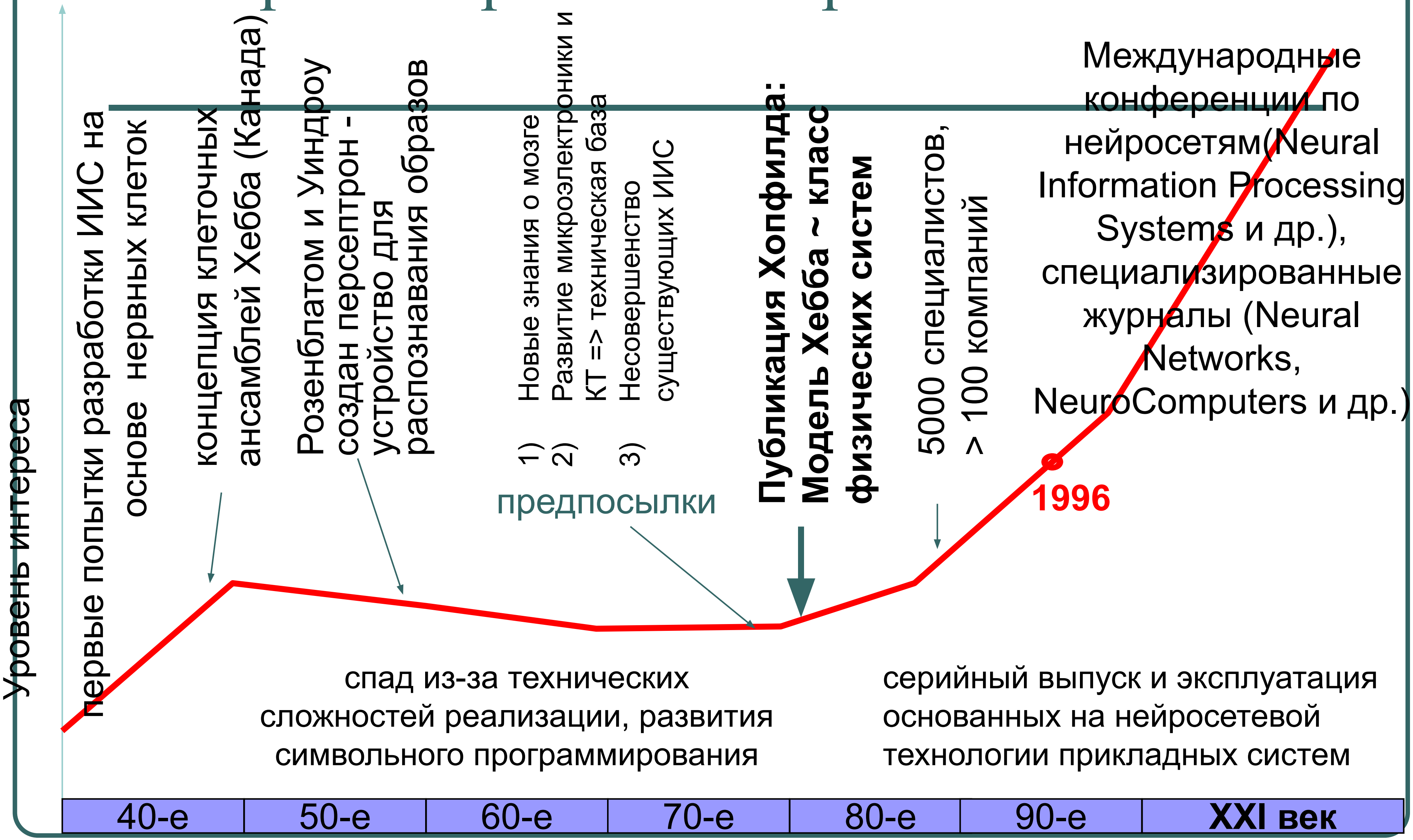
Сферы знаний

Нейрокомпьютер…

- программно-техническая система (ее также можно назвать специализированной ЭВМ), которая реализует, или, как говорят, обеспечивает некоторую формальную модель естественной нейронной сети .

Программирование нейрокомпьютеров осуществляется не заданием последовательности команд, а предъявлением образцов, примеров решения задач из нужной области

История нейрокомпьютера



Некоторые сведения о мозге

Самая сложная из известных систем переработки информации.

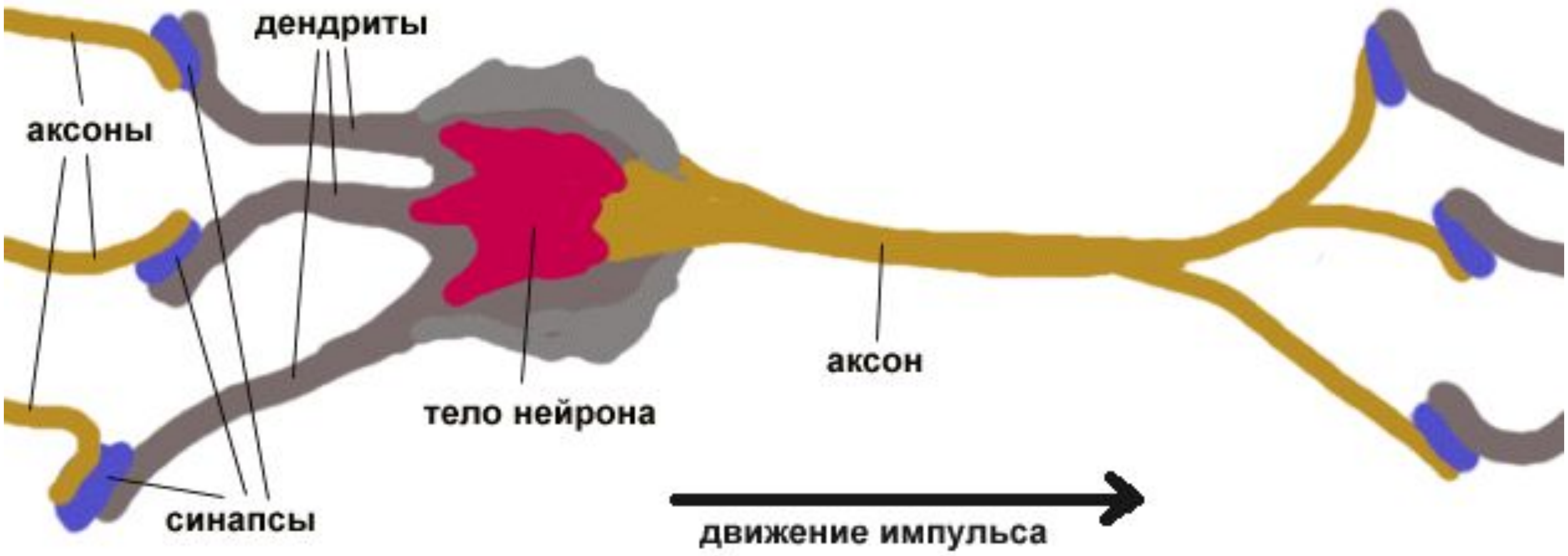
- В нем содержится около 100 млрд. нейронов, каждый из которых имеет в среднем 100 тыс. связей.
- Надежен: функционирует при потере (отмирании) нейронов
- Обработка огромных объемов информации осуществляется за доли секунды, несмотря на то, что время реакции нейрона несколько миллисекунд.

Хорошо изучена структура и функции отдельных нейронов

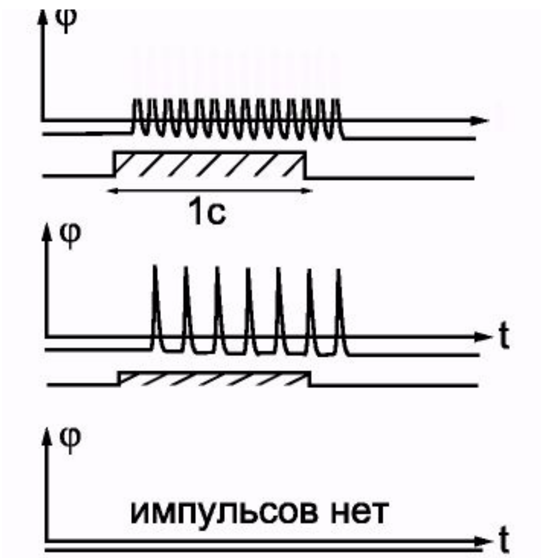
Есть некоторые данные об организации внутренних и внешних связей между нейронами некоторых структурных образований мозга

Мало известно об участии различных структур в процессах переработки информации.

