

# **Раздел 1: Интеллектуальные системы управления**

## **Тема 1.1: Интеллектуальные системы управления – новые технологии и реализации АСУ ТП**

### **Лекция 1:**

- 1. Цель и задачи курса.**
- 2. Эволюция средств проектирования и реализации СУ.**
- 3. Традиционная и интеллектуальная АСУ ТП.**

#### **1. Цель и задачи курса**

В настоящее время существует ряд задач, которые не могут быть решены традиционными методами теории автоматического управления. Это характерно для больших и сложных объектов и систем, алгоритмы работы которых не могут быть формализованы или работают в неопределенных ситуациях. Как правило, управляют такими объектами и системами люди (человек-оператор), эксперты по принятию решений в данной области. Такой тип систем управления называют системами управления на основе знаний или *интеллектуальными системами управления*. Наша задача заключается в изучении такого типа систем, методов проектирования, а также алгоритмов их функционирования.

#### **2. Эволюция средств проектирования и реализации СУ**

#### **3. Традиционная и интеллектуальная АСУ ТП**

На данный момент времени интеллектуальные системы автоматического управления (ИСАУ) можно разделить на три группы: а) ЭС, для решения задач управления в РМВ; б) интеллектуальные системы на основе теории интеллектуальных машин Дж.Саридиса; в) системы экспертного управления с адаптивными и неадаптивными регуляторами.

*Первая группа.* Системы экспертного управления в РМВ можно рассматривать как распространение методов традиционных ЭС на решение задач управления динамическими объектами. Стратегия управления строится в рамках нечеткой или основанной на продукционных правилах методологии. Такое управление базируется на применении логических правил выбора управляющего воздействия в различных ситуациях. Эти правила заменяют традиционные алгоритмы управления. Областью применения этого подхода являются задачи управления сложными процессами, для которых либо не существует формальных моделей, либо эти модели неадекватны процессу ("грубые" модели). Решение подобных задач ориентировано на представление эмпирических знаний о том, как должен управляться конкретный объект.

*Вторая группа.* Теория интеллектуального управления исследует наиболее общие прикладные задачи управления, такие как интеллектуальные роботы, автономные устройства, и формулирует их в рамках единого математического подхода. Понятие интеллектуального управления было введено в 1971 году, как области, лежащей на пересечении теории искусственного интеллекта и автоматического управления. С тех пор ведутся безуспешные попытки довести строгость формализации этой области до уровня специальной научной дисциплины.

Несколько иной подход был предложен в работах / /, где расширен базовый набор научных дисциплин, формирующих интеллектуальное управление. Теперь оно рассматривается как область, лежащая на пересечении теорий искусственного интеллекта, исследования операций и автоматического управления. В соответствии с этим в указанных работах представлены основные понятия и методы данного направления. Так, интеллектуальное управление представляет собой процесс, который ведет некоторую интеллектуальную машину к достижению ею же заданной цели автономно. При этом под интеллектуальными подразумеваются машины, способные заменить человека при выполнении им опасной, утомительной или высокоточной работы, где их высокая эффективность оказывается более выгодной в смысле человеческого или гуманистического измерения.

Системы интеллектуального управления используются для достижения автономными объектами заданных им целей без какого-либо взаимодействия с человеком-оператором. Для этого в системы должны быть заложены

организующие, планирующие и исполнительные функции. При этом структурная организация систем интеллектуального управления строится в соответствии с принципом – при высокой степени интеллектуализации не требуется высокой точности, и наоборот (аналог принципа неопределенности Гейзенберга). В соответствии с этим принципом Дж. Саридисом предложена система иерархического оптимального управления, состоящая из трех основных уровней интеллектуальности или точности:

а) уровня организации с функциями рассуждения, планирования и принятия решений по организации процесса решения задачи (уровень искусственного интеллекта);

б) уровня согласования взаимодействий между верхним и нижним уровнями интеллектуализации, с функциями, направленными на согласование работы нижнего уровня (уровень искусственного интеллекта и исследования операций);

в) уровня исполнения, являющегося нижним уровнем, с высокими требованиями к точности и функциям, которые базируются на методах теории автоматического управления.

Задачи, решаемые на уровне исполнения, включают в себя выбор подходящего регулятора, удовлетворяющего определенным условиям, задаваемым разработчиком. В автономной системе, когда разработчиком является сама интеллектуальная машина, проблему управления можно рассматривать как выбор наилучшего регулятора в смысле удовлетворения условиям поставленной задачи. Выбор проводится среди всех регуляторов, осуществляющих допустимое управление.

*Третья группа.* Эта группа носит название экспертного управления. Оно связано с применением методов ЭС для контроля традиционных ПИД и адаптивных регуляторов. В системах такого типа знания, необходимые для эффективной настройки параметров регуляторов, приобретаются как у специалистов по системам управления, так и у операторов и затем помещаются в БЗ в виде правил настройки. В процессе функционирования системы характер поведения объекта фиксируется классификатором образов. Подсистема вывода использует правила настройки и классифицированные образы для настройки параметров регуляторов. Весь процесс настройки выполняется в РМВ.

Экспертное управление применяется для расширения класса традиционных алгоритмов управления благодаря введению в систему общих знаний по управлению, эвристических методов настройки и адаптации. Такие системы состоят из интеллектуальной комбинации различных алгоритмов управления, идентификации и контроля.

По своему духу экспертное управление ближе к традиционному адаптивному, чем методы управления, развиваемые в рамках первого из названных направлений, в том числе к методам нечеткого управления. Развитие данного подхода обусловлено некоторыми недостатками современных адаптивных регуляторов. В первую очередь это относится к требованию наличия априорных знаний об объекте управления. Рекуррентные алгоритмы идентификации и алгоритмы управления, используемые в традиционных адаптивных регуляторах, могут квалифицироваться как заключительные алгоритмические представления большого количества знаний по управлению – как теоретических, так и практических.

Экспертное управление включает в себя две большие проблемные области. Первая из них рассматривает то, какими знаниями об объекте необходимо располагать для того, чтобы автоматически настроить регулятор и осуществить наблюдение за объектом. Сюда также входит вопрос о том, как эти знания должны быть получены. Вторая область – это представление и использование этих знаний.

Идеально приспособленной для решения задач экспертного управления следовало бы считать ту систему, которая:

- может надлежащим образом управлять произвольным объектом (нестационарным, нелинейным, подверженным возмущениям);
- требует минимального количества априорных знаний об объекте управления;
- способна использовать априорные знания;
- использует качественные термины для характеристики контура управления (например, "как можно быстрее", "малое отклонение" и т.д.);
- увеличивает количество собственных знаний об объекте и соответственно улучшает управление;

- диагностирует ход процесса управления и состояние компонент контура управления;
- может функционировать в диалоге с пользователем и получать от него дополнительную информацию об объекте управления, его характеристиках и т.п.;
- позволяет модифицировать и расширять знания и эвристические методы управления, лежащие в основе ее функционирования.

Образно говоря, метод достижения идеальной цели экспертного управления можно выразить как попытку включить опытного специалиста по управлению в контур управления, предоставив ему возможность использовать любые алгоритмы управления, идентификации, измерения, контроля и создания системы управления. Структурная схема такой системы управления показана на рис.1.1.



Рис.1.1 Структурная схема ИСАУ

## **Тема 1.2: Основные понятия и определения теории интеллектуальных систем управления. Модели и алгоритмы принятия решений при управлении сложными объектами.**

### **Лекция 2:**

#### **Модели представления знаний в СУ.**

Основу интеллектуальных систем управления составляют базы знаний (БЗ).

Вы уже изучали экспертные системы (ЭС) и на первый взгляд экспертные системы и системы управления на основе БЗ весьма похожи. Однако имеется ряд различий между этими системами в методологии их создания и принципах функционирования. В ЭС реализуется принцип консультации с человеком-экспертом. ЭС работают вне реального масштаба времени (РМВ), а человек включен в контур принятия решения.

Системы управления на основе знаний служат для контроля и управления объектами в РМВ. Система работает с объектом, а не с человеком-оператором. Она использует предварительную обработку информации об объекте и при появлении какой-либо неисправности диагностирует объект и принимает решение на коррекцию аномалии. Оператор САУ выключен из контура принятия решения и выполняет лишь общие функции контроля.

На данный момент времени интеллектуальные системы автоматического управления (ИСАУ) можно разделить на три группы: а) ЭС, для решения задач управления в РМВ; б) интеллектуальные системы на основе теории интеллектуальных машин Дж.Саридиса; в) системы экспертного управления с адаптивными и неадаптивными регуляторами.

*Первая группа.* Системы экспертного управления в РМВ можно рассматривать как распространение методов традиционных ЭС на решение задач управления динамическими объектами. Стратегия управления строится в рамках нечеткой или основанной на продукционных правилах методологии. Такое управление базируется на применении логических правил выбора управляющего воздействия в различных ситуациях. Эти правила заменяют

традиционные алгоритмы управления. Областью применения этого подхода являются задачи управления сложными процессами, для которых либо не существует формальных моделей, либо эти модели неадекватны процессу ("грубые" модели). Решение подобных задач ориентировано на представление эмпирических знаний о том, как должен управляться конкретный объект.

*Вторая группа.* Теория интеллектуального управления исследует наиболее общие прикладные задачи управления, такие как интеллектуальные роботы, автономные устройства, и формулирует их в рамках единого математического подхода. Понятие интеллектуального управления было введено в 1971 году, как области, лежащей на пересечении теории искусственного интеллекта и автоматического управления. С тех пор ведутся безуспешные попытки довести строгость формализации этой области до уровня специальной научной дисциплины.

Несколько иной подход был предложен в работах / /, где расширен базовый набор научных дисциплин, формирующих интеллектуальное управление. Теперь оно рассматривается как область, лежащая на пересечении теорий искусственного интеллекта, исследования операций и автоматического управления. В соответствии с этим в указанных работах представлены основные понятия и методы данного направления. Так, интеллектуальное управление представляет собой процесс, который ведет некоторую интеллектуальную машину к достижению ею же заданной цели автономно. При этом под интеллектуальными подразумеваются машины, способные заменить человека при выполнении им опасной, утомительной или высокоточной работы, где их высокая эффективность оказывается более выгодной в смысле человеческого или гуманистического измерения.

Системы интеллектуального управления используются для достижения автономными объектами заданных им целей без какого-либо взаимодействия с человеком-оператором. Для этого в системы должны быть заложены организующие, планирующие и исполнительные функции. При этом структурная организация систем интеллектуального управления строится в соответствии с принципом – при высокой степени интеллектуализации не требуется высокой точности, и наоборот (аналог принципа неопределенности Гейзенберга). В соответствии с этим принципом Дж. Саридисом предложена

система иерархического оптимального управления, состоящая из трех основных уровней интеллектуальности или точности:

а) уровня организации с функциями рассуждения, планирования и принятия решений по организации процесса решения задачи (уровень искусственного интеллекта);

б) уровня согласования взаимодействий между верхним и нижним уровнями интеллектуализации, с функциями, направленными на согласование работы нижнего уровня (уровень искусственного интеллекта и исследования операций);

в) уровня исполнения, являющегося нижним уровнем, с высокими требованиями к точности и функциям, которые базируются на методах теории автоматического управления.

Задачи, решаемые на уровне исполнения, включают в себя выбор подходящего регулятора, удовлетворяющего определенным условиям, задаваемым разработчиком. В автономной системе, когда разработчиком является сама интеллектуальная машина, проблему управления можно рассматривать как выбор наилучшего регулятора в смысле удовлетворения условиям поставленной задачи. Выбор проводится среди всех регуляторов, осуществляющих допустимое управление.

*Третья группа.* Эта группа носит название экспертного управления. Оно связано с применением методов ЭС для контроля традиционных ПИД и адаптивных регуляторов. В системах такого типа знания, необходимые для эффективной настройки параметров регуляторов, приобретаются как у специалистов по системам управления, так и у операторов и затем помещаются в БЗ в виде правил настройки. В процессе функционирования системы характер поведения объекта фиксируется классификатором образов. Подсистема вывода использует правила настройки и классифицированные образы для настройки параметров регуляторов. Весь процесс настройки выполняется в РМВ.

Экспертное управление применяется для расширения класса традиционных алгоритмов управления благодаря введению в систему общих знаний по управлению, эвристических методов настройки и адаптации. Такие системы состоят из интеллектуальной комбинации различных алгоритмов управления, идентификации и контроля.

По своему духу экспертное управление ближе к традиционному адаптивному, чем методы управления, развиваемые в рамках первого из названных направлений, в том числе к методам нечеткого управления. Развитие данного подхода обусловлено некоторыми недостатками современных адаптивных регуляторов. В первую очередь это относится к требованию наличия априорных знаний об объекте управления. Рекуррентные алгоритмы идентификации и алгоритмы управления, используемые в традиционных адаптивных регуляторах, могут квалифицироваться как заключительные алгоритмические представления большого количества знаний по управлению – как теоретических, так и практических.

Экспертное управление включает в себя две большие проблемные области. Первая из них рассматривает то, какими знаниями об объекте необходимо располагать для того, чтобы автоматически настроить регулятор и осуществить наблюдение за объектом. Сюда также входит вопрос о том, как эти знания должны быть получены. Вторая область – это представление и использование этих знаний.

Идеально приспособленной для решения задач экспертного управления следовало бы считать ту систему, которая:

- может надлежащим образом управлять произвольным объектом (нестационарным, нелинейным, подверженным возмущениям);
- требует минимального количества априорных знаний об объекте управления;
- способна использовать априорные знания;
- использует качественные термины для характеристики контура управления (например, "как можно быстрее", "малое отклонение" и т.д.);
- увеличивает количество собственных знаний об объекте и соответственно улучшает управление;
- диагностирует ход процесса управления и состояние компонент контура управления;
- может функционировать в диалоге с пользователем и получать от него дополнительную информацию об объекте управления, его характеристиках и т.п.;

- позволяет модифицировать и расширять знания и эвристические методы управления, лежащие в основе ее функционирования.

Образно говоря, метод достижения идеальной цели экспертного управления можно выразить как попытку включить опытного специалиста по управлению в контур управления, предоставив ему возможность использовать любые алгоритмы управления, идентификации, измерения, контроля и создания системы управления.

### **Лекция 3:**

- 1. Понятие нечетких множеств.**
- 2. Основные определения, правила и действия над нечеткими множествами.**

#### **1. Понятие нечетких множеств**

Пусть  $X$  – некоторое конечное базовое множество или шкала. *Нечетким множеством*  $A$  в  $X$  называется упорядоченная совокупность пар, представленная в виде:

$$A = \{ \langle \mu_A(x), x \rangle \mid x \in X \}, \mu_A(x) \in [0, 1], \quad (3.1)$$

где  $\mu_A: X \rightarrow [0, 1]$  - функция принадлежности множества  $A$  множеству  $X$ ;  $x$  - числовое значение элемента множества  $X$ ;  $\mu_A(x)$  - степень принадлежности данного элемента  $x$  множеству  $A$ . Степень принадлежности представляется числами из интервала  $[0, 1]$ . Функции принадлежности строятся на основе экспертных оценок.

*Пример:* пусть  $N = \{0, 1, 2, 3, 4, 5\}$  совокупность чисел. Тогда множество

$$M \text{ на множестве } N \text{ можно записать как } M = \{ \langle 1; 0 \rangle, \langle 0.8; 1 \rangle, \langle 0.6; 2 \rangle, \langle 0.4; 3 \rangle, \langle 0.1; 4 \rangle, \langle 0; 5 \rangle \}.$$

#### **2. Основные определения, правила и действия над нечеткими множествами**

Над нечеткими множествами выполняются те же операции, что и над обычными, а также операции, введенные для использования нечетких множеств в математическом аппарате принятия решений.

*Объединением* нечетких множеств  $A$  и  $B$  из  $X$  называют нечеткое множество вида

$$A \cup B = \int_x (\mu_A(x) \vee \mu_B(x)) / x, \text{ где } \mu_A(x) \vee \mu_B(x) = \max(\mu_A(x), \mu_B(x)), x \in X.$$

*Пересечением* нечетких множеств A и B в X называют нечеткое множество вида

$$A \cap B = \int_x (\mu_A(x) \wedge \mu_B(x)) / x, \text{ где } \mu_A(x) \wedge \mu_B(x) = \min(\mu_A(x), \mu_B(x)), x \in X.$$

*Дополнением* или *отрицанием* нечеткого множества A называют нечеткое множество вида

$$\bar{A} = \int_x (1 - \mu_A(x)) / x, x \in X.$$

*Концентрирование* нечеткого множества A из X, осуществляющее ограничение числа элементов множества, определяют в виде

$$\text{CON}(A) = A^2 = \int_x (1 - \mu_A(x))^2 / x, x \in X.$$

*Растяжение* нечеткого множества A из X, осуществляющее увеличение числа элементов множества, определяют в виде

$$\text{DIL}(A) = \sqrt{A} = \int_x (1 - \mu_A(x))^{0.5} / x, x \in X.$$

*Симметричной разностью* называют нечеткое множество вида  $A \otimes B$ , имеющее функцию принадлежности

$$\mu_{A \otimes B}(x) = \mu_A(x) - \mu_B(x), x \in X.$$

*Множеством уровня  $\alpha$*  нечеткого множества A из X, называют множество, составленное из элементов x, принадлежащих X, степени принадлежности которых нечеткому множеству A не меньше числа  $\alpha$ . Таким образом, если  $A_\alpha$  - множество уровня  $\alpha$  нечеткого множества A, то

$$\forall \alpha \in [0, 1], A_\alpha = \{x \mid x \in X, \mu_A(x) \geq \alpha\}$$

Функцию принадлежности для произвольного нечеткого множества A можно определить в виде

$$\mu_A(x) = \sup \min(\alpha, \mu_{A_\alpha}(x)), \alpha \in [0, 1]$$

$$\text{где } \mu_{A_\alpha}(x) = 1, \text{ если } x \in A_\alpha,$$

$$\mu_{A_\alpha}(x) = 0, \text{ если } x \notin A_\alpha.$$

Нечеткое множество уровня нечеткого множества A, определяется как

$$\bar{A}_\alpha = \{x, \mu_A(x)\}, x \in A_\alpha.$$

*Включение множеств.* Пусть A и B произвольные множества из множества U.

Говорят, что A включает в себя B ( $B \subseteq A$ ), если

$$\forall x \in X, \mu_B(x) \leq \mu_A(x)$$

Когда последнее неравенство строгое, то говорят, что включение строгое.

Очевидно, что  $A=B$ , если  $A \subseteq B$  и  $B \subseteq A$ .

## **Лекция 4:**

- 1. Модели объектов управления в нечеткой математической среде.**
- 2. Идентификация статических моделей с нечеткими параметрами.**
- 3. Методы идентификации объектов с нечеткими параметрами.**
- 4. Методика построения функции принадлежности.**

### **1. Модели объектов управления в нечеткой математической среде**

Применение детерминированной математики для описания сложных объектов часто приводит к неоправданной идеализации технологического объекта. Поэтому в тех случаях, когда объект является слабо формализуемым, рекомендуется использовать методы построения нечетких моделей. Рассмотрим два таких метода. Первый основывается на построении статистических моделей объектов с нечеткими коэффициентами методом регрессивного анализа. Он может быть рекомендован для моделирования некоторых классов технологических объектов.

Преимуществом такого подхода является возможность его использования при моделировании систем, для которых сбор статической информации затруднен или полностью исключен. Полученная продукционная модель, является продуктом экспертного опроса операторов, технологов и другого обслуживающего персонала. Полученные модели достаточно эффективны при решении задач управления.

### **2. Идентификация статических моделей с нечеткими параметрами**

#### **Постановка задачи.**

Пусть  $X = \{x_1, x_2, \dots, x_n\}$  - входные переменные;  $Y$  - выходная область управления. Зададим для  $x_i$  ( $i = 1, 2, \dots, n$ ) детерминированные значения, для  $Y$  -

нечеткие. Математическую зависимость между указанными параметрами опишем нечетким уравнением регрессии, параметры которого неизвестны

$$\tilde{y} = \tilde{f}(x_1, \dots, x_n) \quad (4.1)$$

$\sim$  - оператор нечеткости.

*Нечетким уравнением* называют уравнение, чьи коэффициенты или переменные являются нечеткими множествами, R-множестве действительных чисел.

Предположим, что в результате наблюдения объекта получено N значений входных (выходных) переменных  $(x_{1i}, x_{2i}, \dots, x_{ni}, y_i)$ , которые представлены таблицей 4.1.

Таблица 4.1.

| N   | X <sub>1</sub>  | x <sub>2</sub>  | ... | x <sub>n</sub>  | Y               |
|-----|-----------------|-----------------|-----|-----------------|-----------------|
| 1   | X <sub>11</sub> | x <sub>11</sub> | ... | x <sub>11</sub> | X <sub>11</sub> |
| 2   | X <sub>11</sub> | x <sub>11</sub> | ... | x <sub>11</sub> | X <sub>11</sub> |
| ... | ...             | ...             | ... | ...             | ...             |
| N   | X <sub>11</sub> | x <sub>11</sub> | ... | x <sub>11</sub> | X <sub>11</sub> |

#### Задачи идентификации:

1) выбрать функцию

$$\tilde{y} = f(x_1, \dots, x_n, \tilde{a}_1, \dots, \tilde{a}_n) = \sum_{j=0}^n \tilde{a}_j x_j \quad (4.2)$$

аппроксимирующую функцию  $\tilde{f}(x_1, \dots, x_n)$  заданную таблицей 4.1.

