

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Кубанский государственный технологический
университет»

Кафедра информационных систем и программирования

СИСТЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Методические указания по выполнению лабораторных работ
для студентов всех форм обучения направления подготовки

09.03.04 Программная инженерия

Краснодар

2019

Составитель: канд. техн. наук, ст. преп. К.Е. Тотухов

Системы искусственного интеллекта: методические указания по выполнению лабораторных работ для студентов всех форм обучения направления подготовки 09.03.04 Программная инженерия / Сост.: К.Е. Тотухов; Кубан. гос. технол. ун-т. Каф. информационных систем и программирования. – Краснодар, 2019. – 23 с.

Содержат перечень лабораторных работ с указанием целей и задач их выполнения, теоретическую информацию к каждой лабораторной работе, варианты заданий, формы итогового контроля, ссылки на литературу.

Рецензенты: д-р техн. наук, ст. науч. сотр., проф. кафедры ИСиП КубГТУ
Л.А. Видовский;

канд. техн. наук, доц., доц. кафедры математики и выч. техники
Академии маркетинга и социально-информационных
технологий Н. С. Нестерова

Содержание

Лабораторная работа №1 Игровые задачи и их решение на основе интеллектуальных алгоритмов.....	4
Лабораторная работа №2 Модели представления знаний.....	6
Лабораторная работа №3 Нечёткий естественно-языковой интерфейс базы знаний	9
Лабораторная работа №4 Однослойный перцептрон и его обучение методом коррекции ошибки.....	11
Лабораторная работа №5 Архитектуры искусственных нейронных сетей.....	13
Лабораторная работа №6 Кластеризация алгоритмом k-средних	15
Лабораторная работа №7 Система на основе генетических алгоритмов	17
Лабораторная работа №8 Модель искусственной жизни с агентами на нейроконтроллерах.....	19
Список литературы	23

Лабораторная работа №1 Игровые задачи и их решение на основе интеллектуальных алгоритмов

Цель работы: освоить методики решения игровых задач с помощью интеллектуальных алгоритмов

Теоретический материал:

1. Альфа-бета отсечение основано на той идее, что анализ некоторых ходов можно прекратить досрочно, игнорируя результат их показаний. Альфа-бета относится к классу методов ветвей и границ.

Оптимизация вводит две дополнительные переменные *alpha* и *beta*, где *alpha* — текущее максимальное значение, меньше которого игрок максимизации никогда не выберет, а *beta* — текущее минимальное значение, больше которого игрок минимизации никогда не выберет. Изначально они устанавливаются в $-\infty$, $+\infty$ соответственно, и по мере получения оценок $f(V_i)$ модифицируются:

- $alpha = \max(alpha, f(V_i))$; для уровня максимизации.
- $beta = \min(beta, f(V_i))$; для уровня минимизации.

Как только условие $alpha > beta$ станет верным, что означает конфликт ожиданий, мы прерываем анализ V_i и возвращаем последнюю полученную оценку этого уровня.

На нашем примере, на втором рисунке, с помощью альфа-бета отсечений, полностью отсекаются 3 нижних ряда, но без самого левого столбца.

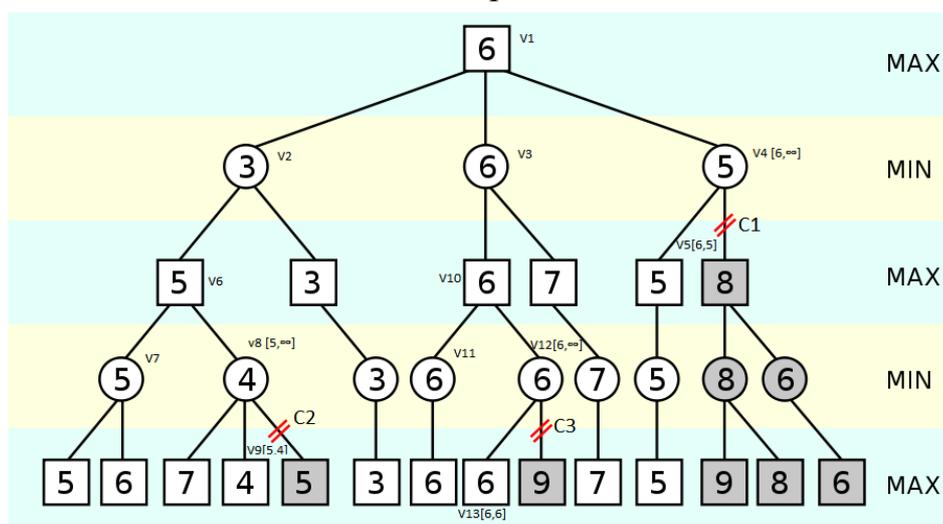


Рисунок 1 – Пример альфа-бета отсечения

- V_i — узлы дерева поиска решений, i никоим образом не коррелирует с порядком обхода дерева.
- $V_i[\alpha, \beta]$ — узлы с указанными текущими значениями α, β .
- C_i — i -ое альфа-бета отсечение.
- MIN, MAX — уровни максимизации и минимизации соответственно. Будьте внимательны, состояния на уровне максимума будут выбирать свои максимумы из состояний следующего уровня, на которых указан MIN, и наоборот. То есть максимальный выбирается из тех, на которых написан MIN, а минимальный из тех, на которых написан MAX, не перепутайте.

2. Эвристический алгоритм (эвристика) — алгоритм решения задачи, включающий практический метод, не являющийся гарантированно точным или оптимальным, но достаточный для решения поставленной задачи. Позволяет ускорить решение задачи в тех случаях, когда точное решение не может быть найдено.

Определение: Эвристический алгоритм — это алгоритм решения задачи, правильность которого для всех возможных случаев не доказана, но про который известно, что он даёт достаточно хорошее решение в большинстве случаев. В действительности может быть даже известно (то есть доказано) то, что эвристический алгоритм формально неверен. Его всё равно можно применять, если при этом он даёт неверный результат только в отдельных, достаточно редких и хорошо выделяемых случаях или же даёт неточный, но всё же приемлемый результат.

Проще говоря, эвристика — это не полностью математически обоснованный (или даже «не совсем корректный»), но при этом практически полезный алгоритм.

Важно понимать, что эвристика, в отличие от корректного алгоритма решения задачи, обладает следующими особенностями.

- Она не гарантирует нахождение лучшего решения.
- Она не гарантирует нахождение решения, даже если оно заведомо существует (возможен «пропуск цели»).
- Она может дать неверное решение в некоторых случаях.