Биологический нейрон



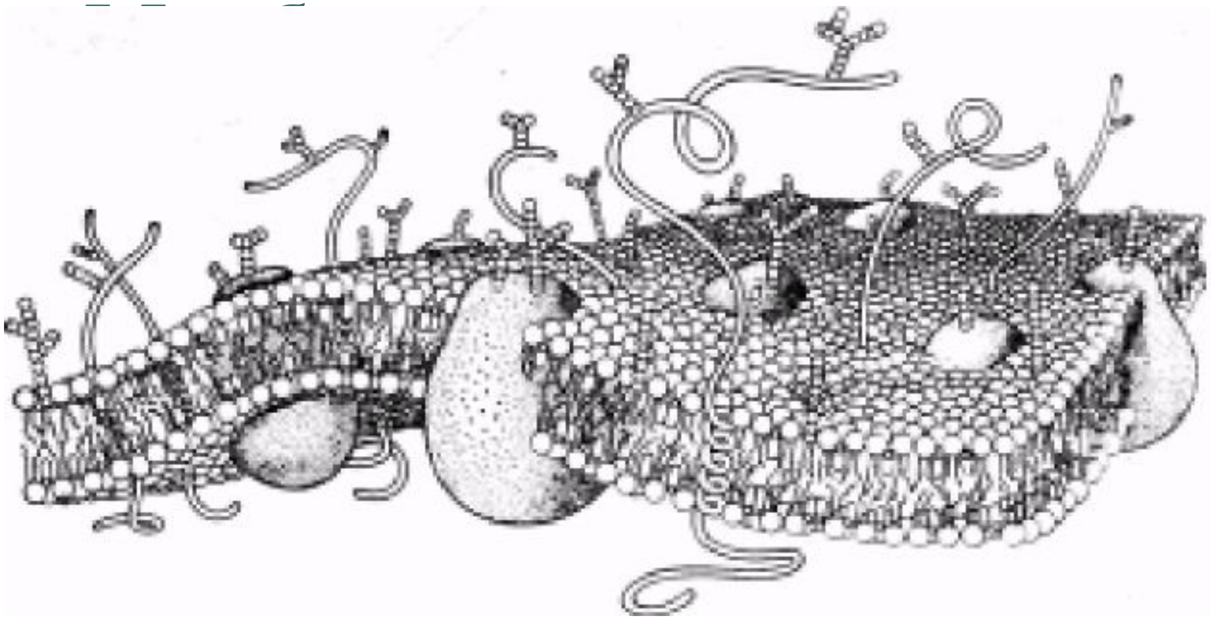
Нервный импульс



- — процесс распространения возбуждения по аксону от тела клетки (аксонного холмика) до окончания аксона.
- - основная единица информации, передаваемая по волокну.
- … передаётся в виде скачков потенциала внутриклеточной среды по отношению к внешней среде, окружающей клетку со скоростью от 1 до 100 м/с.

Рефрактерность – отсутствие возбудимости нервной клетки после предшествующего возбуждения.

Период рефрактерности – минимальный интервал времени между нервными импульсами (10^{-4}.. 10^{-3} с)



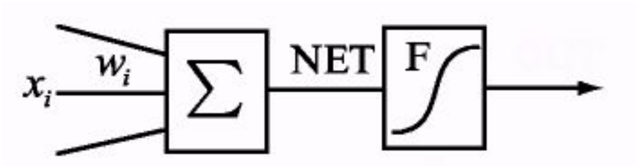
Обеспечивает проведение нервных импульсов по волокну

Толщина мембраны около 10 нм

- Мера возбуждения клетки = уровень поляризации её мембраны, зависящий от суммарного количества нейромедиатора (химической субстанции), выделенной на всех синапсах.

Нейроподобный элемент (НПЭ) или формальный нейрон

- Модель физического нейрона.



НПЭ состоит из взвешенного сумматора и нелинейного элемента. Функционирование определяется формулами:

$$NET = \sum_i w_i x_i \quad \text{и} \quad OUT = F(NET - \theta)$$

где x_i — входные сигналы, совокупность x_i образует вектор X;
w_i — весовые коэффициенты, совокупность w_i образует вектор весов W;
NET — взвешенная сумма входных сигналов, значение NET передается на нелинейный элемент;
Θ — пороговый уровень данного нейрона;
F — нелинейная функция, называемая *функцией активации.*

НПЭ имеет несколько входных сигналов x и один выходной сигнал OUT. Параметры НПЭ: вектор весов W, пороговый уровень *Θ* и вид функции активации F.

Принцип работы НПЭ

1. На НПЭ поступает входной вектор X, представляющий собой выходные сигналы других НПЭ.

   Этот входной сигнал соответствует сигналам, поступающим в синапсы биологических нейронов

2. Каждый входной сигнал умножается на соответствующий вес w1 , w2, ... wn - аналог эффективности сигнала.

   Вес является скалярной величиной, положительной для возбуждающих и отрицательной для тормозящих связей.

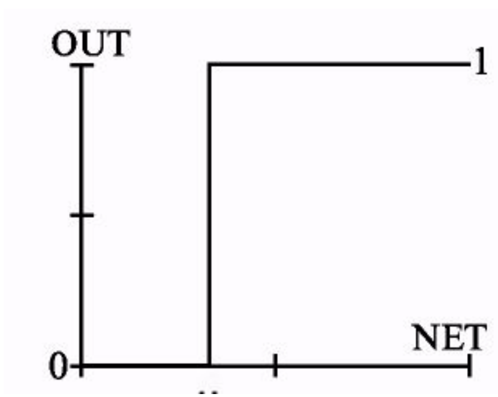
3. Взвешенные весами связей входные сигналы поступают на блок суммирования, соответствующий телу клетки, где осуществляется их алгебраическое суммирование и определяется уровень возбуждения НПЭ.
4. Выходной сигнал нейрона y определяется путем пропускания уровня возбуждения через функцию активации.

Виды функций активации **F**

Жесткая ступенька и пологая ступенька

Жёсткая ступенька

$$OUT = \begin{cases} 0, & NET < \theta \\ 1, & NET \ge \theta \end{cases}$$



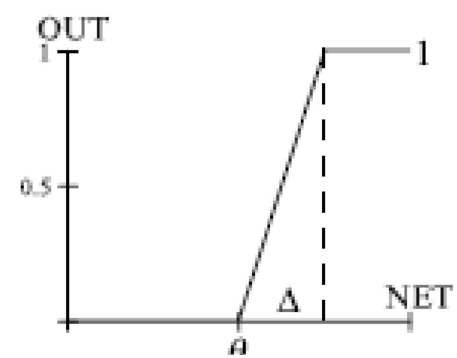
+ простая;
+ реализация требует малых затрат;

-не позволяет моделировать схемы с непрерывными сигналами;

- затруднено обучение нейросетей.

Пологая ступенька

$$OUT = \begin{cases} 0, & NET \le \theta \\ \dfrac{(NET - \theta)}{\Delta}, & \theta \le NET < \theta + \Delta \\ 1, & NET \ge \theta + \Delta \end{cases}$$

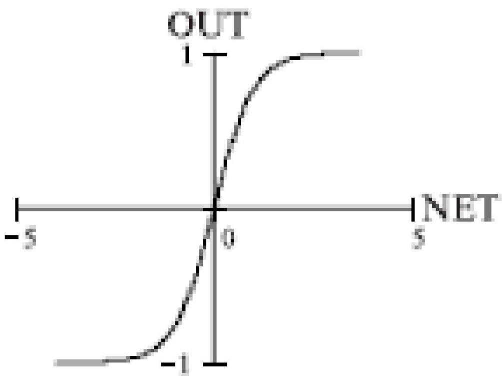


+ легко рассчитывается;
+ обучение затруднено.

Гиперболический тангенс и функция Ферми

Гиперболический тангенс

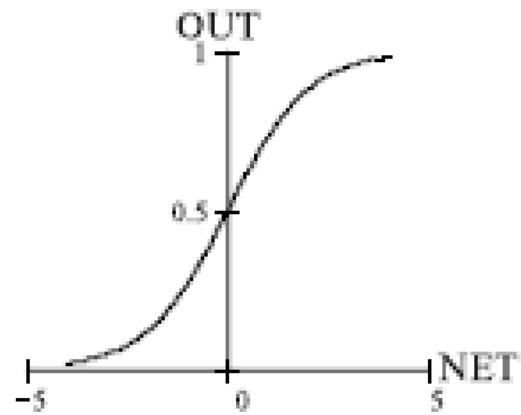
$$OUT = \mathrm{th}(NET) = \frac{e^{NET} - e^{-NET}}{e^{NET} + e^{-NET}}$$



* применяется для сетей с непрерывными сигналами;
\+ легкое обучение.

Логистическая функция (функция Ферми)

$$OUT = \frac{1}{1 + e^{-NET}}$$

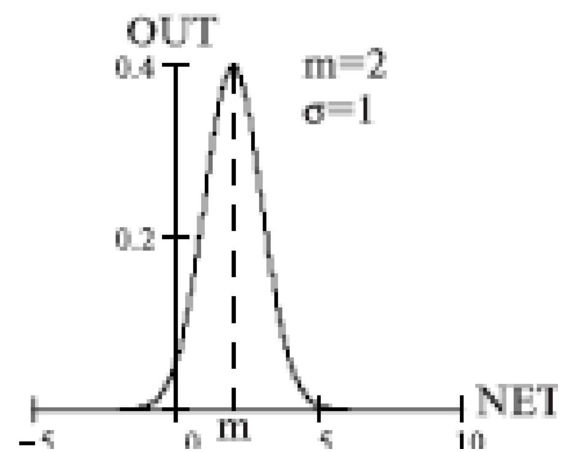


* применяется для многослойных персептронов;
\+ широкий диапазон сигналов;
\+ легкое обучение.

