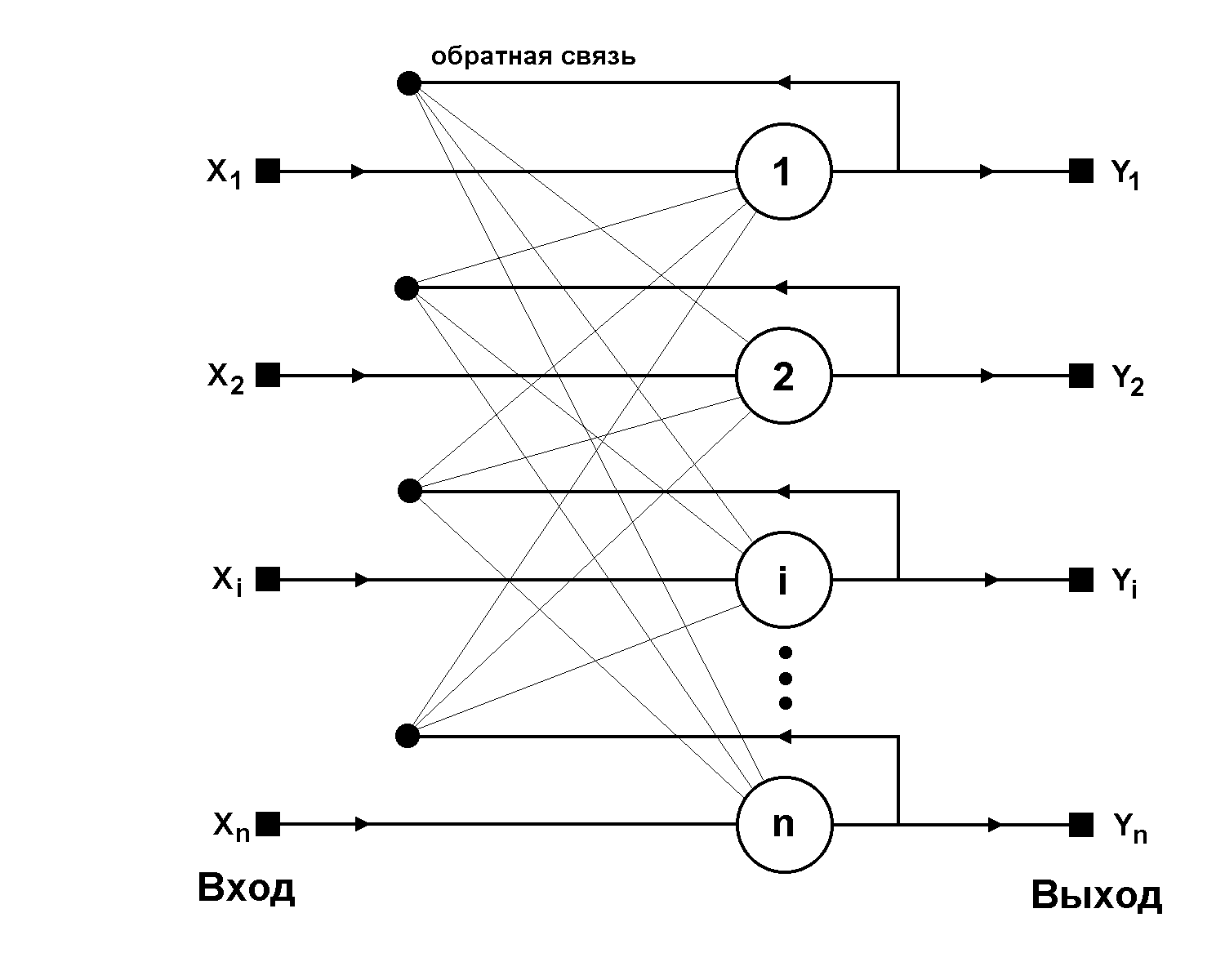
## Лабораторная работа №5.Сети Хопфилда и Хемминга