Задание: применить алгоритм (альфа-бета отсечения если у вас чётный номер в группе, эвристики – если нечётный) к одной из классических игровых задач.

Лабораторная работа №2 Модели представления знаний

Цель работы: разработать базу знаний на основе одной из классических моделей – продукционной, фреймовой, семантической.

Теоретический материал:

1. Продукционная модель

В общем виде под продукцией понимается выражение следующего вида (1):

$$(i); Q; P; A \rightarrow B; N \quad (1)$$

Где i - имя продукции, с помощью которого данная продукция выделяется из всего множества продукций. В качестве имени может выступать некоторая лексема, отражающая суть данной продукции (например, "покупка книги"), или порядковый номер продукций в их множестве, хранящимся в памяти системы.

- Элемент Q характеризует сферу применения продукции. Такие сферы легко выделяются в когнитивных структурах человека. Наши знания как бы "разложены по полочкам". На одной полочке хранятся знания о том, как надо готовить пищу, на другой как добраться до работы, и т.п. Разделение знаний на отдельные сферы позволяет экономить время на поиск нужных знаний. Такое же разделение на сферы в базе знаний ИИ целесообразно и при использовании для представления знаний продукционных моделей.

- Основным элементом продукции является ее ядро: $A \rightarrow B$. Интерпретация ядра продукции может быть различной и зависит от того, что стоит слева и справа от знака секвенции \rightarrow . Обычное прочтение ядра продукции выглядит так: ЕСЛИ A , ТО B , более сложные конструкции ядра допускают в правой части альтернативный выбор, например, ЕСЛИ A , ТО B_1 , ИНАЧЕ B_2 . Секвенция может истолковываться в обычном логическом смысле как знак логического следования B из истинного A (если A не является истинным выражением, то о B ничего сказать нельзя). Возможны и другие интерпретации ядра продукции, например A описывает некоторое условие, необходимое для того, чтобы можно было совершить действие B .

- Элемент P есть условие применимости ядра продукции. Обычно P представляет собой логическое выражение (как правило предикат). Когда P принимает значение "истина", ядро продукции активизируется. Если P "ложно", то ядро продукции не может быть использовано.

- Элемент N описывает постусловия продукции. Они актуализируются

только в том случае, если ядро продукции реализовалось. Постусловия продукции описывают действия и процедуры, которые необходимо выполнить после реализации В. Выполнение N может происходить сразу после реализации ядра продукции.

2. Семантическая сеть.

Приведенная ниже программа реализует простейшую сеть, представляющую собой базу знаний по птицам и самолетам.

Схематически данная сеть имеет следующий вид:

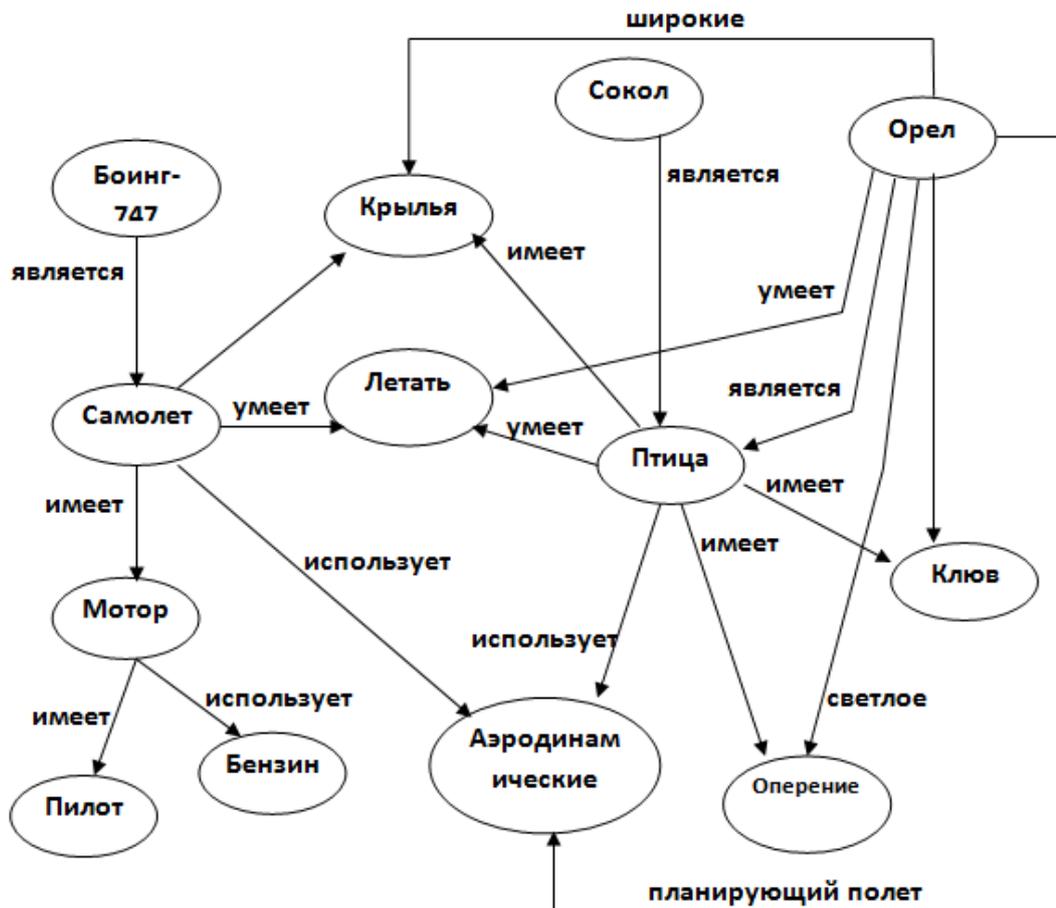


Рисунок 2 – Классифицирующая семантическая сеть

По построенной сети программа позволяет:

- получить все узлы, связанные с отдельным узлом конкретным соотношением;
- получить имена всех отношений для отдельного узла;
- получить все пары узлов, связанных конкретным соотношением;
- показать сеть графически;

- проследить путь от одного узла к другому.

