**Нейронная сеть встречного распространения**

* 1. **Описание нейронной сети встречного распространения**

Многие алгоритмы обучения нейронных сетей (НС) характерны тем, что необходимым участником процесса обучения является «учитель», который заранее произво­дит классификацию используемых при обучении объектов и результаты этой классификации вводит в машину. Однако в биологических системах такой ме­ханизм обучения трудно себе представить. Способность разных людей одинаково проводить классификацию «без учителя» заставляет думать, что различия, на которых основывается такая способность, имеют объектив­ный и безусловный характер.

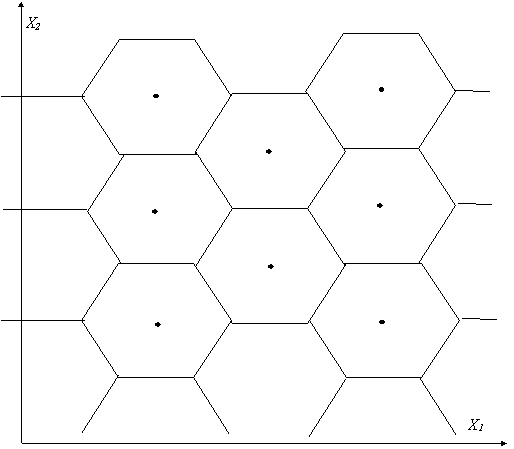
Для реализации этих алгоритмов на практике необходимо решить два воп­роса. Первый связан с выбором способа разбиения объектов на классы без учителя, второй — с выработкой правила отнесения входного образа к опреде­ленному классу. Необходимо описать способ проведения границы между множествами объек­тов и сформулировать алгоритм ее построения. Процесс разбиения некоторого множества объектов на классы называется кластеризацией.

Примером НС, использующей алгоритм обучения без учителя, является введенная Т. Кохоненом (1982) «самоорганизующаяся карта признаков» (Self-Organizing Feature Maps, SOM). Другое, часто используемое название сети Кохонена, — KCN (Kohonen Clustering Networks). KCN используют для отображе­ния нелинейных взаимосвязей данных на достаточно легко интерпретируемые (чаще всего двумерные) сетки, представляющие метрические и топологичес­кие зависимости входных векторов, объединяемых в кластеры.

Однако большей популярностью пользуются гибридные НС, которые представляют собой объединение различного рода сетей и концепций их обучения. Одной из широко используемых комбинированных сетей, использующихся на практике, является двухслойная сеть встречного распространения (CPN — Counterpropagation Network), первым слоем которой является самоорганизующаяся сеть Кохонена, а вторым — выходная звезда С. Гроссберга. Сети данного вида успешно применяют в финансовых и экономических приложениях, таких, как рассмотрение заявок на предоставление займов, пред­сказание трендов цен акций, товаров и курсов обменов валют. Можно ожидать успешного применения CPN-сети в задачах интерполяции.

1.2.1 Нейронные сети  Кохонена

Данные сети позволяют в результате обучения осуществлять топологически непрерывное отображение *F* входного *n*-мерного пространства в выходное *m*-мерное пространство; т.е. . При этом обучение здесь происходит «без учителя» на основе образов поступающих на сеть. Для обучения самоорганизующихся нейронных сетей используется *конкурентный метод*, который был предложен в 1976 году С. Гроссбергом (S. Grossberg) и затем развит в работах финского ученого Т. Кохонена (T. Kohonen) [1]. По мере поступления входных образов на такую сеть посредством обучения происходит разбиение *n*-мерного входного пространства на различные области решений, каждой из которых соответствует отдельный нейрон. Границы отдельной области перпендикулярны линиям, проведенным между центроидами соседних областей решений. Такое разделение пространства называется диаграммой Вороного (Voronoi) или картами Кохонена. Для двухмерного случая () область решений представляет собой правильные шестиугольники (рисунок 1.1), в результате чего получается наименьшая ошибка. Для  наилучшая форма областей решений является неизвестной [2].



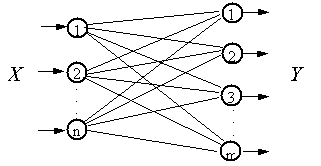
*Рисунок 1.1 — Разбиение входного пространства образов*

Таким образом, самоорганизация таких сетей происходит в результате топологического упорядочивания входной информации по различным кластерам, количество которых является *m*.