Среди различных конфигураций искусственных нейронных сетей (НС) встречаются такие, при классификации которых по принципу обучения, строго говоря, не подходят ни обучение с учителем [1], ни обучение без учителя [2]. В таких сетях весовые коэффициенты синапсов рассчитываются только однажды перед началом функционирования сети на основе информации об обрабатываемых данных, и все обучение сети сводится именно к этому расчету. С одной стороны, предъявление априорной информации можно расценивать, как помощь учителя, но с другой – сеть фактически просто запоминает образцы до того, как на ее вход поступают реальные данные, и не может изменять свое поведение, поэтому говорить о звене обратной связи с "миром" (учителем) не приходится. Из сетей с подобной логикой работы наиболее известны сеть Хопфилда и сеть Хэмминга, которые обычно используются для организации ассоциативной памяти. Далее речь пойдет именно о них.

**НС Хопфилда**

Структурная схема сети Хопфилда приведена на рис.1. Она состоит из единственного слоя нейронов, число которых является одновременно числом входов и выходов сети. Каждый нейрон связан синапсами со всеми остальными нейронами, а также имеет один входной синапс, через который осуществляется ввод сигнала. Выходные сигналы, как обычно, образуются на аксонах.



*Рис.1 Структурная схема сети Хопфилда*

Задача, решаемая данной сетью в качестве ассоциативной памяти, как правило, формулируется следующим образом. Известен некоторый набор двоичных сигналов (изображений, звуковых оцифровок, прочих данных, описывающих некие объекты или характеристики процессов), которые считаются образцовыми. Сеть должна уметь из произвольного неидеального сигнала, поданного на ее вход, выделить ("вспомнить" по частичной информации) соответствующий образец (если такой есть) или "дать заключение" о том, что входные данные не соответствуют ни одному из образцов. В общем случае, любой сигнал может быть описан вектором **X** = { xi: i=0...n-1}, n – число нейронов в сети и размерность входных и выходных векторов. Каждый элемент xi равен либо +1, либо -1. Обозначим вектор, описывающий k-тый образец, через **Xk**, а его компоненты, соответственно, – xik, k=0...m-1, m – число образцов. Когда сеть распознает (или "вспомнит") какой-либо образец на основе предъявленных ей данных, ее выходы будут содержать именно его, то есть **Y** = **Xk**, где **Y** – вектор выходных значений сети: **Y** = { yi: i=0,...n-1}. В противном случае, выходной вектор не совпадет ни с одним образцовым.

Если, например, сигналы представляют собой некие изображения, то, отобразив в графи­ческом виде данные с выхода сети, можно будет увидеть картинку, полностью совпадающую с одной из образцовых (в случае успеха) или же "вольную импровизацию" сети (в случае неудачи).

На стадии инициализации сети весовые коэффициенты синапсов устанавливаются следующим образом [3][4]:

 (1)

Здесь i и j – индексы, соответственно, предсинаптического и постсинаптического нейронов; *xik*, *xjk* – i-ый и j-ый элементы вектора k-ого образца.

Алгоритм функционирования сети следующий (p – номер итерации):

1. На входы сети подается неизвестный сигнал. Фактически его ввод осуществляется непо­сред­ственной установкой значений аксонов:

*yi(0) = xi , i = 0...n-1,* (2)