3. Фрейм

Фрейм – это минимально возможное описание сущности какого-либо события, ситуации, процесса или объекта. В историческом плане развитие фреймовой модели связано с теорией фреймов М. Минского, определяющей способ формализации знаний, используемый при решении задач распознавания образов (сцен) и понимания речи. «Отправным моментом для данной теории служит тот факт, что человек, пытаясь познать новую для себя ситуацию или по-новому взглянуть на уже привычные вещи, выбирает из своей памяти некоторую структуру данных (образ), называемую нами фреймом, с таким расчетом, чтобы путем изменения в ней отдельных деталей сделать ее пригодной для понимания более широкого класса явлений или процессов». Другими словами, *фрейм – это форма описания знаний, очерчивающая рамки рассматриваемого (в текущей ситуации при решении данной задачи) фрагмента предметной области.*

Модель фрейма является достаточно универсальной, поскольку позволяет отобразить все многообразие знаний о мире через:

- *фреймы-структуры, для обозначения объектов и понятий (заем, залог, вексель);*
- *фреймы-роли (менеджер, кассир, клиент);*
- *фреймы-сценарии (банкротство, собрание акционеров, празднование именин);*
- *фреймы-ситуации (тревога, авария, рабочий режим устройства) и др.*

Различают *фреймы-образцы, или прототипы, и фреймы-экземпляры, которые создаются для отображения реальных фактических ситуаций на основе поступающих данных.*

Фрейм имеет имя (название) и состоит из *слов.*

Задание: разработать базу знаний (согласовать тематику с преподавателем) на основе:

1. Продукционной модели, если номер в группе 1,4,7,10,13,16,19
2. Семантической если 2,5,8,11,14,17,20
3. Фреймовой если 3,6,9,12,15,18,21

Лабораторная работа №3 Нечёткий естественно-языковой интерфейс базы знаний

Цель работы: усовершенствовать базу знаний нечётким естественно-языковым интерфейсом.

Теоретический материал:

Среди формальных моделей, развиваемых в рамках лингвистического подхода, наиболее глубоко исследованными и нашедшими широкое практическое применение при создании конкретных приложений, являются грамматики Хомского. В их числе следует выделить трансформационные грамматики, описывающие взаимосвязь глубинного и поверхностного уровней синтаксической структуры, модели типа "Смысл \Leftrightarrow Текст", где язык рассматривается как система кодов, отражающих содержательно-семантическую составляющую текстовой информации, а также так называемые генеративные модели, основанные на базе функциональных теорий и реализующие принцип ограничений на формирование лингвистической структуры для описания грамматики.

При этом учет и проявление грамматики происходит уже не только на уровне реализации, но и самой методологии смысловой обработки текстов, Соответствующая технология подбора "инструментария", используемого лингвистическим процессором, включает несколько основных этапов:

— конкретизацию класса модели, а следовательно и максимальной совокупности лингвистических и нелингвистических средств, необходимых для решения поставленной задачи по обработке текста;

— конкретизацию типа взаимодействия пользователя и системы (активная система - пассивный пользователь, пассивная система — активный пользователь), а следовательно и уточнение перечня средств, обеспечивающих требуемый режим функционирования;

— конкретизацию типа обрабатываемого текста (для перевода с языка на язык, для перефразы, для извлечения знаний, морфологического анализа или коррекции и т.д.), а следовательно, и окончательное определение состава привлекаемых средств.

Очевидно, что дифференциация моделей и их последовательная редукция являются не только одним из возможных, но и достаточно эффективным путем построения лингвистических процессоров, ориентированных на решение задач смысловой обработки текстов.

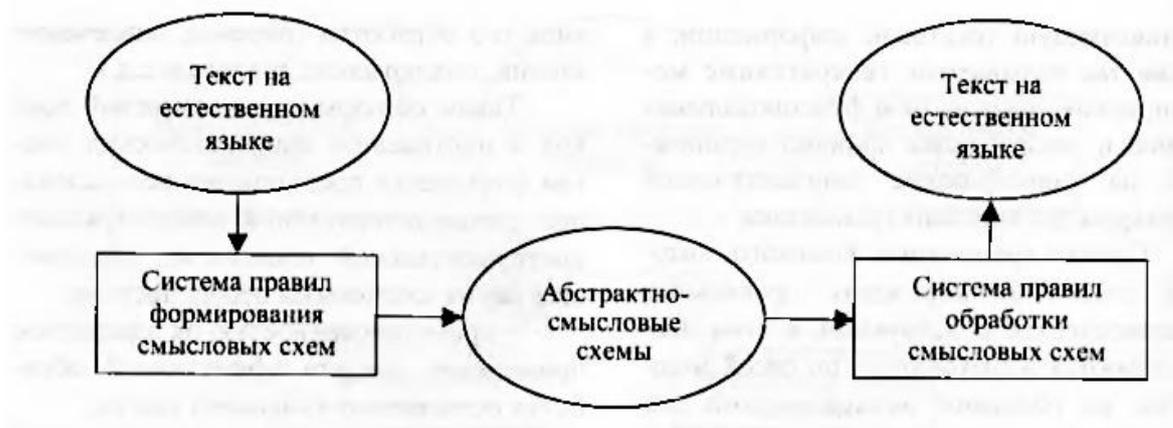


Рисунок 3 – Обобщённое представление процедур обработки и интерпретации естественно-языкового текста

Экспериментально-эвристический подход к построению лингвистических систем связан с попытками комплексного применения методов искусственного интеллекта для реализации механизмов глубинного проникновения в смысловое содержание анализируемой естественно-языковой информации с учетом семантики контекста.

Так, например, в соответствии с теорией концептуальной зависимости процедуры смыслового разбора текста строятся на основе оценки семантической роли слова в зависимости от конкретных ситуаций, свойственных построению отдельных фраз.

Использование упрощенных моделей естественного языка, позволяющих во многом игнорировать сложности синтаксиса, обеспечивает возможность смысловой интерпретации исследуемого текста в виде логико-ситуационных схем, которые на уровне качественных представлений определяют, кто и что делает, по отношению к кому, когда, где и т.д.

Обобщая результаты проведенного анализа, следует отметить, что, как показано на рисунке 3, процесс обработки естественно-языковой информации вне зависимости от типа выбранной лингвистической модели сводится к формированию и интерпретации абстрактно-смысловых схем, в той или иной мере отвечающих семантике имеющегося текстового фрагмента. Подобное представление полностью согласуется с концепцией построения интеллектуальных систем управления, предполагающей использование встроенных или неявно заданных ситуационных моделей для принятия необходимых решений, составления текущих планов и прогнозов.

Задание: усовершенствовать базу знаний нечётким естественно-языковым интерфейсом.

Лабораторная работа №4 Однослойный перцептрон и его обучение методом коррекции ошибки

Цель работы: запрограммировать и обучить однослойный перцептрон.

