

## Лабораторная работа № 4

# Введение в искусственные нейронные сети

**Цель работы.** Разработка и исследование алгоритмов обучения искусственных нейронных сетей.

### ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

**Распознавание образов** — научное направление, связанное с разработкой принципов и построением систем, предназначенных для определения принадлежности объекта к одному из классов. Классы объектов могут быть заранее выделены (задача классификации) или необходимо их выявить в процессе решения задачи (задача кластеризации). Под объектами в распознавании образов (англ. pattern) понимают (**семантическая классификация**):

- различные предметы;
- явления;
- процессы;
- ситуации;
- сигналы;
- и т.п.

Помимо смыслового (семантического) отличия, вкладываемого в понятие образ, образы отличаются друг от друга также по способу представления (**синтаксису**):

- в классических моделях образ обычно описывается набором признаков, каждый из которых характеризует определенное свойство объекта;
- в структурных моделях в качестве образа выступает некоторое высказывание, порожаемое грамматикой, характеризующей класс;
- в задачах обработки текста роль образа выполняет некоторая цепочка символов или шаблонное представление этой цепочки (например, регулярные выражения).

Образы допускают рекурсивное определение (например, символ является образом, список символов также является образом). При представлении образов нередко набор его характеристик делят на две группы: переменные и постоянные характеристики.

Задачи распознавания образов представляют собой, по существу, дискретные аналоги задач поиска оптимальных решений (дискретного программирования). К ним относится широкий класс задач, в которых по некоторой, обычно весьма разнородной, неполной, нечеткой, искаженной и

косвенной информации требуется установить, обладают ли изучаемые объекты (ситуации, явления) фиксированным конечным набором свойств, позволяющим отнести их к определенному классу. Другой из важных областей применения теории распознавания образов является решение задач прогнозирования поведения объектов или развития ситуации. К задачам этого вида относятся задачи технической и медицинской диагностики, геологического прогнозирования, прогнозирования свойств химических соединений, сплавов и новых материалов, прогнозирования урожая и хода строительства крупных объектов, обнаружения лесных пожаров, управления производственными процессами и т.д. Задача распознавания образов возникает, в различных областях искусственного интеллекта, например в понимании естественного языка, символьной обработке алгебраических выражений, экспертных системах и др. В общем случае, любую задачу можно рассматривать как распознавание образа – при известных исходных данных за счет использования определенного алгоритма (методики) требуется получить решение (образ).

Основные типы задач распознавания образов приведены в следующей таблице.

Таблица 1

Типы задач распознавания образов

| Тип задачи  | Примечания  |
|---|---|
| Классификация   | Отнесение предъявленного объекта (ситуации) по его формализованному описанию к одному из заданных классов   |
| Реализация ассоциативной (контекстно-адресуемой) памяти       | Ассоциация искаженного представления образа с оригиналом (восстановление исходного образа)  |
| Кластеризация   | Разбиение множества объектов (ситуаций) по их формализованным описаниям на систему непересекающихся подмножеств (классов)                                       |
| Абстрагирование и формализованное описание                    | Определение набора характеристик для построения формализованного описания объектов распознавания; оценка информативности отдельных характеристик и их сочетаний |
| Приведение исходных данных к виду, удобному для распознавания | Построение формализованного описания объекта распознавания  |

|                       |   |
|-----------------------|---|
| Прогнозирование       | Предсказание значений характеристик или поведения системы в будущем на основании предшествующих и текущих наблюдений          |
| Аппроксимация функций | Поиск функции, наиболее близко соответствующей набору экспериментальных данных  |
| Оптимизация           | Поиск эффективного или оптимального решения целевой функции в условиях действия ограничений                                   |
| Управление            | Расчет входного воздействия, при котором система будет следовать желаемой траектории или соответствовать требуемым параметрам |

Одним из наиболее эффективных и распространенных способов представления и решения перечисленных выше задач являются искусственные нейронные сети (ИНС).

**Искусственная нейронная сеть** – математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей.

Как отмечалось ранее, ИНС разрабатываются в рамках нейрокибернетического подхода к построению ИИС. Основными отличительными особенностями ИНС от логических ИИС являются:

- ИНС подобны структуре мозга. В логических ИИС используются абстрактные структуры (правила, фреймы, сценарии и т.д.), не имеющие аналогов в живой природе;

- для решения задач в логических ИИС необходимо заранее сформулировать весь набор закономерностей, описывающих предметную область. В ИНС используется альтернативный подход, основанный на концепции обучения на примерах. В этом случае при построении ИИС не требуется заранее знать обо всех закономерностях исследуемой области, но необходимо располагать достаточным количеством примеров для настройки разрабатываемой системы, которая после обучения будет способна получать требуемые результаты с определенной степенью достоверности.

Нервная система и мозг человека состоят из нейронов (клеток), соединенных между собой нервными волокнами. Нервные волокна способны передавать электрические импульсы между нейронами. Все процессы передачи раздражений от нашей кожи, ушей и глаз к мозгу, процессы мышления и управления действиями - все это реализовано в живом организме как передача электрических импульсов между нейронами.