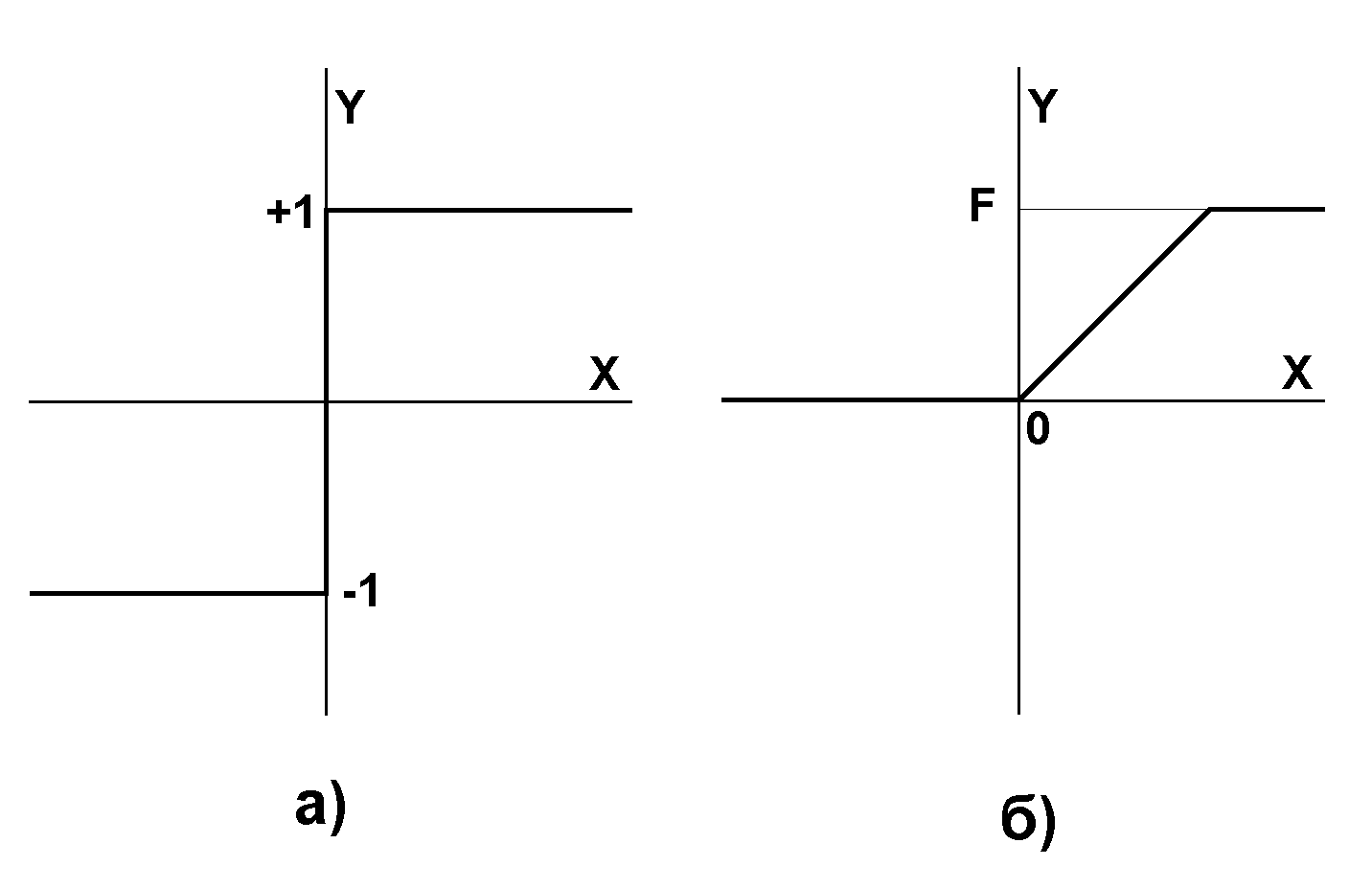
поэтому обозначение на схеме сети входных синапсов в явном виде носит чисто условный характер. Ноль в скобке справа от *yi* означает нулевую итерацию в цикле работы сети.

2. Рассчитывается новое состояние нейронов

*, j=0...n-1* (3)

и новые значения аксонов

 (4)