Теоретический материал:

Модель искусственного нейрона (рисунок 4) представляет собой дискретно-непрерывный преобразователь информации. Информация, поступающая на вход нейрона, суммируется с учётом весовых коэффициентов w_i , сигналов x_i , $i = 1, \dots, n$, где n – размерность пространства входных сигналов. Потенциал нейрона определяется по формуле:

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

Схема кибернетической модели нейрона показана на рисунке 4.

Взвешенная сумма поступивших сигналов (потенциал) преобразуется с помощью передаточной функции $f(P)$ в выходной сигнал нейрона Y , который передается другим нейронам сети, т.е. $Y = f(P)$.

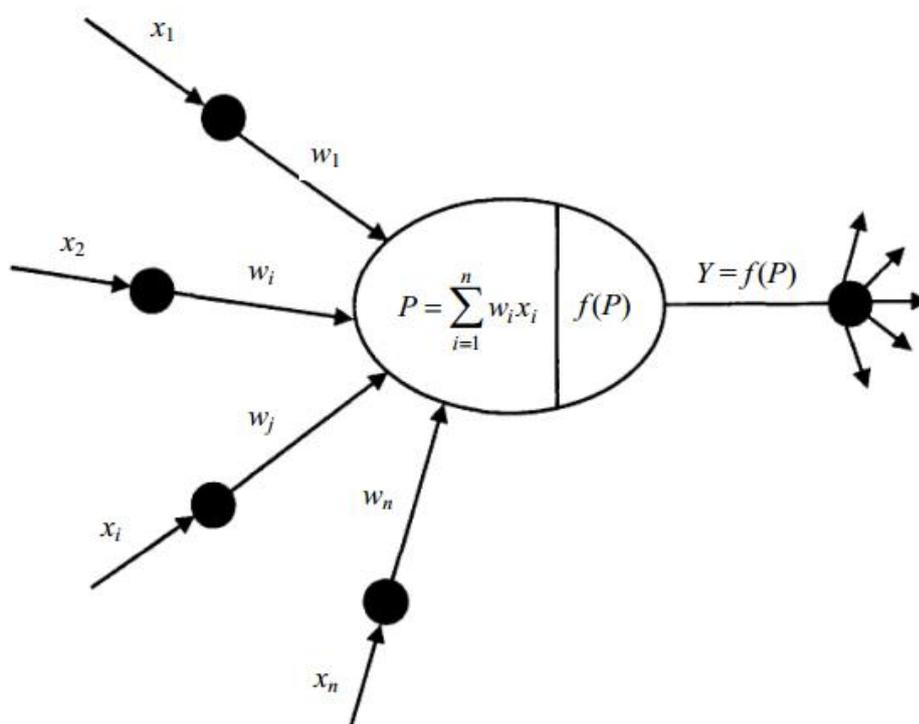


Рисунок 4 – Кибернетическая модель искусственного нейрона МакКаллока –Питтса, 1943 г.

Вид передаточной (активационной) функции является важнейшей

характеристикой нейрона. В общем случае эта функция может быть ступенчатой (пороговой), линейной или нелинейной. Пороговая функция пропускает информацию только в том случае, если алгебраическая сумма входных сигналов превышает некоторую постоянную

величину P^* , например:

$$Y = \begin{cases} 1, & \text{если } P \geq P^*; \\ -1, & \text{если } P < P^*. \end{cases}$$

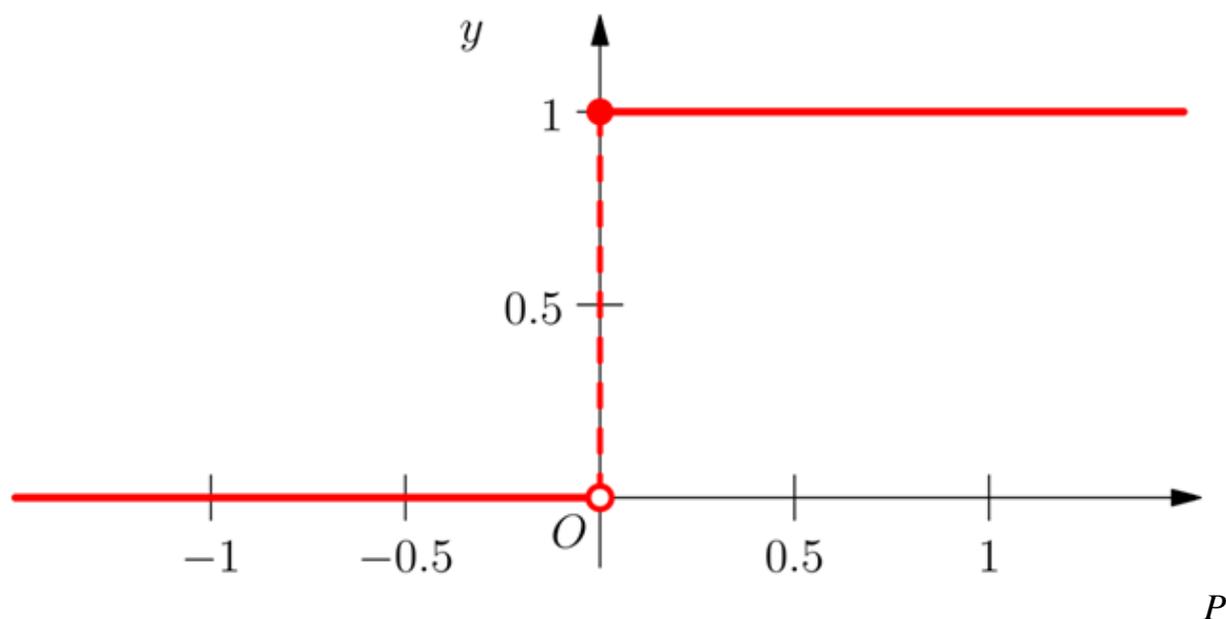


Рисунок 5 – Ступенчатая функция активации нейрона при $P^* = 0$

Алгоритм обучения однослойного перцептрона **методом коррекции ошибки** включает следующие шаги.

1. Системе предъявляется эталонный образ.
2. Если результат распознавания совпадает с заданным, весовые коэффициенты связей не изменяются.
3. Если ИНС неправильно распознаёт результат, то весовым коэффициентам даётся приращение в сторону повышения качества распознавания.

Задание: запрограммировать и обучить однослойный перцептрон

Лабораторная работа №5 Архитектуры искусственных нейронных сетей

Цель работы: запрограммировать одну из сложных архитектур искусственных нейронных сетей.

