Сеть Хеминга

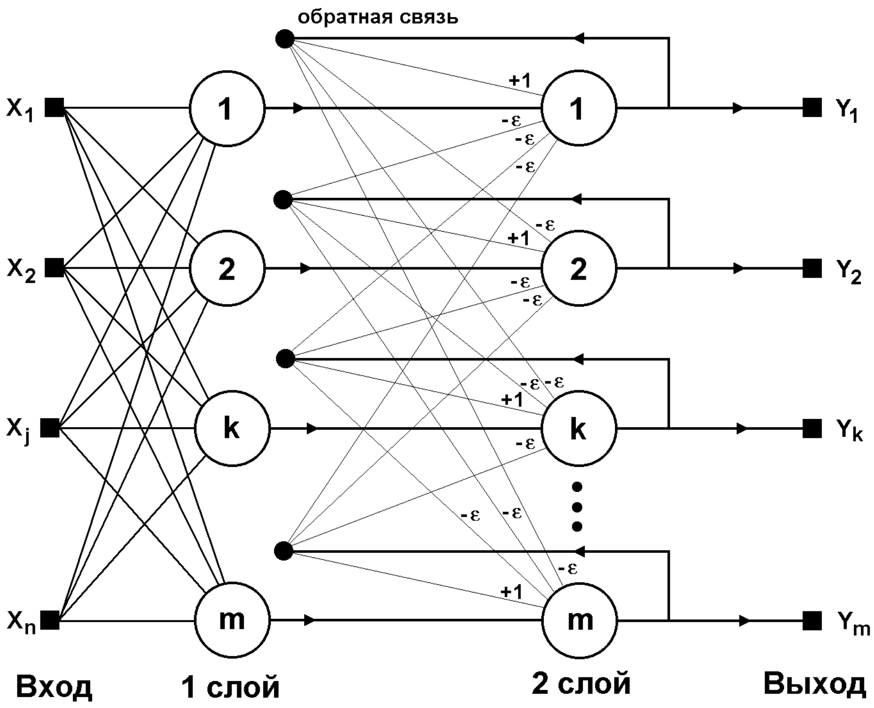

Рис.3 Структурная схема сети Хэмминга

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).

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 со 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:

$w_{ik} = \frac{x_i^k}{2}$, $i=0...n-1$, $k=0...m-1$ (5)

$T_k = n / 2$, $k = 0...m-1$ (6)

Здесь x_i^k – i-ый элемент k-ого образца.

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

Алгоритм функционирования сети Хэмминга следующий:

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

$$y_j^{(1)} = s_j^{(1)} = \sum_{i=0}^{n-1} w_{ij} x_i + T_j, \quad j=0...m-1 \quad (7)$$

После этого полученными значениями инициализируются значения аксонов второго слоя:

$$y_j^{(2)} = y_j^{(1)}, \quad j = 0...m-1 \quad (8)$$

2. Вычислить новые состояния нейронов второго слоя:

$$s_j^{(2)}(p+1) = y_j(p) - \varepsilon \sum_{k=0}^{m-1} y_k^{(2)}(p), k \neq j, j = 0 \ldots m-1 \quad (9)$$

и значения их аксонов:

$$y_j^{(2)}(p+1) = f\left[s_j^{(2)}(p+1)\right], j = 0 \ldots m-1 \quad (10)$$

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), что и было сделано в ее конкретной реализации, описанной ниже.

Программная модель сети Хэмминга строится на основе набора специальных классов NeuronHN, LayerHN и NetHN – производных от классов, рассмотренных в предыдущих статьях цикла [1][2]. Описания классов приведены в листинге 1. Релизации всех функций находятся в файле NEURO_HN (листинг 2). Классы NeuronHN и LayerHN наследуют большинство методов от базовых классов.

В классе NetHN определены следующие элементы:

Nin и Nout – соответственно размерность входного вектора с данными и число образцов;

dx и dy – размеры входного образа по двум координатам (для случая трехмерных образов необходимо добавить переменную dz), dx*dy должно быть равно Nin, эти переменные используются функцией загрузки данных из файла LoadNextPattern;

DX и DY – размеры выходного слоя (влияют только на отображение выходого слоя с помощью функции Show); обе пары размеров устанавливаются функцией SetDxDy;

Class – массив с данными об образцах, заполняется функцией SetClasses, эта функция выполняет общую инициализацию сети, сводящуюся к запоминанию образцовых данных.

Метод Initialize проводит дополнительную инициализацию на уровне тестируемых данных (шаг 1 алгоритма). Метод Cycle реализует шаг 2, а метод IsConverged проверят, застабилизировались ли состояния нейронов (шаг 3).

Из глобальных функций – SetSigmoidAlfaHN позволяет установить параметр F активационной функции, а SetLimitHN задает коэффициент, лежащий в пределах от нуля до единицы и определяющий долю величины 1/m, образующую ε .

На листинге 3 приведена тестовая программа для проверки сети. Здесь конструируется сеть со вторым слоем из пяти нейронов, выполняющая распознавание пяти входных образов, которые представляют собой схематичные изображения букв размером 5 на 6 точек (см.рис.4а). Обучение сети фактически сводится к загрузке и запоминанию идеальных изображений, записанных в файле "charh.img", приведенном на листинге 4. Затем на ее вход поочередно подаются зашумленные на 8/30 образы (см.рис.4б) из файла "charhh.img" с листинга 5, которые она успешно различает.