*Рис.2 Активационные функции*

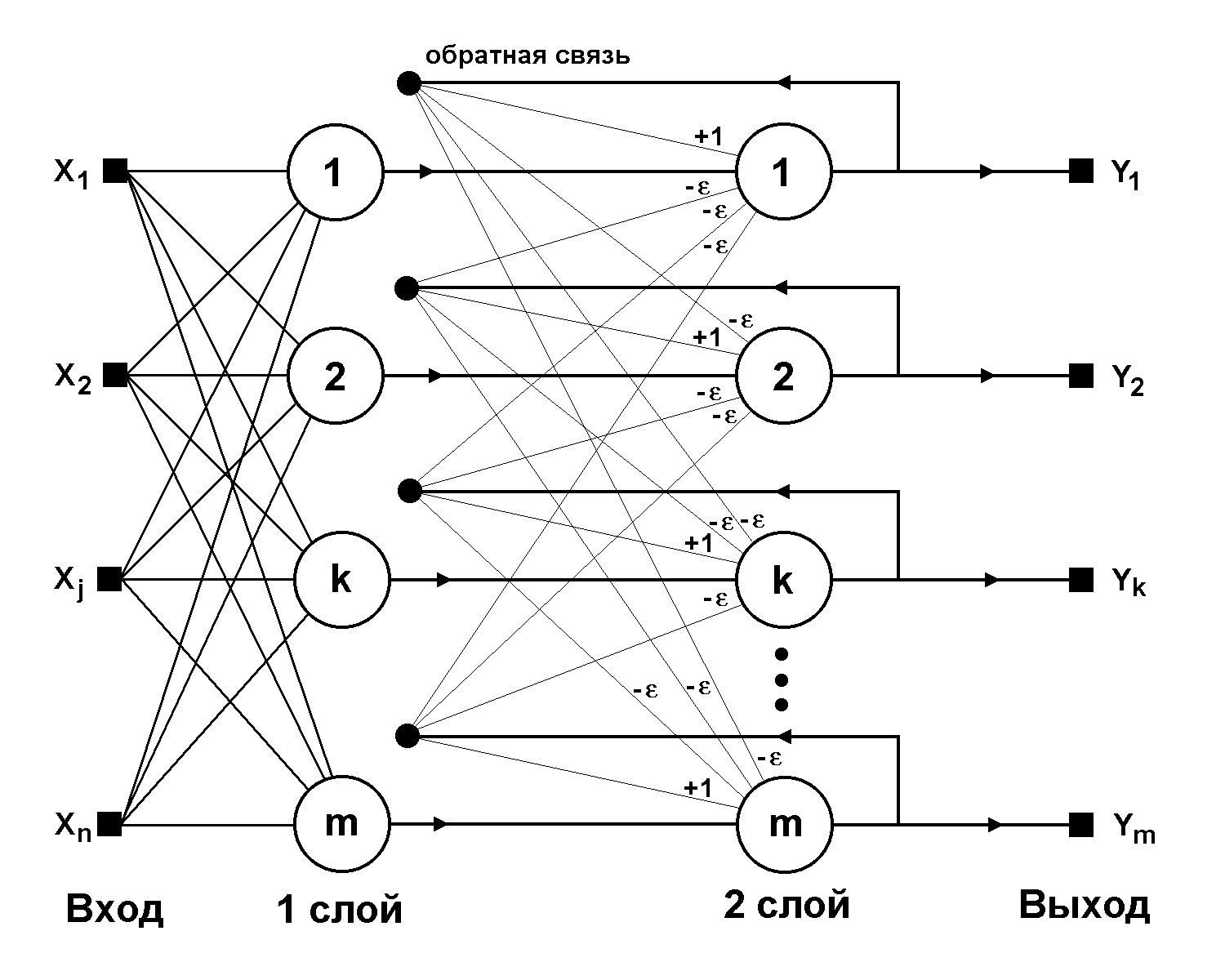
где *f* – активационная функция в виде скачка, приве­денная на рис.2а.

3. Проверка, изменились ли выходные значения аксонов за последнюю итерацию. Если да – переход к пункту 2, иначе (если выходы стабилизировались) – конец. При этом выходной вектор представляет собой образец, наилучшим образом сочетающийся с входными данными.

Как говорилось выше, иногда сеть не может провести распознавание и выдает на выходе несуществующий образ. Это связано с проблемой ограниченности возможностей сети. Для сети Хопфилда число запоминаемых образов m не должно превышать величины, примерно равной 0.15n. Кроме того, если два образа А и Б сильно похожи, они, возможно, будут вызывать у сети перекрестные ассоциации, то есть предъявление на входы сети вектора А приведет к появлению на ее выходах вектора Б и наоборот.

**НС Хемминга**

Когда нет необходимости, чтобы сеть в явном виде выдавала образец, то есть достаточно, скажем, получать номер образца, ассоциативную память успешно реализует сеть Хэмминга. Данная сеть характеризуется, по сравнению с сетью Хопфилда, меньшими затратами на память и объемом вычислений, что становится очевидным из ее структуры (рис. 3).