Теоретический материал:

1. Сеть Хопфилда:

Сеть Хопфилда строится с учётом следующих условий:

- все элементы связаны со всеми;
- $w_{ji} = w_{ij}$
- прямые и обратные связи симметричны;
- $w_{ii} = 0$ – диагональные элементы матрицы связей равны нулю, т.е. исключаются обратные связи с выхода на вход одного нейрона.

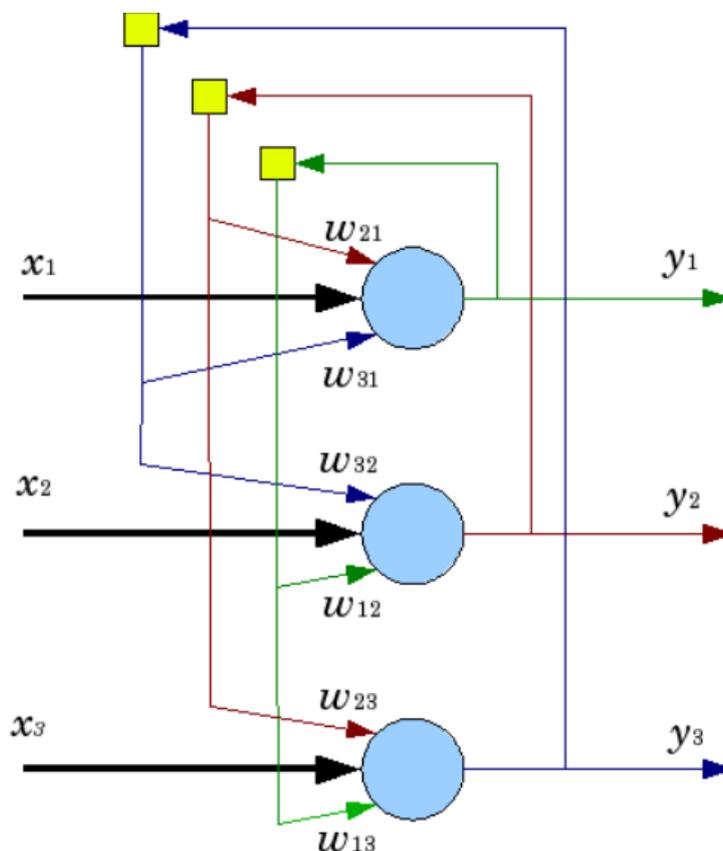


Рисунок 6 – Сеть Хопфилда

2. Многослойный перцептрон:

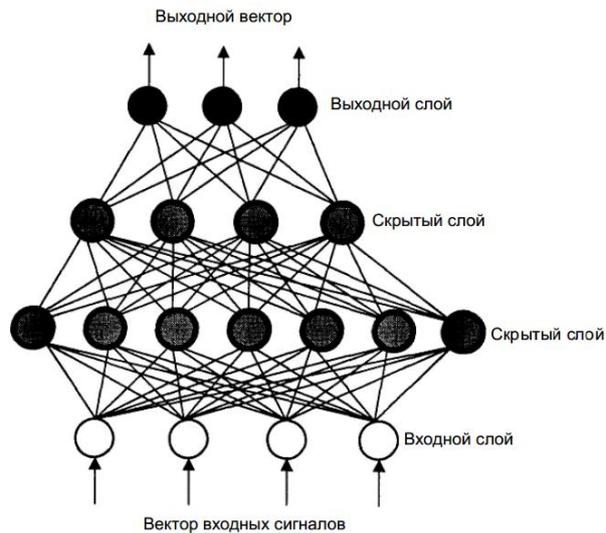


Рисунок 7 – Многослойный перцептрон с двумя скрытыми слоями

3. Сеть Кохонена

Сеть Кохонена, в отличие от многослойной нейронной сети, очень проста; она представляет собой два слоя: входной и выходной. Ее также называют самоорганизующей картой. Элементы карты располагаются в некотором пространстве, как правило, двумерном. Сеть Кохонена изображена на рисунке 8.

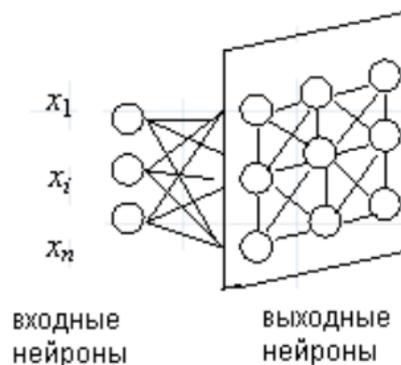


Рисунок 8 – Схема сети Кохонена

Задание: запрограммировать одну из сложных архитектур искусственных нейронных сетей:

1. Сеть Хопфилда, если номер в группе 1,4,7,10,13,16,19
2. Многослойный перцептрон если 2,5,8,11,14,17,20
3. Сеть Кохонена если 3,6,9,12,15,18,21

Условие: обязательно реализовать графическую прорисовку сети.

Лабораторная работа №6 Кластеризация алгоритмом k-средних

Цель работы: Реализовать программно кластеризацию алгоритмом k-средних.

Теоретический материал:

Метод k-средних

Определение: *Центроид* — точка, которая является центром кластера.

k-средних (k-means) — наиболее популярный метод кластеризации. Алгоритму широко отдается предпочтение из-за его простоты реализации, большой скорости (а это очень важно при работе с видео).

Действие алгоритма таково, что он стремится минимизировать суммарное квадратичное отклонение точек кластеров от центров этих кластеров. Проще говоря, это итеративный алгоритм, который делит данное множество пикселей на k кластеров, точки которых являются максимально приближенными к их центрам, а сама кластеризация происходит за счет смещения этих же центров.

Также следует оговорить то, что метод k-средних очень чувствительный к шуму, который может существенно исказить результаты кластеризации. Так что в идеале, перед кластеризацией, нужно прогнать кадры через фильтры, предназначенные для его уменьшения.

Принцип кластеризации методом k-средних:

1. Надо выбрать из множества k пикселей те пиксели, которые будут центроидами соответствующих k кластеров.
2. Выборка начальных центроидов может быть как случайной, так и по определенному алгоритму.
3. Входим в цикл, который продолжается до тех пор, пока центроиды кластеров не перестанут изменять свое положение.
4. Обходим каждый пиксель и смотрим, к какому центроиду какого кластера он является ближайшим.
5. Нашли ближайший центроид? Привязываем пиксель к кластеру этого центроида.
6. Перебрали все пиксели? Теперь нужно вычислить новые координаты центроидов k кластеров.