2) Определить оценки ее параметров:

Для подробной оценки можно воспользоваться критерием минимизации отклонения нечетких значений выходного параметра  $y_i$ , полученных из (4.2) от его выборочных нечетких значений, представленных в таблице 4.1.

$$\tilde{J} = U_{i=1, N} (\tilde{y}_i | - | \hat{y}_i)^2 \rightarrow \min \quad (4.3)$$

здесь  $| - |$  - ограниченная разница нечетких чисел, определенная по формуле

$$\mu_{\tilde{y}}(x) = (\mu_{\tilde{y}} | - | \hat{y}_i(x)) = \max(0, \mu_{\tilde{y}}(x) - \mu_{\hat{y}_i}(x)) \quad (4.4)$$

### **3. Методы идентификации объектов с нечеткими параметрами**

На первом этапе идентификации определяющее значение имеет качественный анализ процесса. На втором этапе, основным вопросом является

выбор способа оценивания, обеспечивающего необходимые свойства модели объекта.

Рассмотрим математическую модель, представленную в виде нечеткого уравнения множественной регрессии:

$$\tilde{y}_i = \tilde{a}_0 + \tilde{a}_1 x_1 + \tilde{a}_2 x_2 + \dots + \tilde{a}_n x_n \quad (4.5)$$

Задача оценивания параметров уравнения (4.5) заключается в определении коэффициентов  $\tilde{a}_i, i = 0 \dots n$ , удовлетворяющих условию (4.4). Выражение  $\tilde{y} = f(x_1, \dots, x_n, \tilde{a}_1, \dots, \tilde{a}_n)$  представляет собой многомерную функцию с нечеткими параметрами. Если учесть в (4), то

$$\tilde{J} = \bigcup_{i=1, N} (\tilde{y}_i | - | f(x_1, \dots, x_n, \tilde{a}_1, \dots, \tilde{a}_n))^2 \rightarrow \min \quad (4.6)$$

т.е. задача оценивания параметров уравнения регрессии (4.5) сводится к минимизации многомерной функции (4.6) нечеткими переменными.

Предположим, что нечеткие коэффициенты  $\tilde{Q}_i (i = 0, n)$  являются нормальными нечеткими множествами на  $R$ .

$$\tilde{Q}_i = \bigcup_{Q_i \in R} \mu_{\tilde{Q}_i}(Q_i) / \alpha_i,$$

Определим  $\alpha$ -уровневые множества нечетких коэффициентов  $\tilde{Q}_i$ :

$$\tilde{Q}_i^\alpha = \left\{ Q_i : Q_i \in R, \mu_{Q_i}(Q_i) \geq \alpha \right\}_{i=0, n}$$

где  $\alpha \in [0, 1]$

Тогда для каждого уровня

$$\alpha: \{ \alpha_0=0, \alpha_1, \dots, \alpha_j, \alpha_p=0 \}$$

можно записать уравнение множественной регрессии (4.5):

$$\begin{cases} y^{\alpha_0} = a_0^{\alpha_0} + a_1^{\alpha_0} x_1 + \dots + a_n^{\alpha_0} x_n, \\ y^{\alpha_1} = a_0^{\alpha_1} + a_1^{\alpha_1} x_1 + \dots + a_n^{\alpha_1} x_n, \\ \dots \\ y^{\alpha_p} = a_0^{\alpha_p} + a_1^{\alpha_p} x_1 + \dots + a_n^{\alpha_p} x_n, \end{cases} \quad (4.7)$$

Это уравнение является обыкновенным уравнением множественной регрессии, представляющим собой корреляционную связь между многими величинами на уровнях  $\alpha_j$ . Для оценивания нечетких коэффициентов  $\tilde{Q}_0, \tilde{Q}_1, \dots, \tilde{Q}_n$  достаточно определить коэффициенты  $a_0^{\alpha_j}, a_1^{\alpha_j}, \dots, a_n^{\alpha_j}$ , на каждом уровне  $\alpha_j$ , которые удовлетворяют условию:

$$J_j = \sum_{j=0}^n (y_j^{\alpha_j} - \hat{y}_j^{\alpha_j})^2 \rightarrow \min, j = 1 \dots p,$$

где  $\hat{y}_i^{\alpha_i} = a_0^{\alpha_i} + a_1^{\alpha_i} x_1 + \dots + a_n^{\alpha_i} x_n$ .

Наблюдаемые детерминированные значения  $J_i^{\alpha_i}$  получены аппроксимацией в таблице 4.1 нечетких значений выходной переменной  $y$   $\alpha$ -уровневыми нечеткими множествами в соответствии с аппроксимацией нечетких коэффициентов  $\tilde{Q}_i (i = 0, n)$

Таким образом, исходная задача оценивания нечетких коэффициентов нечеткого уравнения регрессии (4.5) сводится к классическим задачам оценивания параметров множественной регрессии (4.7).

|       |          |          |     |          |                  |
|-------|----------|----------|-----|----------|------------------|
| $N_0$ | $X_1$    | $X_2$    | ... | $X_n$    | $V_1$            |
| 1     | $X_{11}$ | $X_{21}$ | ... | $X_{n1}$ | $V_1^{\alpha_0}$ |
| 2     | $X_{12}$ | $X_{22}$ | ... | $X_{n2}$ | $V_1^{\alpha_0}$ |
| ...   | ...      | ...      | ... | ...      | ...              |
| N     | $X_{1N}$ | $X_{2N}$ | ... | $X_{nN}$ | $V_1^{\alpha_0}$ |
| 1     | $X_{11}$ | $X_{21}$ | ... | $X_{n1}$ | $V_1^{\alpha_0}$ |
| 2     | $X_{12}$ | $X_{22}$ | ... | $X_{n2}$ | $V_1^{\alpha_0}$ |
| ...   | ...      | ...      | ... | ...      | ...              |
| N     | $X_{1N}$ | $X_{2N}$ | ... | $X_{nN}$ | $V_1^{\alpha_0}$ |
| 1     | $X_{11}$ | $X_{21}$ | ... | $X_{n1}$ | $V_1^{\alpha_1}$ |
| 2     | $X_{12}$ | $X_{22}$ | ... | $X_{n2}$ | $V_1^{\alpha_1}$ |
| ...   | ...      | ...      | ... | ...      | ...              |
| N     | $X_{1N}$ | $X_{2N}$ | ... | $X_{nN}$ | $V_1^{\alpha_1}$ |
| ...   | ...      | ...      | ... | ...      | ...              |
| 1     | $X_{11}$ | $X_{21}$ | ... | $X_{n1}$ | $V_1^{\alpha_p}$ |
| 2     | $X_{12}$ | $X_{22}$ | ... | $X_{n2}$ | $V_1^{\alpha_p}$ |
| ...   | ...      | ...      | ... | ...      | ...              |
| N     | $X_{1N}$ | $X_{2N}$ | ... | $X_{nN}$ | $V_1^{\alpha_p}$ |

#### 4. Методика построения функции принадлежности

Пусть требуется для множества  $A$  найти значения функции принадлежности  $\mu_A(x_i)$ ,  $x \in X$ , для чего получаем следующий эксперимент с  $m$  экспертами.

1. Каждому из экспертов представляется количественное значение  $x_i$  и определяется, относит ли он это значение к множеству  $A$ . Далее находится общее число значений  $c(A)$ , которые эксперты отнесли к множеству  $A$ .
2. Значение функции принадлежности  $\mu_A(x_i)$  в точке  $x_i$  определяется как  $\mu_A(x_i) = c(A)/m$ .

3. Пункты 1 и 2 повторить для всех значений  $x_i$  шкалы  $X$ , тем самым определяя все значения функции  $\mu_A(x_i)$ .

Пример:

Школа  $X = \{1, 2, 3, 4, 5\}$  для оценки (выступили спортсменов)?, шесть судей-экспертов определяют множество  $A$  - «хороший» результат. Допустим ни один из экспертов 1 и 2 балла не отнес к множеству «хороший», один эксперт отнес 3 балла к «хорошему» результату, 4 эксперта отнесли 4 балла к множеству «хороший» и все 5 судей отнесли 5 баллов к множеству «хороший». Тогда в соответствии с описанной методикой функция принадлежности  $\mu_A(x_i)$  для множества  $A$  - «хороший»:  $\mu_A(1)=0$ ;  $\mu_A(2)=0$ ;  $\mu_A(3)=0.2$ ;  $\mu_A(4)=0.8$ ;  $\mu_A(5)=1$ .

Таким образом  $A = \{ \langle 0.2; 3 \rangle, \langle 0.8; 4 \rangle, \langle 1; 5 \rangle \}$ .

## **Лекция 5:**

### **Лингвистические и нечеткие переменные.**

Понятие лингвистической переменной впервые было введено Л.Заде [4]. *Лингвистические переменные* позволяют адекватно отразить приблизительное словесное описание предметов и явлений в том случае, если точное детерминированное описание отсутствует.

Пример: «Молод» - значение лингвистической переменной «Возраст».

Совершенство значений лингвистической переменной составляет *терм-множество* этой переменной.

$T$  (возраст) = {чрезвычайно молод, + молод, + молодой, + среднего возраста, + старый, + чрезвычайно старый}; знак «плюс» означает объединение. Этим значениям лингвистической переменной можно придать значения  $T = \{17, 24, 31, 48, 70, 94\}$ . Лингвистическое значение, например, «старый» можно рассматривать как название некоторого *нечеткого ограничения* на значение базовой переменной, характеризующееся *функцией совместимости*, которая каждому значению базовой переменной ставит в соответствие число из интервала  $[0, 1]$ , символизирующее совместимость этого значения с нечетким ограничением. Функция совместимости идентична функции принадлежности к множеству.

Другой аспект понятия лингвистической переменной состоит в том, что лингвистической переменной соответствуют два правила: *синтаксическое*, которое может быть задано в форме грамматики, порождающей названия значений переменной, и *семантическое*, которое определяет алгоритмическую процедуру для вычисления смысла каждого значения.

Дадим определение:

1) *Обычная* (не нечеткая) переменная характеризуется тройкой  $(X, U, R(X,u))$ , где  $X$  - название переменной;  $U$  - универсальное множество;  $u$  - общее название элементов множества  $U$ ;  $R(X, u)$  - подмножество множества  $U$ , представляющее собой ограничение на значения элементов  $U$ , обусловленное названием  $X$ .

Кроме того, переменной соответствует уравнение назначения  $x=u$ ;  $R(X,u)$ , или, что переменной  $Ч$  назначено значение и с учетом ограничения  $R(X,u)$ .

Пример: «возраст» - переменная;  $U$  - множество чисел  $0...1, 2, 3, 4 \dots$ ;  $R(X,u)$  - подмножество  $\{17, 24, 31, 94\}$ .

2) *Нечеткая переменная* характеризуется тройкой  $(\bar{X}, U, R(X,u))$ , где  $X$  - название переменной;  $R(X,u)$  - нечеткое подмножество множества  $U$ , представляющее собой нечеткое ограничение на значения ограничения на значения переменных  $U$ , обусловленное  $U$ . Неограниченная переменная  $U$  является для  $X$  базовой. Уравнение названия для  $X$  имеет вид  $X=u$ :  $R(X,u)$  и отражает то, что элементу  $ч$  назначается значение  $U$  с учетом ограничения  $R(X,u)$ .

Степень, с которой удовлетворяется это равенство, принято называть совместимостью значения  $u$  с  $R(X,u)$  и обозначать ее  $C(u)$ .

$$C(u) = \mu_{R(X,u)}(u), u \in U$$

где  $\mu_{R(X,u)}(u)$  - степень принадлежности  $u$  ограничению  $R(X,u)$ .

3) *Лингвистическая переменная* характеризуется набором  $(x, T(x), U, G, M)$ , где  $X$  - название переменной;  $T(x)$  - терм-множество переменной  $X$ , т.е. множество названий лингвистических значений переменной  $X$ , причем каждое из таких значений является переменной  $X$  со значениями из универсального множества  $U$  с базовой переменной  $u$ ;  $G$  - синтаксическое правило, порождающее название  $X$  значений переменной  $x$ ;  $M$  - семантическое правило, которое ставит в соответствие каждой нечеткой

переменной  $X$  ее смысл  $M(X)$ , т.е. нечеткое подмножество  $M(X)$  универсального множества  $U$ .

Конкретное название  $X$ , порожденное синтаксическим правилом  $G$ , называют *термом*. Терм, состоящий из одного или нескольких слов, фигурирующих всегда вместе, называют *атомарным термом*. Терм, состоящий из одного или более атомарных термов, называют *составным термом*.

Смысл  $M(X)$  термина  $X$  определяется как ограничение  $R(X, u)$  на базовую переменную  $U$ , обусловленное нечеткой переменной  $X$ :

$$M(X)=R(X,u)$$

$$\text{или } M(X)=R(\text{терм в } T(X))$$

Смысл термина  $X$  получается путем применения семантического правила  $M$  к значению термина  $X$ , назначенному согласно правой части уравнения (1.2).

## **Лекция 6:**

- 1. Элементы нечеткой логики. Операции над высказываниями.**
- 2. Многозначная логика.**

### **1. Элементы нечеткой логики. Операции над высказываниями**

Одной из простейших операций над высказываниями является отрицание:

$\neg A$  - «не  $A$ »;

Другой операцией - *конъюнкция* И -  $A \wedge B$  -  $A \& B$ .

Третья операция - *дизъюнкция* ИЛИ -  $A \vee B$  -  $A \times B$ .

Четвертая операция - *следование*: Если  $A$  то  $B$  - это высказывание ложно, когда посылка  $A$  истинна, а заключение  $B$  - ложно. Обозначение -  $A \rightarrow B$  - это выражение называется *импликацией*.

Выражение « $A$  тогда и только тогда, когда  $B$ » обозначается через  $A \equiv B$ , и называется *эквивалентностью*.  $A \equiv B$  истинно тогда и только тогда, когда  $A$  и  $B$  истинно.

Таблица истинности под высказыванием:

| A | B | $\neg A$ | $A \wedge B$ | $A \vee B$ | $A \rightarrow B$ | $A \equiv B$ |
|---|---|----------|--------------|------------|-------------------|--------------|
| И | И | Л        | И            | И          | И                 | И            |
| Л | И | И        | Л            | И          | И                 | Л            |
| И | Л | Л        | Л            | И          | Л                 | Л            |
| И | Л | И        | Л            | Л          | И                 | И            |

Символы называют  $\neg$ ,  $\wedge$ ,  $\vee$ ,  $\rightarrow$ ,  $\equiv$  пропозициональными связками.

*Пропозициональной формой* называют выражение, состоящее из букв (выражений) A, B, C и связок.

Пропозициональная форма определяет некоторую истинную функцию, которая может быть представлена таблицей истинности для этой формы. Истинностной функцией от n аргументов называют всякую функцию от n аргументов, принимающую значение «истина» и «ложь», если аргументы ее пробегают те же значения.

Пропозициональную форму, которая истинна независимо от значения входных переменных, называют *тавтологией*.

Пропозициональную форму, которая ложна при всех возможных значениях ее входных переменных, называют *противоречием* ( $A \equiv \neg B$ ) или ( $A \wedge (\neg A)$ ).

Правило отделения или правило гипотетического силлогизма:

Если  $(A \rightarrow B)$  истинно и A истинно, то B истинно. Под *силлогизмом* подразумевается дедуктивное умозаключение, в котором одно суждение является необходимым следованием двух других. Все это булева логика.

## 2. Многозначная логика

Многозначная логика основывается на теории нечетких множеств.

Пусть P является высказыванием, а  $\dots(P)$  - его значением истинности, при этом  $\dots(P) \dots [0, 1]$ .

Отрицание для высказывания P будет:  $\delta(\neg P) = 1 - \delta(P)$ ;  $\delta(\neg \neg P) = \delta(P)$ .

Импликация:  $\delta(P \rightarrow Q) = \delta(\neg P \vee Q)$ ;

Эквивалентность :  $\delta(P \leftrightarrow Q) = \delta(P \rightarrow Q \wedge (P \rightarrow Q))$ ;

Тавтология:  $\delta(P) = \delta(P \vee \neg Q)$ ;

противоречие:  $\delta(P) = \delta(P \wedge \neg Q)$ .

Так как связок достаточно большое количество, то на их основе формируется значительное количество логик.

Рассмотрим две из них.

1. Логика, основанная на  $(\tilde{\Phi}(x), \cup, \cap, -)$ , является многозначной и ее обычно называют к-логикой. Многозначная логика является размытием (в смысле нечеткости) стандартного исчисления высказываний P. В этой логике каждому высказыванию P ставится в соответствие нормализованное нечеткое множество в  $[0, 1]$ , т.е. пара  $[\mu_P(0), \mu_P(1)]$  как степень логичности и степень истинности соответственно. Так как логические связки стандартного исчисления высказываний являются функционалами истинности, т.е. представляются в виде функций, то их можно размыть.

В этой логике дизъюнкцию и конъюнкцию определяют как:

$$\delta(P \vee Q) = \max(\delta(P), \delta(Q)); \delta(P \wedge Q) = \min(\delta(P), \delta(Q)).$$

Имеется 16 связок.

Конъюнкция и дизъюнкция удовлетворяют закону поглощения и закону Де-Моргана.

Эквивалентность определяется как:

$$\delta(\neg(P \vee Q) \wedge \delta(P \vee \neg Q)) = \delta[(P \wedge Q) \vee \delta(\neg P \wedge \neg Q)];$$

$$\text{Импликация: } \delta[P \rightarrow (Q \rightarrow R)] = \delta[(P \wedge Q) \rightarrow R].$$

$$\text{Тавтология и противоречие: } \delta(P \rightarrow P) = \delta(P); \delta(P \leftrightarrow P) = \delta(P).$$

- 2) Логика основанная на  $(\tilde{\Phi}(x), \dot{\cup}, \dot{\cap}, -)$

В этой логике дизъюнкция и конъюнкция определяются как:  $\delta(P \vee Q) = \min(1, \delta(P) + \delta(Q))$ ;  $\delta(P \wedge Q) = \max(0, \delta(P) + \delta(Q) - 1)$ .

Имеется 16 связок.

В большинстве случаев работают с многозначными логиками, в которых логические переменные принимают значения из интервала  $[0, 1]$ , разбитого на 10 подинтервалов, т.е. используя множество  $v[0, 0.1, 0.2, \dots, 1]$ .

Используются:

S\*-логика  $a \rightarrow b$  1, если  $a \neq 1$ , или  $b = 1$ ; 0 - в противном случае;

S-логика  $a \rightarrow b$  1, если  $a \leq b$ ; 0 - в противном случае;

G-логика  $a \rightarrow b$  1, если  $a \neq b$ ; b - в противном случае;

L-логика  $a \rightarrow b = \min(1, 1-a+b)$ ;

KD-логика  $a \rightarrow b = (1-a) \vee b = \max(1-a, b)$ .

$Z_1 - Z_3$  - логики:

$Z_1$  - логика  $a \rightarrow b$   $(1 - a)$ , если  $a < b$ ; 1, если  $a = b$ ;  $b$ , если  $a > b$ ;

$Z_2$  - логика  $a \rightarrow b$  1, если  $a \leq b$ ;  $(1 - a) \wedge b$ , если  $a > b$ ;

$Z_2$  - логика  $a \rightarrow b$  1, если  $a \leq b$ ;  $b/(a + (1 - b))$ , если  $a > b$ .

Существуют таблицы импликативных переходов для каждой из этих логик.

## Лекция 7-8:

1. **Нечеткие алгоритмы.**
2. **Основные принципы построения нечетких алгоритмов.**
3. **Алгоритмы классификации и композиции.**
4. **Модели принятия решений на основе алгоритмов классификации и композиции**

### 1. Нечеткие алгоритмы

Лингвистические модели описывают разбиение многомерного пространства признаков факторов, наиболее существенно влияющих на выбор управляющих решений, на нечеткие области (эталонные классы).

Лингвистические модели представляют собой тройку множеств  $(W, \Psi, H)$  и  $(W, T, H)$ , в которой  $W = X \times Y \times Z \times \dots \times S$  - множество признаков-факторов;  $\Psi = \{L_1, \dots, L_i, \dots, L_k\}$  - разбиение  $W$  на нечеткие эталонные классы  $L_i$ ,  $K\{h_1, \dots, h_i, \dots, h_k\}$  - множество управляющих решений  $h_i$ , соответствующих классам  $L_i$ .

« $X$  - произведение».

Таким образом, нечеткая модель включает в себя (в отличие от обычной и регрессионной модели с нечеткими параметрами) и управляющие алгоритмы.

*Нечеткими алгоритмами управления* называется упорядоченная последовательность нечетких инструкций (могут иметь место отдельные четкие инструкции), обеспечивающая функционирование некоторого объекта или процесса.

Пример фрагмента нечеткого алгоритма:

$x$  мало,  $y$  велико.

Если  $x$  мало и  $x$  слегка увеличить, то  $y$  слабо увеличится.

Если  $x$  мало и  $y$  велико, то  $z$  очень мало, иначе  $z$  немало.

Если  $y$  не очень велико, то  $x$  немного уменьшить.

Если  $y$  очень велико, то стоп.

Каждый шаг этого алгоритма представляет собой нечеткий оператор - нечеткую инструкцию, записанную на естественном языке (возможно формализованном).

## 2. Основные принципы построения нечетких алгоритмов

Нечеткие инструкции формируются на основе обобщения опыта специалиста или на основе тщательного изучения и содержательного анализа.

Для построения Нечетких алгоритмов учитываются все ограничения и критерии, вытекающие из рассмотрения задачи, однако полученные нечеткие инструкции используются не все: выделяются наиболее существенные из них, исключаются возможные противоречия и устанавливается порядок их выполнения, приводящий к решению задачи.

При переходе от нечетких инструкций к их формализованному представлению используется математический аппарат теории нечетких множеств и нечеткой логики.

При формализации связей нечетких операторов между собой применяются различные операции над нечеткими множествами, например:  $\cap$ ,  $\times$ ,  $^{\circ}$ .