Топологическое упорядочивание информации напоминает процессы происходящие в головном мозге при его развитии (введение), когда осуществляется формирование топологически упорядоченных нейронных структур.

1.2.2 Конкурентное обучение

Конкурентное обучение (*competitive learning*) является основным методом для обучения нейронных сетей Кохонена. Конкурентная нейронная сеть в общем случае представляет собой двухслойную нейронную сеть с прямыми связями (рисунок 1.2). Первый слой выполняет чисто распределительные функции, причем каждый нейрон его имеет соединения со всеми нейронными элементами выходного слоя. Во втором слое происходит конкуренция между нейронными элементами, в результате которой определяется нейрон-победитель.



*Рисунок 1.2 — Топология конкурентной нейронной сети*

Для *нейрона-победителя* синаптические связи усиливаются, а для остальных нейронов не изменяются или могут уменьшаться. Победителем в конкуренции является нейрон, который в результате подачи на вход сети определенного образа имеет максимальную взвешенную активность



где  — входной образ,  — вектор столбец весовых коэффициентов *j*-го выходного нейрона. Пусть

.

Тогда активность выходных нейронов определяется по формуле .



где 

Таким образом, после обучения нейронной сети при подаче входного образа активность нейрона-победителя принимается равной единице, а остальных нейронов нулю. Это правило известно под названием ***«****победитель берет все*» (winner take all). Выражение эквивалентно скалярному произведению вектора весов соответствующего нейронного элемента на входной вектор нейронной сети

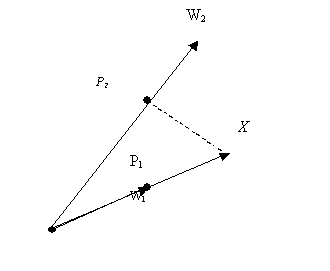


где ,  и  — модули векторов и .

Обозначим , где *Р* — проекция вектора *Х* на вектор *W*. Тогда



Если векторы и  не нормированы, то происходит неадекватное определение нейрона победителя (рисунок 1.3). Как следует из рисунка, нейрон, вектор весов которого  больше отличается от входного образа *Х* чем нейрон, имеющий весовой вектор , становится победителем.



*Рисунок 1.3 — Геометрическая интерпретация определения нейрона победителя при ненормированном весовом векторе и входном образе*

Поэтому, при определении нейрона-победителя по взвешенной активности , необходимо нормализовать весовые и входные векторы для каждого нейрона. Нормализация осуществляется следующим образом:





Тогда взвешенную активность *j-*го нейрона можно представить как

.

Из этого выражения следует, что максимальную активность будет иметь тот нейрон, весовой вектор которого коллинеарен входному вектору. Концы векторов при этом находятся на поверхности *n-*мерной сферы (гиперсферы), радиус которой равен 1.

В выражении взвешенная сумма эквивалентна коэффициенту взаимной корреляции между входным и весовым вектором. Он будет равен 1, когда угол между векторами равен нулю. Отсюда следует, что правило настройки весовых коэффициентов нейрона-победителя должно соответствовать вращению вектора *Wk* в сторону вектора *X* .

В результате можно записать правило обучения для вектора весов нейрона-победителя .

,

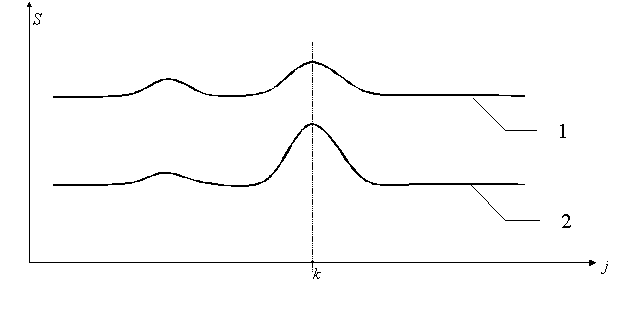
где  характеризует скорость обучения.

В качестве нейрона победителя выбирается такой нейрон, весовой вектор которого наиболее близок к входному вектору. В обычной форме правило обучения для *k*-го нейрона-победителя можно представить, как



где 

При применении данного правила к *k*-му нейрону усиливается его выходная активность (рисунок 1.4).