*Рис.3 Структурная схема сети Хэмминга*

Сеть состоит из двух слоев. Первый и второй слои имеют по m нейронов, где m – число образцов. Нейроны первого слоя имеют по n синапсов, соединенных с входами сети (образующими фиктивный нулевой слой). Нейроны второго слоя связаны между собой ингибиторными (отрицательными обратными) синаптическими связями. Единственный синапс с положительной обратной связью для каждого нейрона соединен с его же аксоном.

Идея работы сети состоит в нахождении расстояния Хэмминга от тестируемого образа до всех образцов. Расстоянием Хэмминга называется число отличающихся битов в двух бинарных векторах. Сеть должна выбрать образец с минимальным расстоянием Хэмминга до неизвестного входного сигнала, в результате чего будет активизирован только один выход сети, соответствующий этому образцу.

На стадии инициализации весовым коэффициентам первого слоя и порогу активационной функции присваиваются следующие значения:

*, i=0...n-1, k=0...m-1* (5)

*Tk* = *n / 2, k = 0...m-1* (6)

Здесь *xik* – i-ый элемент k-ого образца.

Весовые коэффициенты тормозящих синапсов во втором слое берут равными некоторой величине 0 < ε < 1/m. Синапс нейрона, связанный с его же аксоном имеет вес +1.

Алгоритм функционирования сети Хэмминга следующий:

1. На входы сети подается неизвестный вектор **X** = {xi:i=0...n-1}, исходя из которого рассчитываются состояния нейронов первого слоя (верхний индекс в скобках указывает номер слоя):

*, j=0...m-1* (7)

После этого полученными значениями инициализируются значения аксонов второго слоя:

*yj(2) = yj(1), j = 0...m-1* (8)

2. Вычислить новые состояния нейронов второго слоя:

 (9)

и значения их аксонов:

 (10)

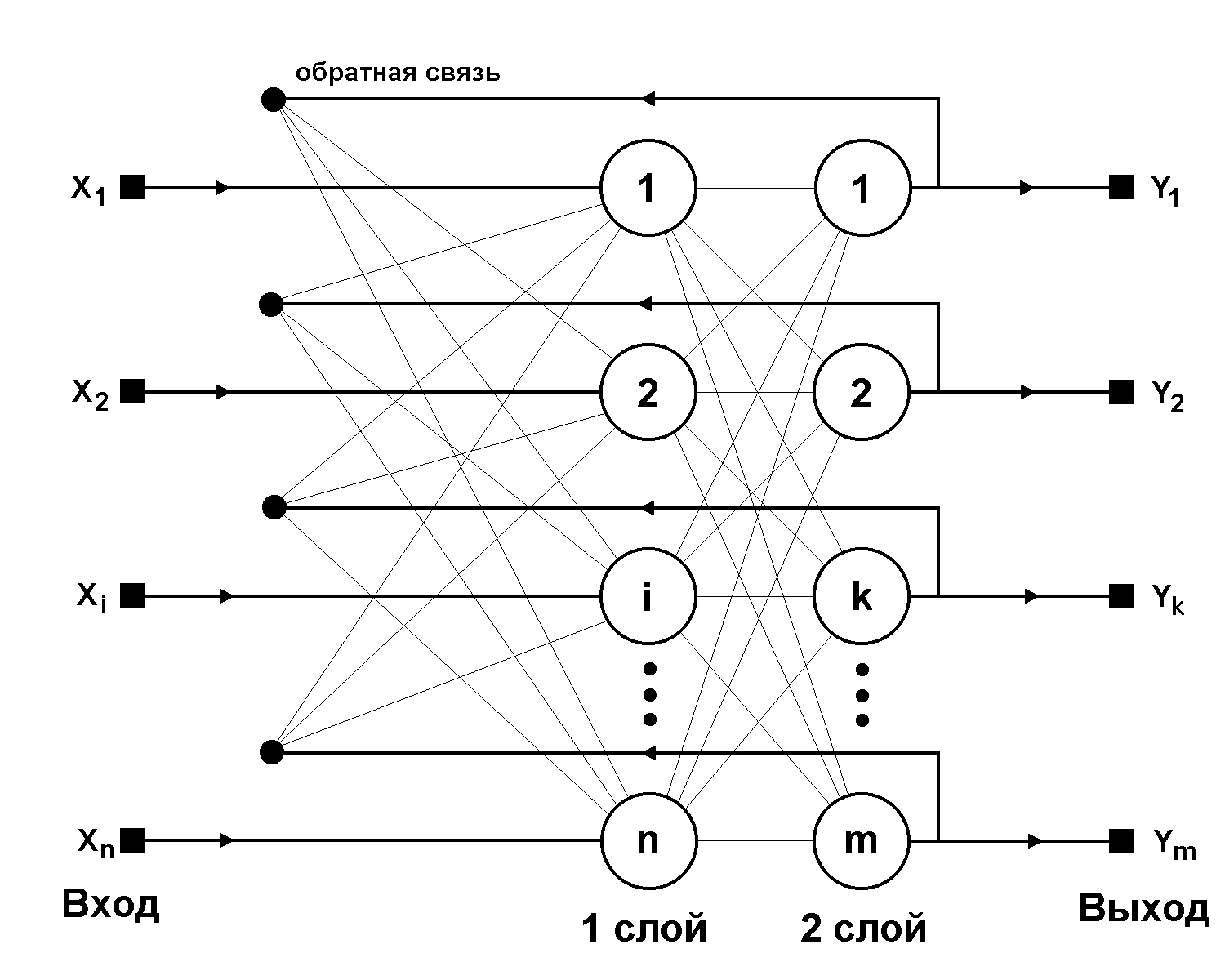
Активационная функция f имеет вид порога (рис. 2б), причем величина F должна быть достаточно большой, чтобы любые возможные значения аргумента не приводили к насыщению.