при построении нечетких алгоритмов перечисления, произведения, композиции на основе множеств  $\Pi$  выполняются преобразования над их степенями принадлежности  $\mu(u)$ ,  $\mu(v)$ ,  $\mu(u, v)$  для множеств  $I$ , а затем осуществляется переход от множества  $I$  к множеству  $\Pi$

Для адаптации и обучения нечетких алгоритмов предусматривается возможность введения в них новых нечетких инструкций и подбора вида функций принадлежности нечетких множеств.

Примечание: нечеткие алгоритмы целесообразно разрабатывать в том случае, когда сложность ситуации исключает возможность выполнения только четких инструкций.

## 3. Алгоритмы классификации и композиции

Наиболее распространенными типами нечетких алгоритмов являются алгоритмы классификации и композиции. Первые строятся на основе сравнения текущих нечетких операторов с эталонными и принятия в зависимости от результатов сравнения управляющего решения. Алгоритмы композиции основываются на последовательной связи нечетких операторов между собой и получении результирующего нечеткого оператора, соответствующего управляющему решению.

Обозначение и классификация нечетких операторов-инструкций:

Виды:

назначающие (и.и. -  $x$  не мало и не очень велико)

условные (если  $x$  положительное, то  $u$  незначительно уменьшить, иначе  $u$  увеличить)

безусловные ( $x$  уменьшить на  $x$ , несколько раз)

Нечеткие операторы могут иметь в своем составе и четкую часть (если  $x$  очень велико, то  $u$  увеличить в 3 раза, иначе стоп).

Обычно в алгоритмах классификации применяются назначающие операторы, в алгоритмах композиции - все виды нечетких операторов.

Обозначения:

Назначающий -  $P$ ;

Условный -  $R$ ;

Безусловный -  $Q$ .

*Назначающий* оператор представляет собой множество  $P$ , над которым могут выполняться операции  $\bar{\cap}$ ,  $\cup$ ,  $\cap$ ,  $P^\alpha$ ; например,  $x$  не мало и не очень велико -  $\bar{\cap}P_1(x) \cap \bar{\cap}P_2(x)$ .

*Безусловный* оператор представляет собой множество  $Q$ , выражающее оцениваемое на интервале действие, над которым могут выполняться операции  $\bar{\cap}$ ,  $\cup$ ,  $\cap$ ,  $Q^\alpha$ ; например,  $x$  незначительно переместить вперед -  $Q(\Delta x)$ .

*Условный* оператор записывается как предложение «если  $A$ , то  $B$ , иначе  $K$ », которое описывает нечеткое отношение между значениями лингвистических переменных  $A$ ,  $B$  и математически интерпретируется двумя способами. В первом способе используется отношение, определяемое как  $R=A \times B \cup \bar{\cap}A \times K$ . Во втором случае используется функция нечеткой логики  $F$ , определяемая как

$F=(a \rightarrow b)(\bar{1}a \rightarrow k)$ ,  $X, Y, U$  соответственно и принимающие на них те же значения, что и функция принадлежности множеств  $A, B, K$ .

Для реализации алгоритмов классификации с учетом концепции нечеткого включения и нечеткого равенства требуется выполнять соответственно операции  $A \subseteq B$  и  $A \approx B$ , при этом степени  $\mu(A \subseteq B)$  и  $\mu(A \approx B)$  находятся на основе  $\bar{1}, \cup, \cap$ .

При реализации алгоритмов композиции имеют место 9 видов связи операторов  $P_1 \sigma P_2, P \sigma Q, P \sigma R, R \sigma P, Q \sigma P, R_1 \sigma R_2, R \sigma Q, Q \sigma R, Q_1 \sigma Q_2$ , где  $\sigma$  - обобщенная операция. Анализ видов связи операторов показал, что в большинстве случаев имеют место 4 операции:

$\times$  - для связи назначающих операторов между собой и безусловных операторов между собой;

$\circ$  - (композиция) для связи назначающих и безусловных операторов с условными операторами;

$A^{\mu B}$  - для связи назначающих и безусловных операторов между собой;

$\cap$  - для связи условных операторов между собой.

В свою очередь, указанные операции требуют выполнения операций  $\cup, \cap, A^\alpha$  (множество уровня  $\alpha$ ). В результате будем утверждать, что для синтеза нечетких алгоритмов управления достаточно использовать такие типовые операции над нечеткими множествами:  $\bar{1}, \cup, \cap, A^\alpha, A^{\mu B}, \times$  (перемножение),  $\circ, R$  (нечеткого отношения).

#### **4. Модели принятия решений на основе алгоритмов классификации и композиции**

##### **Модель и алгоритм класификации**

Путем экспертного опроса из содержательного анализа задачи выделяются признаки-факторы,  $X, Y, Z$  (анализ будем проводить относительно трех признаков, которые, по мнению экспертов, наиболее существенно влияют на выбор решений  $h_i$ ) и формируется пространство  $W=X*Y*Z$ . С каждым из выделенных признаков связываем лингвистическую переменную со своими значениями. Лингвистические переменные, соответствующие признакам  $X, Y, Z$ , обозначим через  $A, B, C$ , а их значения - через  $\{\alpha_r, r=1 \dots n\}, \{\beta_j, j=1 \dots k\}, \{\gamma_i, i=1 \dots p\}$ . Для всех выделенных значений  $\alpha_r, \beta_j, \gamma_i$  путем экспериментального

опроса строятся функции принадлежности  $\mu_\alpha$ ,  $\mu_\beta$ ,  $\mu_\gamma$  на соответствующих базовых шкалах  $X$ ,  $Y$ ,  $Z$ .

Далее строится структура модели в виде решений таблицы, имеющей  $n*m*r$  строк и 4 столбца.

Строки таблицы соответствуют всевозможным наборам  $(\alpha_i, \beta_j, \gamma_k)$ , первые три столбца обозначены символами лингвистических переменных  $A$ ,  $B$ ,  $C$ . Четвертый -  $H$ . В столбце  $H$  для каждого набора  $A$ ,  $B$ ,  $C$  простраивается одно из возможных управляющих решений  $h_i$ , которое принял бы эксперт в ситуации, словесно описанной набором  $A$ ,  $B$ ,  $C$ . В итоге формируется  $(W, \Psi, H)$ , в которой каждый класс  $L_i$ , характеризуется функцией принадлежности  $\mu_{L_i}$ , задаваемой следующей нечеткой логической формулой:

$$\mu_{L_i}(x, y, z) = \bigvee_{(\alpha\beta\gamma) \in L_i} (\mu_\alpha(x) \wedge \mu_\beta(y) \wedge \mu_\gamma(z)),$$

где  $L_i$  - множество наборов  $(\alpha, \beta, \gamma)$ , которым в решающей таблице соответствует решение  $h_i$ ,  $x \dots X$ ,  
 $y \dots Y$ ,  $z \dots Z$ .

Опишем алгоритм выработки управляющих решений на основе этой модели.

Если ситуация, характеризующаяся состоянием системы, представляется точкой  $W_0(x_0, y_0, z_0)$  в пространстве  $W$ , тогда алгоритм сводится к следующей формальной процедуре:

Подставим точку  $(x_0, y_0, z_0)$  в функции принадлежности  $\mu_{L_i}$  эталонных классов  $L_i$  и вычислим значение  $\mu_{L_i}(x_0, y_0, z_0)$ .

Среди всех значений  $\mu_{L_i}(x_0, y_0, z_0)$  находим такое, которое является максимальным в точке  $(x_0, y_0, z_0)$ .

Принимаем решение  $h_s$ , соответствующее эталонному классу  $L_s$ .

Конец.

Алгоритм позволяет определить для данной ситуации, характеризуемой точкой в многомерном пространстве признаков, наиболее близкий эталонный класс в этом пространстве и выработать соответствующее ему решение.

Модель и алгоритм композиции

Модель композиции описывает связь всех возможных состояний сложной системы с управляющими решениями и представляет собой тройку  $(W, T, H)$ , в

которой  $W=X \times Y \times Z \times \dots \times S$  - многомерное пространство признаков-факторов наиболее существенно влияющих на выбор управляющих решений;

$T$  - Нечеткое отношение на  $W \times H$ , где  $H$  - множество управляющих решений.

Определение нечеткого отношения  $T$ , связывающего  $W$  с  $H$ , строится на основе словесной информации эксперта путем непосредственной формализации его нечетких стратегий. Эксперт описывает особенности функционирования системы в виде ряда высказываний  $\{\pi_i\}$ , которые формализуются с помощью операторов  $P, Q, R$ ; при этом условные операторы могут интерпретироваться как отношениям  $R$ , так и функцией  $F$ . В общем случае каждый оператор задается значениями функции принадлежности  $\mu_{T(\pi_i)}(x_0, y_0, z_0)$ , а для отношения  $T$  эти значения находятся, как  $\mu_T(x_0, y_0, z_0) = \sigma_i(\pi_i)(x_0, y_0, z_0)$ , где  $\sigma_i$  - обобщенная операция.

Для этой модели алгоритм выработки управляющих решений для ситуации, характеризующей состояние системы в точке  $W_0(x_0, y_0, z_0)$  в пространстве  $W$ , сводится к следующей формальной процедуре:

- Подставляем точку  $W_0(x_0, y_0, z_0)$  в функцию принадлежности  $\mu_T$ , получаем значения функции  $\mu_T(x_0, h_i)$  нечеткого решения, зависящий только от одной переменной.
- Перебираем последовательно все  $h_i \dots H$  и ищем такое  $h_i$ , которое доставляет максимум функции  $\mu_T$ .
- Выбираем значение  $h_s$ , в качестве искомого управляющего решения.
- Конец.

Алгоритм позволяет выработать управляющее решение путем композиции нечетких отношений, одно из которых описывает текущее состояние системы, а другое - модель управления.

Рассмотрим пример построения и реализации модели и алгоритма: предположим, что требуется оперативно управлять запуском партий заготовок некоторого изделия на вход технологической линии. Управление заключается в принятии управляющего решения  $H$  о величине запускаемой партии. На выбор этого решения влияют следующие факторы:

$X$  - планируемый объем;

$Y$  - класс точности этого изделия;

Признаки X и Y и решение H сопоставим соответственно следующим лингвистическим переменным: A - план; B - класс точности; C - запуск. Для переменной A - лингвистическое значение  $\alpha_1$  - завышенный; для переменной B -  $\beta_1$  - высокий,  $\beta_2$  - низкий; для переменной C -  $\gamma_1$  - большой,  $\gamma_2$  - средний. Представим особенности функционирования данного производственного процесса в виде следующих высказываний

$\pi_1$  - если A - зависимый и B - высокий, то C - большой,

$\pi_2$  - если B - низкий, то C - средний.

Исходя из функций принадлежности соответствующих лингвистических значений, запишем модель:

$(W, T, H): W=X \times Y, H=\{h\}; T=T(\pi_1) \cap T(\pi_2);$

$\mu_T(x, y, z) = \mu_{T(\pi_1)}(x, y, z) \wedge \mu_{T(\pi_2)}(x, y, z) = (\mu_{\alpha}(x) \wedge \mu_{\beta}(y) \rightarrow \mu_{\gamma_1}(h)) \vee (\mu_{\beta_2}(y) \rightarrow \mu_{\gamma_2}(h))$

Для точки  $(x_0, y_0)$  получим  $\mu_T(x_0, y_0, z_0) = (\mu_{\alpha}(x_0) \wedge \mu_{\beta_1}(x_0) \wedge \mu_{\beta_1}(y_0) \rightarrow \mu_{\gamma_1}(h)) \wedge (\mu_{\beta_2}(y_0) \rightarrow \mu_{\gamma_2}(h))$ .

## **Лекция 9:**

**Нечеткие модели принятия решений при управлении сложными системами:**

- 1. Модели принятия решений на основе действий оператора.**
- 2. Лингвологическая и автолингвологическая модели принятия решений.**
- 3. Ситуационное управление.**

### **1. Модели принятия решений на основе действий оператора**

Модель задается в виде тройки  $(W, R, H)$ , где  $W = \{w_1, \dots, w_i, \dots, w_n\}$ , а  $H = \{h_1, \dots, h_i, \dots, h_n\}$  - базовые множества, на которых заданы соответственно входы  $s_j$  и выходы  $t_j$  системы; R - нечеткое отношение «вход-выход», являющееся условным оператором. Предполагается способ формирования отношения R на основе идентификации его по внешнему описанию нечеткой системы  $(s_j, t_j) \dots W \times H$ .

## 2. Лингвологическая и автолингвологическая модели принятия решений

Типовая структура *лингвологической* модели записывается следующим образом:

если  $a_{11}, a_{12}, \dots, a_{1j}, \dots, a_{1m}$ , то  $b_1$ , иначе,

...

если  $a_{i1}, a_{i2}, \dots, a_{ij}, \dots, a_{im}$ , то  $b_i$ , иначе,

...

если  $a_{n1}, a_{n2}, \dots, a_{nj}, \dots, a_{nm}$ , то  $b_k$ , иначе,

если  $a_1, a_2, \dots, a_j, \dots, a_m$ , то  $b$ .

Здесь  $a_{ij}$ ,  $a_j$ ,  $b_i$ ,  $b$  - значения лингвистических переменных, которым соответствуют множества  $a_{ij}$ ,  $a_j \dots X_j$ ,  $b_i$ ,  $b \dots Y$ , где  $X_j$  и  $Y$  - базовые множества входных и выходного параметров.

Для построения лингвистической модели необходимо найти новое значение переменной  $b_i$ , которое определяется в результате лингвистической аппроксимации множества  $b$ , вычисляемого несколькими способами:

1) В терминах нечетких множеств множество  $b$  определяется как

$$b = \bigcup_{i=1}^n (U_{i=1}^m (a_j \dots R_{ij})), \text{ где } R_{ij} = a_{ij} \dots b_i.$$

2) В терминах нечетких векторов и векторных матриц множество  $b$  определяется как соответствующий ему нечеткий вектор  $b^T$  следующим образом:

$$b^T = \bigoplus_{i=1}^m (\text{Rep}(A^T \circ A_i) I_0) \circ b^T,$$

где  $A^T$  - нечеткая матрица, составленная из нечетких векторов  $a_j$ ;  $A_i$  - нечеткие матрицы составленные из нечетких векторов  $a_{ij}$ , соответствующих множествам  $a_{ij}$ ;  $b_i^T$  - нечеткий вектор, соответствующий множеству  $b_i$ ;  $\text{Rep}$  - репроекция диагонального вектора;  $I_0$  - единичная матрица;  $\bigoplus_i$  - символ операции прямой суммы.

Структура *автолингвологической* модели задается в виде:

если  $a(t+1), a(t+2), \dots, a(t+l+1)$ , то  $a(t+l+2)$  иначе,

...

если  $a(t+k), a(t+k+1), \dots, a(t+k+l)$ , то  $a(t+k+l+1)$  иначе,

если  $a(t+k+1), a(t+k+2), \dots, a(t+k+l+1)$ , то  $a(t+k+l+2)$ .

Здесь  $a(t)$  для всех  $t \dots T$  - значение лингвистической переменной, которой ставятся в соответствие множество  $a(t) \dots P(x) \dots T$ , где  $X$  - базовое множество,  $P(x)$

- множество всех нечетких множеств,  $T$  - множество временных промежутков. Новое значение  $a(t+k+1+2)=b$  определяется как множество  $b$  по формулам (1, 2).

### 3. Ситуационное управление

Модели ситуационного управления, основываются на анализе нечетких ситуаций, возникающих на объекте управления. Под нечеткой ситуацией  $X$  понимается описание функционирования объекта управления в данный момент времени соответствующих значений признаков, определенных в терминах естественного языка [Мелихов А.Н., Барштейн Л.С., Коровин С.Я., Различные ситуационные модели принятия решений. Таганрог, 1986г., 92с.].

Предлагаются 2 модели: ситуация - действие (С-Д) и ситуация - стратегия управления - действие (С-СУ-Д). Суть модели С-Д состоит в анализе нечеткой текущей ситуации  $x_0$  путем сравнения ее с эталонными ситуациями:  $x_3^j$ ,  $j=1 \dots n$ ; определение эталонной ситуации  $x_3^k$ ,  $k=1 \dots n$  с максимальной степенью нечеткого равенства ... и немедленном принятии решения  $h_i^k$ , соответствующего  $x_3^k$ . Возникшая после применения  $h_i^k$  ситуация  $x_{ок}$  считается текущей и процесс управления объектом продолжается.

Модель С-СУ-Д предполагает определение стратегии управления объектом с последующим использованием ее для приведения объекта в целевую ситуацию  $X_c$ . Под стратегией управления понимается кратчайшая последовательность подцелевых ситуаций и элементарных управлений для достижения ситуации  $X_c$ .

## Тема 1.3: Нечеткие регуляторы

### Лекция 10:

1. **Общая структура СУ с нечеткими регуляторами. Принцип работы. База знаний регулятора.**
2. **Таблица лингвистических правил.**

## 1. Общая структура СУ с нечеткими регуляторами. Принцип работы. База знаний регулятора

В основе проектирования интеллектуальных нечетких регуляторов (НР) лежит конструирование “знаний” с применением методов представления и поиска знаний, которые базируются на принципах искусственного интеллекта. Структура ИСУ с НР представлена на рис. 10.1.

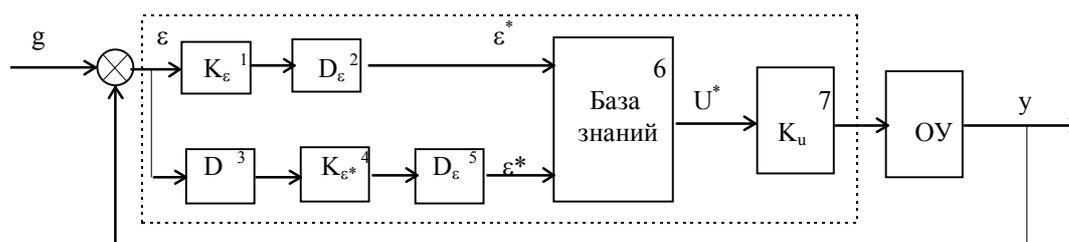


Рисунок 10.1

Выходная переменная объекта управления  $y$  сравнивается с ее заданным значением  $g$ , ошибка рассогласования  $\varepsilon$  поступает в масштабный элемент 1 с коэффициентом  $k_\varepsilon$ , и в дифференциатор 3, выход которого подается на  $K_\varepsilon$  в масштабном элементе 4. Элементы 2-5 предназначены для преобразования текущих значений рассогласования и производной от рассогласования в их лингвистические значения. Нечеткие значения  $\varepsilon^*$  и  $\dot{\varepsilon}$  поступают в главный элемент нечеткого регулятора 6 - базу знаний БЗ. Как правило, БЗ НР строится на основе продукционной модели знаний, имеющих конструкцию вида “если ..., то ...”. Каждая продукция, представляющая собой множество пар “С-D”, позволяет ставить в соответствие сложившейся ситуации действие регулятора в виде значения регулирующего воздействия на объект.

Найденное лингвистическое значение управления после умножения на масштабный коэффициент  $k_u$  в элементе 7 и преобразования его в четкое значение  $U$  поступает на исполнительный элемент ОУ.

При синтезе НР основной центр тяжести падает на конструкцию из БЗ. Из теории ИС известно, что представление знаний в управляющих системах осуществляется логическим, реляционными, фреймовыми и продукционными языками. Мы будем основываться на продукционной модели описания знаний.

Каждая продукция представляет собой фрагмент знаний вида - ситуация-действие. Левая часть каждой продукции рассматривается как конъюнкция элементарных условий, а правая часть как множество элементарных действий.

Для регулятора, представленного на рис. 10.1 любое правило в БЗ может быть записано как:

Если ( $\varepsilon$  есть  $\varepsilon_1^*$ ) и ( $\dot{\varepsilon}$  есть  $\dot{\varepsilon}_1^*$ ), то ( $u$  есть  $u_1^*$ ), где  $\varepsilon, \dot{\varepsilon}, u$  - переменные,  $\varepsilon_1^*, \dot{\varepsilon}_1^*, u_1^*$  - их лингвистические значения.