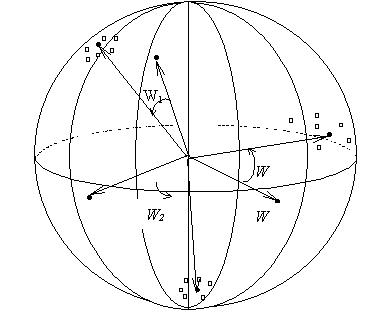
*Рисунок 1.4 — Профили активности нейронов.*

*1 — профиль активности нейронов до обучения; 2 — после обучения.*

Так как весовой вектор должен быть нормированным, то правило изменения весового вектора модифицируется по выражению .

.

При применении этого правила для обучения нейронной сети весовые векторы нейронов будут вращаться в направлении кластеров входных образов, как показано на рисунке 1.5.



*Рисунок 1.5 — Изменение весовых векторов в процессе обучения:*

*р — вектор образцов; w — вектор весов.*

В случае использования ненормализованных векторов для определения нейрона победителя нужно оперировать вместо взвешенной активности евклидовым расстоянием.

.

При помощи определяется нейрон победитель с номером *k*, который соответствует минимальному евклидовому расстоянию между входным и весовым вектором.

.

Тогда настройка весового вектора нейрона-победителя происходит по формуле .



При использовании выражения для обучения одного нейронного элемента на интервале входных значений [a,b] (x∈ [a,b]) вес нейрона с течением времени стремиться к середине интервала.

Недостатком описанного выше метода обучения является то, что при случайной инициализации весовых векторов, может получиться так, что некоторые нейроны никогда не будут победителями. Для нейтрализации этого недостатка можно расширить правило обучения по формуле .

,

где . Данное правило позволяет отобразить весовые векторы побежденных нейронов в такую область, где увеличиваются их шансы в конкуренции. Другим вариантом является частотно-чувствительное конкурентное обучение (sensitive competitive learning). Здесь для каждого нейрона ведется статистика его побед. Пусть  — частота нахождения *j*-го нейрона в состоянии победителя. Тогда нейрон победитель определяется по формуле .

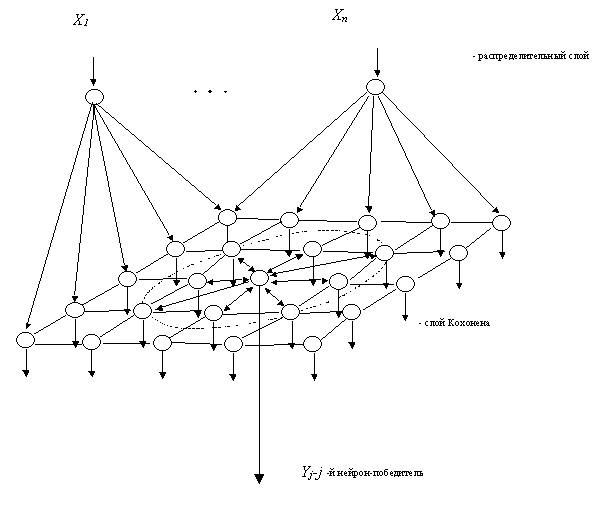
.

Чем чаще нейрон становится победителем, тем меньше шансов он имеет в конкуренции.

Итак, при конкурентном обучении все множество входных образов разбивается на кластеры, каждому из которых соответствует свой нейрон. При поступлении на вход НС неизвестного образа, она будет его относить к такому кластеру, на который он больше всего похож. В этом заключается обобщающая способность такого типа НС. В 1978 г. С. Гроссберг доказал сходимость конкурентных методов обучения [3].

1.2.3 Самоорганизующиеся карты Кохонена

Самоорганизующиеся карты Кохонена являются дальнейшим расширением нейронных сетей с конкурентным обучением. Они были разработаны Кохоненом в 1982 году. Топология НС Кохонена состоит из двух слоев. Первый слой выполняет распределительные функции, а нейроны второго слоя расположены на плоскости образуя матрицу (рисунок 1.6).