Рассмотрим строение биологического нейрона. Каждый нейрон имеет отростки нервных волокон двух типов - **дендриты**, по которым принимаются импульсы, и единственный **аксон**, по которому нейрон может передавать импульс. Аксон, который в конце разветвляется на волокна, контактирует с

дендритами других нейронов через специальные образования - **синапсы**, которые влияют на силу импульса.

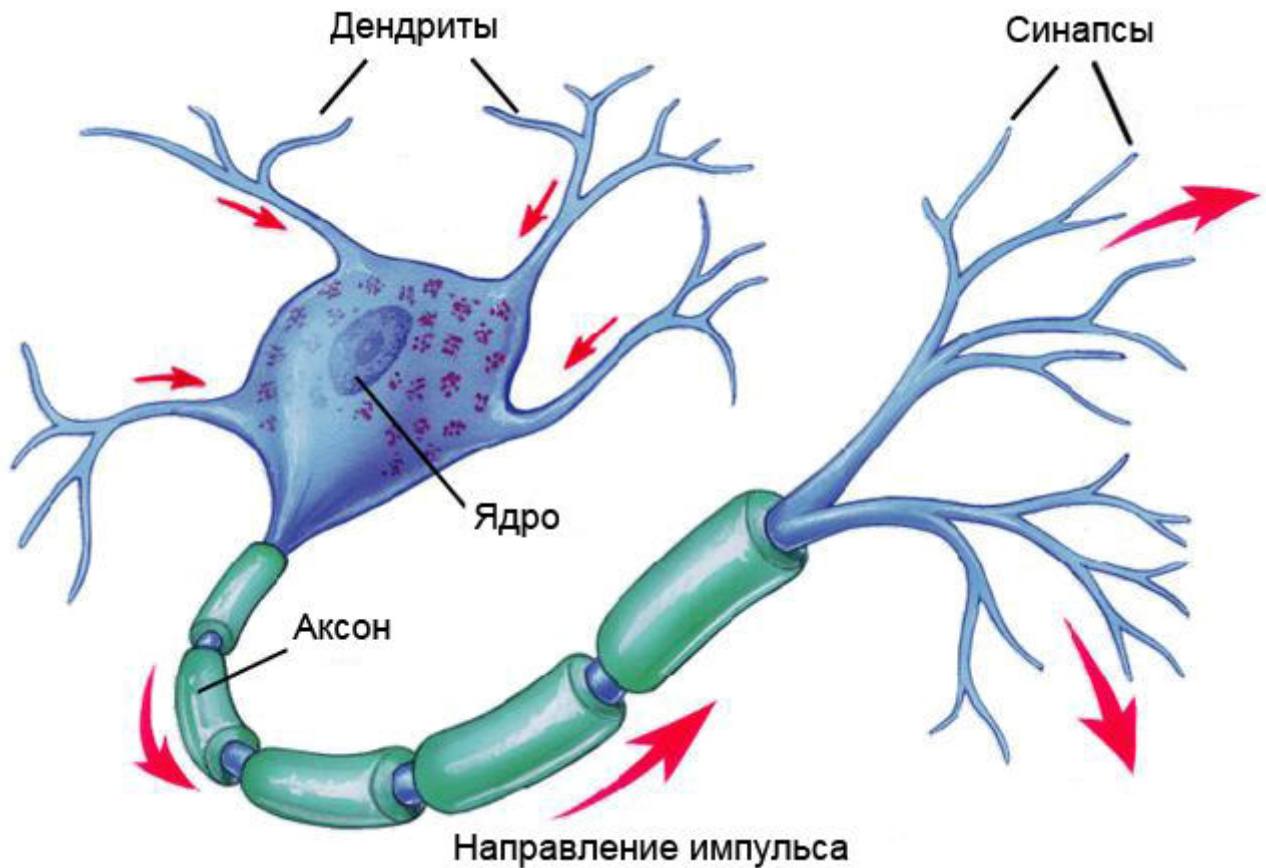


Рис.1. Биологический нейрон

Можно считать, что при прохождении синапса сила импульса меняется в определенное число раз, которое мы будем называть весом синапса. Импульсы, поступившие к нейрону одновременно по нескольким дендритам, суммируются.

Если суммарный импульс превышает некоторый порог, нейрон возбуждается, формирует собственный импульс и передает его далее по аксону.

Важно отметить, что веса синапсов могут изменяться со временем, а значит, меняется и поведение соответствующего нейрона.

Кора головного мозга человека является протяженной, образованной нейронами поверхностью толщиной от 2 до 3 мм с площадью около 2200 см<sup>2</sup>, что вдвое превышает площадь поверхности стандартной клавиатуры.

Общее количество нейронов в мозге человека около  $10^{11}$ , что приблизительно равно числу звезд Млечного пути.

Каждый нейрон связан с  $10^3$  -  $10^4$  другими нейронами. В целом мозг человека содержит приблизительно от  $10^{14}$  до  $10^{15}$  взаимосвязей.