Особые фу тивации

- Экспонента $OUT = e^{-NET}$
- SOFTMAX-функция (выходы-вероятности) $OUT = \dfrac{e^{NET}}{\sum_i e^{NET_i}}$
- Линейная функция (не требуется последовательное соединение слоёв $OUT = K\,NET,\ K\text{=const}$
- Гауссова кривая (реакция НПЭ должна быть максимальна для некоторого значения)

$$OUT = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(NET-m)}{2\sigma^2}}$$



Выбор функции активации

определяется…

1. спецификой задачи.
2. удобством реализации на ЭВМ, в виде электрической схемы или другим способом.
3. алгоритмом обучения: некоторые алгоритмы накладывают ограничения на вид функции активации, их нужно учитывать.

Чаще всего вид нелинейности не оказывает принципиального влияния на решение задачи. Однако удачный выбор может сократить время обучения в несколько раз

Ограничения модели нейрона

- Вычисления выхода нейрона предполагаются мгновенными, не вносящими задержки.
- В модели отсутствуют нервные импульсы.
- Нет модуляции уровня сигнала плотностью импульсов, как в нервной системе.
- Не появляются эффекты синхронизации, когда скопления нейронов обрабатывают информацию синхронно, под управлением периодических волн возбуждения-торможения.
- Нет четких алгоритмов для выбора функции активации.
- Нет механизмов, регулирующих работу сети в целом (пример - гормональная регуляция активности в биологических нервных сетях).
- Чрезмерная формализация понятий: "порог", "весовые коэффициенты".
- Не поддерживается многообразие синапсов. Тормозные и возбуждающие синапсы реализуются в данной модели в виде весовых коэффициентов противоположного знака, но это далеко не все виды.
- В модели не прослеживается различие между градуальными потенциалами и нервными импульсами.

Нейроподобная сеть

- совокупность нейроподобных элементов, определенным образом соединенных друг с другом и с внешней средой.

Входной вектор (кодирующий входное воздействие или образ внешней среды) подается на сеть путем активации входных нейроподобных элементов.

Множество выходных сигналов нейронной сети $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n$ называется **вектором выходной активности**, или **паттерном активности** нейронной сети.

Особенности архитектуры нейросети

- топология межнейронных связей;
- выбор определенного подмножества НПЭ для ввода и вывода информации;
- наличие или отсутствие конкуренции;
- направление и способ управления и синхронизации информационных потоков между нейронами

обуславливают конкретный вид выполняемого сетью преобразования информации

Искусственные нейронные сети

- Состоят из слоев искусственных нейронов, взаимосвязанных между собой
- Имеются входы и выходы сети – входной и выходной векторы
- Сеть учится правильно реагировать на сигналы, поступающие на входы
- Реализуются аппаратно или программно (чаще)
- Существуют много разных моделей, отличающихся структурой, особенностями реализации нейронов, методами и алгоритмами обучения
- Используются для задач распознавания, классификации, предсказания в системах диагностики, финансового анализа, мониторинга, интеллектуальных роботах и т.д.

Важнейшие свойства биологических нейросетей

- Способность к полной обработке информации: ассоциативность (сеть может восстанавливать полный образ по его части), способность к классификации, обобщению, абстрагированию и множество других.
- Надежность. Биологические НС обладают фантастической надежностью: выход из строя даже 10% нейронов в нервной системе не прерывает ее работы. По сравнению с последовательными ЭВМ, основанными на принципах фон Неймана, где сбой одной ячейки памяти или одного узла в аппаратуре приводит к краху системы.
- Параллельность обработки информации.
- Самоорганизация. В процессе работы биологические НС самостоятельно, под воздействием внешней среды, обучаются решению разнообразных задач. Неизвестно никаких принципиальных ограничений на сложность задач, решаемых биологическими нейронными сетями. Нервная система сама формирует алгоритмы своей деятельности, уточняя и усложняя их в течение жизни.
- Биологические НС являются аналоговыми системами

Отличия между биологическими НС и ЭВМ на архитектуре фон Неймана

	Машина фон Неймана	Биологическая нейронная система
Процессор	Сложный	Простой
	Высокоскоростной	Низкоскоростной
	Один или несколько	Большое количество
Память	Отделена от процессора	Интегрирована в процессор
	Локализована	Распределенная
	Адресация не по содержанию	Адресация по содержанию
Вычисления	Централизованные	Распределенные
	Последовательные	Параллельные
	Хранимые программы	Самообучение
Надежность	Высокая уязвимость	Живучесть
Специализация	Численные и символьные операции	Проблемы восприятия
Среда функционирования	Строго определенная	Плохо определенная
	Строго ограниченная	Без ограничений

Подходы к созданию нейронных сетей

- Информационный подход: безразлично, какие механизмы лежат в основе работы искусственных нейронных сетей, важно лишь, чтобы при решении задач информационные процессы в НС были подобны биологическим.
- Биологический подход: при моделировании важно полное биоподобие, и необходимо детально изучать работу биологического нейрона.

Крупные работы в исследованиях биологических нейронных сетей принадлежат Эндрю Хаксли, Алану Ходжкину, Бернарду Катцу, Джону Экклзу, Стивену Куффлеру и др.

Методы исследования нейроподобных сетей

Метод	Особенности
аналитическое исследование	- сложность из-за большого количества НПЭ + интересные аналитические результаты получены для многих моделей нейроподобных сетей
математическое (имитационное моделирование)	+ дает возможность создать практически любые модели - из-за последовательного характера их работы удается исследовать модели ограниченного размера
физическое моделирование	+ позволяет быстро получить достоверные результаты работы модели - техническая сложность аппаратной реализации большого количества НПЭ с многими адаптивными связями

Категории моделей нейронных сетей

- модели отдельных нейронов;
- модели небольших групп нейронов;
- модели нейронных сетей;
- модели мыслительной деятельности и мозга в целом.

Виды обучения нейронных сетей

- Обучение с учителем
  – Пример при обучении – входной вектор и соответствующий ему выходной вектор
- Обучение без учителя
  – Нейронная сеть занимается кластеризацией (разбиением на классы) входных векторов с учетом их похожести в пространстве признаков
- Обучение с учителем-внешней средой
  – Нейронная сеть, взаимодействуя с внешней средой, получает от нее поощрения и наказания, и стремится уменьшить вероятность поучения наказаний и увеличить - поощрений

Алгоритмы обучения

	С учителем	Без учителя
Дано	вектор X, ожидаемые выходные сигналы нейрона $dj \in D$	вектор X
Подбор значений	фактические выходные сигналы нейрона должны принимать значения, как можно более близкие к ожидаемым	сеть учится давать наилучшие значения выходов. Что понимается под "наилучшими" — определяется алгоритмом обучения.
Новые значения	.. за счет способности к обобщению сетью, если подать на вход вектор, который не встречался при обучении.	всегда

Методы обучения МСП

Алгоритм обратного распространения ошибки

классический

Градиентные

Выявление градиента целевой функции

Алгоритм переменной метрики

Алгоритм наискорейшего спуска

Алгоритм сопряжения градиентов

Алгоритм Левенберга-Марквардта

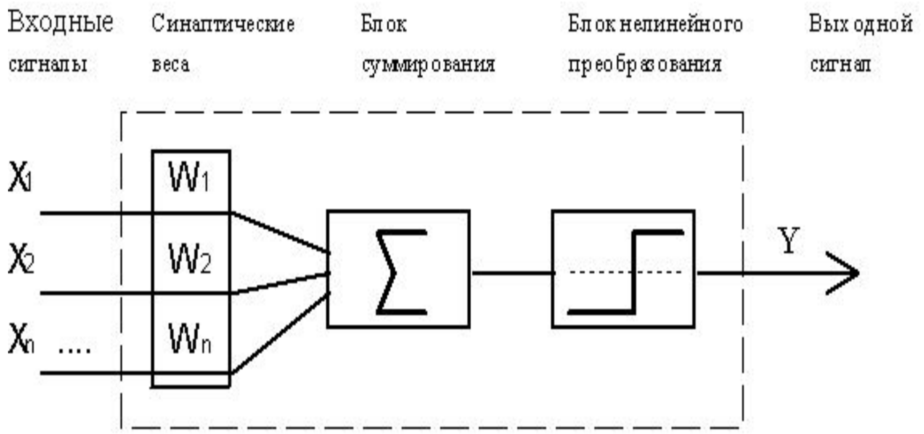
Эвристические методы

На основе личного опыта автора в области обучения нейронных сетей

+ элементы глобальной оптимизации (имитации отжига, генетические алгоритмы)

1943 г.