*Рисунок 1.6 — Топология нейронной сети Кохонена*

В своем простейшем виде сеть Кохонена функционирует по принципу «победитель берет все». При этом она должна выполнять топологически упорядоченное отображение входных векторов на матрицу нейронов второго слоя. Соседние наиболее похожие входные образы должны отображаться на соседние нейроны матрицы. Это достигается путем введения области притяжения *G* для нейрона победителя, в радиусе действия которой нейроны активно изменяют свои весовые векторы в сторону входного образа. Область притяжения можно описать функцией притяжения, которую обозначим . Здесь *t* — время, *k* — номер нейрона-победителя, *p* — номер искомого нейрона. В дискретном варианте функция притяжения определяется следующим образом:



В область притяжения нейрона *k* входят все нейроны, находящиеся на определенном расстоянии от нейрона победителя. В непрерывном варианте часто используется функция Гаусса (рисунок 1.7).

*Рисунок 1.7 — Двухмерная функция Гаусса*

Она определяется при помощи выражения .

,

где  — расстояние между нейронами, *σ*(*t*) — среднеквадратичное отклонение (радиус области притяжения). Положение каждого нейрона в матрице характеризуется его координатами.

.

Тогда

.

В процессе обучения нейронной сети Кохонена изменяются весовые коэффициенты не только нейрона победителя, но и всех нейронов внутри области притяжения. Так, для *p*-ого нейрона весовой вектор изменяется по закону .

.

С увеличением времени обучения радиус области притяжения уменьшается. В результате нейронные элементы сжимаются около нейрона победителя, пока он не останется один. Это изображено на рисунке 1.8, где  — область притяжения в момент времени *t*. Введем декартову систему координат, так, что каждый нейрон имеет координаты , где .

*Рисунок 1.8 — Изменение области притяжения с течением времени*

Поставим в соответствие нейрону с координатами (*i*, *j*) весовой вектор *W*ij, который определяется по формуле .

,

где *n* — количество нейронных элементов входного слоя.

Тогда процедуру обучения сети Кохонена для непрерывной функции притяжения можно представить в виде следующей последовательности действий [4]:

1. Случайно инициализируются весовые коэффициенты *W* нейронной сети.

2. Задается начальное значение радиуса притяжения *s* и момент времени .

3. Подается входной образ  на нейронную сеть и вычисляется норма вектора: , где  — весовой вектор нейрона с координатами .

4. Определяется нейрон победитель: , где  — координаты нейрона победителя.

5. Для каждого нейрона производится вычисление функции притяжения.

,

где .

6. В соответствии с функцией притяжения осуществляется модификация весовых коэффициентов: , где .

7. Переходим к пункту 3 и повторяем процедуру для где *L*- общее количество входных образов.

8. Увеличиваем на единицу квант времени, уменьшаем радиус области притяжения *s* и повторяем процесс, начиная с пункта 3.

Обучение производится до получения желаемой степени согласования между весовыми и выходными векторами. Начальное значение области притяжения *s* может охватывать всю матрицу нейронов, а затем последовательно уменьшается, как показано на рисунке 1.8. Для дискретной функции притяжения процедура обучения является аналогичной. Функционирование такой сети происходит путем определения нейрона победителя и присвоения ему единичного значения, а остальным нейронам нулевого значения.

В процессе обучения происходит упорядочивание весовых коэффициентов таким образом, что уменьшается разница между весами соседних нейронов. Покажем это. Пусть весовые векторы соседних нейронов изменяется как

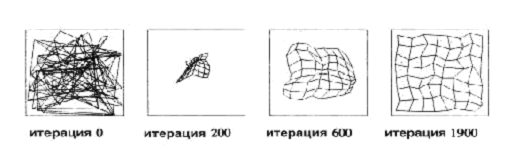
.

Расстояние между ними равно

.

Так как , то

.

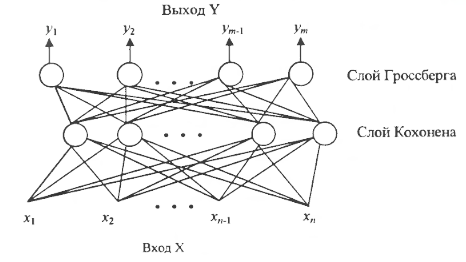


*Рисунок 1.9 — Изменение весовых коэффициентов сети Кохонена*

Таким образом, в процессе обучения разница между весовыми векторами топологически близких нейронов уменьшается. Это показано на рисунке 1.9 для сети с двумя входными нейронами и  выходными нейронами. Линии соединяют значения весов нейрона с координатами с весами нейрона и . В начале  веса выбраны случайно и беспорядочно распределены на плоскости. Затем веса изменяются, так что их плотность приблизительно соответствует плотности вероятности входных векторов.