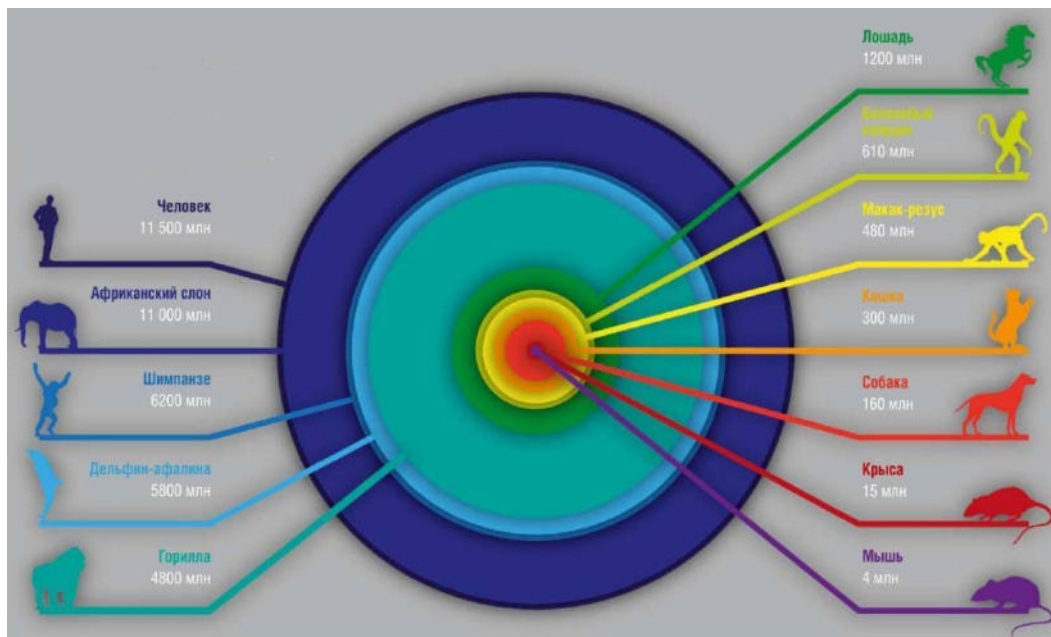


Рис.2. Количество нейронов у человека и животных

### Математическая модель нейрона

Обобщенная математическая модель нейрона выглядит следующим образом:

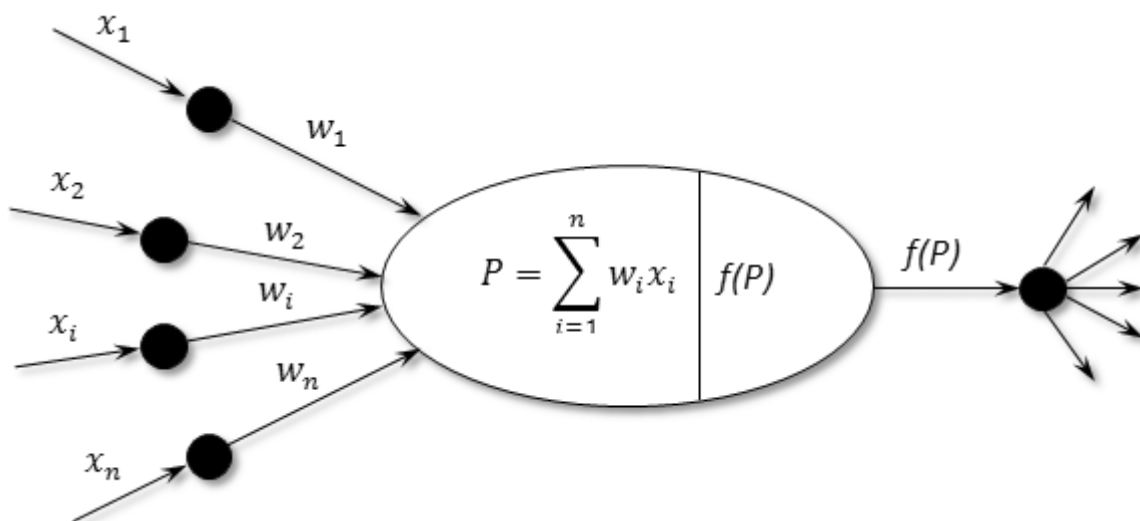


Рис. 3. Математическая модель нейрона

- входные сигналы  $x_i$  – данные, поступающие из окружающей среды или от других активных нейронов. Входные значения могут быть дискретными из множеств  $[0, 1]$  или  $[-1, 1]$  либо принимать любые вещественные значения;

- вещественные весовые коэффициенты  $w_i$  – определяют силу связи между нейронами;

- уровень активации (потенциал) нейрона  $P = \sum_{i=1}^n w_i x_i$ ;
- функция активации  $f(P)$  – предназначена для вычисления выходного значения сигнала, передаваемого другим нейронам.

### Классификация искусственных нейронных сетей

Искусственная нейронная сеть состоит из нескольких слоев, в которые группируются нейроны. Схематично её можно представить следующим образом.