Модель МакКаллока-Питса



Выходной сигнал:

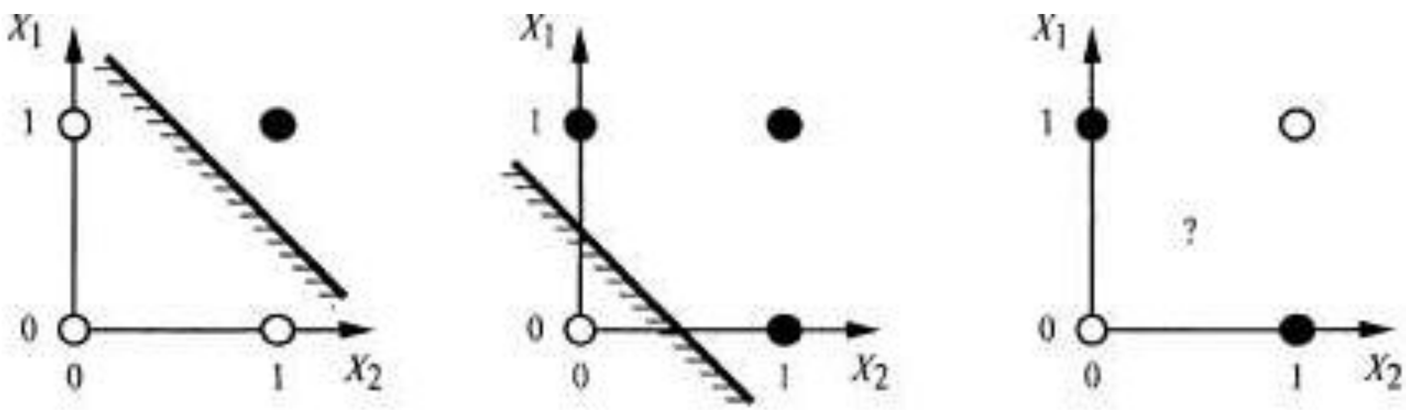
$$net = \sum_{i=1}^{n} W_i x_i$$

Пороговая функция:

$$Y = f(net) = \begin{cases} 1, & net > \Theta \\ 0, & net \le \Theta \end{cases}$$

Построение дискретной модели обосновывается проявлением рефракции у биологических нейронов, приводящей к тому, что нейрон может изменять свое состояние с конечной частотой, причем длительность периодов бездействия зависит от частоты его срабатывания.

Логические операции

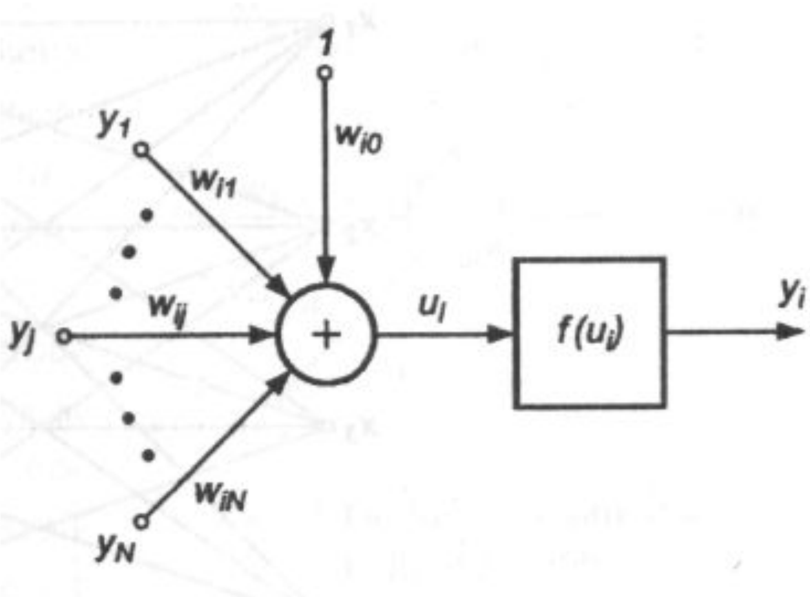


а) X_1 and X_2 б) X_1 or X_2 в) X_1 xor X_2

Алгоритм обучения персептрона Маккалока-Питтса

- Случайно выбираются веса w_{ij}
- На входы подается обучающий вектор x и рассчитывается выходной сигнал y_i с использованием пороговой функции
- Если ожидаемое значение d_i совпадает с y_i, то веса не изменяются
- Если $y_i=0$ и $d_i=1$, то $w_{ij}(t+1)=w_{ij}(t)+x_j$
- Если $y_i=1$ и $d_i=0$, то $w_{ij}(t+1)=w_{ij}(t)-x_j$
- Повторяются шаги, начиная со второго, для новых примеров

Модель нейрона Хебба



Правило Хебба: вес w_{ij} нейрона изменяется пропорционально произведению его входного и выходного сигналов:

$$\Delta w_{ij} = \eta y_j y_i ,$$

где η – коэффициент обучения, значение которого меняется (0,1)

При обучении с учителе $\Delta w_{ij} = \eta y_j d_i .$

Стабилизация процесса обучения: $w_{ij}(t+1) = w_{ij}(t)(1-\gamma) + \Delta w_{ij}$

Классификация нейронных сетей

Однонаправленные

Рекуррентные
(с обратной связью)

простой перцептрон

Способ объединения нейронов

Нейронная сеть

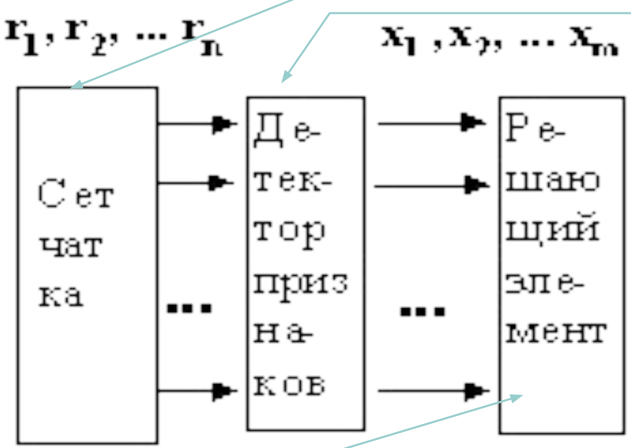
Количество слоёв нейронов

сеть Хопфилда

Однослойные

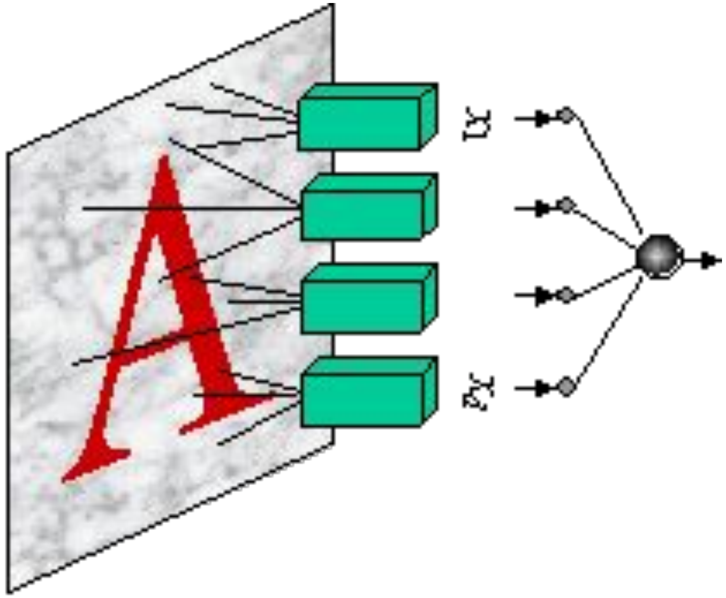
Многослойные

Простой персептрон



- матрица бинарных входов (сенсорных нейронов или "сетчатка") r_1, r_2, ... r_n, куда подаются входные образы;
- набор нейроподобных элементов x_1 , x_2, ... x_m, с фиксированными связями к подмножествам сетчатки ("детекторы признаков");
- "решающий элемент"- бинарный НПЭ с модифицируемыми связями с "детекторами". Обычно число решающих элементов выбирается равным количеству классов, на которое необходимо разбить предъявляемые персептрону образы.

Персептрон Розенблатта



Простой персептрон, для которого справедливы условия:

$$\mathbf{n=m \text{ и } x_i = r_i,}$$

при этом детекторы признаков могут рассматриваться как входной слой.

Персептрон Розенблатта имел один слой обучаемых весов, на входы которого подавались сигналы с d = 512 ассоциирующих нейронов со случайными фиксированными весами, образующие признаковое пространство для 400-пиксельных образов

?обучение?

Алгоритм обучения персептрона Розенблатта

процедура сходимости персептрона Розенблатта

1. Вектор весов w_i устанавливается в произвольное состояние.
2. На сетчатку поочередно подают образы из обучающей выборки, которые трансформируются в выходной сигнал у решающего элемента.
3. При правильном отклике ничего не изменяется.
4. При неправильном отклике y=0 веса всех связей от активных элементов сетчатки увеличивают, а при неправильном отклике y=1 – уменьшают на величину.

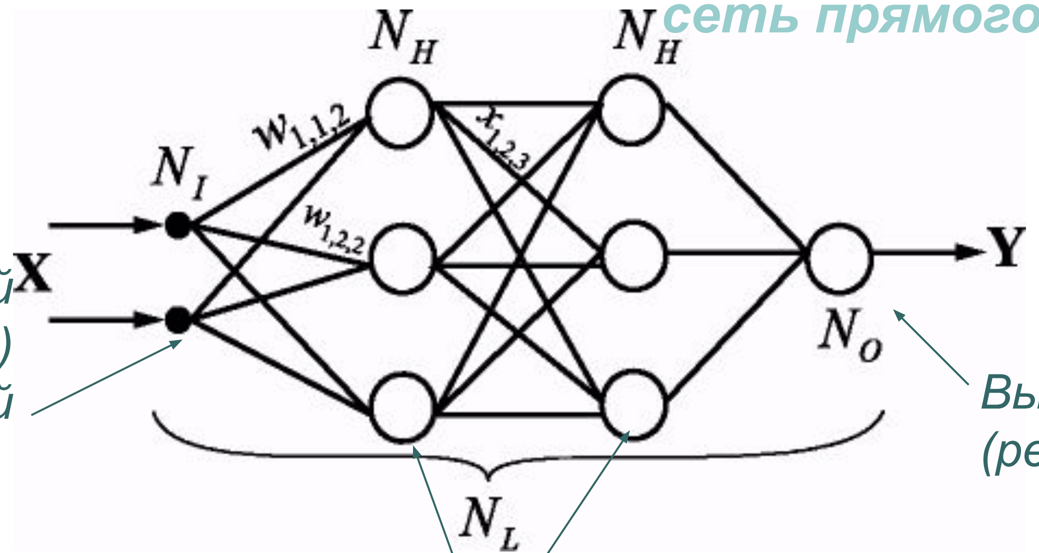
Если решение существует, оно будет достигнуто за конечное число шагов при начальном выборе связей.