Теперь проверяем координаты новых центроидов. Если они соответственно равны предыдущим центроидам — выходим из цикла, если нет, возвращаемся к пункту 3.

На рисунке 2 показан результат метода к-средних в общем виде

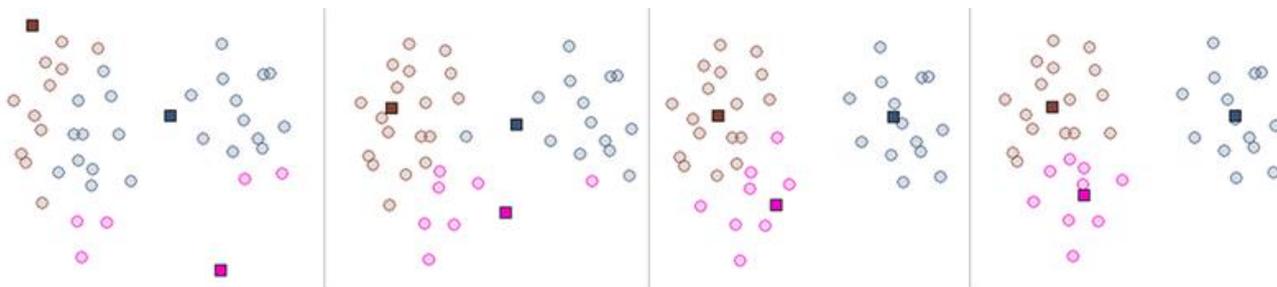


Рисунок 9 – Результат кластеризации по к-средних в общем виде

Задание: Реализовать программно кластеризацию алгоритмом к-средних

Лабораторная работа №7 Система на основе генетических алгоритмов

Цель работы: Реализовать программно систему на основе генетических алгоритмов.

Теоретический материал:

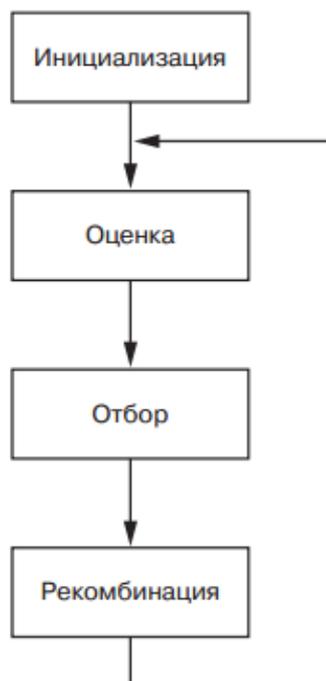


Рисунок 10 – Общая схема работа генетического алгоритма

Выполнение генетического алгоритма включает три основных шага (причем для каждого шага предусмотрен большой разброс возможных вариантов). Эти шаги показаны на рисунке 10.

Генетический алгоритм выполняется в три этапа (если не учитывать начальное создание популяции). Во время оценки определяется здоровье популяции. Далее производится отбор подгруппы хромосом на основании предварительно заданного критерия. Наконец, выбранная подгруппа рекомбинируется, в результате чего получается новая популяция.

Теперь рассмотрим итерацию алгоритма, чтобы понять принцип его работы. В этом примере выполняется поиск параметров, которые позволяют максимизировать уравнение:

$$f(x,y) = \frac{1}{1+x^2+y^2}$$

Это уравнение графически представлено на рисунке 11 (трехмерный график и контурный график)

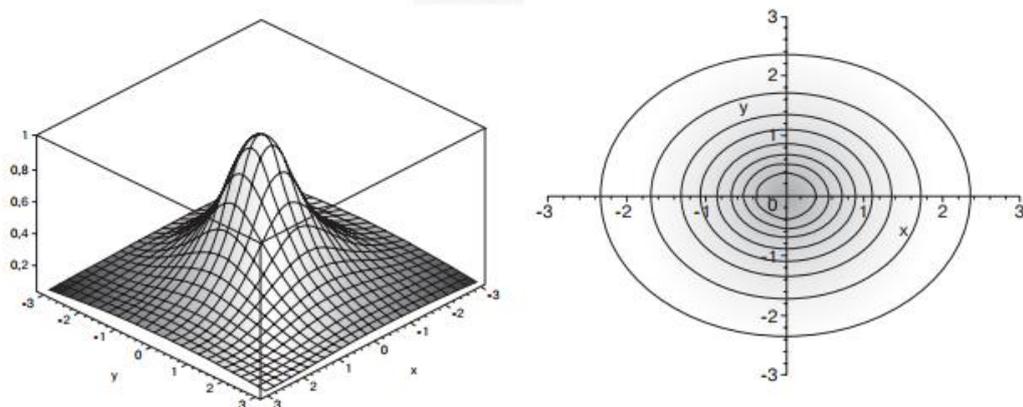


Рисунок 11 – Трёхмерный и контурный график работы генетического алгоритма

Задание: Реализовать программно систему на основе генетических алгоритмов.

Лабораторная работа №8 Модель искусственной жизни с агентами на нейроконтроллерах

Цель работы: Реализовать программно искусственную жизнь с агентами на нейроконтроллерах.

Теоретический материал:

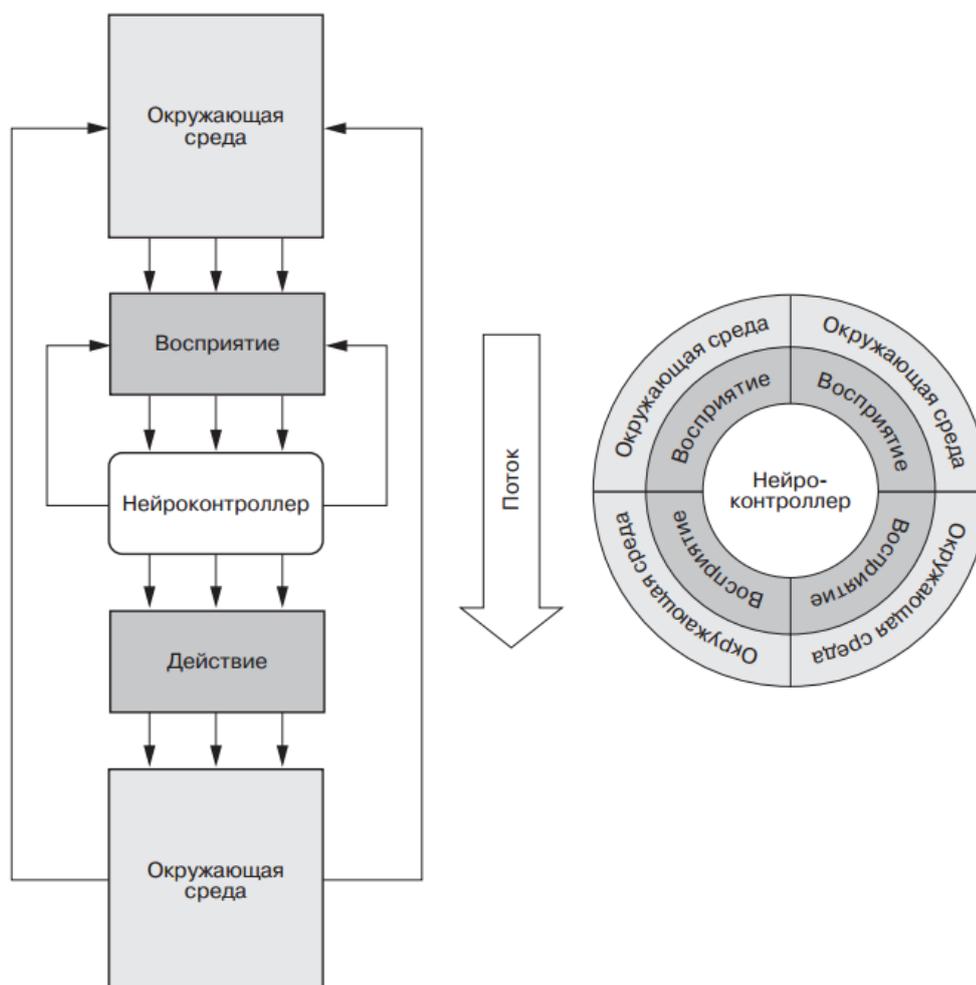


Рисунок 12 - Пример нейроконтроллера в окружающей среде

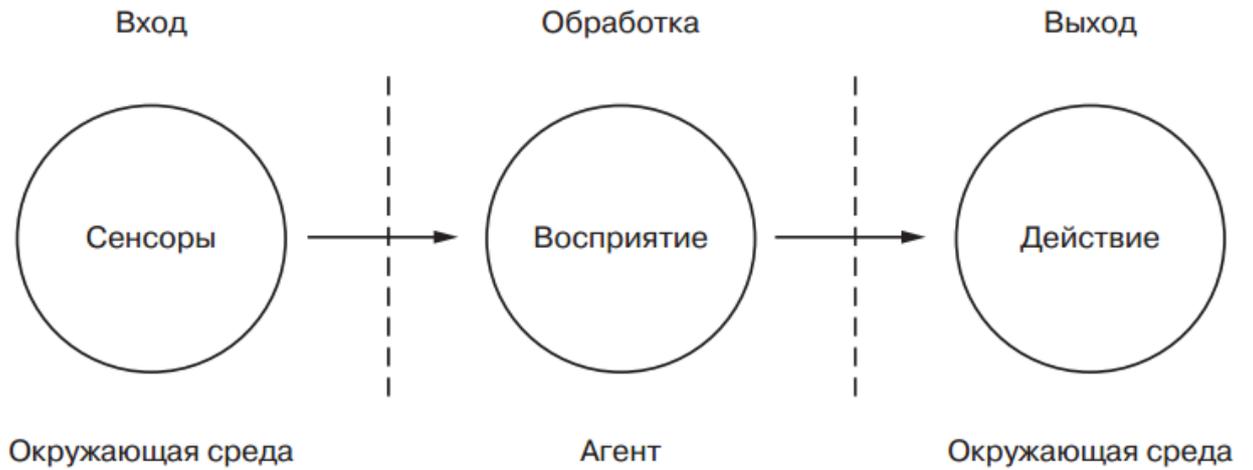


Рисунок 13 – Модель систем с агентами

	Фронт	Фронт	Фронт	Фронт	Фронт	
	Слева	Близость	Близость	Близость	Справа	
	Слева	Близость	Агент	Близость	Справа	