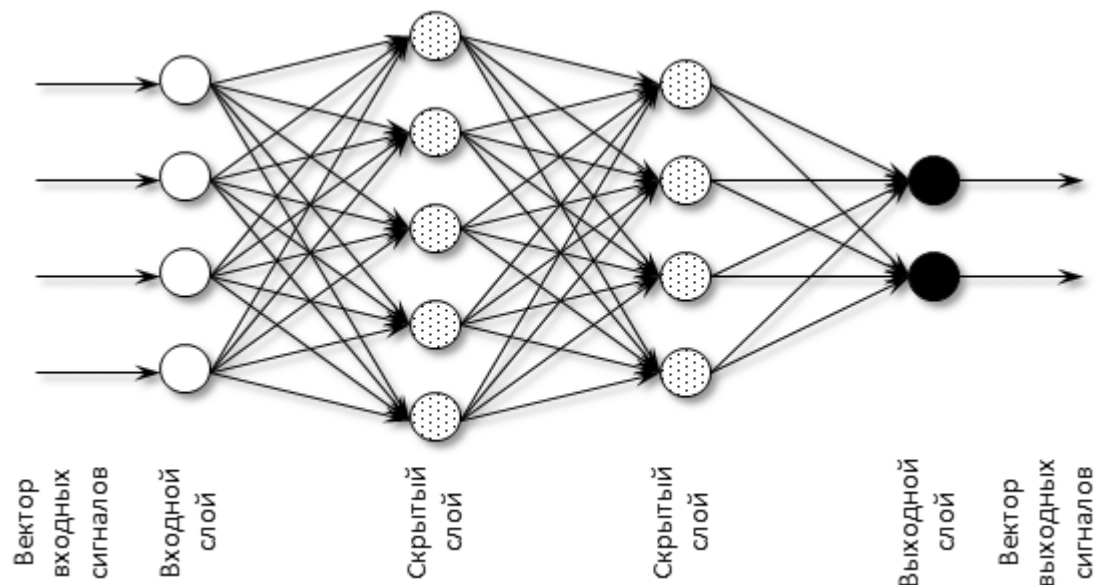


Рис. 4. Искусственная нейронная сеть

На нейроны входного слоя подаются измеренные и преобразованные значения характеристик объекта (образа).

Сигналы, поступившие к нейронам входного слоя, передаются на следующий слой (скрытый или выходной) **без преобразования (т.е. к ним не применяется функция активации)**. Скрытые слои и выходной слой, отражающие специфику знаний, преобразуют входные данные.

Выходной слой генерирует скаляр (одно значение) или вектор (несколько значений), являющийся решением задачи (в частности распознаваемым образом).

В зависимости от **количества преобразующих слоев (скрытых и выходного)** различают одно- (в сети отсутствуют скрытые слои) и многослойные сети.

В зависимости от **функции активации** различают следующие ИНС (Табл. 2).

Типы ИНС в зависимости от функции активации

| Функция                 | Способ определения Y   |
|-------------------------|--|
| Ступенчатая пороговая   | $Y = 0$ , при $P < P^*$<br>$Y = 1$ , при $P \geq P^*$  |
| Линейная пороговая      | $Y = 0$ , при $P < P_1^*$<br>$Y = a + b \cdot P$ , при $P_1^* \leq P < P_2^*$<br>$Y = 1$ , при $P \geq P_2^*$<br>$P_2^* = P_1^* + 1 / b$ |
| Сигмоидальная           | $Y = 1 / (1 + e^{-a(P-P^*)})$  |
| Гиперболический тангенс | $Y = a \operatorname{th}(b P) = a (e^{bP} - e^{-bP}) / (e^{bP} + e^{-bP})$   |
| Арктангенс              | $Y = 2 \operatorname{arctg}(P) / \pi$  |
| Линейная                | $Y = a + b \cdot P$  |
| Гауссиана               | $Y = e^{-b(P-P^*)}$  |

Примечания.

1.  $P^*$ ,  $P_1^*$ ,  $P_2^*$  - пороговые значения.
2.  $a$ ,  $b$  – коэффициенты.
3.  $e$  – основание натурального логарифма.

В зависимости от **типа межнейронных связей** различают ИНС:

- с прямыми связями (рис 5);
- с перекрестными связями;

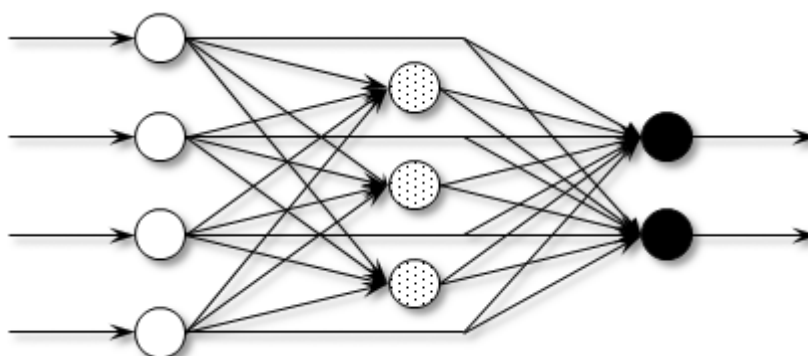


Рис. 6. ИНС с перекрестными связями

- с обратными (рекуррентными) связями. В таких сетях нейрон может посылать сигналы сам себе, нейронам того же слоя или нейронам предыдущих слоев.