1.2.4 Сети встречного распространения

Сеть со встречным распространением CPN представляет собой соединение самоорганизующейся сети Кохонена и выходной звезды Гроссберга. Тополо­гия сети встречного распространения приведена на рисунке 1.10. В рамках этой архитектуры элементы сети Кохонена не являются выходом сети, а служат лишь входами для выходного слоя — выходной звезды Гроссберга. Создатель сети встречного распространения Р. Хехт-Нильсен (R. HehtNilsen, 1987) рекомен­дует использовать эти архитектуры для решения задач аппроксимации функ­ций и заполнения пробелов в таблице данных.



*Рисунок 1.10 — Топология сети встречного распространения.*

Идеи С. Гроссберга, связанные с исследованиями в области биологической кибернетики, реализуются во многих НС. В частности, конфигурации входных и выходных звезд Гроссберга используют для созда­ния иерархических НС.

Нейрон в форме входной звезды имеет и входов, которым соответствуют весовые коэффициенты  и один выход **Y**, являющийся взве­шенной суммой входов. Входная звезда обучается выдавать сигнал на выходе всякий раз, когда на входы поступает определенный вектор. Таким образом, входная звезда является своего рода детектором состояния входов и реагирует только на свой входной вектор. Подстройка весовых коэффициентов прово­дится по следующей формуле:



где  — весовой вектор *i*-й входной звезды на t-м такте обучения;  — вход­ной вектор;  — скорость обучения. Скорость обучения  имеет начальное зна­чение в пределах 0,1…0,2 и постепенно уменьшается в процессе обучения.

Выходная звезда Гроссберга выполняет противоположную функцию — при поступлении сигнала на вход выдается определенный вектор. Нейрон этого типа имеет один вход и т выходов с весами , которые подстраиваются в соответствии с формулой



где  — весовой вектор *i*-й выходной звезды на t-м такте обучения;  — выходной вектор;  — скорость обучения. Рекомендуется начать обучение со значения  в пределах единицы, постепенно уменьшая до значений, близких к нулю. Итерационный процесс будет сходить­ся к некоторому усредненному образу, полученному из совокупности обучаю­щих векторов.

Особенностью нейронов в форме звезд Гроссберга является избирательность памяти. Каждый нейрон в форме входной звезды помнит «свой» относящийся к нему образ и игнорирует остальные. Каждой выходной звезде соответствует некоторая конкретная командная функция. Образ памяти связывается с опре­деленным нейроном, а не возникает вследствие взаимодействия между нейро­нами в сети.

Обучение CPN состоит из двух шагов. На первом шаге весовые векторы слоя Кохонена настраиваются таким образом, чтобы провести распределение входных векторов по классам, каждый из которых соответствует одному ней­рону-победителю. Обучение проводится без учителя. Точность кластеризации в этом случае будет гарантирована только тогда, когда обучающая выборка яв­ляется представительной.

На втором шаге осуществляется обучение с учителем. Проводится под­стройка весовых коэффициентов выходного слоя Гроссберга на примерах с за­данным выходом с использованием формулы . При этом настраиваются только веса, соответствующие связям с теми элементами слоя Кохонена, кото­рые являются «победителями» в текущем такте обучения (выигравшие элементы посылают выходной сигнал, равный единице). Темпы обучения нейро­нов Кохонена и Гроссберга должны быть согласованы; кроме того, в слое Ко­хонена подстраиваются также веса всех нейронов в окрестности победителя, которая постепенно сужается до одного нейрона.

При функционировании сети в режиме распознавания нейроны слоя Грос­сберга по сигналу нейрона-победителя в слое Кохонена воспроизводят на вы­ходах сети образ в соответствии со значениями его весовых коэффициентов. В случае, когда слой Гроссберга состоит из единственного элемента, получаю­щийся скалярный выход равен одному из весов, соответствующих связям это­го элемента (с выигравшим нейроном).

1.2.5 Алгоритм обучения CPN-сети

Окончательно алгоритм обучения сети встречного распространения можно сформулировать следующим образом:

1. Случайно инициализируются весовые коэффициенты *W* всех слоев.