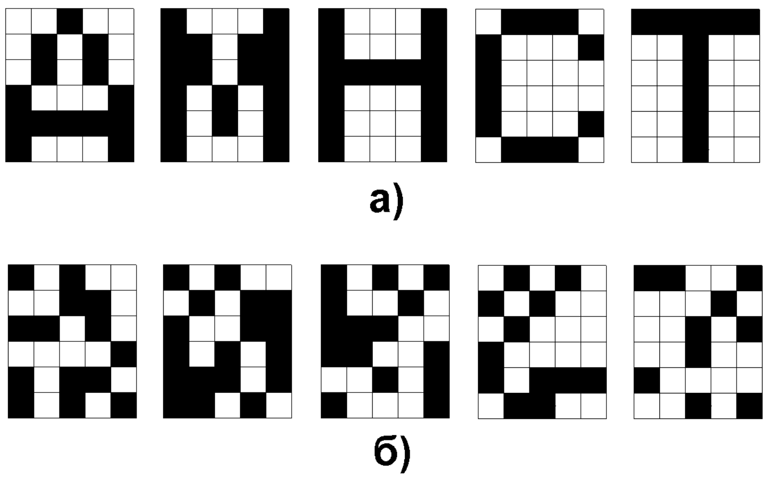

Рис. 4 Образцовые и тестовые образы

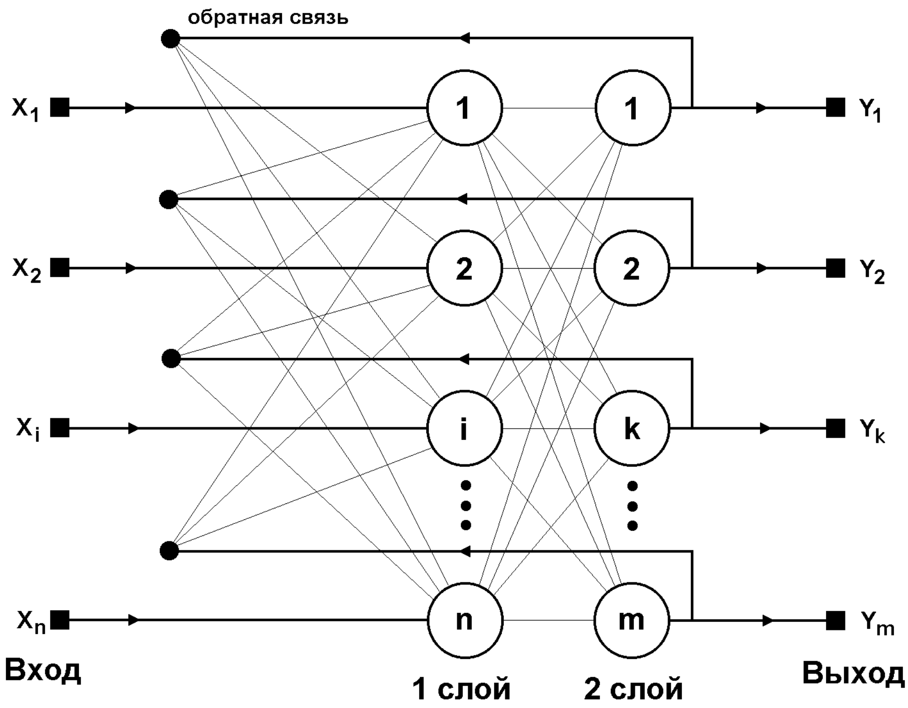

Рис.5 Структурная схема ДАП

В проект кроме файлов NEURO_HN и NEUROHAM входят также SUBFUN и NEURO_FF, описанные в [1]. Программа тестировалась в среде Borland C++ 3.1.

Предложенные классы позволяют моделировать и более крупные сети Хэмминга. Увеличение числа и сложности распознаваемых образов ограничивается фактически только объемом ОЗУ. Следует отметить, что обучение сети Хэмминга представляет самый простой алгоритм из всех рассмотренных до настоящего времени алгоритмов в этом цикле статей.

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. Сеть способна запоминать пары ассоциированных друг с другом образов. Пусть пары образов записываются в виде векторов $\mathbf{X^k} = \{x_i^k\text{:}i=0...n\text{-}1\}$ и $\mathbf{Y^k} = \{y_j^k\text{: } j=0...m\text{-}1\}$, k=0...r-1, где r – число пар. Подача на вход первого слоя некоторого вектора $\mathbf{P} = \{p_i\text{:}i=0...n\text{-}1\}$ вызывает образование на входе второго слоя некоего другого вектора $\mathbf{Q} = \{q_j\text{:}j=0...m\text{-}1\}$, 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 $\mathbf{W^{(1)}}$ первого слоя. Весовая матрица второго слоя $\mathbf{W^{(2)}}$ равна транспонированной первой $(\mathbf{W^{(1)}})^T$. Процесс обучения, также как и в случае сети Хопфилда, заключается в предварительном расчете элементов матрицы **W** (и соответственно $\mathbf{W^T}$) по формуле:

$$w_{ij} = \sum_k x_i y_j,\ i = 0 \ldots n-1,\ j = 0 \ldots m-1 \qquad (11)$$

Эта формула является развернутой записью матричного уравнения

$$\mathbf{W} = \sum_{k} \mathbf{X}^{\mathbf{T}}\mathbf{Y} \qquad (12)$$

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 обощение, 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.

Литература

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