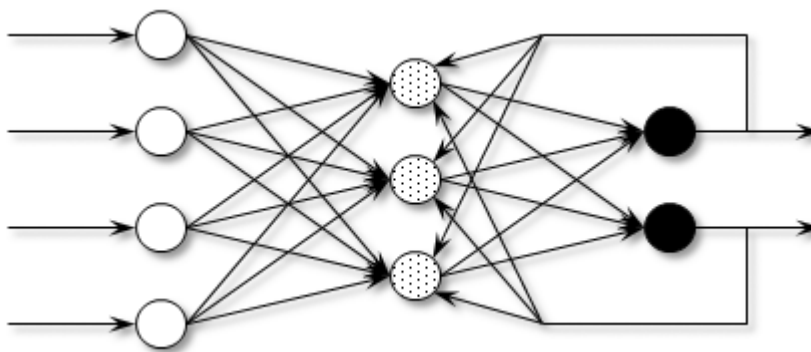


Рис. 7. ИНС с обратными связями

Перед непосредственным использованием, т.е. перед решением конкретной задачи распознавания образов, необходимо настроить (обучить) сеть. Процесс обучения сети заключается в определении набора связей и коэффициентов связей между нейронами. В зависимости от **способа обучения** различают следующие типы ИНС:

- обучаемые с учителем (контролируемое или наблюдаемое обучение). При обучении с учителем все примеры обучающей выборки содержат правильные ответы (выходы), соответствующие исходным данным (входам). В процессе обучения веса (коэффициенты) настраиваются так, чтобы сеть порождала ответы, наиболее близкие к правильным;

- обучаемые без учителя (неконтролируемое или ненаблюдаемое обучение). Обучение без учителя используется, когда не для всех примеров обучающей выборки известны правильные ответы. В этом случае предпринимаются попытки определения коэффициентов сети с целью определения категорий (классов) образцов и дальнейшего их распределения по категориям. Используется, в частности, для решения задач кластеризации;

- со смешанным (гибридное) обучением. При смешанном обучении часть весов определяется посредством обучения с учителем, а другая часть получается с помощью алгоритмов самообучения.

В зависимости от **правила коррекции коэффициентов ( $w_i$ )** ИНС делят на следующие категории:

- правило коррекции по ошибке (дельта-правило). Разность между известным значением результата и реакцией сети используется для корректировки весовых коэффициентов. Корректировка заключается в небольшом (обычно менее 1%) увеличении весов тех связей, которые усиливают правильные реакции, и уменьшении весов тех связей, которые способствуют ошибочным. Обычно используется для однослойных сетей;

- правило обратного распространения ошибки. При обучении корректировка распространяется обратно по сети на все весовые коэффициенты. Обычно используется для многослойных сетей. В случае,



когда корректировка весовых коэффициентов выполняется после прогона одного образа, говорят о последовательном режиме обучения. Режим, когда корректировка выполняется не сразу, а после прогона нескольких обучающих образов, называется пакетным (правило обратного распространения ошибки во времени);

- синхронное обучение (правило Хебба). Это правило опирается на следующие нейрофизиологическое наблюдение: если нейроны с обеих сторон синапса активизируются одновременно и регулярно, то сила синаптической связи возрастает. Таким образом, корректировке подлежат коэффициенты только тех связей, выход которых отличен от нуля;

- конкурентное обучение (правило Кохонена, «победитель забирает все»). В каждом слое корректируются весовые коэффициенты только одного нейрона-победителя, у которого выход наиболее точно соответствует предъявленному образцу;

- правило Больцмана. Алгоритм обучения основан на идее моделирования отжига - способа выжигания дефектов в кристаллической решетке. Атомы, занимающие в ней неправильное место, при низкой температуре не могут сместиться в нужное положение - им не хватает кинетической энергии для преодоления потенциального барьера. При этом система в целом находится в состоянии локального энергетического минимума. Для выхода из него металл нагревают до высокой температуры, а затем медленно охлаждают, позволяя атомам занять правильные положения в решетке, соответствующее глобальному минимуму энергии. Имитация отжига в нейронной сети выполняется по следующей процедуре:

- на вход сети подается обучающий образ и вычисляется выход;
- вычисляется значение средней квадратичной ошибки между желаемым и полученным выходными векторами;
- весовые коэффициенты изменяются случайным образом, затем вычисляются новый выход и результирующая ошибка. Если ошибка уменьшилась, оставляют измененные веса; если ошибка увеличилась, оставляют измененные веса с вероятностью, определяемой распределением Больцмана. Если ошибка осталась неизменной, то весовые коэффициенты возвращают к его предыдущему значению.

## **Процедура построения искусственных нейронных сетей**

Процедура построения искусственных нейронных сетей, используемых в задачах классификации, включает следующие этапы.

**Этап 1.** Определение типа распознаваемых образов (фотография, видео, совокупность числовых характеристик и т.п.), набора измеряемых параметров образов, информативных с точки зрения распознавания, и классов распознаваемых образов.

**Этап 2.** Выбор способа преобразования (применение фильтров, подавление шумов, сегментация и т.п.) измеряемых параметров образов во входной вектор числовых величин и представления набора классов образов в виде вектора выходных величин. Эффективность нейросетевой модели повышается, если диапазоны изменения величин входного и выходного векторов будут нормализованы, например, в диапазоне  $[0,1]$  или  $[-1,1]$ .