## 2. Таблица лингвистических правил

Лингвистические правила, являющиеся БЗ НР могут быть представлены таблицей. Например, табл. 10.1, где использованы следующие обозначения:

- |                             |                              |
|-----------------------------|------------------------------|
| ОБ - отрицательное большое; | ПН - положительный нуль;     |
| ОНБ - - " - ниже большого;  | ПНМ - - " - ниже маленького; |
| ОС - - " - среднее;         | ПМ - - " - маленького;       |
| ОНС - - " - ниже большого;  | ПНС - - " - ниже среднего;   |
| ОМ - - " - ниже большого;   | ПС - - " - среднее;          |
| ОНМ - - " - ниже большого;  | ПНБ - - " - ниже большого;   |
| ОН - - " - ниже большого;   | ПБ - - " - большое;          |

Таблица 10.1

| Σ   | Σ   |     |     |     |     |     |     |     |     |     |     |     |     |    |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|
|     | ОБ  | ОНБ | ОС  | ОНС | ОМ  | ОНМ | ОН  | ПО  | ПНМ | НМ  | ПНС | ПС  | ПНБ | ПБ |
| ОБ  | ПБ  | ПБ |
| ОНБ | ПНБ | ПБ |
| ОС  | ПНБ | ПНБ | ПНБ | ПНБ | ПНБ | ПНС | ПНС | ПНС | ПНС | ПНС | ПНБ | ПНБ | ПБ  | ПБ |
| ОНС | ПНБ | ПНБ | ПНБ | ПНБ | ПС  | ПНС | ПНС | ПНС | ПНС | ПС  | ПНБ | ПНБ | ПБ  | ПБ |
| ОМ  | ПНБ | ПНБ | ПНБ | ПС  | ПНС | ПМ  | ПМ  | ПМ  | ПС  | ПНС | ПС  | ПНБ | ПБ  | ПБ |
| ОНМ | ПНБ | ПНБ | ПНБ | ПНС | ПМ  | ПНМ | ПНМ | ПНМ | ПНС | ПНС | ПС  | ПНБ | ПБ  | ПБ |
| ОН  | ПНБ | ПНБ | ПС  | ПНС | ПНМ | ПН  | ПН  | ПН  | ПНМ | ПНС | ПБ  | ПНБ | ПНБ | ПБ |
| ПО  | ОНБ | ОНБ | ОС  | ПНМ | ОНС | ОНМ | ОНМ | ОНМ | ОН  | ОНС | ОС  | ОНБ | ОНБ | ОБ |
| ПНМ | ОНБ | ОНБ | ОС  | ОНС | ОНМ | ОНМ | ОНМ | ОНМ | ОН  | ОНС | ОС  | ОБ  | ОБ  | ОБ |
| НМ  | ОНБ | ОНБ | ОС  | ОНС | ОМ  | ОМ  | ОМ  | ОМ  | ОНМ | ОС  | ОНБ | ОБ  | ОБ  | ОБ |
| ПНС | ОБ  | ОНБ | ОС  | ОНС | ОНС | ОНС | ОНС | ОНС | ОНС | ОНБ | ОБ  | ОБ  | ОБ  | ОБ |
| ПС  | ОБ  | ОНБ | ОНБ | ОНБ | ОС  | ОС  | ОС  | ОС  | ОС  | ОС  | ОБ  | ОБ  | ОБ  | ОБ |
| ПНБ | ОБ  | ОБ  | ОНБ | ОБ  | ОБ  | ОБ  | ОБ  | ОБ |
| ПБ  | ОБ  | ОБ |

Масштабные коэффициенты  $K, K_{\varepsilon}, K_u$  являются параметрами универсальных множеств  $E, E'$  и  $U$ , на которых определяются нечеткие множества  $\varepsilon_1^*, \dot{\varepsilon}_1^*, u_1^*$ . Например, если универсальным множеством  $E$  является  $(\pm 10)$  (шкала) и требуется, чтобы рассогласование в системе находилось в

диапазоне ( $\pm 1$ ), то  $K_\varepsilon=10,0$  с тем, чтобы нечеткий регулятор мог использовать все универсальное множество, на котором определено нечеткое множество.

## **Лекция 11:**

- 1. Аналитические методы синтеза НР.**
- 2. Лингвистический синтез НР.**
- 3. Программная реализация НР.**

### **1. Аналитические методы синтеза НР**

Рассмотрим систему управления по возмущению. Структура системы показана на рис 11.1.

...

Система функционирует следующим образом. Возмущение  $f$ , проходя по двум каналам через ОУ, влияет на выходные величины  $y$ . Первый канал, являясь естественным, реагирует на значение возмущения  $f$  в соответствии с передаточной функцией  $W(p)$ . По второму каналу системы возмущение  $f$  проходит следующим образом: текущее значение  $u$  производится от  $f$ , после умножения на масштабные коэффициенты  $k$ ,  $k$  . и последующего преобразования в элементах  $D$  и  $D$ , в их лингвистические значения, поступают на вход нечеткого компенсатора  $\Gamma$ . Выход последнего после умножения на масштабный коэффициент  $K_u$  поступает на вход объекта, изменяя выходную величину  $y$  в соответствии с передаточной функцией  $W_0(p)$ . Задачей синтеза рассматриваемой системы по критерию инвариантности  $y$  относительно изменения  $f$  заключается в конструировании такого закона регулирования /// который обеспечивает независимость выходной переменной при произвольных изменениях возмущения  $f$ . Идея данного метода синтеза подобных систем заключается в аппроксимации нечетких отношений  $R$  функциями в четкой области, с дальнейшими использованием аналитических методов их оптимального конструирования.

Из ТАУ известно, что в классических СУ по возмущению закон регулирования описывает обычное отображение

...

где  $F$ - пространство входов,  $U$ - пространство управлений. Если регулятор реализован в классе нечетких отображений типа

Если  $x_1$  и если  $x_2$ , то

Если  $x_1$  и если  $x_3$ , то

...

Если  $x_1$  и если  $x_4$ , то

То статически аппроксимируя их, можно сконструировать отображение  $\Phi_0$ . Построенное отображение  $\Phi_0$  нечеткого компенсатора. Позволяет исследовать систему управления на качество работы методами ТАУ,

*Пример.* Пусть нечеткий компенсатор определяется ТЛП типа:

Если значение возмущения ПБ и если значение скорости изменения возмущения ОМ, то значение управления ПО, ПНВЧС. Если значение возмущения ОБ и если значение скорости изменения возмущения ПБ, то значение управления ПС.

Ил ...

или ...

Аппроксимируя нечеткий компенсатор аналитические выражения ...), получим четкую структуру СУ эквивалентную нечеткой, для которой условия инвариантности определяются как

...

где ... - соответственные передаточные функции по возмущающему и управляющему каналам.

...

Аналитическая методика может быть применена и к комбинированной нечеткой СУ, схема которой показана на рис. 5. В этом случае нечеткие компенсатор и регулятор, представленные ТЛП, аппроксимируем аналитическими функциями вида

...

## 2. Лингвистический синтез НР

Рассмотрим методику лингвистического синтеза НР с двумя входом и выходом для объектов 1-го и 2-го порядков при наличии их нечеткого описания. ЛС НР для объекта 1-го порядка.

Пусть динамика ОУ описывается некоторой таблицей лингвистических правил (ТЛП) (табл. 11.2.),

| U* | X* |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|
|    | ОБ | ОС | ОМ | НО | ПМ | ПС | ПБ |
| ОБ | НО | ОМ | ОС | ОБ | ОБ | ОБ | ОБ |
| ОС | ПМ | НО | ОМ | ОС | ОБ | ОБ | ОБ |
| ОМ | ПС | ПМ | НО | ОМ |    |    |    |
| НО | ПБ | ПС | ПМ | НО | ОМ | ОС | ОБ |
| ПМ |    |    |    | ПМ | НО | ОМ | ОС |
| ПС |    |    |    | ПС | ПМ | НО | ОМ |
| ПБ |    |    |    | ПБ | ПС | ПМ | НО |

построенной на основе реакции объекта на скачкообразное изменение управления, и представлена в виде нечетких продукций:

Если  $\tilde{u}$  есть ОБ, если  $\tilde{x}$  есть НО, то  $\dot{\tilde{x}}$  есть ОБ иначе,

Если  $\tilde{u}$  есть ОБ, если  $\tilde{x}$  есть ОМ, то  $\dot{\tilde{x}}$  есть ОС иначе,

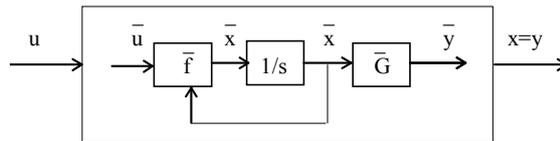
Если  $\tilde{u}$  есть ПБ, если  $\tilde{x}$  есть ПБ, то  $\dot{\tilde{x}}$  есть НО иначе и т.д.,

где  $\tilde{x}, \dot{\tilde{x}}$  их - базовые переменные соответственно нечетких множеств выхода и скорости изменения выхода:  $\tilde{u}$  - базовая переменная нечеткого множества управления, при этом соответствующая лингвистической переменной  $U^\pi = \{\tilde{u}, \Psi, \tilde{u}\}$ , ОБ, ОС, ОМ, НО, ПМ, ПС, ПБ - значения лингвистических переменных, описываемых соответствующими нечеткими подмножествами, заданными степенями принадлежности на универсальных меньшинствах (рис. 11.2). В таблице 11.2. установившиеся режимы ОУ определяются теми состояниями  $\tilde{u}, \tilde{x}, \dot{\tilde{x}}$ , для которых они равны НО. Из ТЛП видно, что для каждого термина управляющего воздействия существует установившееся состояние объекта, которое будет иметь место при  $\dot{\tilde{x}} = \text{НО}$ .

Лингвистическую модель 11.1. можно формально представить в следующем виде:

$$\dot{\tilde{x}} = f^\pi(\tilde{x}, \tilde{u}); \quad \tilde{y} = G^\pi(\tilde{x}) - \tilde{x}; \quad (11.2)$$

где  $f^\pi$  - ТЛП с входами  $\tilde{u}$  и  $\tilde{x}$ . Уравнение 11.2 схематично показано на рис. 11.3:



Пусть входное замкнутое системы является вид  $u$ , поступающее на сравнивающее устройство

$$\varepsilon = g - x - \text{ошибка}$$

Лингвистическое описание сравнивающего устройства имеет вид

$$\tilde{\varepsilon} = C^\pi(\tilde{x}, \tilde{y});$$

причем базовые переменные  $y^\pi$  и  $x^\pi$  принимают вид следующего значения

|                        |    |    |    |    |    |    |
|------------------------|----|----|----|----|----|----|
| $\tilde{y} \in T^*(y)$ |    |    |    |    |    |    |
| ОБ                     | ОС | ОН | НО | ПМ | ПС | ПБ |
| $\tilde{x} \in T^*(x)$ |    |    |    |    |    |    |
| ОБ                     | ОС | ОН | НО | ПМ | ПС | ПБ |

термы ошибки и термы выхода оборота и задания идентичны.

Определяем лингвистическое описание сравнивающего устройства, которое представим в виде таблицы 11.4.

Пусть информация изменения скорости ошибки определяется выражением

$$\dot{\varepsilon} = -\frac{dx}{dt} = -x;$$

при  $g = \text{const}$ , а значение лингвистической переменной  $\dot{\varepsilon}^\pi = \text{ОБ} \rightarrow \dot{x}^\pi = \text{ПБ}$ ; т.е термы  $\dot{\varepsilon}^\pi$  и  $\dot{x}^\pi$  противоположны по смыслу.

| $\tilde{x}$ | $\tilde{g}$ |    |    |    |    |    |    |
|-------------|-------------|----|----|----|----|----|----|
|             | ОБ          | ОС | ОМ | НО | ПМ | ПС | ПБ |
| ОБ          | НО          | ПМ | ПС | ПБ | ПБ | ПБ | ПБ |
| ОС          | ОМ          | НО | ПМ | ПС | ПБ | ПБ | ПБ |
| ОН          | ОС          | ОМ | НО | ПМ | ПС | ПБ | ПБ |
| НО          | ОБ          | ОС | ОМ | НО | ПМ | ПС | ПБ |
| ПМ          | ОБ          | ОБ | ОС | ОН | НО | ПМ | ПС |
| ПС          | ОБ          | ОБ | ОБ | ОС | ОМ | НО | ПМ |

|                       |    |    |    |    |    |    |    |
|-----------------------|----|----|----|----|----|----|----|
| ПБ                    | ОБ | ОБ | ОБ | ОБ | ОС | ОМ | НО |
| $\tilde{\varepsilon}$ |    |    |    |    |    |    |    |

Предположим, что желаемый переходной процесс замкнутой системы управления дом в виде табл. 11.5. лингвистических правил, которая может быть получена из градиента желаемой переходной характеристики.

$$\dot{\tilde{x}} = W^\pi(\tilde{x}, \tilde{y})$$

|                   |             |    |    |    |    |    |    |
|-------------------|-------------|----|----|----|----|----|----|
| $\tilde{g}$       | $\tilde{x}$ |    |    |    |    |    |    |
|                   | ОБ          | ОС | ОМ | НО | ПМ | ПС | ПБ |
| ОБ                | НО          | ОМ | ОС | ОБ |    |    |    |
| ОС                | ПН          | НО | ОМ | ОС |    |    |    |
| ОН                | ПС          | ПМ | НО | ОМ |    |    |    |
| НО                | ПБ          | ПС | ПМ | НО | ОМ | ОС | ОБ |
| ПМ                |             |    |    | ПН | НО | ОМ | ОС |
| ПС                |             |    |    | ПС | ПМ | НО | ОМ |
| ПБ                |             |    |    | ПБ | ПС | ПМ | НО |
| $\dot{\tilde{x}}$ |             |    |    |    |    |    |    |

Табл.11.5 можно представить в координатах ошибки и скорости ее изменения, или что тоже самое - в виде ТЛП желаемой замкнутой системы(см. Табл. 11.6.)

|                             |                       |    |    |    |    |    |    |
|-----------------------------|-----------------------|----|----|----|----|----|----|
| $\tilde{y}$                 | $\tilde{\varepsilon}$ |    |    |    |    |    |    |
|                             | ОБ                    | ОС | ОМ | НО | ПМ | ПС | ПБ |
| ОБ                          | ПС                    | ПС | ПМ | НО |    |    |    |
| ОС                          |                       | ПС | ПС | НО | ОМ |    |    |
| ОН                          |                       |    | ПМ | НО | ОМ | ОС |    |
| НО                          | ПБ                    | ПС | ПМ | НО | ОМ | ОС | ОБ |
| ПМ                          |                       | ПС | ПМ | НО | ОМ |    |    |
| ПС                          |                       |    | ПМ | НО | ОС | ОС |    |
| ПБ                          |                       |    |    | НО | ОМ | ОБ | НО |
| $\dot{\tilde{\varepsilon}}$ |                       |    |    |    |    |    |    |

где для установившегося состояния  $\tilde{\varepsilon} = \dot{\tilde{\varepsilon}} = \text{НО}$  составляем ТЛП разомкнутой системы при  $g^\pi = \text{НО}$ , т.е. объекта управления со сравнивающим устройством

$$\tilde{\varepsilon} = -\tilde{x}; \quad \dot{\tilde{\varepsilon}} = -\dot{\tilde{x}} \rightarrow F^\pi(\tilde{\varepsilon}, \tilde{u}).$$

|             |                       |    |    |    |    |    |    |
|-------------|-----------------------|----|----|----|----|----|----|
| $\tilde{u}$ | $\tilde{\varepsilon}$ |    |    |    |    |    |    |
|             | ОБ                    | ОС | ОМ | НО | ПМ | ПС | ПБ |
| ОБ          |                       |    |    | ПБ | ПС | ПМ | НО |
| ОС          |                       |    |    | ПС | ПМ | НО | ОМ |
| ОН          |                       |    |    | ПМ | НО | ОМ | ОС |
| НО          | ПБ                    | ПС | ПМ | НО | ОМ | ОС | ОБ |

|    |                       |    |    |    |  |  |  |
|----|-----------------------|----|----|----|--|--|--|
| ПМ | ПС                    | ПМ | НО | ОМ |  |  |  |
| ПС | ПМ                    | НО | ОМ | ОС |  |  |  |
| ПБ | НО                    | ОМ | ОС | ОБ |  |  |  |
|    | $\tilde{\varepsilon}$ |    |    |    |  |  |  |

Теперь задачу синтеза можно сформулировать следующим образом: требуется определить структуру ТЛП НР, которая переводит ОУ (табл. 11.6) из начального состояния  $\tilde{x}_0, \dot{\tilde{x}}_0$  и  $\tilde{\varepsilon}, \dot{\tilde{\varepsilon}}$  в конечное  $\tilde{x}_k, \dot{\tilde{x}}_k$  или  $\tilde{\varepsilon}_k, \dot{\tilde{\varepsilon}}_k$ , например, система из начального состояния  $(x_0^\pi = \text{НО}; \dot{x}_0^\pi = \text{НО}; g^\pi = \text{ПС} \quad (\varepsilon_0^\pi = \text{ПС}; \dot{\varepsilon}_0^\pi = \text{НО}; g^\pi = \text{ПС})$  переводится в конечное  $(x_k^\pi = \text{ПС}; \dot{x}_k^\pi = \text{НО}; g^\pi = \text{ПС} \quad (\varepsilon_k^\pi = \text{НО}; \dot{\varepsilon}_k^\pi = \text{НО}; g^\pi = \text{ПС})$  с желаемым качеством переходного процесса (табл. 6.6.) Исходя из рассмотренного лингвистический синтез НР можно осуществлять по следующей логической схеме

- 1) НОУ × НР → желаемая замкнутая система
- 2) НОУ × СУ × желаемая замкнутая система → нечеткий регулятор (НР) замкнутой системы управления, а) схема 2. Для лингвистического синтеза СУ.

Теперь нетрудно вывести лингвистические правила регулирования: объединив табл. 11.6. и 11.7 получим табл. 11.8 , а на ее основе

$$\tilde{u} = H^\pi(\tilde{\varepsilon}, \dot{\tilde{\varepsilon}})$$

т.е. лингвистически синтезирующего ТП НР (см. Табл. 11.9.)

Табл. 11.8.

|             |                             |    |    |    |    |    |    |
|-------------|-----------------------------|----|----|----|----|----|----|
| $\tilde{u}$ | $\tilde{\varepsilon}$       |    |    |    |    |    |    |
|             | ОБ                          | ОС | ОМ | НО | ПМ | ПС | ПБ |
| ОБ          |                             |    |    |    |    |    |    |
| ОС          |                             |    |    |    |    |    |    |
| ОН          |                             |    |    |    |    |    |    |
| НО          |                             |    |    |    |    |    |    |
| ПМ          |                             |    |    |    |    |    |    |
| ПС          |                             |    |    |    |    |    |    |
| ПБ          |                             |    |    |    |    |    |    |
|             | $\dot{\tilde{\varepsilon}}$ |    |    |    |    |    |    |

Табл. 11.9.

| $\dot{\tilde{e}}$ | $\tilde{e}$       |    |    |    |    |    |    |
|-------------------|-------------------|----|----|----|----|----|----|
|                   | ОБ                | ОС | ОМ | НО | ПМ | ПС | ПБ |
| ОБ                |                   |    |    |    |    |    |    |
| ОС                |                   |    |    |    |    |    |    |
| ОН                |                   |    |    |    |    |    |    |
| НО                |                   |    |    |    |    |    |    |
| ПМ                |                   |    |    |    |    |    |    |
| ПС                |                   |    |    |    |    |    |    |
| ПБ                |                   |    |    |    |    |    |    |
|                   | $\dot{\tilde{u}}$ |    |    |    |    |    |    |

Теперь при  $\tilde{e} = \text{ПС}$  и  $\dot{\tilde{e}} = \text{НО}$  из табл. 11.9 определяем  $u^\pi$ , которое есть ПС. Затем подставим в табл. 6.1.  $u^\pi = \text{ПС}$  и определим лингвистические термы  $x^\pi$  и  $\dot{x}^\pi$ . из этой табл. следует, что для ПС возможны 4 состояния:

$x^\pi = \text{НО}$ ,  $\dot{x}^\pi = \text{ПС}$ ;  $x^\pi = \text{ПМ}$ ,  $\dot{x}^\pi = \text{ПМ}$ ;  $x^\pi = \text{ПС}$ ,  $\dot{x}^\pi = \text{НО}$ ;  $x^\pi = \text{ПБ}$ ,  $\dot{x}^\pi = \text{ОМ}$ ;

Учитывая, что движение системы будет в сторону увеличения  $x$ , скорость  $\dot{x}$  будет положительной, следовательно 4 варианта отпадает.

Рассмотрим первые три варианта:

$x^\pi = \text{НО}$ ,  $\dot{x}^\pi = \text{ПС}$ ; определяем  $\varepsilon^\pi = \text{ПС}$ ,  $\tilde{\varepsilon}^\pi = \text{ОС}$ . При новых значениях  $e^\pi, \dot{e}^\pi$  из табл. 11.9. определим, что  $u^\pi = \text{НО}$ . Это значить, что управление остается неизменным, т.е.  $u^\pi = \text{ПС}$ ; при  $x^\pi = \text{ПМ}$ ,  $\dot{x}^\pi = \text{ПМ}$  получаем  $e^\pi = \text{ПМ}$ ;  $\dot{e}^\pi = \text{ОМ}$ . При этих значениях  $e^\pi$  и  $\dot{e}^\pi$  из табл. 11.4. определим, что  $u^\pi = \text{НО}$ ; при  $x^\pi = \text{НО}$ ,  $\dot{x}^\pi = \text{ПС}$   $x^\pi = \text{ПС}$ ,  $\dot{x}^\pi = \text{НО}$ ; получаем  $e^\pi = \text{НО}$ ;  $\dot{e}^\pi = \text{НО}$ ,  $u^\pi = \text{НО}$ .

## Тема 1.4: Нейронные сети в искусственных системах управления

### Лекция 15:

1. Биологический нейрон. Модель нейрона.
2. Персептроны. Структура и принцип действия персептрона.
3. Основные понятия и определения.