3. Проверить, изменились ли выходы нейронов второго слоя за последнюю итерацию. Если да – перейди к шагу 2. Иначе – конец.

Из оценки алгоритма видно, что роль первого слоя весьма условна: воспользовавшись один раз на шаге 1 значениями его весовых коэффициентов, сеть больше не обращается к нему, поэтому первый слой может быть вообще исключен из сети (заменен на матрицу весовых коэффициентов).

**Двунаправленная ассоциативная память**

Обсуждение сетей, реализующих ассоциативную память, было бы неполным без хотя бы краткого упоминания о двунаправленной ассоциативной памяти (ДАП). Она является логичным развитием парадигмы сети Хопфилда, к которой для этого достаточно добавить второй слой. Структура ДАП представлена на рис.5.



*Рис.5 Структурная схема ДАП*

Сеть способна запоминать пары ассоциированных друг с другом образов. Пусть пары образов записываются в виде векторов **Xk** = {xik:i=0...n-1} и **Yk** = {yjk: j=0...m-1}, k=0...r-1, где r – число пар. Подача на вход первого слоя некоторого вектора **P** = {pi:i=0...n-1} вызывает образование на входе второго слоя некоего другого вектора **Q** = {qj:j=0...m‑1}, который затем снова поступает на вход первого слоя. При каждом таком цикле вектора на выходах обоих слоев приближаются к паре образцовых векторов, первый из которых – **X** – наиболее походит на **P**, который был подан на вход сети в самом начале, а второй – **Y** – ассоциирован с ним. Ассоциации между векторами кодируются в весовой матрице **W(1)** первого слоя. Весовая матрица второго слоя **W(2)** равна транспонированной первой (**W(1)**)T. Процесс обучения, также как и в случае сети Хопфилда, заключается в предварительном расчете элементов матрицы **W** (и соответственно **WT**) по формуле:

 (11)

Эта формула является развернутой записью матричного уравнения

 (12)

для частного случая, когда образы записаны в виде векторов, при этом произведение двух матриц размером соответственно [n\*1] и [1\*m] приводит к (11).

В заключении можно сделать следующее обобщение. Сети Хопфилда, Хэмминга и ДАП позволяют просто и эффективно разрешить задачу воссоздания образов по неполной и искаженной информации. Невысокая емкость сетей (число запоминаемых образов) объясняется тем, что, сети не просто запоминают образы, а позволяют проводить их обобщение, например, с помощью сети Хэмминга возможна классификация по критерию максимального правдоподобия [3]. Вместе с тем, легкость построения программных и аппаратных моделей делают эти сети привлекательными для многих применений.