**Этап 3.** Проектирование архитектуры искусственной нейронной сети:

- определение количества слоев. В сети должно быть, как минимум, два слоя: входной и выходной. Количество скрытых слоев определяется, как правило, экспертным или опытным путем;

- определение количества нейронов в каждом слое. Количество нейронов во входном слое соответствует количеству преобразованных измеряемых параметров распознаваемых образов. Количество нейронов в выходном слое, как правило, соответствует количеству классов распознаваемых образов. Определение количества нейронов в каждом скрытом слое является неформальной проблемой, при решении которой можно использовать эвристическое правило: число нейронов в следующем слое должно быть в два раза меньше, чем в предыдущем;

- выбор типа связей между нейронами. Зависит от специфики решаемой задачи;

- выбор функции активации. Зависит от специфики решаемой задачи.

**Этап 4.** Обучение сети – уточнение значений весовых коэффициентов связей на основе многократного прогона обучающих примеров через сеть. Каждому обучающему примеру (образу) соответствует строго определенные вектора входных и выходных величин. Обучающие примеры, как правило, представляют собой эталонные (идеальные) представления образов, а также их незначительные модификации. Они должны быть подобраны для каждого класса образов. Существует эмпирическое правило, которое устанавливает рекомендуемое соотношение  $\lambda$  между количеством обучающих примеров и числом связей в нейронной сети:  $\lambda \leq 10$ .

**Этап 5.** Тестирование сети с помощью контрольного набора примеров (образов) для оценки качества выбранной архитектуры и степени обучения. Контрольный набор должен содержать, как минимум, по одному примеру для каждого класса образов и включать в себя как эталонные, так и значительно искаженные образы.

Этапы с третьего по пятый могут совместно применяться для уточнения архитектуры сети на основе конструктивного или деструктивного подхода. В соответствии с первым подходом обучение начинается на сети небольшого размера, который постепенно увеличивается до достижения требуемой точности по результатам тестирования. Деструктивный подход базируется на принципе «прореживания дерева», в соответствии с которым из сети с заведомо избыточным объемом постепенно удаляют «лишние» слои,

нейроны и примыкающие к ним связи. Этот подход дает возможность исследовать влияние удаляемых элементов на точность распознавания сети.

### **Алгоритм обратного распространения ошибки**

Способность к обучению является уникальным свойством мозга. Для ИНС под обучением понимается процесс настройки (корректировки) весовых коэффициентов связей ( $w$ ) и порогов нейронов ( $P^*$ ) для эффективного решения поставленной задачи.

Обычно обучение нейронной сети осуществляется на основе некоторого множества эталонных примеров, для которых известны входные и выходные вектора (входные сигналы и соответствующие им результаты).

По мере процесса обучения, который происходит по некоторому алгоритму, сеть должна все лучше и лучше реагировать на входные сигналы (выдавать правильные результаты).

К сожалению, **не существует универсального алгоритма обучения**, подходящего для всех архитектур нейронных сетей.

Для решения задачи принадлежности предъявленного образа одному из заранее выделенных классов образов хорошо себя зарекомендовали многослойные нейронные сети с прямыми связями.

Самым известным и популярным методом их обучения является **алгоритмом обратного распространения ошибки** (англ. backpropagation), который является модификацией классического метода градиентного спуска. Этот алгоритм был изложен в диссертации Пола Вербоса (Paul Werbos) 1974 г., но тогда не привлек к себе должного внимания. Рождение алгоритма для широкой публики связано с работой группы PDP (Parallel Distributed Processing), освещенной в двухтомном труде 1986г. Именно там, в статье Румельхарта, Хинтона и Уильямса, была изложена теория обучения многослойного персептрона.

Помимо многослойности и прямых связей, дополнительное требование его применимости – функция активации должна быть дифференцируема, т.е. иметь первую производную.

Алгоритм подразумевает два потока сигналов (данных) при прогоне одного обучающего образа через сеть:

- прямой (от входного до выходного слоя) распространение сигнала, соответствующего значениям входного вектора обучающего образа;
- обратный (от выходного до входного слоя) распространение сигнала ошибки для корректировки весовых коэффициентов связей.

Блок-схема алгоритма обратного распространения ошибки показана на следующем рисунке.