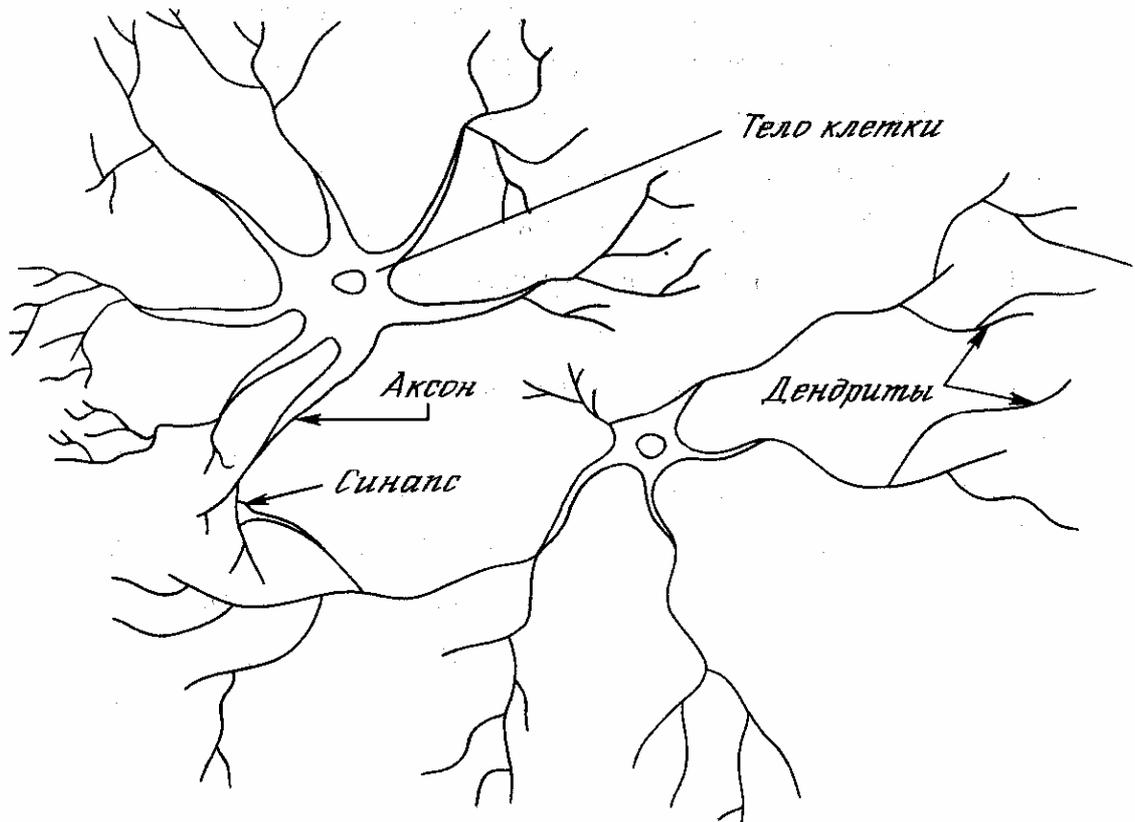
## 1. Биологический нейрон. Модель нейрона

Развитие искусственных нейронных сетей вдохновляется биологией. То есть, рассматривая сетевые конфигурации и алгоритмы, исследователи мыслят их в терминах организации мозговой деятельности. Но на этом аналогия может и закончиться. Наши знания о работе мозга столь ограничены, что мало бы нашлось руководящих ориентиров для тех, кто стал бы ему подражать. Поэтому разработчикам сетей приходится выходить за пределы современных биологических знаний в поисках структур, способных выполнять полезные функции. Во многих случаях это приводит к необходимости отказа от биологического правдоподобия, мозг становится просто метафорой, и создаются сети, невозможные в живой материи или требующие неправдоподобно больших допущений об анатомии и функционировании мозга.

Несмотря на то, что связь с биологией слаба и зачастую несущественна, искусственные нейронные сети продолжают сравниваться с мозгом. Их функционирование часто напоминает человеческое познание, поэтому трудно избежать этой аналогии. К сожалению, такие сравнения неплодотворны и создают неоправданные ожидания, неизбежно ведущие к разочарованию. Исследовательский энтузиазм, основанный на ложных надеждах, может испариться, столкнувшись с суровой действительностью, как это уже однажды было в шестидесятые годы, и многообещающая область снова придет в упадок, если не будет соблюдаться необходимая сдержанность.

Несмотря на сделанные предупреждения, полезно все же знать кое-что о нервной системе млекопитающих, так как она успешно решает задачи, к выполнению которых лишь стремятся искусственные системы.

Нервная система человека, построенная из элементов, называемых нейронами, имеет ошеломляющую сложность. Около  $10^{11}$  нейронов участвуют в примерно  $10^{15}$  передающих связях, имеющих длину метр и более. Каждый нейрон обладает многими качествами, общими с другими элементами тела, но его уникальной способностью является прием, обработка и передача электрохимических сигналов по нервным путям, которые образуют коммуникационную систему мозга.



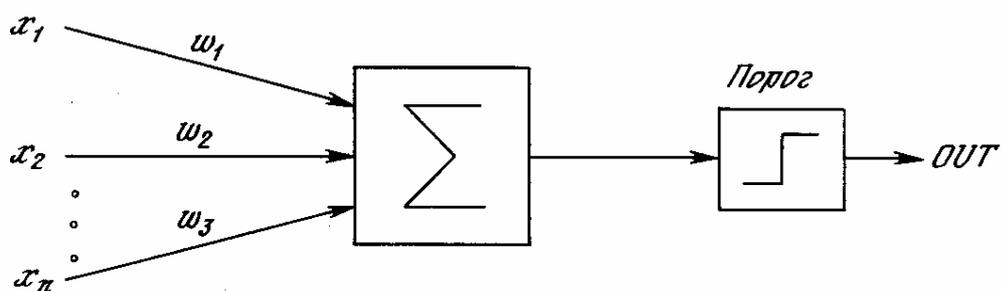
**Рис. 15.1. Биологический нейрон**

На рис. 15.1 показана структура пары типичных биологических нейронов. Дендриты идут от тела нервной клетки к другим нейронам, где они принимают сигналы в точках соединения, называемых синапсами. Принятые синапсом входные сигналы подводятся к телу нейрона. Здесь они суммируются, причем одни входы стремятся возбудить нейрон, другие – воспрепятствовать его возбуждению. Когда суммарное возбуждение в теле нейрона превышает некоторый порог, нейрон возбуждается, посылая по аксону сигнал другим нейронам. У этой основной, функциональной схемы много усложнений и исключений, тем не менее, большинство искусственных нейронных сетей моделируют лишь эти простые свойства.

## **2. Перцептроны. Структура и принцип действия перцептрона**

В качестве научного предмета искусственные нейронные сети впервые заявили о себе в 40-е годы. Стремясь воспроизвести функции человеческого мозга, исследователи создали простые аппаратные (а позже программные)

модели биологического нейрона и системы его соединений. Когда нейрофизиологи достигли более глубокого понимания нервной системы человека, эти ранние попытки стали восприниматься как весьма грубые аппроксимации. Тем не менее, на этом пути были достигнуты впечатляющие результаты, стимулировавшие дальнейшие исследования, приведшие к созданию более изощренных сетей.

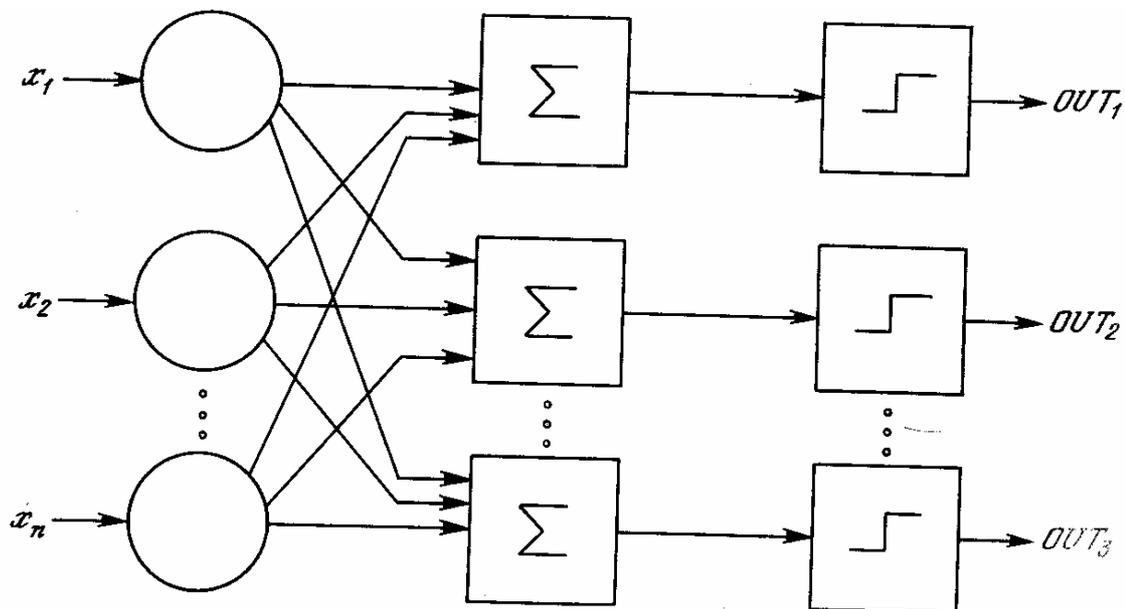


**Рис. 15.2. Персептронный нейрон**

Первое систематическое изучение искусственных нейронных сетей было предпринято Маккаллоком и Питтсом в 1943 г. Позднее они исследовали сетевые парадигмы для распознавания изображений, подвергаемых сдвигам и поворотам. Простая нейронная модель, показанная на рис. 15.2, использовалась в большей части их работы. Элемент  $\Sigma$  умножает каждый вход  $x$  на вес  $w$  и суммирует взвешенные входы. Если эта сумма больше заданного порогового значения, выход равен единице, в противном случае – нулю. Эти системы (и множество им подобных) получили название *персептронов*. Они состоят из одного слоя искусственных нейронов, соединенных с помощью весовых коэффициентов с множеством входов (см. рис. 15.3), хотя в принципе описываются и более сложные системы.

В 60-е годы персептроны вызвали большой интерес и оптимизм. Розенблатт доказал теорему об обучении персептронов. Уидроу дал ряд убедительных демонстраций систем персептронного типа, и исследователи во всем мире стремились изучить возможности этих систем. Однако потом оказалось, что персептроны не способны обучиться решению ряда простых задач. Минский строго проанализировал эту проблему и показал, что имеются жесткие ограничения на то, что могут выполнять однослойные персептроны, и, следовательно, на то, чему они могут обучаться. Так как в то время методы обучения многослойных сетей не были известны, исследователи перешли в

более многообещающие области, и исследования в области нейронных сетей пришли в упадок. Недавнее открытие методов обучения многослойных сетей в большей степени, чем какой-либо иной фактор, повлияло на возрождение интереса и исследовательских усилий.



**Рис. 2.2. Персептрон со многими выходами**

Несмотря на свои ограничения персептроны широко изучались (хотя не слишком широко использовались). Теория персептронов является основой для многих других типов искусственных нейронных сетей, и персептроны иллюстрируют важные принципы. В силу этих причин они являются логической исходной точкой для изучения искусственных нейронных сетей.

### **3. Основные понятия и определения**

*Представляемость* - способность персептрона (или другой сети) моделировать определенную функцию.

*Обучаемость* требует наличия систематической процедуры настройки весов сети для реализации этой функции.

*Линейная разделимость* ограничивает однослойные сети задачами классификации, в которых множества точек (соответствующих входным значениям) могут быть разделены геометрически.

## Лекция 16:

1. Процесс обучения персептрона.
2. Возможности многослойных персептронов.

### 1. Процесс обучения персептрона

Способность искусственных нейронных сетей обучаться является их наиболее важным свойством. Подобно биологическим системам, которые они моделируют, эти нейронные сети сами моделируют себя в результате попыток достичь лучшей модели поведения.

Используя критерий линейной разделимости, можно решить, способна ли однослойная нейронная сеть реализовывать требуемую функцию. Даже в том случае, когда ответ положительный, это принесет мало пользы, если у нас нет способа найти нужные значения для весов и порогов. Чтобы сеть представляла практическую ценность, нужен систематический метод (алгоритм) для вычисления этих значений. Розенблатт сделал это в своем алгоритме обучения персептрона вместе с доказательством того, что персептрон может быть обучен всему, что он может реализовывать.

Обучение может быть *с учителем* или *без него*.

Для обучения с учителем нужен «внешний» учитель, который оценивал бы поведение системы и управлял ее последующими модификациями.

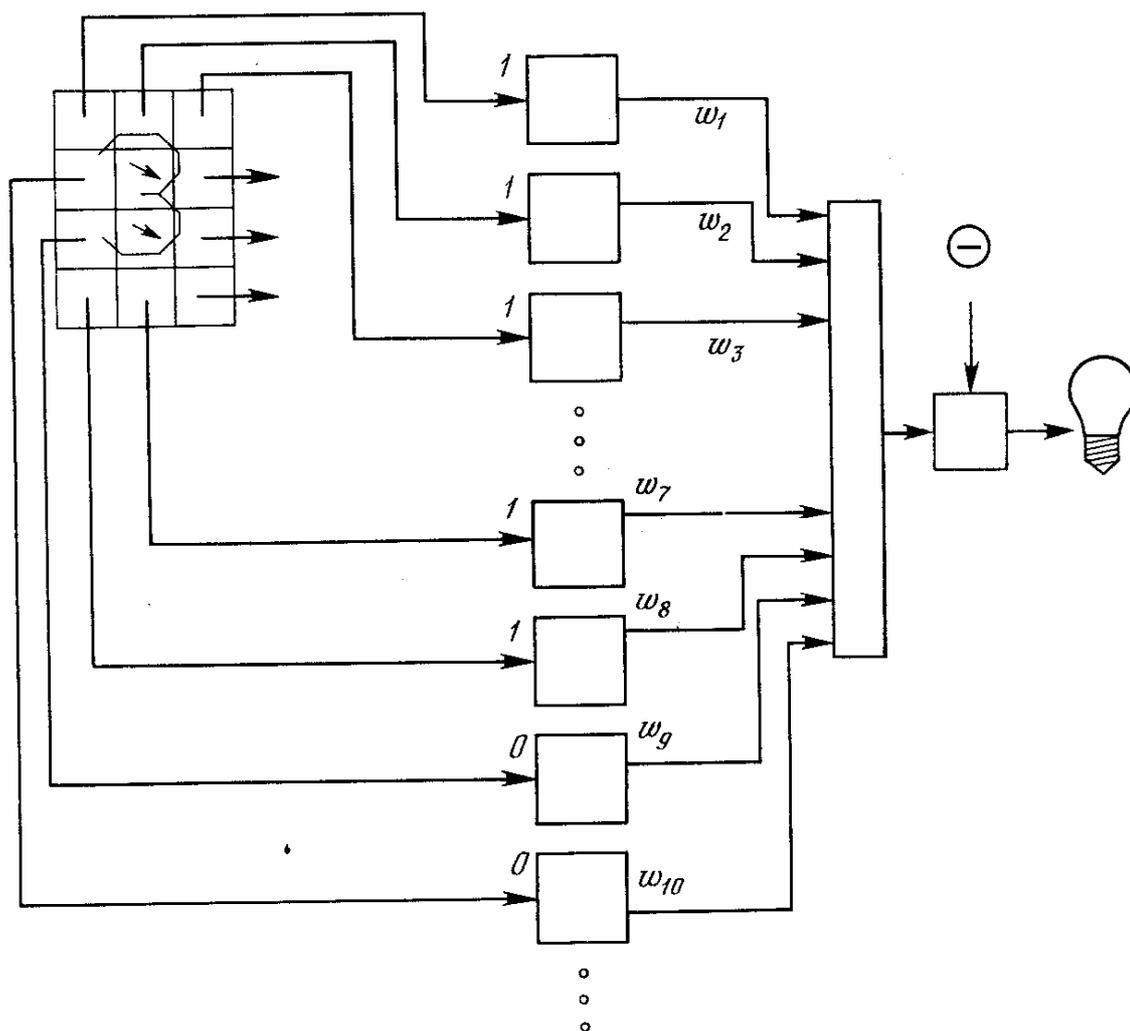
При обучении без учителя, рассматриваемого в последующих главах, сеть путем самоорганизации делает требуемые изменения. Обучение персептрона является обучением с учителем.

Алгоритм обучения персептрона может быть реализован на цифровом компьютере или другом электронном устройстве, и сеть становится в определенном смысле самоподстраивающейся. По этой причине процедуру подстройки весов обычно называют «обучением» и говорят, что сеть «обучается».

#### АЛГОРИТМ ОБУЧЕНИЯ ПЕРСЕПТРОНА

Персептрон обучают, подавая множество образов по одному на его вход и подстраивая веса до тех пор, пока для всех образов не будет достигнут требуемый выход. Допустим, что входные образы нанесены на

демонстрационные карты. Каждая карта разбита на квадраты и от каждого квадрата на персептрон подается вход. Если в квадрате имеется линия, то от него подается единица, в противном случае – ноль. Множество квадратов на карте задает, таким образом, множество нулей и единиц, которое и подается на входы персептрона. Цель состоит в том, чтобы научить персептрон включать индикатор при подаче на него множества входов, задающих нечетное число, и не включать в случае четного.



**Рис. 16.1. Персептронная система распознавания изображений**

На рис. 16.1 показана такая персептронная конфигурация. Допустим, что вектор  $\mathbf{X}$  является образом распознаваемой демонстрационной карты. Каждая компонента (квадрат)  $\mathbf{X} = (x_1, x_2, \dots, x_n)$  – умножается на соответствующую компоненту вектора весов  $\mathbf{W} = (w_1, w_2, \dots, w_n)$ . Эти произведения суммируются. Если сумма превышает порог  $\Theta$ , то выход нейрона  $Y$  равен единице (индикатор загорается), в противном случае он – ноль. Эта операция компактно

записывается в векторной форме как  $Y = XW$ , а после нее следует пороговая операция.

Для обучения сети образ  $X$  подается на вход и вычисляется выход  $Y$ . Если  $Y$  правилен, то ничего не меняется. Однако если выход неправилен, то веса, присоединенные к входам, усиливающим ошибочный результат, модифицируются, чтобы уменьшить ошибку.

Чтобы увидеть, как это осуществляется, допустим, что демонстрационная карта с цифрой 3 подана на вход и выход  $Y$  равен 1 (показывая нечетность). Так как это правильный ответ, то веса не изменяются. Если, однако, на вход подается карта с номером 4 и выход  $Y$  равен единице (нечетный), то веса, присоединенные к единичным входам, должны быть уменьшены, так как они стремятся дать неверный результат. Аналогично, если карта с номером 3 дает нулевой выход, то веса, присоединенные к единичным входам, должны быть увеличены, чтобы скорректировать ошибку.

Этот метод обучения может быть подытожен следующим образом:

2. Подать входной образ и вычислить  $Y$ .

2а. Если выход правильный, то перейти на шаг 1;

б. Если выход неправильный и равен нулю, то добавить все входы к соответствующим им весам; или

в. Если выход неправильный и равен единице, то вычесть каждый вход из соответствующего ему веса.

3. Перейти на шаг 1.

За конечное число шагов сеть научится разделять карты на четные и нечетные при условии, что множество цифр линейно делимо. Это значит, что для всех нечетных карт выход будет больше порога, а для всех четных – меньше.

## **2. Возможности многослойных персептронов**

## Лекция 17:

1. Искусственные нейронные сети. Структура.
2. Принципы обучения и тренировки нейронных сетей.
3. Алгоритм обратного распространения.

### 1. Искусственные нейронные сети. Структура

Искусственный нейрон имитирует в первом приближении свойства биологического нейрона. На вход искусственного нейрона поступает некоторое множество сигналов, каждый из которых является выходом другого нейрона. Каждый вход умножается на соответствующий вес, аналогичный синаптической силе, и все произведения суммируются, определяя уровень активации нейрона. На рис. 17.1 представлена модель, реализующая эту идею. Хотя сетевые парадигмы весьма разнообразны, в основе почти всех их лежит эта конфигурация. Здесь множество входных сигналов, обозначенных  $x_1, x_2, \dots, x_n$ , поступает на искусственный нейрон. Эти входные сигналы, в совокупности обозначаемые вектором  $\mathbf{X}$ , соответствуют сигналам, приходящим в синапсы биологического нейрона. Каждый сигнал умножается на соответствующий вес  $w_1, w_2, \dots, w_n$  и поступает на суммирующий блок, обозначенный  $\Sigma$ . Каждый вес соответствует «силе» одной биологической синаптической связи. (Множество весов в совокупности обозначается вектором  $\mathbf{W}$ .) Суммирующий блок, соответствующий телу биологического элемента, складывает взвешенные входы алгебраически, создавая выход, который мы будем называть NET. В векторных обозначениях это может быть компактно записано следующим образом:

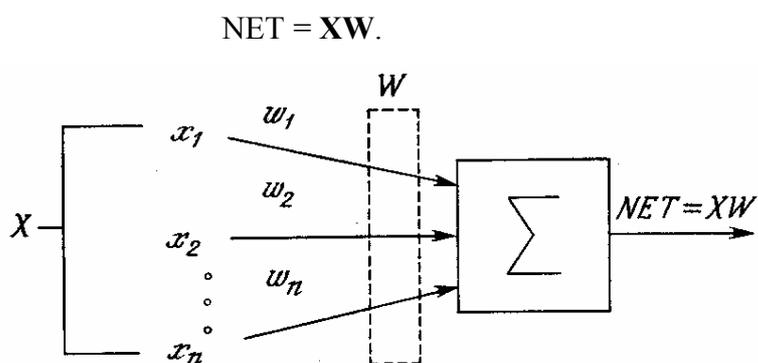


Рис. 17.1. Искусственный нейрон

## 2. Принципы обучения и тренировки нейронных сетей.

Искусственные нейронные сети обучаются самыми разнообразными методами. Однако, большинство методов обучения исходят из общих предпосылок и имеет много идентичных характеристик. Целью данного приложения является обзор некоторых фундаментальных алгоритмов, как с точки зрения их текущей применимости, так и с точки зрения их исторической важности. После ознакомления с этими фундаментальными алгоритмами другие, основанные на них, алгоритмы будут достаточно легки для понимания и новые разработки также могут быть лучше поняты и развиты.

### ОБУЧЕНИЕ С УЧИТЕЛЕМ И БЕЗ УЧИТЕЛЯ

Обучающие алгоритмы могут быть классифицированы как алгоритмы обучения с учителем и без учителя. В первом случае существует *учитель*, который предъявляет входные образы сети, сравнивает результирующие выходы с требуемыми, а затем настраивает веса сети таким образом, чтобы уменьшить различия. Трудно представить такой обучающий механизм в биологических системах; следовательно, хотя данный подход привел к большим успехам при решении прикладных задач, он отвергается исследователями, полагающими, что искусственные нейронные сети обязательно должны использовать те же механизмы, что и человеческий мозг.