Характеристики персептрона

- Тип входных сигналов: бинарные или аналоговые (действительные).
- Размерности входа и выхода ограничены при программной реализации только возможностями вычислительной системы, на которой моделируется нейронная сеть, при аппаратной реализации - технологическими возможностями.
- Емкость сети совпадает с числом нейронов.
- Модификации. Многослойные персептроны дают возможность строить более сложные разделяющие поверхности и поэтому имеют более широкое применение при решении задач распознавания.
- Достоинства. Программные или аппаратные реализации модели очень просты. Простой и быстрый алгоритм обучения.
- Недостатки. Примитивные разделяющие поверхности (гиперплоскости) дают возможность решать лишь самые простые задачи распознавания.
- Области применения. Распознавание образов, классификация.

Многослойный персептрон

сеть прямого распространения



Сенсорный (входной) слой

Выходной (результативный) слой

Скрытые (ассоциативные) слои

- Принцип связи между нейронами - "каждый с каждым".
- Количество нейронов в слоях может быть произвольным.
- Обычно во всех скрытых слоях одинаковое количество нейронов.
- Входной слой только распределяет сигналы.

Классификация

- Сеть с одним скрытым слоем, содержащим H нейронов со *ступенчатой* функцией активации, способна осуществить произвольную классификацию Hd точек d-мерного пространства (т.е. классифицировать Hd примеров).

- Одного скрытого слоя нейронов с *сигмоидной* функцией активации достаточно для аппроксимации любой границы между классами со сколь угодно высокой точностью.

Регрессия (аппроксимация)

- Одного скрытого слоя нейронов с *сигмоидной* функцией активации достаточно для аппроксимации любой функции со сколь угодно высокой точностью. (Более того, такая сеть может одновременно аппроксимировать и саму функцию и ее производные.)

Точность аппроксимации возрастает с числом нейронов скрытого слоя.

При H нейронах ошибка оценивается как $O(1/H)$.

Алгоритм решения задач с помощью МСП

1. Определить, какой смысл вкладывается в компоненты входного вектора x. Входной вектор должен содержать формализованное условие задачи, т.е. всю информацию, необходимую для получения ответа.
2. Выбрать выходной вектор y таким образом, чтобы его компоненты содержали полный ответ поставленной задачи.
3. Выбрать вид нелинейности в нейронах (функцию активации).
4. Задать диапазон изменения входов, выходов, весов и пороговых уровней, учитывая множество значений выбранной функции активации.
5. Присвоить начальные значения весовым коэффициентам и пороговым уровням и дополнительным параметрам (например, крутизне функции активации, если она будет настраиваться при обучении).
6. Провести обучение, т.е. подобрать параметры сети так, чтобы задача решалась наилучшим образом. По окончании обучения сеть готова решить задачи того типа, которым она обучена.
7. Подать на вход сети условия задачи в виде вектора x. Рассчитать выходной вектор y, который и даст формализованное решение задачи.

Алгоритм обратного распространения ошибки

Error backpropagation

Основа метода – целевая функция, формулируемая в виде квадратичной суммы разностей между фактическими и ожидаемыми значениями выходных сигналов.

В случае единичной одинарной выборки ***(x,d)*** целевая функция определяется в виде:

$$E(w) = \frac{1}{2}\sum_{k=1}^{M}(y_k - d_k)^2$$

При большом количестве обучающих выборок j (j = 1,2,.. p) целевая функция превращается в сумму по всем выборкам:

$$E(w) = \frac{1}{2}\sum_{j=1}^{p}\sum_{k=1}^{M}(y_k^{(j)} - d_k^{(j)})^2$$

одна & все

Этапы выполнения алгоритма обратного распространения ошибки

1. **Анализ нейронной сети в прямом направлении передачи информации при генерации входных сигналов, составляющих очередной вектор X.**
2. **Создание сети обратного распространения ошибок**
3. **Уточнение весов**
4. **Описанный в п. 1, 2 и 3 процесс следует повторить для всех обучающих выборок.**

К 1. рассчитываются значения выходных сигналов нейронов скрытых слоев и выходного слоя, а также соответствующие производные функций активации каждого слоя.

$$\frac{df(u_i^{(1)})}{du_i^{(1)}}, \frac{df(u_i^{(2)})}{du_i^{(2)}}, \ldots, \frac{df(u_i^{(m)})}{du_i^{(m)}}$$

К 2. путем изменения направлений передачи сигналов, замена функций активации их производными и подача на бывший выход возбуждения в виде разности между фактическим и ожидаемым значением. Для определенной таким образом сети необходимо рассчитать значения требуемых обратных разностей.

К 3. по формулам на основе результатов, полученных в п. 1 и 2, для оригинальной сети и для сети обратного распространения ошибки

К 4. Действие алгоритма завершается в момент, когда норма градиента упадет ниже априори заданного значения точности обучения *e*.

Сравнение градиентных методов обучения

Алгоритм	Описание	Время, (с)	Количество циклов	Количество операций, $x10^6$
Наискорейшего спуска	Использование линейного приближения целевой функции	57,1	980	2,50
Сопряжённых градиентов	Расчёт коэффициента сопряжения на каждом шаге	19,16	89	0,75
Переменной метрики BFGS	Направленная минимизация целевой функции	10,86	44	1,02
Левенберга-Марквардта	Замена Г. аппроксимированным значением (параметром Л.-М.)	1,87	6	0,46
RPROP	При уточнении весов учитывается только знак градиентной составляющей	12,96	185	0,56

Переобучение нейросети

Функция-учитель, порождающая обучающие примеры, N<∞

Нейросеть с нулевой ошибкой обучения

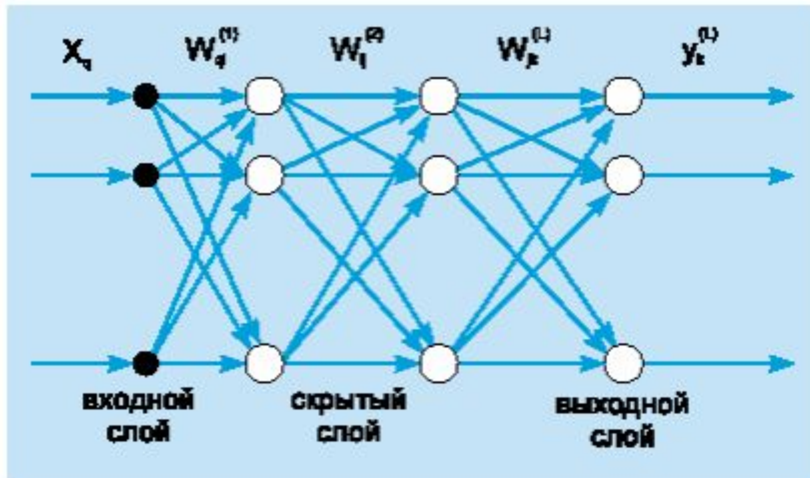
Проблема: недостаточно информации, чтобы выбрать единственное правильное решение : функцию-учителя.