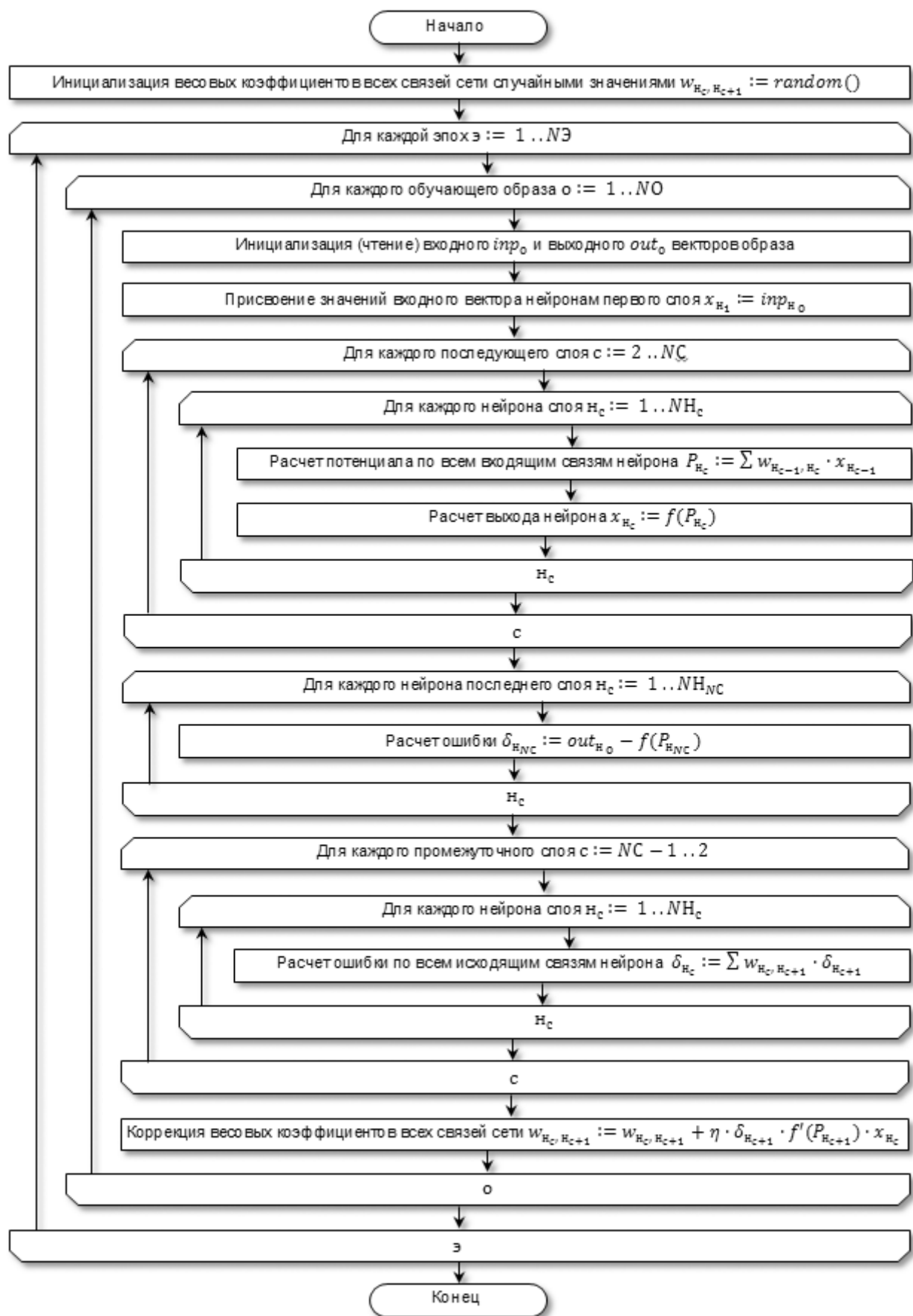


Рис. 8. Блок-схема алгоритма обратного распространения ошибки

На рисунке приняты следующие обозначения:

- $\varepsilon$  – номер эпохи;
- $\mathbf{o}$  – номер обучающего образа;
- $\mathbf{c}$  – номер слоя;
- $\mathbf{n}_c$  – номер нейрона в слое  $\mathbf{c}$ ;
- $N\varepsilon$  – количество эпох;
- $N\mathbf{O}$  – количество обучающих образов;
- $N\mathbf{C}$  – количество слоев;
- $N\mathbf{H}_c$  – количество нейронов в слое  $\mathbf{c}$ ;
- $\mathbf{inp}_o$  – вектор входных значений обучающего образа  $\mathbf{o}$ ;
- $\mathbf{inp}_{\mathbf{n}_o}$  – значение элемента  $\mathbf{n}$  входного вектора обучающего образа  $\mathbf{o}$ ;
- $\mathbf{out}_o$  – вектор ожидаемых выходных значений обучающего образа  $\mathbf{o}$ ;
- $\mathbf{out}_{\mathbf{n}_o}$  – значение элемента  $\mathbf{n}$  выходного вектора обучающего образа  $\mathbf{o}$ ;
- $P_{\mathbf{n}_c}$  – потенциал нейрона  $\mathbf{n}$  слоя  $\mathbf{c}$ ;
- $x_{\mathbf{n}_c}$  – выход нейрона  $\mathbf{n}$  слоя  $\mathbf{c}$ . Для нейронов входного слоя равен соответствующему элементу вектора входных значений  $\mathbf{inp}_{\mathbf{n}_o}$ , для остальных слоев – результату расчета функции активации  $f(P_{\mathbf{n}_c})$ ;
- $w_{i,j}$  – весовой коэффициент связи нейрона  $\mathbf{i}$  предыдущего слоя и нейрона  $\mathbf{j}$  последующего слоя;
- $\delta_{\mathbf{n}_c}$  – ошибка (сигнал ошибки) нейрона  $\mathbf{n}$  слоя  $\mathbf{c}$ ;
- $f'(P_{\mathbf{n}_{c+1}})$  – результат расчета первой производной функции активации нейрона  $\mathbf{n}$  слоя  $\mathbf{c}+1$ ;
- $\eta$  – норма обучения.