2. Задается начальное значение радиуса притяжения *s0*, скорость обучения выходной звезды Гроссберга , максимальное число итераций *N*, минимальная ошибка *E* и момент времени .

3. Подается входной образ  на вход нейронной сети и вычисляется норма вектора: , где  — весовой вектор нейрона с координатами .

4. Определяется нейрон победитель: , где  — координаты нейрона победителя.

5. Для каждого нейрона производится вычисление функции притяжения , где .

6. В соответствии с функцией притяжения осуществляется модификация весовых коэффициентов слоя Кохонена по формуле , где  — скорость обучения сети Кохонена, изменяющаяся по закону .

7. Осуществляется модификация весовых коэффициентов слоя Гроссберга по формуле 

8. Переходим к пункту 3 и повторяем процедуру для  где *L —* общее количество входных образов.

9. Рассчитываем усредненное расстояние между выходными и весовыми векторами по формуле , где  — реальный выход НС,  — желаемый выход НС, *М* — число выходов НС.

10. Рассчитать ошибку .

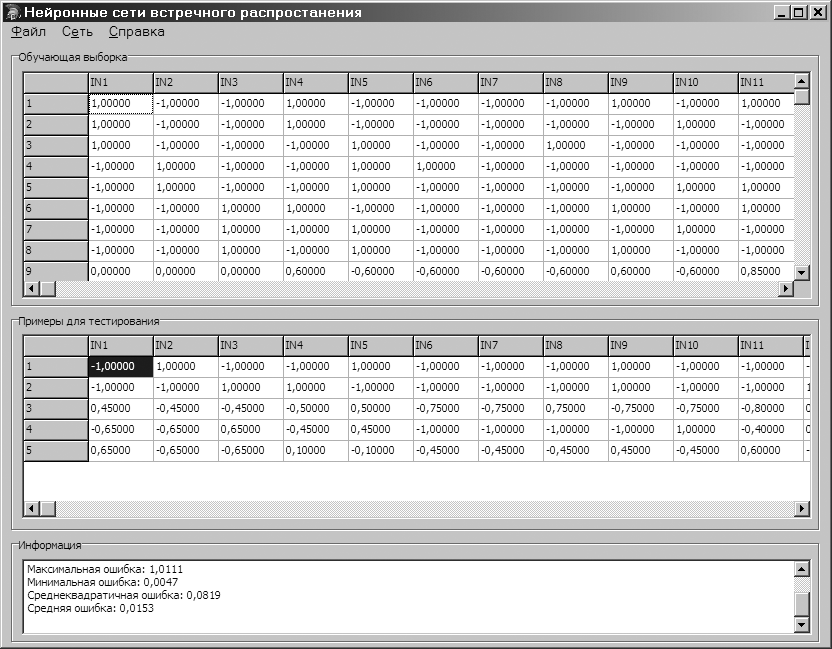
11. Проверяем условия , . Если выполнено хотя бы одно из условий, переходим к шагу 13. Иначе — к шагу 12.

12. Увеличиваем на единицу квант времени, уменьшаем радиус области притяжения *s* по формуле , скорость обучения  и повторяем процесс, начиная с пункта 3.

13. Конец.

* 1. Интерфейс главного окна

При запуске программы открывается главное окно, представленное на рисунке 3.1:

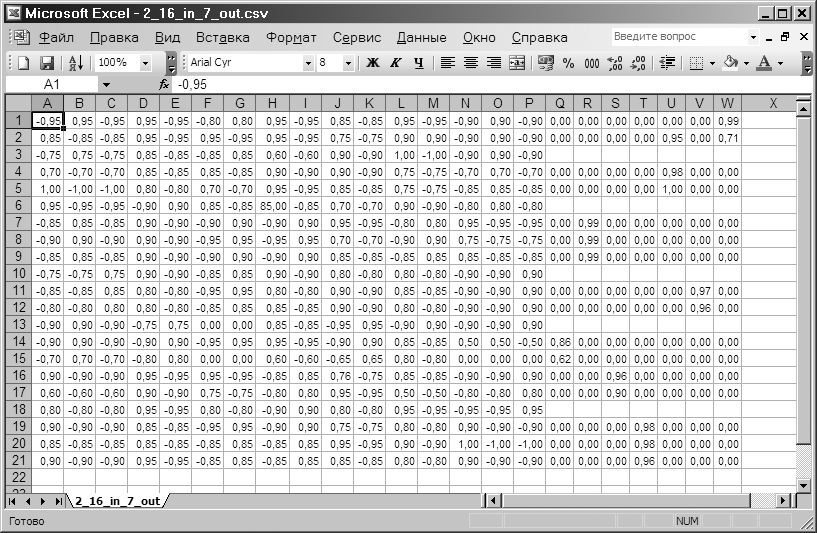


*Рисунок 3.1 — Главное окно программы.*

На главном окне программы располагаются:

* таблица с обучающей выборкой;
* таблица с примерами для тестирования (прогнозирования);
* поле с дополнительной информацией о сети. Эту информацию можно сохранить в текстовый файл, выбрав в контекстном меню пункт «Сохранить…»;
* главное меню.

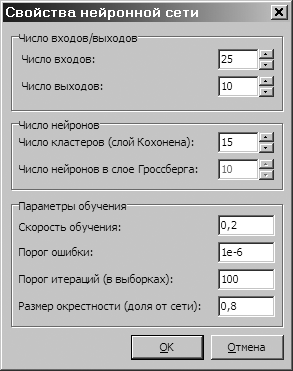
Справка по работе с программой вызывается по пункту «Справка» меню «Справка». При выборе пункта «Новая сеть…» меню «Файл» пользователю будет предложено выбрать файл в формате CVS( вариант DOS) с данными. Такой файл данных можно создать средствами любого табличного процессора, например, Microsoft Excel, OpenOffice.org Calc (рисунок 3.2). Чтобы указать программе, что данная строка предназначена для тестирования, а не для обучения, достаточно удалить значения всех выходов в соответствующей строке.



*Рисунок 3.2 — Окно программы Microsoft Excel* *с данными*

1.3.2 Интерфейс вспомогательных окон

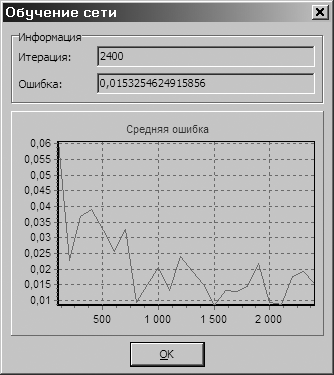
После выбора пользователем нужного файла, откроется окно «Свойства нейронной сети», представленное на рисунке 3.3:



*Рисунок 3.3 — Окно свойств НС.*

Далее пользователь должен задать необходимые параметры будущей НС (число входов/выходов (оно должно совпадать с числом входов и выходов, содержащихся в файле с данными), число нейронов и параметры обучения). После нажатия на кнопку «ОК» будет создана НС встречного распространения выбранной конфигурации и считан файл с данными.

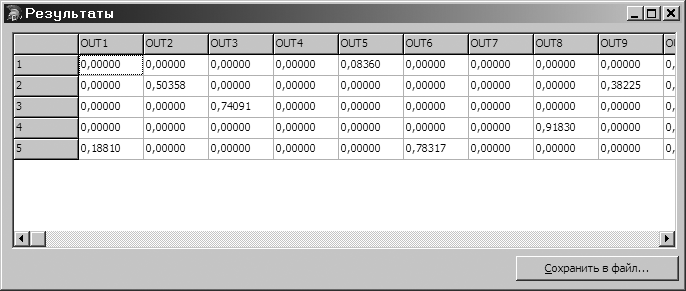
При выборе пункта «Обучить…» меню «Сеть» откроется следующее окно:



*Рисунок 3.4 — Окно «Обучение сети».*

При нажатии на кнопку «Обучить» начнется процесс обучения НС, при этом будет обновляться информация о номере текущей итерации, текущей средней ошибке и график ее изменения. Процесс обучения можно остановить, нажав на кнопку «Прервать».