⇓

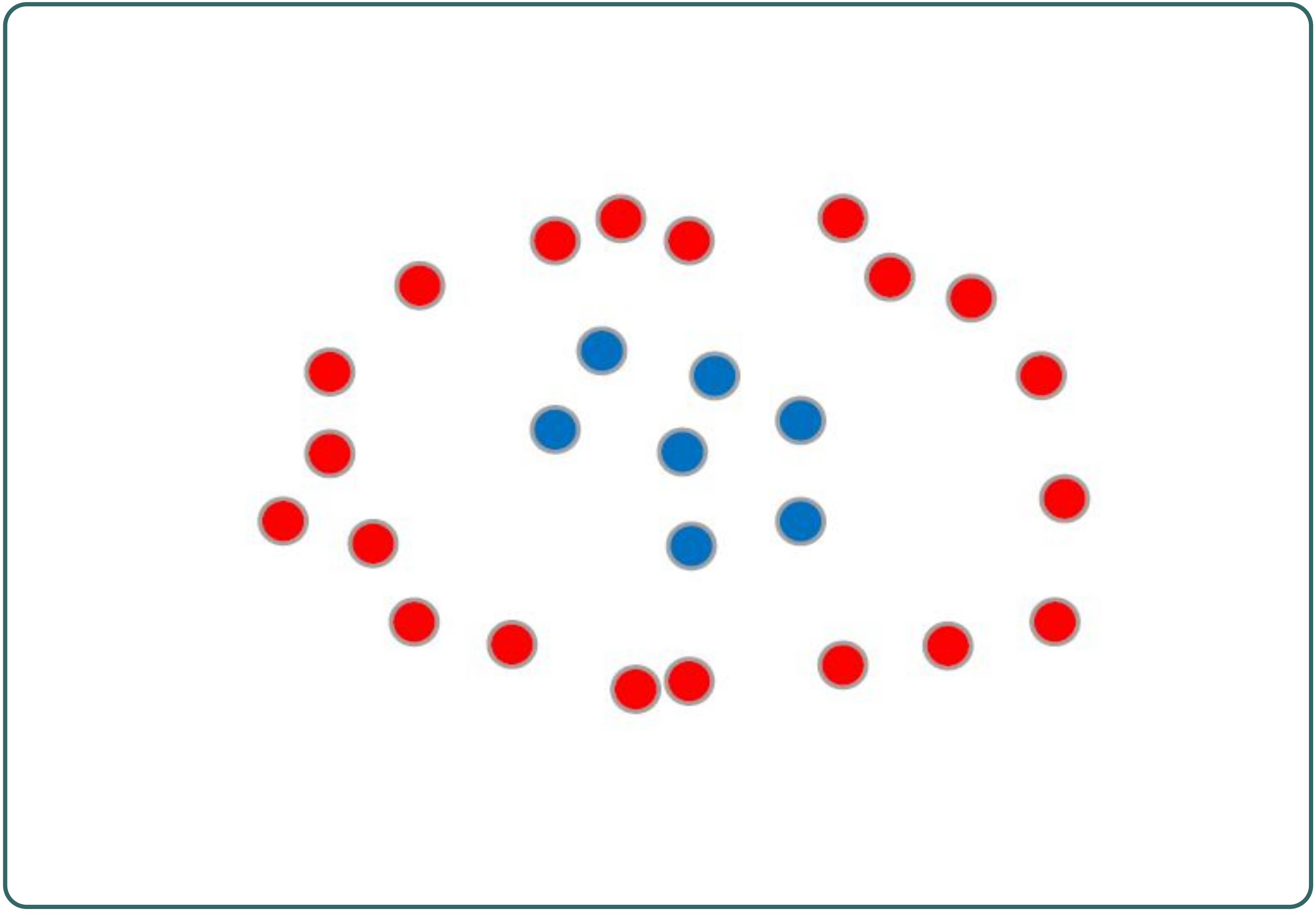
выбранная случайным образом функция дает плохие предсказания на новых примерах, отсутствовавших в обучающей выборке, хотя последнюю сеть воспроизвела без ошибок.

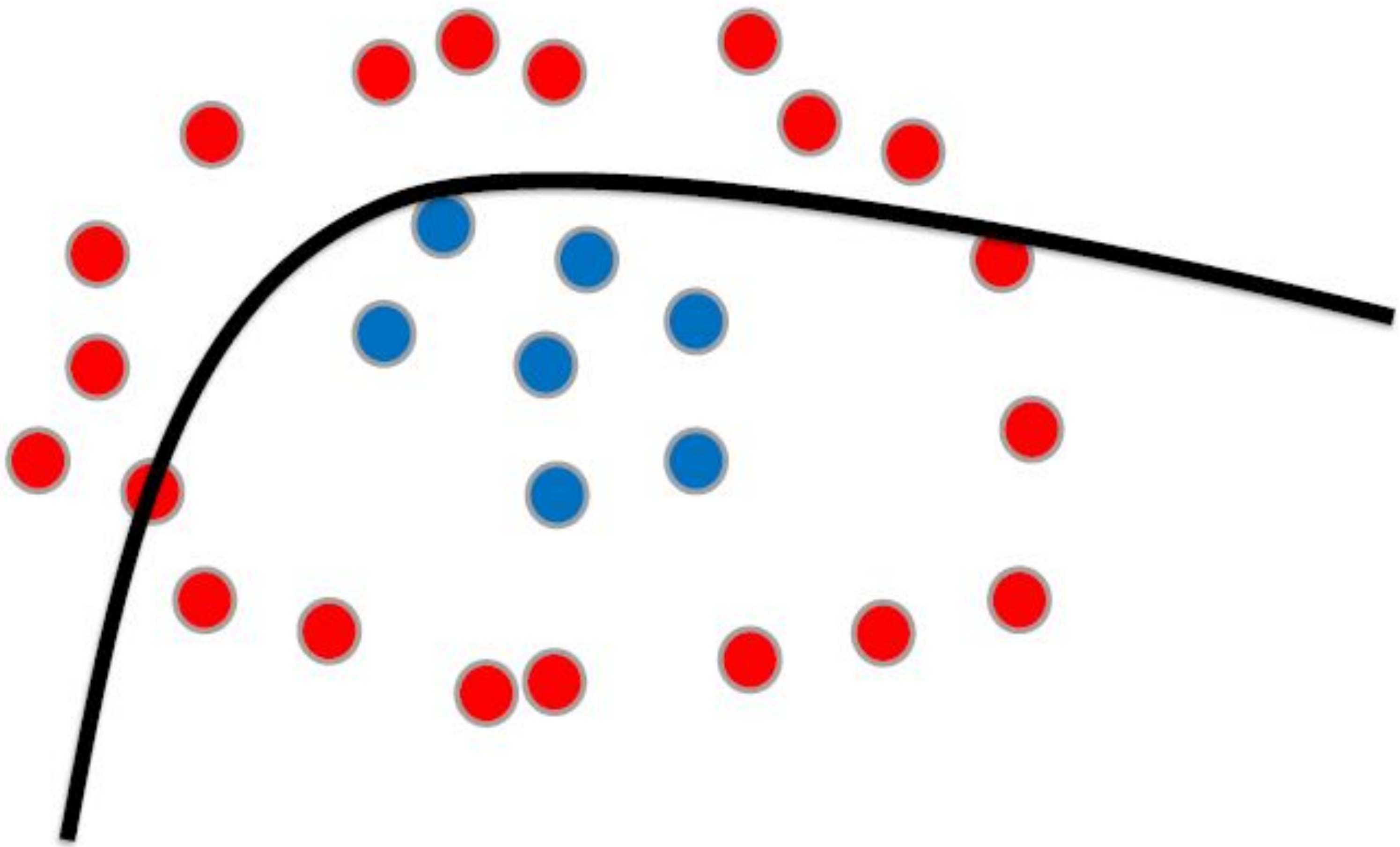
Вместо того, чтобы обобщить известные примеры, сеть запомнила их