#### Задания для самостоятельной работы

* По заданию преподавателя на основе архитектур и алгоритмов обучения указанных выше реализовать НС для интерпретации входных данных представляющих буквы какого-либо алфавита.
* Результаты оформить в виде отчета и защитить на следующем занятии.

#### Контрольные вопросы

1. Изложите основные концепции архитектуры сети Хопфилда.
2. Изложите основные концепции архитектуры сети Хемминга.
3. Изложите основные концепции архитектуры ДАП.
4. Что можно сказать о емкости приведенных выше НС?
5. Каковы достоинства процедуры сетей Хемминга и Хопфилда?
6. Каковы недостатки сетей Хопфилда, Хемминга, ДАП и как они преодолеваются?

#### Литература

1. С. Короткий, Нейронные сети: алгоритм обратного распространения.
2. С. Короткий, Нейронные сети: обучение без учителя.
3. Artificial Neural Networks: Concepts and Theory, IEEE Computer Society Press, 1992.
4. Ф.Уоссермен, Нейрокомпьютерная техника, М.,Мир, 1992.

# Список литературы:

1. Интеллектуальные информационные системы : учебное пособие / А. А. Смагин, С. В. Липатова, А. С. Мельниченко. – Ульяновск : УлГУ, 2010. – 136 с.
2. [Нейроинформатика](http://neuroschool.narod.ru/books/neurinf.html) / А. Н. Горбань, В. Л. Дунин-Барковский, А. Н. Кирдин и др. — Новосибирск: Наука. Сибирское предприятие РАН, 1998. — 296 с
3. Нейронные сети и нейрокомпьютеры: Учебное пособие по курсу «Микропроцессоры» / Круг П.Г. – М.: Издательство МЭИ, 2002. – 176 с.
4. Ф. Уоссермен. Нейрокомпьютерная техника: теория и практика. М. Мир - 1992.
5. McCulloch W. W., Pitts W. 1943. A logical calculus of the ideas imminent in nervous activiti. Bulletin of Mathematical Biophysics 5:115-33. (Русский перевод: Маккаллок У. С., Питтс У. Логическое исчисление идей, относящихся к нервной деятельности. Автоматы. – М.: ИЛ. – 1956).
6. Колмогоров А.Н. О представлении непрерывных функций нескольких переменных суперпозициями непрерывных функций меньшего числа переменных // Докл. АН СССР, том 108, с. 2, 1956.
7. Арнольд В.И. // Докл. АН СССР, том 114, N 4, 1957.
8. Колмогоров А.Н. О представлении непрерывных функций нескольких переменных в виде суперпозиций непрерывных функций одного переменного и сложения // Докл. АН СССР, том 114, с. 953-956, 1957.
9. Kolmogorov A.N. On the Representation of Continuous Functions of Many Variables by Superposition of Continuous Functions of One Variable and Addition, American Math. Soc. Transl., 28 (1963), pp. 55-63.
10. Hecht-Nielsen R. Kolmogorov's Mapping Neural Network Existence Theorem // IEEE First Annual Int. Conf. on Neural Networks, San Diego, 1987, Vol. 3, pp. 11-13.
11. Muller B., Reinhart J. Neural Networks: an introduction, Springer-Verlag, Berlin Heidelberg, 1990.
12. Hebb D. O. 1949. Organization of behavior. New York: Science Editions.
13. Parker D. B. 1987. Second order back propagation: Implementing an optimal *0(n)* approximation to Newton's method as an artificial newral network. Manuscript submitted for publication.
14. Pineda F. J. 1988. Generalization of backpropagation to recurrent and higher order networks. In Newral information processing systems, ed. Dana Z. Anderson, pp. 602-11. New York: American Institute of Phisycs.
15. Rumelhart D. E., Hinton G. E., Williams R. J. 1986. Learning internal reprentations by error propagation. In Parallel distributed processing, vol. 1, pp. 318-62. Cambridge, MA: MIT Press.