Во втором случае обучение проводится *без учителя*, при предъявлении входных образов сеть самоорганизуется посредством настройки своих весов согласно определенному алгоритму. Вследствие отсутствия указания требуемого выхода в процессе обучения результаты непредсказуемы с точки зрения определения возбуждающих образов для конкретных нейронов. При этом, однако, сеть организуется в форме, отражающей существенные характеристики обучающего набора. Например, входные образы могут быть классифицированы согласно степени их сходства так, что образы одного класса активизируют один и тот же выходной нейрон.

### МЕТОД ОБУЧЕНИЯ ХЭББА

Работа [2] обеспечила основу для большинства алгоритмов обучения, которые были разработаны после ее выхода. В предшествующих этой работе

трудах в общем виде определялось, что обучение в биологических системах происходит посредством некоторых физических изменений в нейронах, однако отсутствовали идеи о том, каким образом это в действительности может иметь место. Основываясь на физиологических и психологических исследованиях, Хэбб в [2] интуитивно выдвинул гипотезу о том, каким образом может обучаться набор биологических нейронов. Его теория предполагает только локальное взаимодействие между нейронами при отсутствии глобального учителя; следовательно, обучение является неуправляемым. Несмотря на то, что его работа не включает математического анализа, идеи, изложенные в ней, настолько ясны и непринужденны, что получили статус универсальных допущений. Его книга стала классической и широко изучается специалистами, имеющими серьезный интерес в этой области.

### Алгоритм обучения Хэбба

По существу Хэбб предположил, что синаптическое соединение двух нейронов усиливается, если оба эти нейрона возбуждены. Это можно представить как усиление синапса в соответствии с корреляцией уровней возбужденных нейронов, соединяемых данным синапсом. По этой причине алгоритм обучения Хэбба иногда называется корреляционным алгоритмом.

Идея алгоритма выражается следующим равенством:

$$w_{ij}(t+1) = w_{ij}(t) + \text{NET}_i \text{NET}_j,$$

где  $w_{ij}(t)$  – сила синапса от нейрона  $i$  к нейрону  $j$  в момент времени  $t$ ;  $\text{NET}_i$  – уровень возбуждения пресинаптического нейрона;  $\text{NET}_j$  – уровень возбуждения постсинаптического нейрона.

Концепция Хэбба отвечает на сложный вопрос, каким образом обучение может проводиться без учителя. В методе Хэбба обучение является исключительно локальным явлением, охватывающим только два нейрона и соединяющий их синапс; не требуется глобальной системы обратной связи для развития нейронных образований.

Последующее использование метода Хэбба для обучения нейронных сетей привело к большим успехам, но наряду с этим показало ограниченность метода; некоторые образы просто не могут использоваться для обучения этим методом. В результате появилось большое количество расширений и

нововведений, большинство из которых в значительной степени основано на работе Хэбба.

## Метод сигнального обучения Хэбба

Как мы видели, выход NET простого искусственного нейрона является взвешенной суммой его входов. Это может быть выражено следующим образом:

$$NET_j = \sum_i OUT_i w_{ij}$$

где  $NET_j$  – выход NET нейрона  $j$ ;  $OUT_i$  – выход нейрона  $i$ ;  $w_{ij}$  – вес связи нейрона  $i$  с нейроном  $j$ .

Можно показать, что в этом случае линейная многослойная сеть не является более мощной, чем однослойная сеть; рассматриваемые возможности сети могут быть улучшены только введением нелинейности в передаточную функцию нейрона. Говорят, что сеть, использующая сигмоидальную функцию активации и метод обучения Хэбба, обучается по сигнальному методу Хэбба. В этом случае уравнение Хэбба модифицируется следующим образом:

$$OUT_i = \frac{1}{1 + \exp(-NET_i)} = F(NET_i)$$

$$w_{ij}(t+1) = w_{ij}(t) + OUT_i OUT_j$$

где  $w_{ij}(t)$  – сила синапса от нейрона  $i$  к нейрону  $j$  в момент времени  $t$ ;  $OUT_i$  – выходной уровень пресинаптического нейрона равный  $F(NET_i)$ ;  $OUT_j$  – выходной уровень постсинаптического нейрона равный  $F(NET)$ .

## Метод дифференциального обучения Хэбба

Метод сигнального обучения Хэбба предполагает вычисление свертки предыдущих изменений выходов для определения изменения весов. Настоящий метод, называемый методом дифференциального обучения Хэбба, использует следующее равенство:

$$w_{ij}(t+1) = w_{ij}(t) + [OUT_i(t) - OUT_i(t-1)][OUT_j(t) - OUT_j(t-1)],$$

где  $w_{ij}(t)$  – сила синапса от нейрона  $i$  к нейрону  $j$  в момент времени  $t$ ;  $OUT_i(t)$  – выходной уровень пресинаптического нейрона в момент времени  $t$ ;  $OUT_j(t)$  – выходной уровень постсинаптического нейрона в момент времени  $t$ .

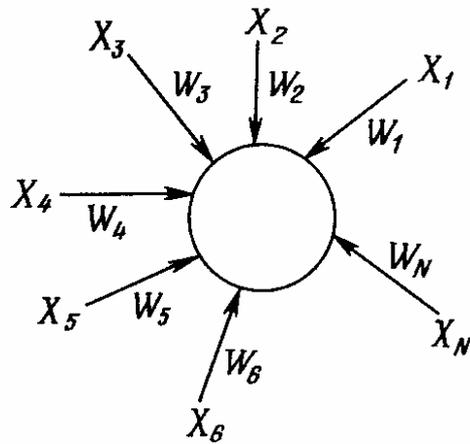


Рис. 17.2. Сеть «Инстар» Гроссберга

### 3. Алгоритм обратного распространения

Обратное распространение – это систематический метод для обучения многослойных искусственных нейронных сетей. Он имеет солидное математическое обоснование. Несмотря на некоторые ограничения, процедура обратного распространения сильно расширила область проблем, в которых могут быть использованы искусственные нейронные сети, и убедительно продемонстрировала свою мощь.

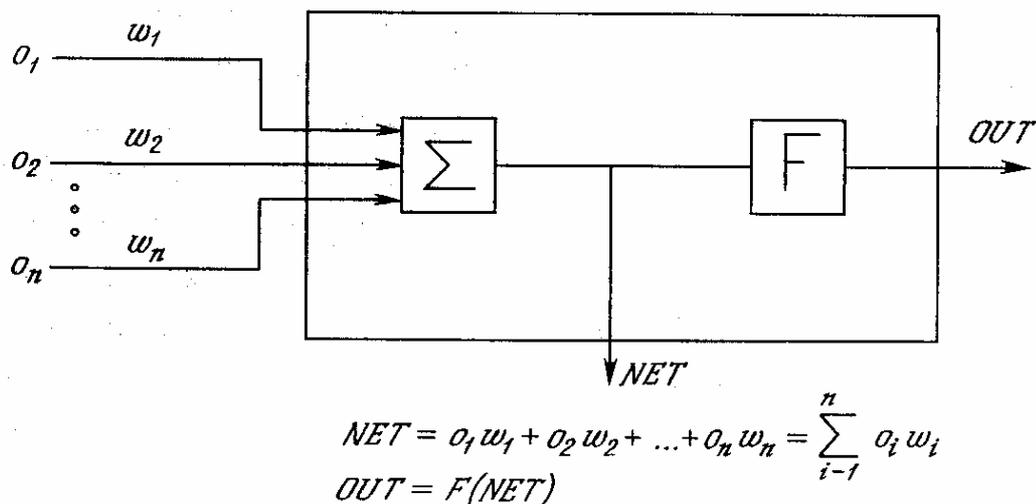


Рис. 17.3. Искусственный нейрон с активационной функцией

На рис. 17.3 показан нейрон, используемый в качестве основного строительного блока в сетях обратного распространения. Подается множество входов, идущих либо извне, либо от предшествующего слоя. Каждый из них умножается на вес, и произведения суммируются. Эта сумма, обозначаемая NET, должна быть вычислена для каждого нейрона сети. После того, как величина NET вычислена, она модифицируется с помощью активационной функции и получается сигнал OUT.

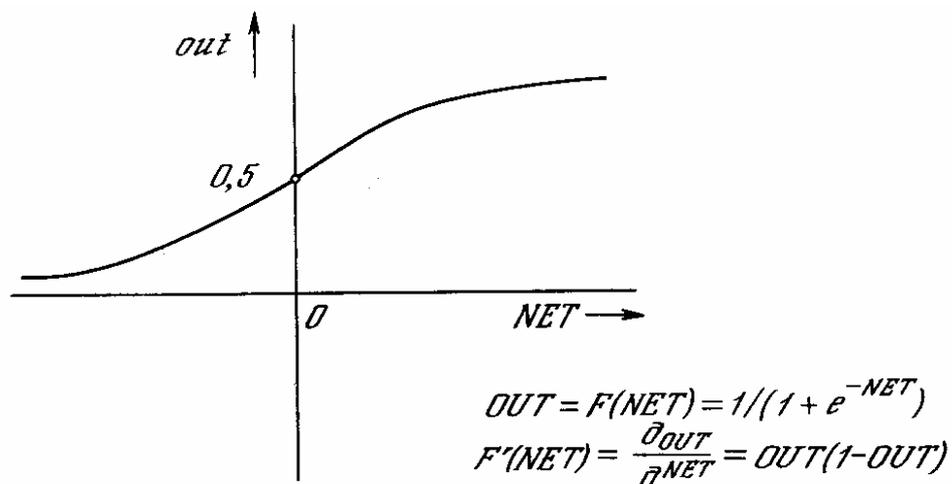


Рис. 17.4. Сигмоидальная активационная функция.

На рис. 17.4 показана активационная функция, обычно используемая для обратного распространения.

$$OUT = \frac{1}{1 + e^{-NET}} \quad (17.1)$$

Эта функция, называемая *сигмоидом*, весьма удобна, так как имеет простую производную, что используется при реализации алгоритма обратного распространения.

$$\frac{\partial OUT}{\partial NET} = OUT(1 - OUT) \quad (17.2)$$

Сигмоид, который иногда называется также *логистической*, или *сжимающей функцией*, сужает диапазон изменения NET так, что значение OUT лежит между нулем и единицей. Как указывалось выше, многослойные нейронные сети обладают большей представляющей мощностью, чем

однослойные, только в случае присутствия нелинейности. Сжимающая функция обеспечивает требуемую нелинейность.

В действительности имеется множество функций, которые могли бы быть использованы. Для алгоритма обратного распространения требуется лишь, чтобы функция была всюду дифференцируема. Сигмоид удовлетворяет этому требованию. Его дополнительное преимущество состоит в автоматическом контроле усиления. Для слабых сигналов (величина NET близка к нулю) кривая вход-выход имеет сильный наклон, дающий большое усиление. Когда величина сигнала становится больше, усиление падает. Таким образом, большие сигналы воспринимаются сетью без насыщения, а слабые сигналы проходят по сети без чрезмерного ослабления.

### **Многослойная сеть.**

На рис. 17.5 изображена многослойная сеть, которая может обучаться с помощью процедуры обратного распространения. (Для ясности рисунок упрощен.) Первый слой нейронов (соединенный с входами) служит лишь в качестве распределительных точек, суммирования входов здесь не производится. Входной сигнал просто проходит через них к весам на их выходах. А каждый нейрон последующих слоев выдает сигналы NET и OUT, как описано выше.

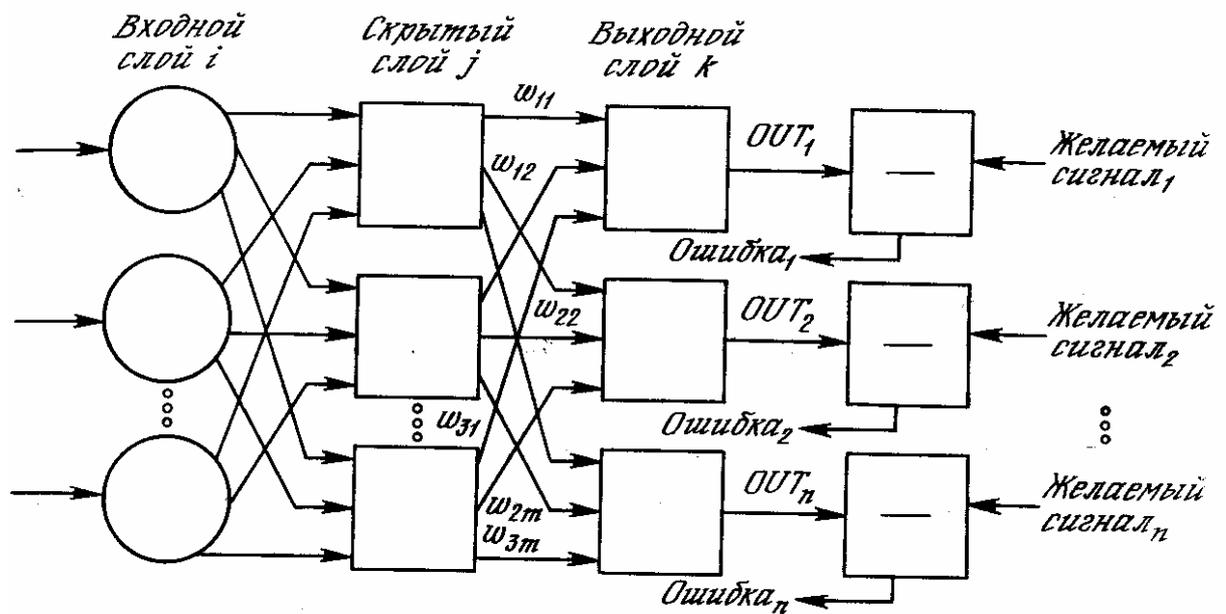


Рис. 17.5. Двухслойная сеть обратного распространения ( $\epsilon$  – желаемый сигнал).

В литературе нет единообразия относительно того, как считать число слоев в таких сетях. Одни авторы используют число слоев нейронов (включая несуммирующий входной слой), другие – число слоев весов. Так как последнее определение функционально описательное, то оно будет использоваться на протяжении книги. Согласно этому определению, сеть на рис. 3.3 рассматривается как двухслойная. Нейрон объединен с множеством весов, присоединенных к его входу. Таким образом, веса первого слоя оканчиваются на нейронах первого слоя. Вход распределительного слоя считается нулевым слоем.

Процедура обратного распространения применима к сетям с любым числом слоев. Однако для того, чтобы продемонстрировать алгоритм, достаточно двух слоев. Сейчас будут рассматриваться лишь сети прямого действия, хотя обратное распространение применимо и к сетям с обратными связями. Эти случаи будут рассмотрены в данной главе позднее.

## ОБЗОР ОБУЧЕНИЯ

Целью обучения сети является такая подстройка ее весов, чтобы приложение некоторого множества входов приводило к требуемому множеству выходов. Для краткости эти множества входов и выходов будут называться *векторами*. При обучении предполагается, что для каждого входного вектора

существует парный ему целевой вектор, задающий требуемый выход. Вместе они называются *обучающей парой*. Как правило, сеть обучается на многих парах. Например, входная часть обучающей пары может состоять из набора нулей и единиц, представляющего двоичный образ некоторой буквы алфавита. На рис. 17.6 показано множество входов для буквы «А», нанесенной на сетке. Если через квадрат проходит линия, то соответствующий нейронный вход равен единице, в противном случае он равен нулю. Выход может быть числом, представляющим букву «А», или другим набором из нулей и единиц, который может быть использован для получения выходного образа. При необходимости распознавать с помощью сети все буквы алфавита, потребовалось бы 26 обучающих пар. Такая группа обучающих пар называется *обучающим множеством*.

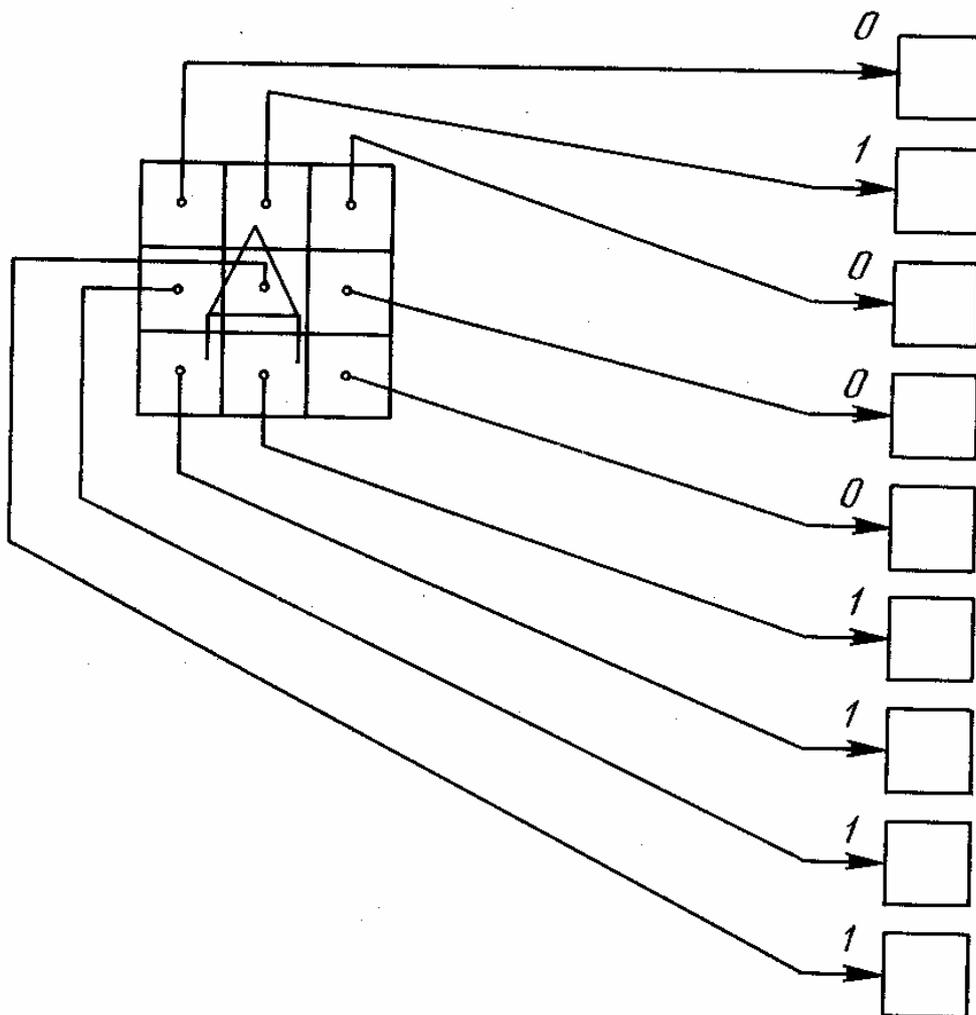


Рис. 17.6 Распознавание изображений

Перед началом обучения всем весам должны быть присвоены небольшие начальные значения, выбранные случайным образом. Это гарантирует, что в сети не произойдет насыщения большими значениями весов, и предотвращает ряд других патологических случаев. Например, если всем весам придать одинаковые начальные значения, а для требуемого функционирования нужны неравные значения, то сеть не сможет обучиться.

Обучение сети обратного распространения требует выполнения следующих операций:

1. Выбрать очередную обучающую пару из обучающего множества; подать входной вектор на вход сети.
2. Вычислить выход сети.
3. Вычислить разность между выходом сети и требуемым выходом (целевым вектором обучающей пары).
4. Подкорректировать веса сети так, чтобы минимизировать ошибку.
5. Повторять шаги с 1 по 4 для каждого вектора обучающего множества до тех пор, пока ошибка на всем множестве не достигнет приемлемого уровня.

Операции, выполняемые шагами 1 и 2, сходны с теми, которые выполняются при функционировании уже обученной сети, т. е. подается входной вектор и вычисляется получающийся выход. Вычисления выполняются послойно. На рис. 17.4 сначала вычисляются выходы нейронов слоя  $j$ , затем они используются в качестве входов слоя  $k$ , вычисляются выходы нейронов слоя  $k$ , которые и образуют выходной вектор сети.

На шаге 3 каждый из выходов сети, которые на рис. 17.4 обозначены OUT, вычитается из соответствующей компоненты целевого вектора, чтобы получить ошибку. Эта ошибка используется на шаге 4 для коррекции весов сети, причем знак и величина изменений весов определяются алгоритмом обучения (см. ниже).

После достаточного числа повторений этих четырех шагов разность между действительными выходами и целевыми выходами должна уменьшиться до приемлемой величины, при этом говорят, что сеть обучилась. Теперь сеть используется для распознавания и веса не изменяются.

На шаги 1 и 2 можно смотреть как на «проход вперед», так как сигнал распространяется по сети от входа к выходу. Шаги 3, 4 составляют «обратный

проход», здесь вычисляемый сигнал ошибки распространяется обратно по сети и используется для подстройки весов.

## **Лекция 18:**

- 1. Алгоритм встречного распространения**
- 2. Сетки Хопфилда. Характеристики специальных нейронных сетей**
- 3. Сетки многошаровых распознавательных систем типа MLP. Сетки радиальных базисных функций типа RBF.**

### **1. Алгоритм встречного распространения**

Возможности сети встречного распространения превосходят возможности однослойных сетей. Время же обучения, по сравнению с обратным распространением может уменьшаться в сто раз. Встречное распространение не столь общо, как обратное распространение, но оно может давать решение в тех приложениях, где долгая обучающая процедура невозможна. Будет показано, что помимо преодоления ограничений других сетей встречное распространение обладает собственными интересными и полезными свойствами.