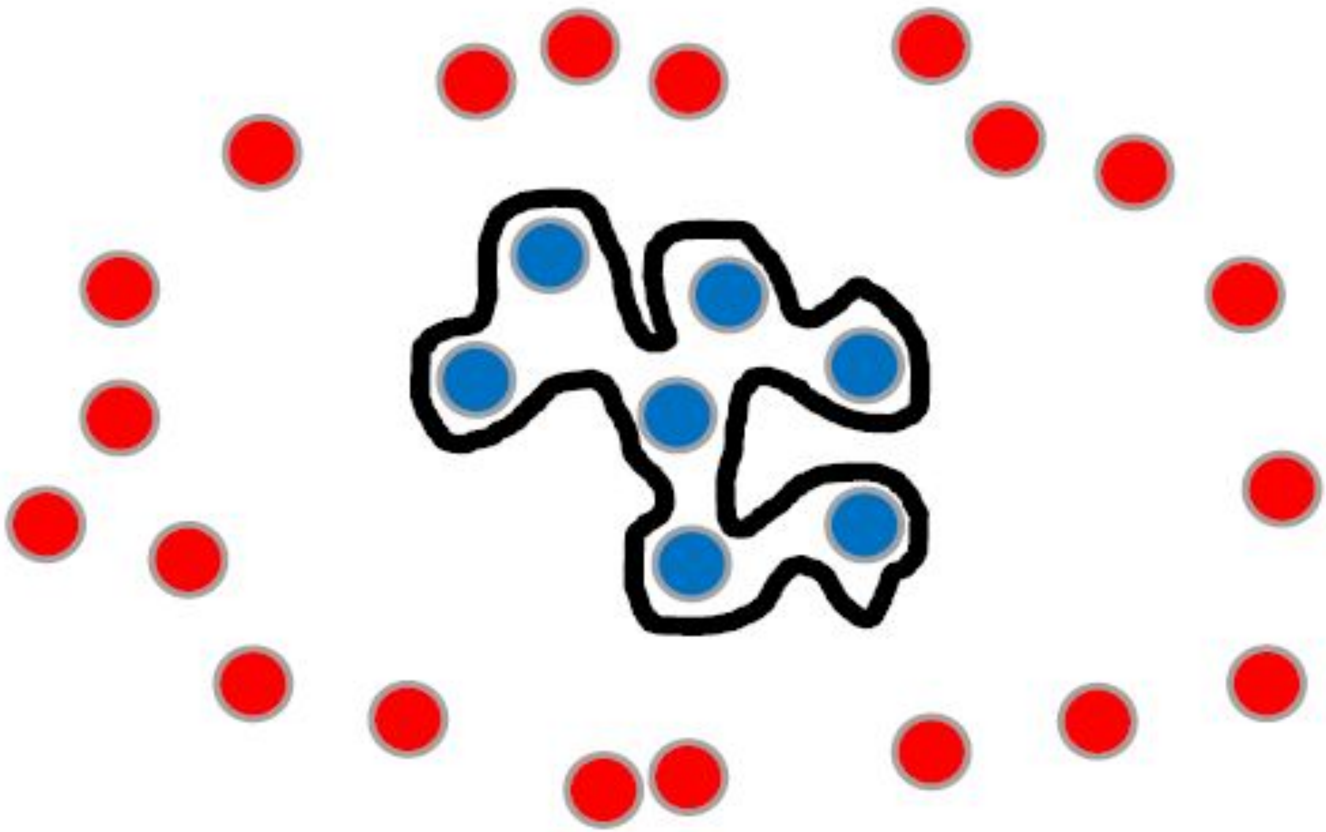
Многослойный персептрон

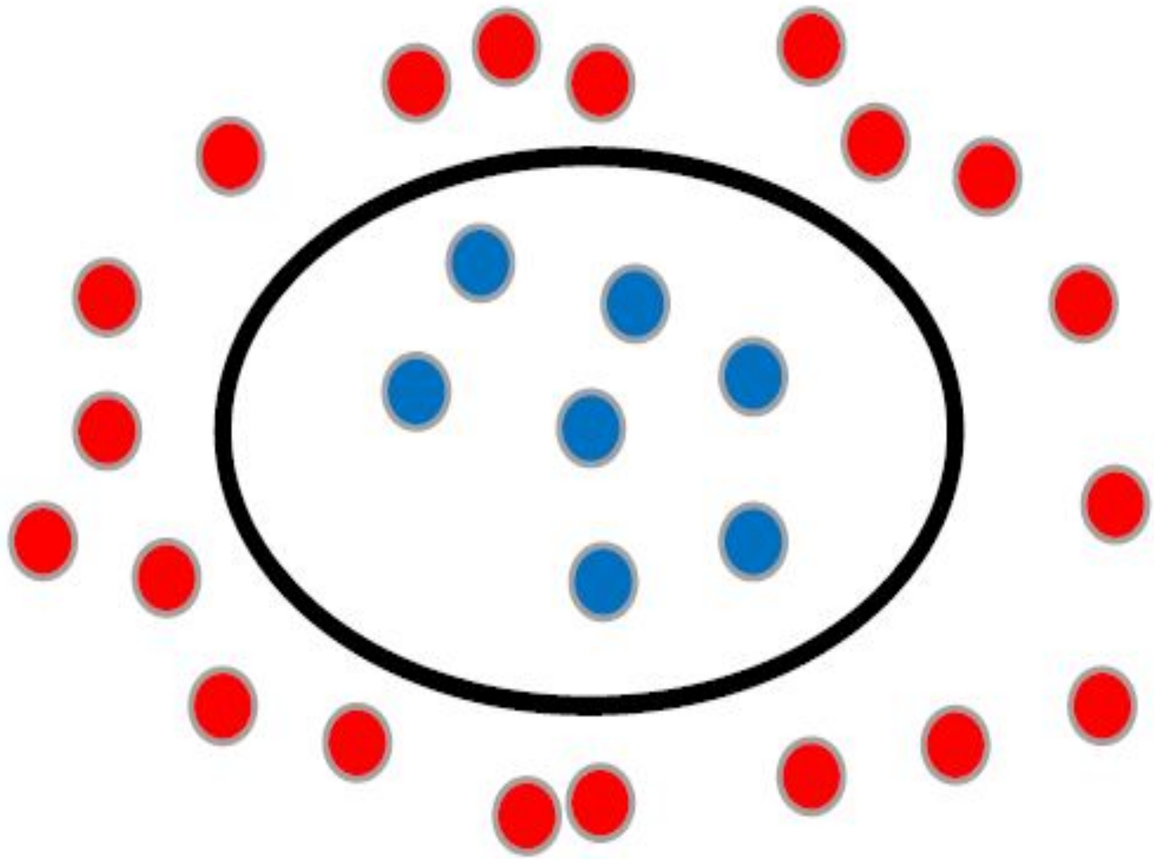


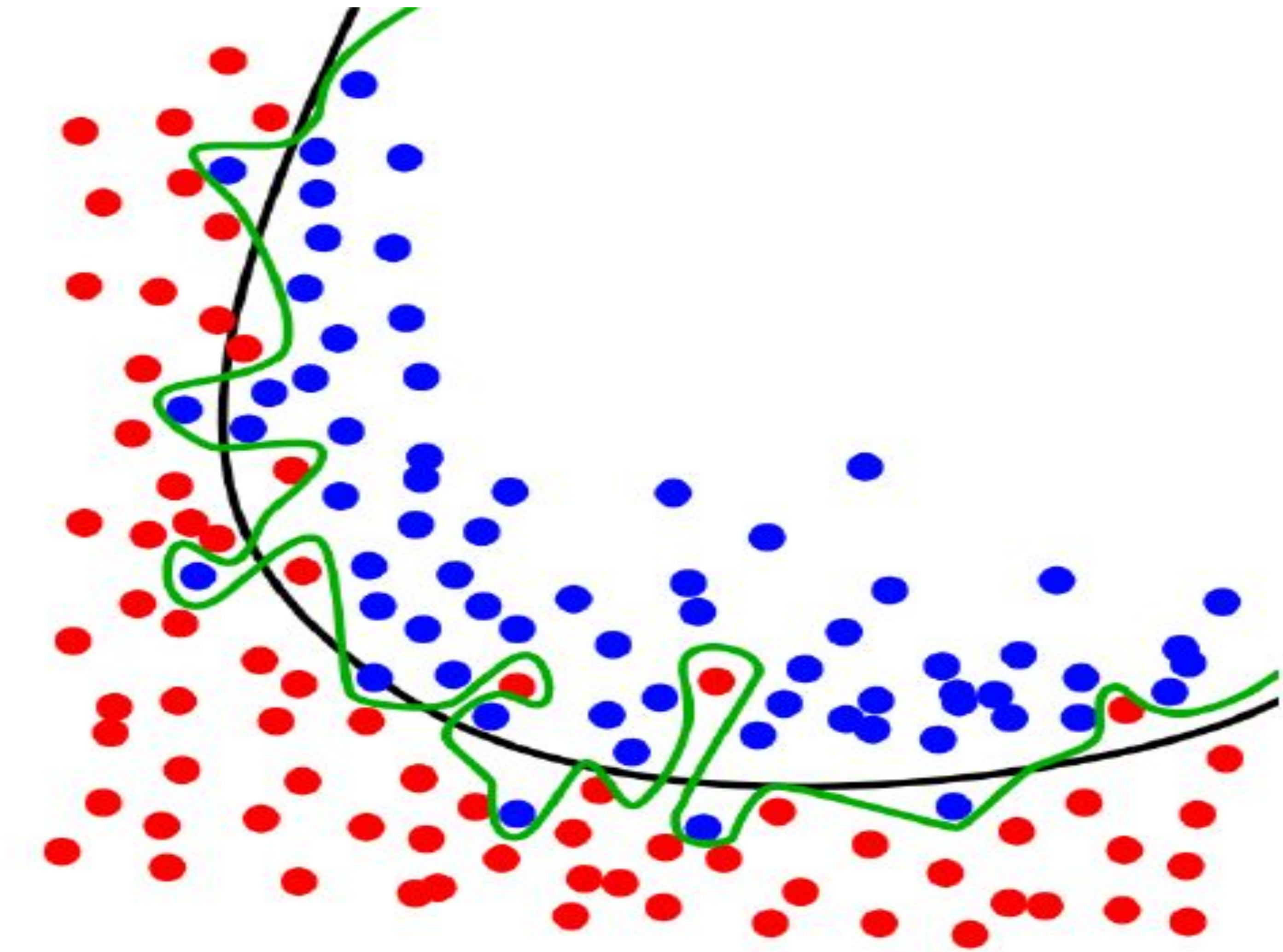
- *Классификация* (дискретный набор выходных значений)
- *Регрессия* (непрерывные выходные значения)







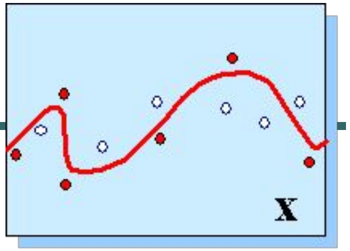




Борьба с переобучением

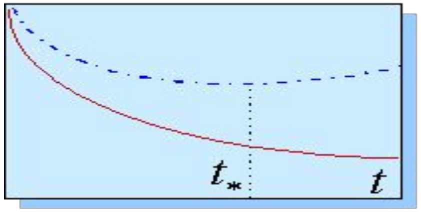
- Подходы:
  - ранняя остановка обучения;
  - прореживание связей (метод от большого - к малому);
  - поэтапное наращивание сети (от малого - к большому).