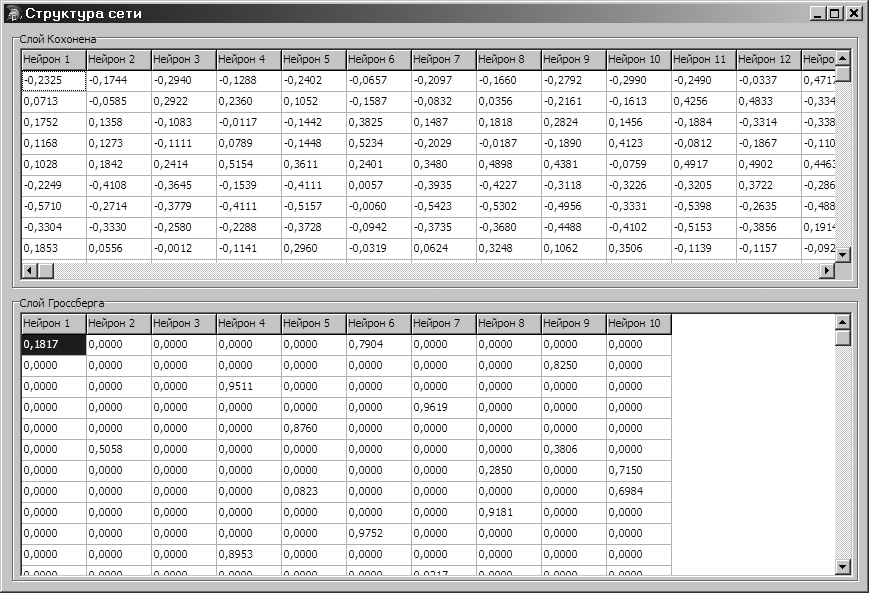
При выборе пункта «Тестирование…» меню «Сеть» откроется окно, представленное на рисунке 3.5:



*Рисунок 3.5 — Результаты тестирования.*

В данном окне расположена таблица с данными, полученными программой. Их можно сохранить в CSV-файл, нажав на кнопку «Сохранить в файл…».

При выборе пункта «Структура сети…» меню «Сеть» откроется окно, представленное на рисунке 3.6:



*Рисунок 3.6 — Окно структуры НС.*

В окне расположены две таблицы, содержащие веса синапсов всех нейронов со слоев Кохонена и Гроссберга.

При выборе пункта «Параметры сети…» меню «Сеть» откроется окно, показанное ранее на рисунке 3.3. Здесь пользователь может изменить параметры уже созданной НС, после чего ее необходимо переобучить.

2.4 Порядок работы с программой

1. В любом табличном процессоре (Microsoft Excel, OpenOffice.org Calc) подготовить файл с исходными данными. При этом должны быть удалены те выходы, которые будут прогнозироваться с помощью программы.
2. Сохранить данный файл в формате CSV.
3. Запустить cpn.exe
4. В меню «Файл» выбрать пункт «Новая сеть…» и указать путь к созданному файлу с данными.
5. В появившемся окне «Свойства нейронной сети» задать необходимые параметры:
   1. число входов и выходов сети;
   2. число кластеров, на которые будет разбиваться входные образы;
   3. скорость обучения;
   4. порог ошибки (минимальное значение средней ошибки, при достижении которого прерывается процесс обучения сети);
   5. максимальное число итераций обучения;
   6. размер окрестности выигравшего нейрона (в долях от числа кластеров).

Нажать на кнопку «ОК».

1. После этого, если входной файл не содержал ошибок, будут заполнены таблицы «Обучающая выборка» и «Примеры для тестирования».
2. В меню «Сеть» выбрать пункт «Обучить».
3. В появившемся окне «Обучение сети» нажать на кнопку «Начать». Процесс обучения можно остановить, нажав на кнопку «Прервать». В процессе обучения выводится номер текущей итерации, величина средней ошибки и строится график, показывающий изменение средней ошибки в процессе обучения. По завершении обучения нажать на кнопку «ОК».
4. После обучения в поле «Информация» главного окна будут отображены все данные о сети. Эти данные можно сохранить в файл, выбрав в контекстном меню поля пункт «Сохранить…».
5. Дополнительно структуру сети (веса синапсов всех нейронов всех слоев) можно посмотреть, выбрав пункт «Структура сети…» меню «Сеть».
6. Чтобы запустить протестировать сеть, необходимо выбрать пункт «Тестирование» меню «Сеть». В появившемся окне «Результаты» будет отображена таблицы со значениями, предсказанными сетью. Результаты можно сохранить в файл, нажав на соответствующую кнопку.
7. Параметры уже созданной НС можно изменить, выбрав пункт «Параметры…» меню «Сеть».