Рисунок 14 – Область предчувствия агента. Ориентация на север

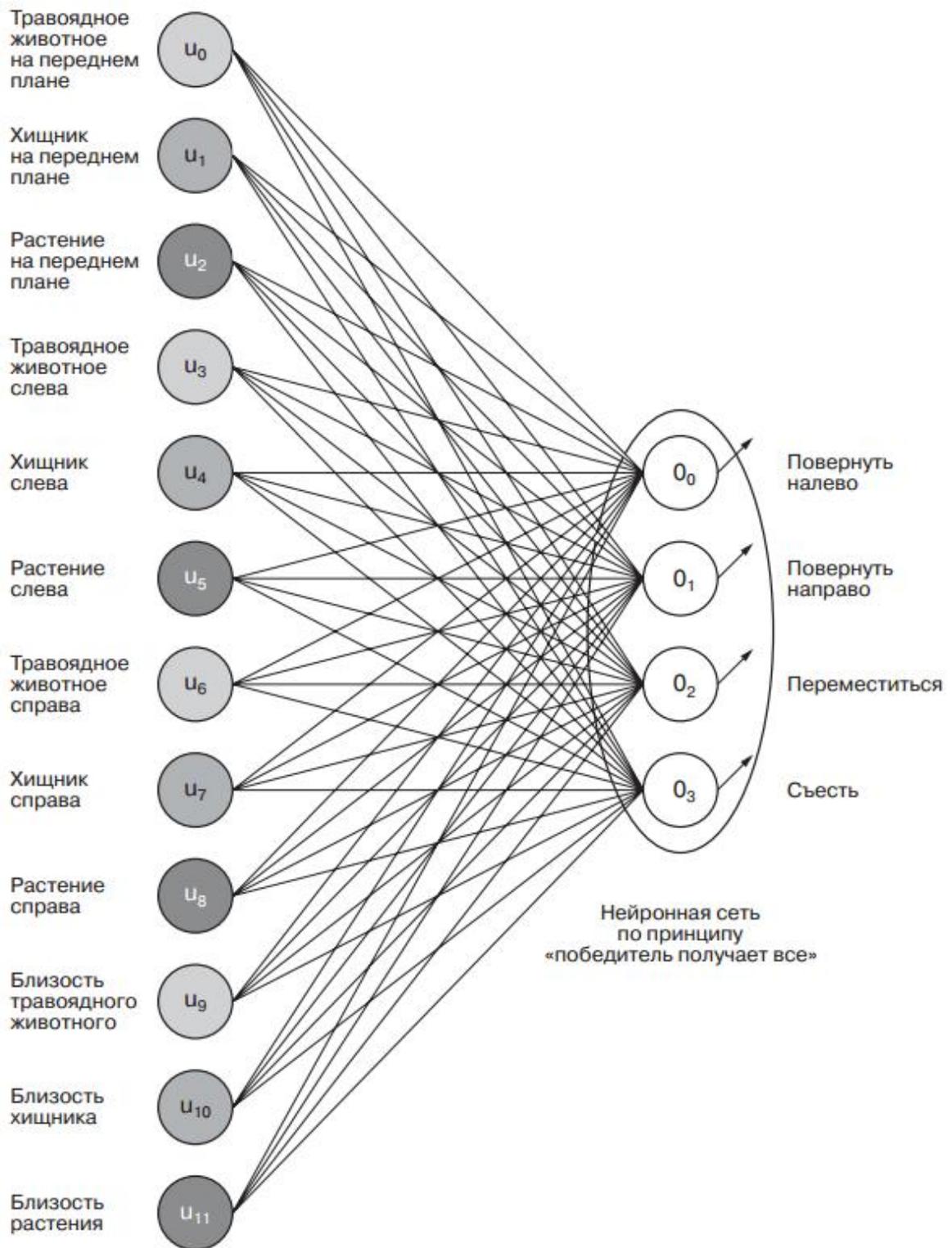


Рисунок 15 – нейроконтроллер агента в общем виде

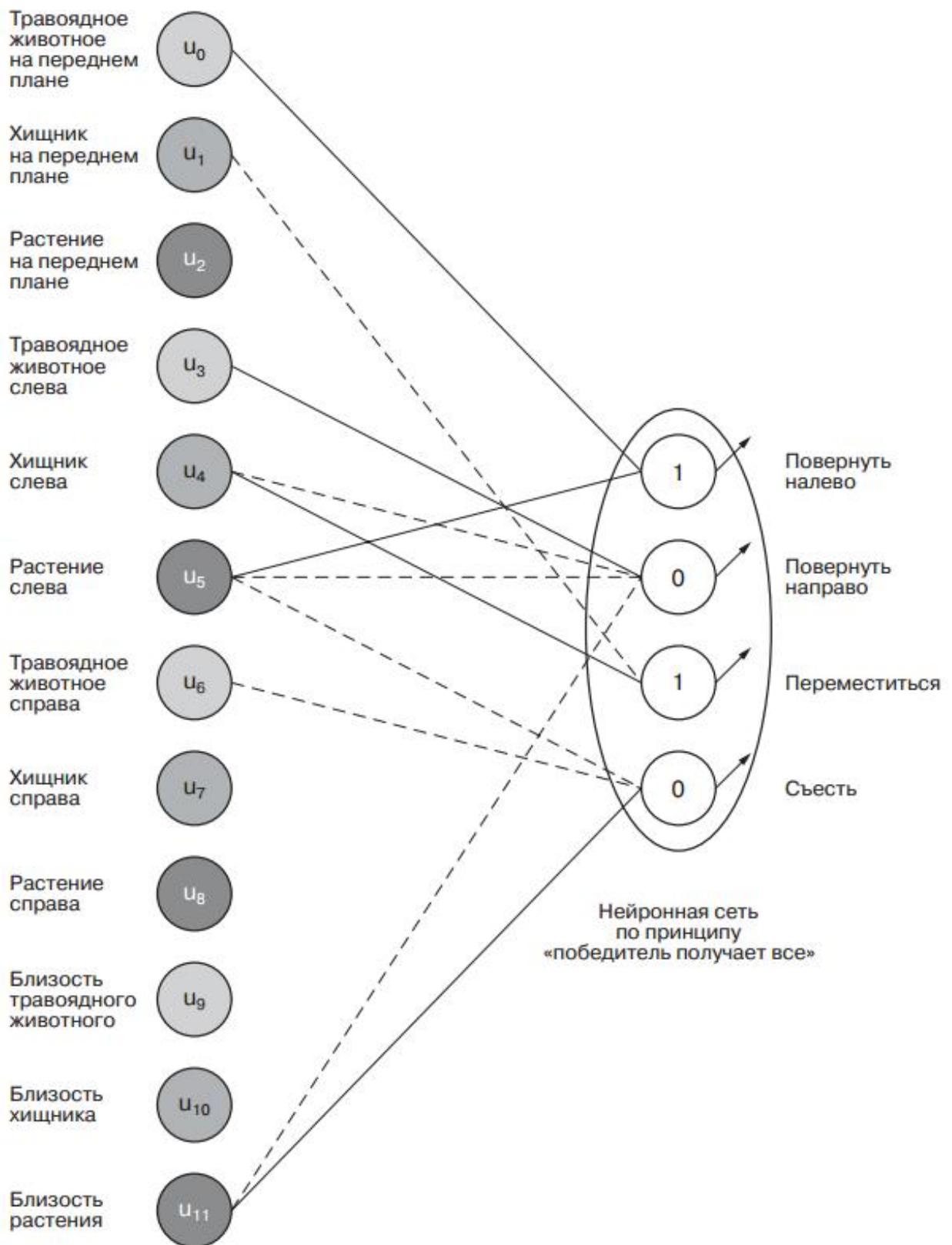


Рисунок 16 – Нейроконтроллер обученного травоядного

Задание: Реализовать программно искусственную жизнь с агентами на нейроконтроллерах.

Список литературы

1. Основы искусственного интеллекта [Электронный ресурс] / Боровская Е.В., Давыдова Н.А., - 3-е изд., (эл.) - М.:Лаборатория знаний, 2016. - 130 с.: ISBN 978-5-00101-421-8 - Режим доступа: <http://znanium.com/catalog/product/366789>

2. Автономный искусственный интеллект [Электронный ресурс] / Жданов А.А., - 4-е изд., (эл.) - М.:БИНОМ. ЛЗ, 2015. - 362 с.: ISBN 978-5-9963-2540-5 - Режим доступа: <http://znanium.com/catalog/product/539035>

3. Джонс М. Т. Программирование искусственного интеллекта в приложениях [Электронный ресурс] / М. Тим Джонс ; Пер. с англ. Осипов А. И. - М. : ДМК Пресс, 2011. - 312 с.: ил.: <http://znanium.com/bookread2.php?book=410211>