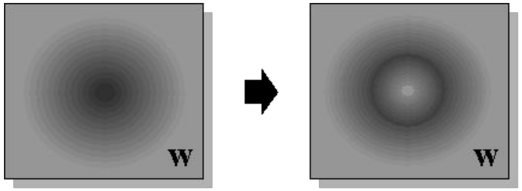
обучение "по частям"



Разделение данных на обучающее и валидационное множества примеров

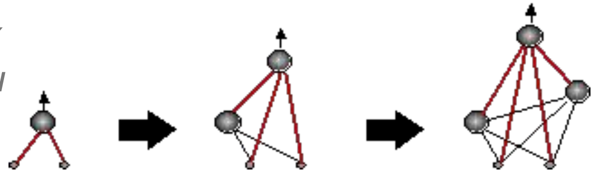
в момент минимума ошибки валидации. При этом обычно ошибка обучения продолжает понижаться



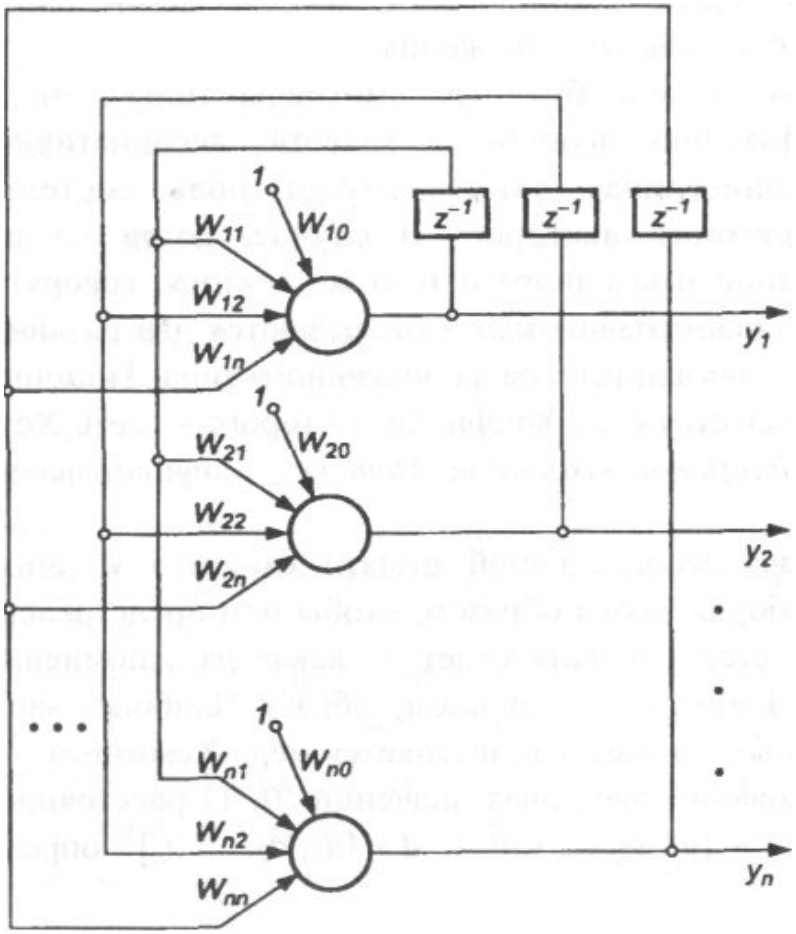


сократить разнообразие возможных конфигураций обученных нейросетей при минимальной потере их аппроксимирующих способностей

добавление промежуточных нейронов с фиксированными весами



Сеть Хопфилда



- выходные сигналы нейронов являются одновременно входными сигналами сети, при этом возбуждающий вектор особо не выделяется.
- отсутствует связь нейрона с собственным выходом

Выходной сигнал i-го нейрона:

$$y_i = \operatorname{sgn}\left(\sum_{j=0}^{N} w_{ij} x_j + b_i\right),$$

где b_i- пороговое значение, заданное внешним источником,
N – количество нейронов.

Решение задач с помощью сетей Хопфилда

1. Построить функцию энергии таким образом, чтобы точка глобального минимума этой функции совпадала с решением задачи. При этом градиент функции энергии должен допускать вычисление с помощью НС.
2. Записать формулы для расчета параметров сети (весовых коэффициентов и пороговых уровней) для расчета градиента функции энергии.
3. Разорвать цепочку обратной связи и предъявить сети входной вектор. Рассчитать значения выходов.
4. Замкнуть обратную связь и предоставить сети возможность самостоятельно менять свое состояние (релаксация). Остановить процесс релаксации после того, как выходной вектор перестанет меняться, т.е. по достижении минимума функции энергии. Полученные выходы сети дают решение задачи.

Режим обучения сети Хопфилда

Фаза обучения ориентирована на формирование таких значений весов, при которых в режиме функционирования задание начального состояния нейронов, близкого к одному из обучающих векторов *x, при* соблюдении зависимости

$$y_i(k) = \mathrm{sgn}\left(\sum_{j=1, i \neq j}^{N} w_{ij} y_j (k-1) \right)$$

приводит к стабильному состоянию, в котором реакция нейронов *y*= *x* остается неизменной в любой момент времени.

Псевдоинверсия (метод проекций)

Пусть при правильно подобранных весах каждая поданная на вход выборка *x* генерирует на выходе саму себя, мгновенно приводя к состоянию:

Тогда решение в результате всех преобразований примет вид:

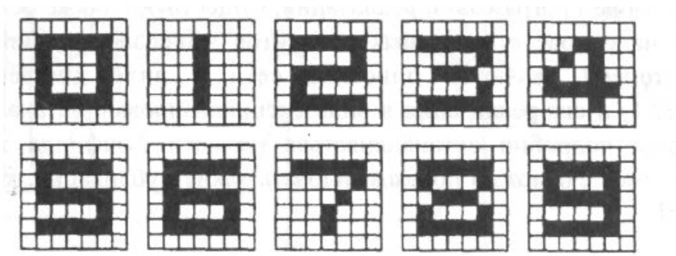
$$\mathbf{W}^{(i)} = \mathbf{W}^{(i-1)} + \frac{1}{[\boldsymbol{x}^{(i)}]^T \boldsymbol{x}^{(i)} - [\boldsymbol{x}^{(i)}]^T \mathbf{W}^{(i-1)} \boldsymbol{x}^{(i)}} \times [\mathbf{W}^{(i-1)} \boldsymbol{x}^{(i)} - \boldsymbol{x}^{(i)}] \times$$

$$\times [\mathbf{W}^{(i-1)} \boldsymbol{x}^{(i)} - \boldsymbol{x}^{(i)}]^T.$$ при $\mathbf{W}^{(0)} = 0.$

агает однократное предъявление всех *p* обучающих выборок, в результате чего матрица весов сети принимает фиксированное значение.

Режим распознавания сети Хопфилда

Обучение

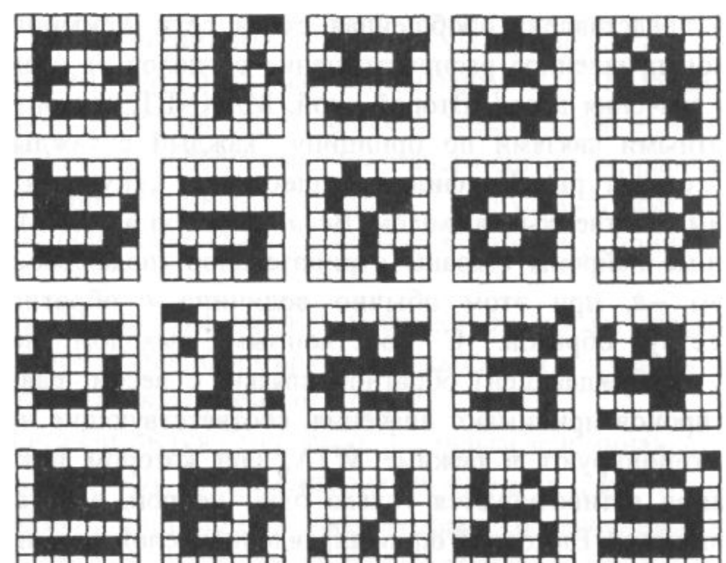


Образцы - 10 цифр, представленных в пиксельной форме размерностью 7x7.
⇒ количество нейронов сети Хопфилда составляет 49, а количество обучающих выборок - 10.

Обучение по Хеббу: безошибочно распознан **один** образ.

Обучение по методу проекций: почти безошибочно распознан **каждый** из запомненных образов.

Тестирование



Свойства современных нейросетей

- Обучаемость. Выбрав одну из моделей НС, создав сеть и выполнив алгоритм обучения, мы можем обучить сеть решению задачи, которая ей по силам. Нет никаких гарантий, что это удастся сделать при выбранных сети, алгоритме и задаче, но если все сделано правильно, то обучение бывает успешным.
- Способность к обобщению. После обучения сеть становится нечувствительной к малым изменениям входных сигналов (шуму или вариациям входных образов) и дает правильный результат на выходе.
- Способность к абстрагированию. Если предъявить сети несколько искаженных вариантов входного образа, то сеть сама может создать на выходе идеальный образ, с которым она никогда не встречалась.

Различие экспертных и НС систем по характеру знаний

	Экспертные системы (ЭС)	Нейросетевые системы (НС)
Источник знаний	Формализованный опыт эксперта, выраженный в виде логических утверждений - правил и фактов, безусловно принимаемых системой	Совокупный опыт эксперта-учителя, отбирающего примеры для обучения + индивидуальный опыт обучающейся на этих примерах нейронной сети
Характер знаний	Формально-логическое “левополушарное” знание в виде правил	Ассоциативное “правополушарное” знание в виде связей между нейронами сети
Развитие знаний	В форме расширения совокупности правил и фактов (базы знаний)	В форме дообучения на дополнительной последовательности примеров, с уточнением границ категорий и формированием новых категорий
Роль эксперта	Задает на основе правил полный объем знаний экспертной системы	Отбирает характерные примеры, не формулируя специально обоснование своего выбора
Роль искус. сист.	Поиск цепочки фактов и правил для доказательства суждения	Формирование индивидуального опыта в форме категорий, получаемых на основе примеров и категоризация образов