Во встречном распространении объединены два хорошо известных алгоритма: самоорганизующаяся карта Кохонена и звезда Гроссберга. Их объединение ведет к свойствам, которых нет ни у одного из них в отдельности. Методы, которые подобно встречному распространению, объединяют различные сетевые парадигмы как строительные блоки, могут привести к сетям, более близким к мозгу по архитектуре, чем любые другие однородные структуры. Похоже, что в мозгу именно каскадные соединения модулей различной специализации позволяют выполнять требуемые вычисления.

Сеть встречного распространения функционирует подобно столу справок, способному к обобщению. В процессе обучения входные векторы ассоциируются с соответствующими выходными векторами. Эти векторы могут быть двоичными, состоящими из нулей и единиц, или непрерывными. Когда сеть обучена, приложение входного вектора приводит к требуемому выходному

вектору. Обобщающая способность сети позволяет получать правильный выход даже при приложении входного вектора, который является неполным или слегка неверным. Это позволяет использовать данную сеть для распознавания образов, восстановления образов и усиления сигналов.

### СТРУКТУРА СЕТИ

На рис. 18.1 показана упрощенная версия прямого действия сети встречного распространения. На нем иллюстрируются функциональные свойства этой парадигмы. Полная двунаправленная сеть основана на тех же принципах, она обсуждается в этой главе позднее.

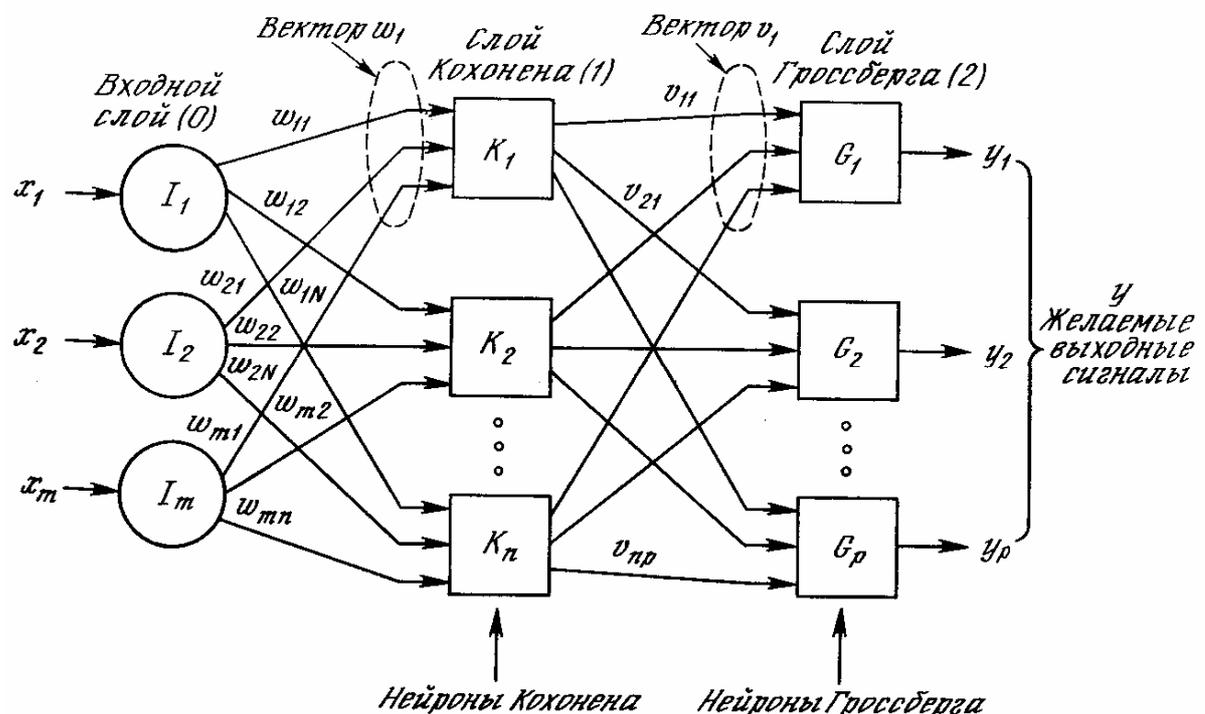


Рис. 18.1. Сеть с встречным распознаванием без обратных связей

Нейроны слоя 0 (показанные кружками) служат лишь точками разветвления и не выполняют вычислений. Каждый нейрон слоя 0 соединен с каждым нейроном слоя 1 (называемого слоем Кохонена) отдельным весом  $w_{mn}$ . Эти веса в целом рассматриваются как матрица весов  $\mathbf{W}$ . Аналогично, каждый нейрон в слое Кохонена (слое 1) соединен с каждым нейроном в слое Гроссберга (слое 2) весом  $v_{np}$ . Эти веса образуют матрицу весов  $\mathbf{V}$ . Все это весьма напоминает другие сети, встречавшиеся в предыдущих главах, различие, однако, состоит в операциях, выполняемых нейронами Кохонена и Гроссберга.

Как и многие другие сети, встречное распространение функционирует в двух режимах: в нормальном режиме, при котором принимается входной вектор  $\mathbf{X}$  и выдается выходной вектор  $\mathbf{Y}$ , и в режиме обучения, при котором подается входной вектор и веса корректируются, чтобы дать требуемый выходной вектор.

## НОРМАЛЬНОЕ ФУНКЦИОНИРОВАНИЕ

### *Слой Кохоненна*

В своей простейшей форме слой Кохонена функционирует в духе «победитель забирает все», т. е. для данного входного вектора один и только один нейрон Кохонена выдает на выходе логическую единицу, все остальные выдают ноль. Нейроны Кохонена можно воспринимать как набор электрических лампочек, так что для любого входного вектора загорается одна из них.

Ассоциированное с каждым нейроном Кохонена множество весов соединяет его с каждым входом. Например, на рис. 18.1 нейрон Кохонена  $K_1$  имеет веса  $w_{11}$ ,  $w_{21}$ , ...,  $w_{m1}$ , составляющие весовой вектор  $\mathbf{W}_1$ . Они соединяются через входной слой с входными сигналами  $x_1, x_2, \dots, x_m$ , составляющими входной вектор  $\mathbf{X}$ . Подобно нейронам большинства сетей выход NET каждого нейрона Кохонена является просто суммой взвешенных входов. Это может быть выражено следующим образом:

$$\text{NET}_j = w_{1j}x_1 + w_{2j}x_2 + \dots + w_{mj}x_m \quad (18.1)$$

где  $\text{NET}_j$  – это выход NET нейрона Кохонена  $j$ ,

$$\text{NET}_j = \sum_i x_i w_{ij} \quad (18.2)$$

или в векторной записи

$$\mathbf{N} = \mathbf{XW}, \quad (18.3)$$

где  $\mathbf{N}$  – вектор выходов NET слоя Кохонена.

Нейрон Кохонена с максимальным значением NET является «победителем». Его выход равен единице, у остальных он равен нулю.

### Слой Гроссберга

Слой Гроссберга функционирует в сходной манере. Его выход NET является взвешенной суммой выходов  $k_1, k_2, \dots, k_n$  слоя Кохонена, образующих вектор  $\mathbf{K}$ . Вектор соединяющих весов, обозначенный через  $\mathbf{V}$ , состоит из весов  $v_{11}, v_{21}, \dots, v_{np}$ . Тогда выход NET каждого нейрона Гроссберга есть

$$\text{NET}_j = \sum_i k_i w_{ij},$$

(18.4)

где  $\text{NET}_j$  – выход  $j$ -го нейрона Гроссберга, или в векторной форме

$$\mathbf{Y} = \mathbf{KV},$$

(18.5)

где  $\mathbf{Y}$  – выходной вектор слоя Гроссберга,  $\mathbf{K}$  – выходной вектор слоя Кохонена,  $\mathbf{V}$  – матрица весов слоя Гроссберга.

Если слой Кохонена функционирует таким образом, что лишь у одного нейрона величина NET равна единице, а у остальных равна нулю, то лишь один элемент вектора  $\mathbf{K}$  отличен от нуля, и вычисления очень просты. Фактически каждый нейрон слоя Гроссберга лишь выдает величину веса, который связывает этот нейрон с единственным ненулевым нейроном Кохонена.

### СЕТЬ ВСТРЕЧНОГО РАСПРОСТРАНЕНИЯ ПОЛНОСТЬЮ

На рис. 18.2 показана сеть встречного распространения целиком. В режиме нормального функционирования предъявляются входные векторы  $\mathbf{X}$  и  $\mathbf{Y}$ , и обученная сеть дает на выходе векторы  $\mathbf{X}'$  и  $\mathbf{Y}'$ , являющиеся аппроксимациями соответственно для  $\mathbf{X}$  и  $\mathbf{Y}$ . Векторы  $\mathbf{X}$  и  $\mathbf{Y}$  предполагаются здесь нормализованными единичными векторами, следовательно, порождаемые на выходе векторы также будут иметь тенденцию быть нормализованными.

В процессе обучения векторы  $\mathbf{X}$  и  $\mathbf{Y}$  подаются одновременно и как входные векторы сети, и как желаемые выходные сигналы. Вектор  $\mathbf{X}$  используется для обучения выходов  $\mathbf{X}'$ , а вектор  $\mathbf{Y}$  – для обучения выходов  $\mathbf{Y}'$  слоя Гроссберга. Сеть встречного распространения целиком обучается с использованием того же самого метода, который описывался для сети прямого действия. Нейроны Кохонена принимают входные сигналы как от векторов  $\mathbf{X}$ , так и от векторов  $\mathbf{Y}$ .

Но это неотличимо от ситуации, когда имеется один большой вектор, составленный из векторов  $X$  и  $Y$ , и не влияет на алгоритм обучения.

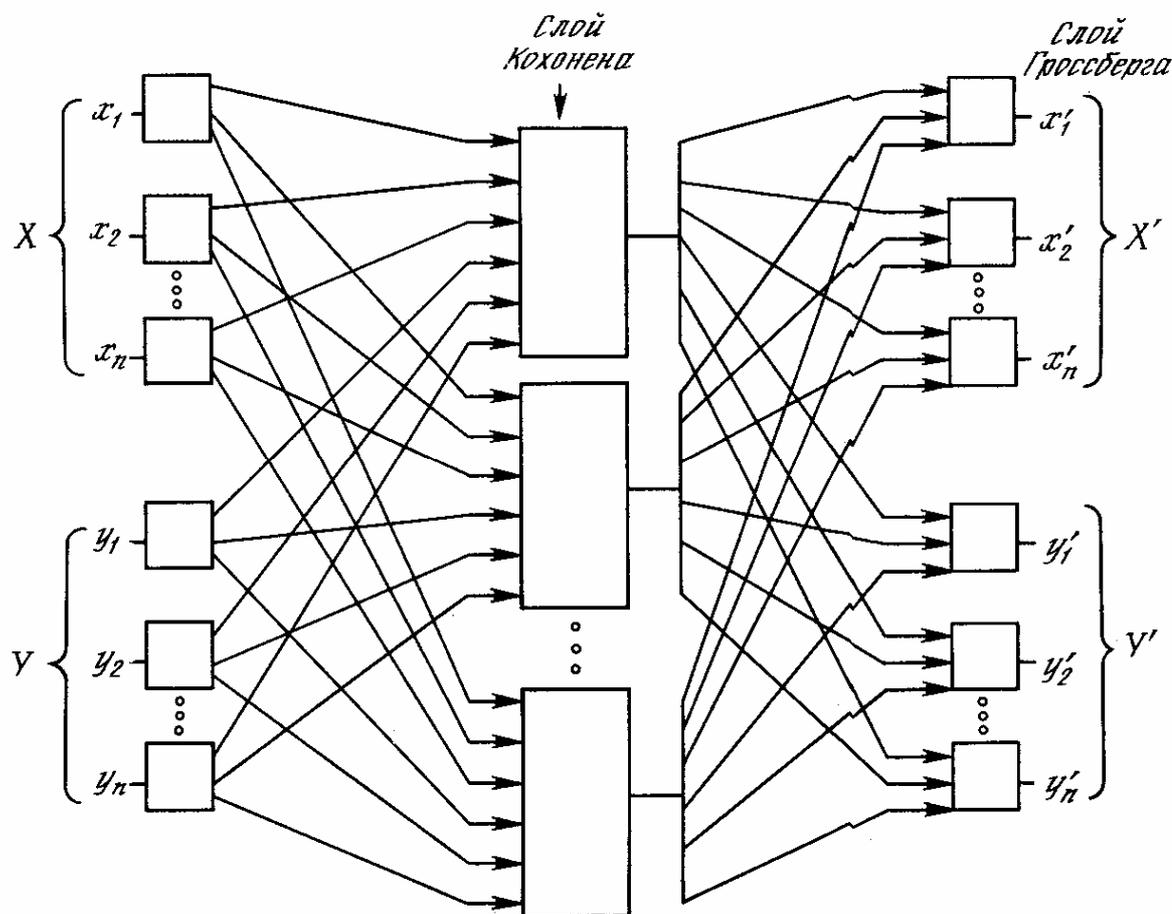


Рис. 18.2. Полная сеть встречного распространения

В качестве результирующего получается единичное отображение, при котором предъявление пары входных векторов порождает их копии на выходе. Это не представляется особенно интересным, если не заметить, что предъявление только вектора  $X$  (с вектором  $Y$ , равным нулю) порождает как выходы  $X'$ , так и выходы  $Y'$ . Если  $F$  – функция, отображающая  $X$  в  $Y'$ , то сеть аппроксимирует ее. Также, если  $F$  обратима, то предъявление только вектора  $Y$  (приравнявая  $X$  нулю) порождает  $X'$ . Уникальная способность порождать функцию и обратную к ней делает сеть встречного распространения полезной в ряде приложений.

Рис. 18.2 в отличие от первоначальной конфигурации [5] не демонстрирует противоток в сети, по которому она получила свое название.

Такая форма выбрана потому, что она также иллюстрирует сеть без обратных связей и позволяет обобщить понятия, развитые в предыдущих главах.

## **2. Сети Хопфилда. Характеристики специальных нейронных сетей**

Рассмотренные выше сети не имели обратных связей, т. е. связей, идущих от выходов сетей и их входам. Отсутствие обратной связи гарантирует безусловную устойчивость сетей. Они не могут войти в режим, когда выход беспрерывно блуждает от состояния к состоянию и не пригоден к использованию. Но это весьма желательное свойство достигается не бесплатно, сети без обратных связей обладают более ограниченными возможностями по сравнению с сетями с обратными связями.

Так как сети с обратными связями имеют пути, передающие сигналы от выходов к входам, то отклик таких сетей является динамическим, т. е. после приложения нового входа вычисляется выход и, передаваясь по сети обратной связи, модифицирует вход. Затем выход повторно вычисляется, и процесс повторяется снова и снова. Для устойчивой сети последовательные итерации приводят к все меньшим изменениям выхода, пока в конце концов выход не становится постоянным. Для многих сетей процесс никогда не заканчивается, такие сети называют неустойчивыми. Неустойчивые сети обладают интересными свойствами и изучались в качестве примера хаотических систем. Однако такой большой предмет, как хаос, находится за пределами этой книги. Вместо этого мы сконцентрируем внимание на устойчивых сетях, т. е. на тех, которые в конце концов дают постоянный выход.

Проблема устойчивости ставила в тупик первых исследователей. Никто не был в состоянии предсказать, какие из сетей будут устойчивыми, а какие будут находиться в постоянном изменении. Более того, проблема представлялась столь трудной, что многие исследователи были настроены пессимистически относительно возможности ее решения. К счастью, в работе [2] была получена теорема, описавшая подмножество сетей с обратными связями, выходы которых в конце концов достигают устойчивого состояния. Это замечательное достижение открыло дорогу дальнейшим исследованиям и сегодня многие ученые занимаются исследованием сложного поведения и возможностей этих систем.

Дж. Хопфилд сделал важный вклад как в теорию, так и в применение систем с обратными связями. Поэтому некоторые из конфигураций известны как сети Хопфилда. Из обзора литературы видно, что исследованием этих и сходных систем занимались многие. Например, в работе [4] изучались общие свойства сетей, аналогичных многим, рассмотренным здесь. Работы, цитируемые в списке литературы в конце главы, не направлены на то, чтобы дать исчерпывающую библиографию по системам с обратными связями. Скорее они являются лишь доступными источниками, которые могут служить для объяснения, расширения и обобщения содержания этой книги.

### КОНФИГУРАЦИИ СЕТЕЙ С ОБРАТНЫМИ СВЯЗЯМИ

На рис. 18.3 показана сеть с обратными связями, состоящая из двух слоев. Способ представления несколько отличается от использованного в работе Хопфилда и других, но эквивалентен им с функциональной точки зрения, а также хорошо связан с сетями, рассмотренными в предыдущих главах. Нулевой слой, как и на предыдущих рисунках, не выполняет вычислительной функции, а лишь распределяет выходы сети обратно на входы. Каждый нейрон первого слоя вычисляет взвешенную сумму своих входов, давая сигнал NET, который затем с помощью нелинейной функции  $F$  преобразуется в сигнал OUT. Эти операции сходны с нейронами других сетей.

#### Бинарные системы

В первой работе Хопфилда функция  $F$  была просто пороговой функцией. Выход такого нейрона равен единице, если взвешенная сумма выходов с других нейронов больше порога  $T_j$ , в противном случае она равна нулю. Он вычисляется следующим образом:

$$NET_j = \sum_{i \neq j} w_{ij} OUT_i + IN_j,$$

(18.6)

$$OUT_j = 1, \text{ если } NET_j > T_j,$$

$$OUT_j = 0, \text{ если } NET_j < T_j,$$

$$OUT_j \text{ не изменяется, если } NET_j = T_j,$$

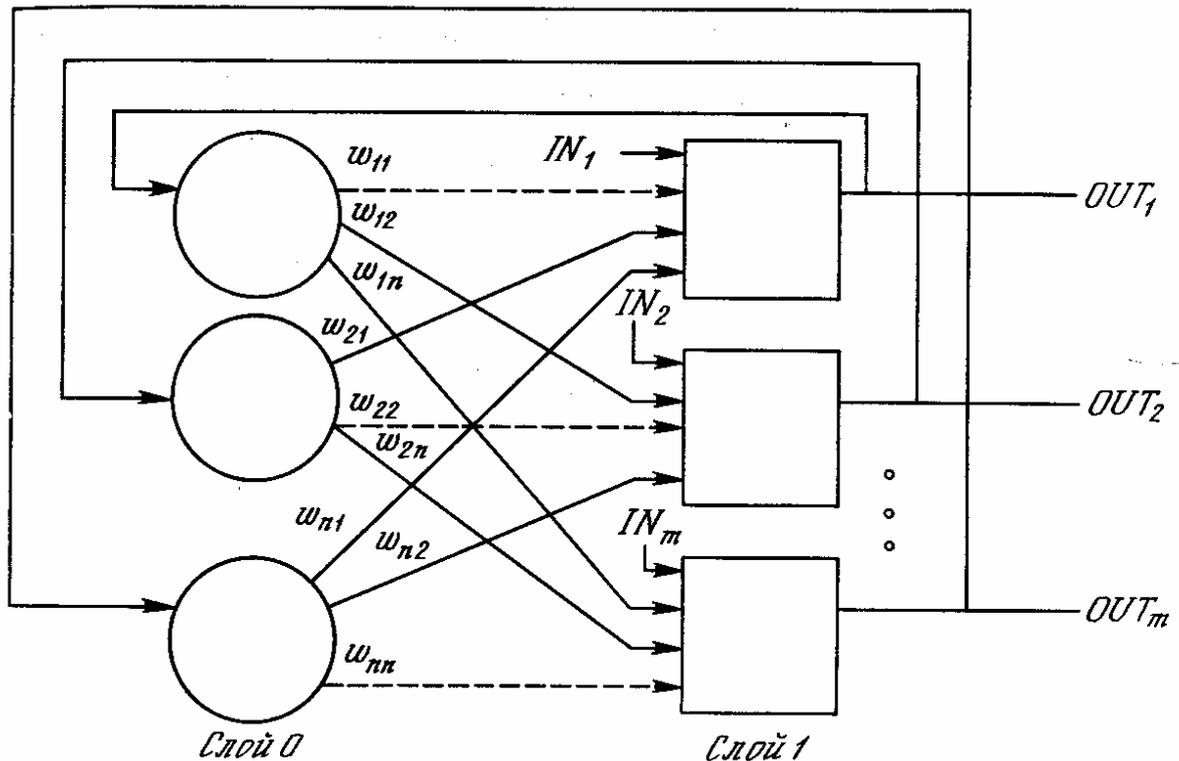


Рис. 18.3. Однослойная сеть с обратными связями.

Пунктирные линии обозначают нулевые веса

*Состояние сети* – это просто множество текущих значений сигналов OUT от всех нейронов. В первоначальной сети Хопфилда состояние каждого нейрона менялось в дискретные случайные моменты времени, в последующей работе состояния нейронов могли меняться одновременно. Так как выходом бинарного нейрона может быть только ноль или единица (промежуточных уровней нет), то текущее состояние сети является двоичным числом, каждый бит которого является сигналом OUT некоторого нейрона.

Функционирование сети легко визуализируется геометрически. На рис. 18.4 показан случай двух нейронов в выходном слое, причем каждой вершине квадрата соответствует одно из четырех состояний системы (00, 01, 10, 11). На рис. 18.5 показана трехнейронная система, представленная кубом (в трехмерном пространстве), имеющим восемь вершин, каждая из которых помечена трехбитовым бинарным числом. В общем случае система с  $n$  нейронами имеет  $2^n$  различных состояний и представляется  $n$ -мерным гиперкубом.

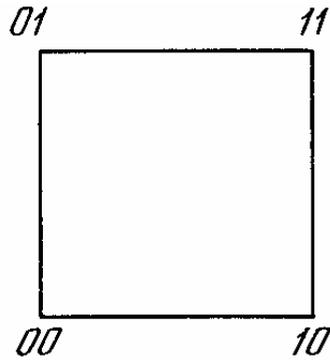


Рис. 18.4. Два нейрона порождают систему с четырьмя состояниями

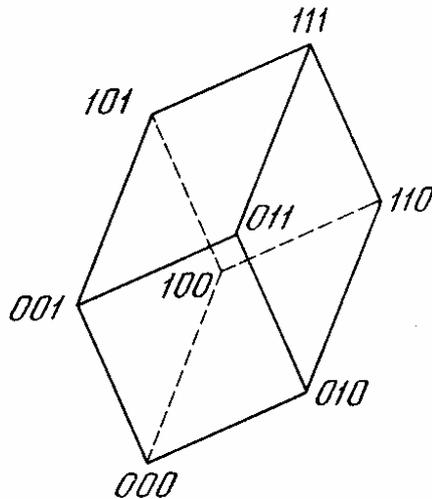


Рис. 18.5 Три нейрона порождают систему с восемью состояниями

Когда подается новый входной вектор, сеть переходит из вершины в вершину, пока не стабилизируется. Устойчивая вершина определяется сетевыми весами, текущими входами и величиной порога. Если входной вектор частично неправилен или неполон, то сеть стабилизируется в вершине, ближайшей к желаемой.

#### Устойчивость

Как и в других сетях, веса между слоями в этой сети могут рассматриваться в виде матрицы  $\mathbf{W}$ . В работе [2] показано, что сеть с обратными связями является устойчивой, если ее матрица симметрична и имеет нули на главной диагонали, т. е. если  $w_{ij} = w_{ji}$  и  $w_{ii} = 0$  для всех  $i$ .

Устойчивость такой сети может быть доказана с помощью элегантного математического метода. Допустим, что найдена функция, которая всегда убывает при изменении состояния сети. В конце концов эта функция должна

достичь минимума и прекратить изменение, гарантируя тем самым устойчивость сети. Такая функция, называемая функцией Ляпунова, для рассматриваемых сетей с обратными связями может быть введена следующим образом:

$$E = -\frac{1}{2} \sum_i \sum_j w_{ij} \text{OUT}_i \text{OUT}_j - \sum_j I_j \text{OUT}_j + \sum_j T_j \text{OUT}_j$$

(18.7)

где  $E$  – искусственная энергия сети;  $w_{ij}$  – вес от выхода нейрона  $i$  к входу нейрона  $j$ ;  $\text{OUT}_j$  – выход нейрона  $j$ ;  $I_j$  – внешний вход нейрона  $j$ ;  $T_j$  – порог нейрона  $j$ .

Изменение энергии  $E$ , вызванное изменением состояния  $j$ -нейрона, есть

$$\delta E = \left[ \sum_{i \neq j} (w_{ij} \text{OUT}_i) + I_j - T_j \right] \delta \text{OUT}_j = -[\text{NET}_j - T_j] \delta \text{OUT}_j$$

(18.8)

где  $\delta \text{OUT}_j$  – изменение выхода  $j$ -го нейрона.

Допустим, что величина NET нейрона  $j$  больше порога. Тогда выражение в скобках будет положительным, а из Уравнения (6.1) следует, что выход нейрона  $j$  должен измениться в положительную сторону (или остаться без изменения). Это значит, что  $\delta \text{OUT}_j$  может быть только положительным или нулем и  $\delta E$  должно быть отрицательным. Следовательно, энергия сети должна либо уменьшиться, либо остаться без изменения.

Далее, допустим, что величина NET меньше порога. Тогда величина  $\delta \text{OUT}_j$  может быть только отрицательной или нулем. Следовательно, опять энергия должна уменьшиться или остаться без изменения.

И окончательно, если величина NET равна порогу,  $\delta_j$  равна нулю и энергия остается без изменения.

Это показывает, что любое изменение состояния нейрона либо уменьшит энергию, либо оставит ее без изменения. Благодаря такому непрерывному стремлению к уменьшению энергия в конце концов должна достигнуть минимума и прекратить изменение. По определению такая сеть является устойчивой.

Симметрия сети является достаточным, но не необходимым условием для устойчивости системы. Имеется много устойчивых систем (например, все сети прямого действия!), которые ему не удовлетворяют. Можно продемонстрировать примеры, в которых незначительное отклонение от симметрии может приводить к непрерывным осцилляциям. Однако приближенной симметрии обычно достаточно для устойчивости систем.

#### Ассоциативная память

Человеческая память ассоциативна, т. е. некоторое воспоминание может порождать большую связанную с ним область. Например, несколько музыкальных тактов могут вызвать целую гамму чувственных воспоминаний, включая пейзажи, звуки и запахи. Напротив, обычная компьютерная память является локально адресуемой, предъявляется адрес и извлекается информация по этому адресу.

Сеть с обратной связью формирует ассоциативную память. Подобно человеческой памяти по заданной части нужной информации вся информация извлекается из «памяти». Чтобы организовать ассоциативную память с помощью сети с обратными связями, веса должны выбираться так, чтобы образовывать энергетические минимумы в нужных вершинах единичного гиперкуба.

Хопфилд разработал ассоциативную память с непрерывными выходами, изменяющимися в пределах от +1 до -1, соответствующих двоичным значениям 0 и 1, Запоминаемая информация кодируется двоичными векторами и хранится в весах согласно следующей формуле:

$$w_{ij} = \sum_{d=1...m} (\text{OUT}_{i,d} \text{OUT}_{j,d})$$

(18.8)

где  $m$  – число запоминаемых выходных векторов;  $d$  – номер запоминаемого выходного вектора;  $\text{OUT}_{i,j}$  –  $i$ -компонента запоминаемого выходного вектора.

Это выражение может стать более ясным, если заметить, что весовой массив  $\mathbf{W}$  может быть найден вычислением внешнего произведения каждого запоминаемого вектора с самим собой (если требуемый вектор имеет  $n$  компонент, то эта операция образует матрицу размером  $n \times n$ ) и суммированием матриц, полученных таким образом. Это может быть записано в виде

$$\mathbf{W} = \sum_i \mathbf{D}_i^t \mathbf{D}_i,$$

(18.9)

где  $\mathbf{D}_i$  –  $i$ -й запоминаемый вектор-строка.

Как только веса заданы, сеть может быть использована для получения запомненного выходного вектора по данному входному вектору, который может быть частично неправильным или неполным. Для этого выходам сети сначала придают значения этого входного вектора. Затем входной вектор убирается и сети предоставляется возможность «расслабиться», опустившись в ближайший глубокий минимум. Сеть идущая по локальному наклону функции энергии, может быть захвачена локальным минимумом, не достигнув наилучшего в глобальном смысле решения.

## **Тема 1.6: Примеры практического применения интеллектуальных систем управления**

### **Лекция 23:**

- 1. Роботы. Общие сведения. Интеллектуальные роботы.**
- 2. Системы управления роботами.**
- 3. Базы знаний интеллектуального робота.**

#### **1. Роботы. Общие сведения. Интеллектуальные роботы.**

Начало развития промышленной робототехники было положено в 60-х годах.

Первыми промышленными роботами были ‘ Версатран ’и ‘ Юнимейт -1900’ (США), в СССР - УМ-1, ‘ Универсал 50 ’, УПК-1.

Робот как ОУ представляет собой электромеханическую систему, состоящей из многозвенной механической конструкции (рабочих механизмов), исполнительных устройств и собственно СУ. Рабочий механизм непосредственно воздействует на среду. ИУ включает совокупность приводов,

датчиков, операционных усилителей, преобразующих и корректирующих элементов. Всем этим управляет СУ.

Задача управления роботами заключается в формировании управляющих воздействий для исполнительных двигателей, отработка которых гарантировала бы прохождение захватным устройством заданной пространственной траектории с заданной точностью. Задача формирования управляющих воздействий сводится к построению программной траектории  $q_p(t)$ , т.е. закона изменения вектора относительно положения звеньев манипулятора

$q = [q_1, q_2, \dots, q_n]^T$ , и последующему синтезу собственно закона управления, обеспечивающего устойчивость движения относительно программной траектории.

## 2. Системы управления роботами

Системы управления роботами делятся на человеко-машинные (системы дистанционного и интерактивного управления с оператором в контуре управления) и автоматические. Первые разделяются на 6 групп:

- 1) системы командного управления - оператор управляет приводом.
- 2) системы копирующего управления - оператор управляет задающим устройством.
- 3) системы полуавтоматического управления - оператор рукояткой управляет манипулятором.
- 4) автоматизированные интерактивные - часть операций выполняется автоматически, а остальные оператором.
- 5) интерактивные супервизорного управления - оператор наблюдает за ходом работы робота и коррекции движения робота.
- 6) интерактивные диалогового управления - робот выполняет команды, но и дает советы оператору.

Системы автоматического управления делятся на системы:

- программного управления - синтез движения робота по заранее рассчитываемой программе
- адаптивного управления - движение робота организуется по гибко изменяемым программам, перестраивание программы происходит в ответ на изменение условий внешней среды. Адаптивные системы управления содержат разнообразные средства оцувствления.

■ интеллектуального управления - программа движения робота не задается, а синтезируется СУ на основе описания внешней среды, совокупности правил возможного поведения в этой среде с целью решения поставленной задачи.

Методы управления положены в основу классификации поколений роботов:

первое - с программным управлением;

второе - с адаптивным управлением;

третье - интеллектуальные роботы (с элементами искусственного интеллекта).

Траектория движения робота первого поколения является жестко определенной и дискретной. Каждая точка траектории соответствует одной комбинации программы и находится между двумя крайними положениями манипулятора, такие роботы требуют точной организации окружающей среды (определенного расположения деталей, оснастки и др.). при переналадке программы требуется реорганизация окружающей среды.

СУ робота цикловая, т.е. сигналы управления непосредственно из программного устройства поступают на привод робота. Датчики информации обычно не используются за исключением концевых выключателей или фиксаторы пополнения. Алгоритмы управления формируются в виде логической последовательности срабатываний звеньев манипулятора.

Программирование осуществляется путем физической настройки концевиков, набора циклограмм или подбора машинных кодов.

Для второго и третьего поколения роботов способно позиционирование в любой точке рабочей зоны. Параметры движения (скорость, ускорение, торможение) можно изменять в процессе перемещения по траектории. СУ могут организовывать замкнутые контуры управления по информации о динамике состояния рабочей среды.

Имеют датчики информации о внутреннем состоянии систем робота и датчики внешнего очувствления (например механическое зрение). В интеллектуальных системах средства очувствления объединяются с базой знаний.

Алгоритмы управления синтезируются в результате решения дифференциальных уравнений, описывающих динамику робота. Программирование осуществляется специальными языками высокого уровня.

Рассмотрим более подробно систему управления роботов с программным управлением. Она включает в себя управляюще-вычислительный модуль, программноноситель, блоки сопряжения робота с технологией обучения, панель управления и пульт ручного управления.

Ядром системы является управляюще-вычислительный модуль, основная функция которого заключается в формировании микроопераций (управляющих импульсов), соответствующих заданному алгоритму работы. В качестве модуля используются микропрограммные автоматы, например, на основе микропроцессорных наборов и микроЭВМ. Программноносители могут быть любого типа, от механических до электронных БИС. Блоки сопряжения выполняют функции формирования команд управления приводом, опрос состояний датчиков, обмена информацией.

Более высокий уровень имеют СУ позиционно-контурного программного управления. Они позволяют расширить функциональные возможности роботов за счет исключения ограничения связанного с числом точек позиционирования захватного устройства манипулятора. Эти СУ используют замкнутые системы позиционирования на основе следящих приводов и используют различные датчики положения (аналоговые, кодовые, фазовые, растровые и др.). Строятся как одно- так и многопроцессорные системы, решают задачи качества и точности работы следящих приводов. Реализуются на основе программируемых контроллеров и мини- и микроЭВМ.

### **3. Базы знаний интеллектуального робота**

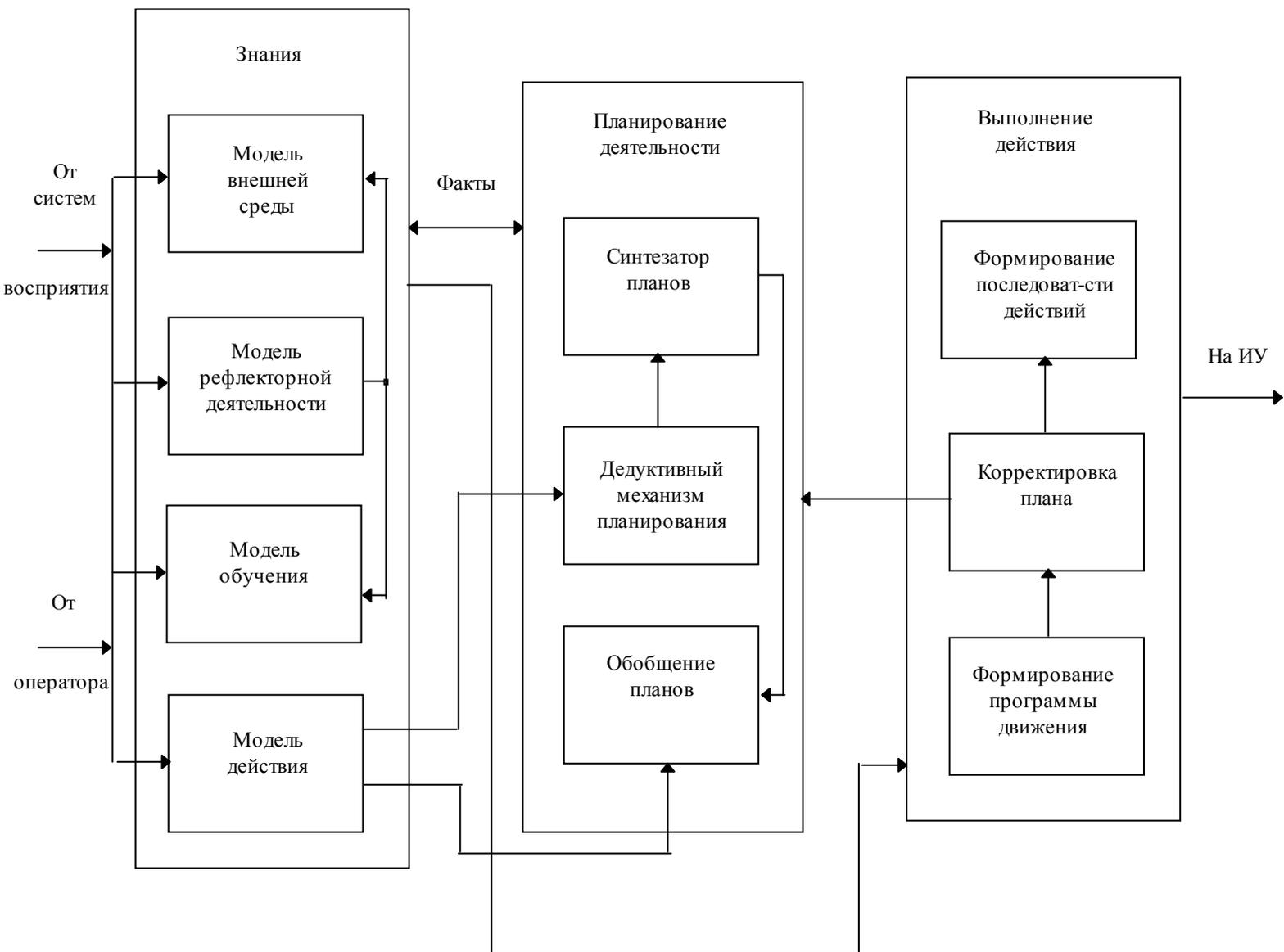
Необходимость в интеллектуальном управлении роботами возникает из практических требования повышения точности и автономности функционирования роботов в условиях не строго организованной технической среды. Наличие таких роботов может повлиять на снижение затрат, расходуемых на подготовку и организацию рабочей среды.

Интеллектуальные роботы содержат всю гамму средств роботов с адаптивным управлением и дополнительно базу знаний. Именно *база знаний* меняет всю стратегию поведения робота и его функциональные возможности. ИР состоит из трех систем: системы восприятия, системы решения задач эффекторной системы (системы управления).

*Система восприятия* связана с двумя источниками информации: внешний мир и человек - оператор. С первым ИР общается посредством датчиков и систем ввода информации. С оператором ИР общается посредством стандартных терминалов диалоговых систем (телетайпов, дисплеев, устройств речевого ввода и др.)

Эффекторная система - это все то, что раньше мы называли системой управления робота (это система выполнения действий). Естественно, нас в ИР больше всего интересует система решения задач, которая и использует базу знаний.

СРЗ промышленного ИР состоит из трех главных подсистем: представления знаний, планирующей и исполнительной (рис.). В свою очередь, знания системы включают в себя модели внешней среды, рефлексивной деятельности и обучения, а также набор операторов преобразующих внешний мир (модели действия).



В связи с использованием знаний возникает ряд задач:

- 1) форма представления знаний;
- 2) накопление знаний;
- 3) корректировка знаний;
- 4) использование знание в процессе принятия решений;

Представление знаний - ? (мы немного рассмотрели).

Накопление знаний - накопление фактов и установление между ними отношений (за счет системы восприятия), накопление методов решения задач.

Корректировка знаний связана с проблемой обучения робота и выражается в разработке механизмов 'отбраковки' знаний по различным

критериям, например, шина логической противоречивости или непротиворечивости.

Синтез планов осуществляется на основе поставленных целей и накопленного знания (цели могут ставиться оператором или самим роботом). При синтезе планов используются дедуктивные механизмы, при которых план определяется как последовательность применимых и пригодных операторов. Обобщение планов - робот должен обобщать построенные планы, чтобы использовать их в дальнейшем, накапливая модели действий.

На основе поступающей от системы восприятия информации о внешней среде формируется ее модель, необходимая для планирования целенаправленного поведения робота. Для реализации спланированной роботом стратегии поведения в условиях изменения состояния внешней среды необходима модель его рефлекторной деятельности (БЗ). Принятое роботом решение в соответствии с выработанным планом поведения и его рефлекторной деятельности передаются в эффекторную систему, где и отрабатывается.

#### *Особенности БЗ ИР*

1. Терпимость к противоречиям. Это означает, что при появлении в БЗ ошибок и противоречий можно допустить определенное снижение эффективности деятельности робота, но нельзя допустить полное прекращение этой деятельности. Новая информация, воспринимаемая роботом, может содержать ошибки или не соответствовать информации, уже имеющейся в БЗ. Проблемы связанные с решаются двумя путями: первый связан с попыткой защитить БЗ от проникновения ошибок (вводятся анализаторы входной информации); второй - связан с созданием систем, способных решить свои задачи на основе противоречивых данных.
2. Обеспечение вывода. БЗ не имеющие полной информации о внешнем мире, должна быть способна к логическому выводу. Различают два типа выводов: свободный и направленный. Направленный запускается при поиске ответа на поступивший запрос, а свободный осуществляется всегда при поступлении в БЗ новой информации.
3. Критичность к новой информации, т.е. способность проверить достоверность новой информации и согласовать ее с уже имеющимися знаниями.

4. Дробность БЗ. Это значит, что при поступлении новой информации БЗ должна обладать способностью выделить некоторый фрагмент знаний, обеспечивающий эффективную обработку этой информации.
5. Обучаемость и способность к перестраиванию БЗ. Совместно с механизмом критичности обучения должно противодействовать накоплению в ЮЗ ошибок и противоречий.

Анализ указанных особенностей, показывает, что основной задачей в области представления знаний, для ИР является адаптация средств и методов представления знаний. В процессе такой адаптации должны быть выработаны: способ представления знаний и элементная база системы знаний.

Лекція 17 Штучні нейронні сітки. Структура штучних нейронних сіток. Різновиди передаточних функцій нейрона. Принципи навчання і тренування нейронних сіток. Алгоритм зворотної передачі.

Л.3 с. 5-14, 41-55, Л.4 с.133-138.

Лекція 18 Алгоритм зустрічної передачі. Сітки Хопффілда. Характеристики спеціальних нейронних сіток. Сітки багатосарових розпізнавальних систем типу MLP. Сітки радіальних базисних функцій типу RBF.

Л.3 с. 55-98. Л.4 с. 139-148.

Лекція 19 Нечіткий нейрон. Нечіткі нейронні сітки. Адаптивні системи керування складними об'єктами на основі нечітких нейронних сіток. Методи генерування моделей об'єктів.

Л.4 с.164-181

Лекція 20 Імітаційне моделювання ТЛП нечітких регуляторів нейронними сітками.. Використання нейронних сіток для оптимізації та навчання ТЛП нечітких регуляторів.

Тема 2.5 Проектування та реалізація лінгвістичних елементів та пристроїв

Лекція 21 – 22 Лінгвістичні комплекси, пристрої, блоки. Структурно-функціональна організація лінгвістичних блоків ЛДБ, ЛБО, ЛБС, ЛБНС, ЛБДП,

ЛБНО, ЛБОС, ЛБОР, ЛБОП, ЛБПР, ЛБСМ. Синтез лінгвістичних елементів, блоків та пристроїв.

Л.2 с. 115-167.

**Тема 2.6** Приклади практичного застосування інтелектуальних систем управління

**Лекція 23 -24** Інтелектуальні роботи. Покоління роботів. Системи керування роботів. Бази знань інтелектуального робота.

Елементи систем керування роботів. Системи технічного зору роботів. Класифікація, структура, функціональні можливості, вирішуванні задачі. Основи обробки та розпізнавання зображень. Методи опису зображень.

**ЛЕКЦІЯ 25** ІНТЕЛЕКТУАЛЬНА АДАПТИВНА СИСТЕМА УПРАВЛІННЯ ТЕХНОЛОГІЧНИМ ПРОЦЕСОМ В УМОВАХ НЕПОВНОЇ ІНФОРМАЦІЇ