Перед процедурой обучения весовые коэффициенты всех связей инициализируются случайными малыми значениями (например, в диапазоне  $[-0.3, 0.3]$ ).

При прогоне через сеть одного обучающего образа выполняется коррекция весовых коэффициентов. Прогон всех обучающих образов называется **эпохой**. Для настройки весовых коэффициентов (обучения) сети должно пройти достаточно большое количество эпох (несколько сотен, тысяч и т.п. – в зависимости от масштабов сети). Количество эпох может быть определено автоматически по заданной максимальной разнице средней ошибки (погрешности) между текущей эпохой и предыдущей, задано экспертом (как на рис.8) и т.п.

При прямом распространении сигнала выполняется стандартная процедура, как и при распознавании образа:

- нейронам первого (входного) слоя присваиваются значения входного вектора  $inp_0$  без преобразования. Можно условно принять, что эти значения  $x_{n_1}$  являются одновременно потенциалами  $P_{n_1}$  и выходами  $f(P_{n_1})$  нейронов. В виду отсутствия преобразования при подсчете количества слоев сети входной слой не учитывается и на схемах эти нейроны часто отображают иначе, чем остальные;

- для нейронов промежуточных и выходного слоев с учетом текущих значений весовых коэффициентов сети выполняется расчет потенциала  $P_{n_c}$  и выхода  $x_{n_c} := f(P_{n_c})$ .

При проходе слоев в обратном направлении выполняется расчет сигнала ошибки:

- для нейронов последнего (выходного) слоя как разница между ожидаемым для образа значением выхода и полученным:

$$\delta_{n_{NC}} := out_{n_o} - f(P_{n_{NC}});$$

- для нейронов промежуточных слоев с учетом весовых коэффициентов связей с нейронами последующего слоя и их сигналов ошибки:

$$\delta_{n_c} := \sum w_{n_c, n_{c+1}} \cdot \delta_{n_{c+1}}.$$

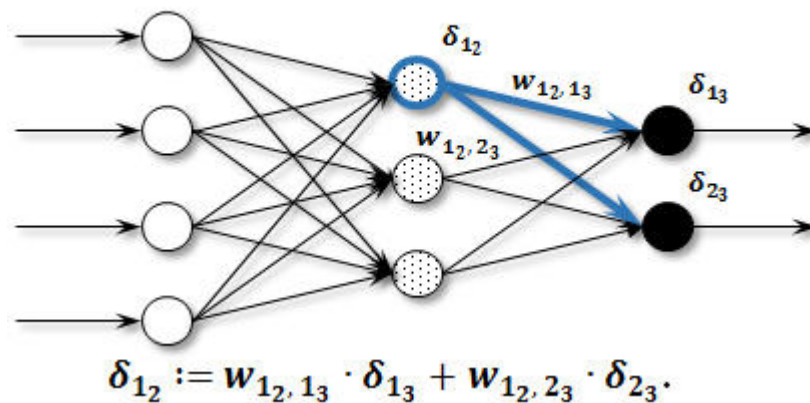


Рис. 9. Пример расчета сигнала ошибки для нейрона 1 слоя 2

Заключительной операцией является корректировка весовых коэффициентов всех связей на величину:

$$\Delta w_{n_c, n_{c+1}} := \eta \cdot \delta_{n_{c+1}} \cdot f'(P_{n_{c+1}}) \cdot x_{n_c};$$

где  $\eta$  – норма обучения;

$\delta_{n_{c+1}} \cdot f'(P_{n_{c+1}})$  – локальный градиент. Произведение сигнала ошибки и результата расчета первой производной функции активации для нейрона, в который входит связь;

$x_{n_c} := f(P_{n_c})$  – результат расчета функции активации для нейрона, из которого выходит связь.

**Норма обучения  $\eta$**  – коэффициент, от которого зависит величина изменения веса (скорость обучения). Выбирается в диапазоне  $0 < \eta < 1$ .

Значения потенциалов  $P_{n+1}$  и  $P_n$  принимаются полученными при прямом распространении сигнала. В связи с этим корректировка весовых коэффициентов может выполняться независимо и в любом направлении (прямом или обратном).

### **Пример построения искусственной нейронной сети.**

На основе предъявленного образа (графического файла) требуется автоматическое определение цифры, изображенной в файле.

Поэтапно процедура разработки и обучения сети выглядит следующим образом.

**Этап 1.** Определение характеристик распознаваемых образов:

- тип распознаваемых образов – растровый графический файл;
- набор измеряемых параметров образов – образ задается совокупностью пикселей, каждый из которых характеризуется интенсивностью красного, зеленого и синего цветов (модель RGB);
- классы распознаваемых образов – цифры от 0 до 9 (10 классов).

**Этап 2.** Изображение, находящееся в файле, можно пропустить через «черно-белый» фильтр, разбить на стандартное количество одинаковых сегментов (например, заданных матрицей  $8 \times 10$ ) и определить для каждого сегмента преобладающий цвет (черный или белый).

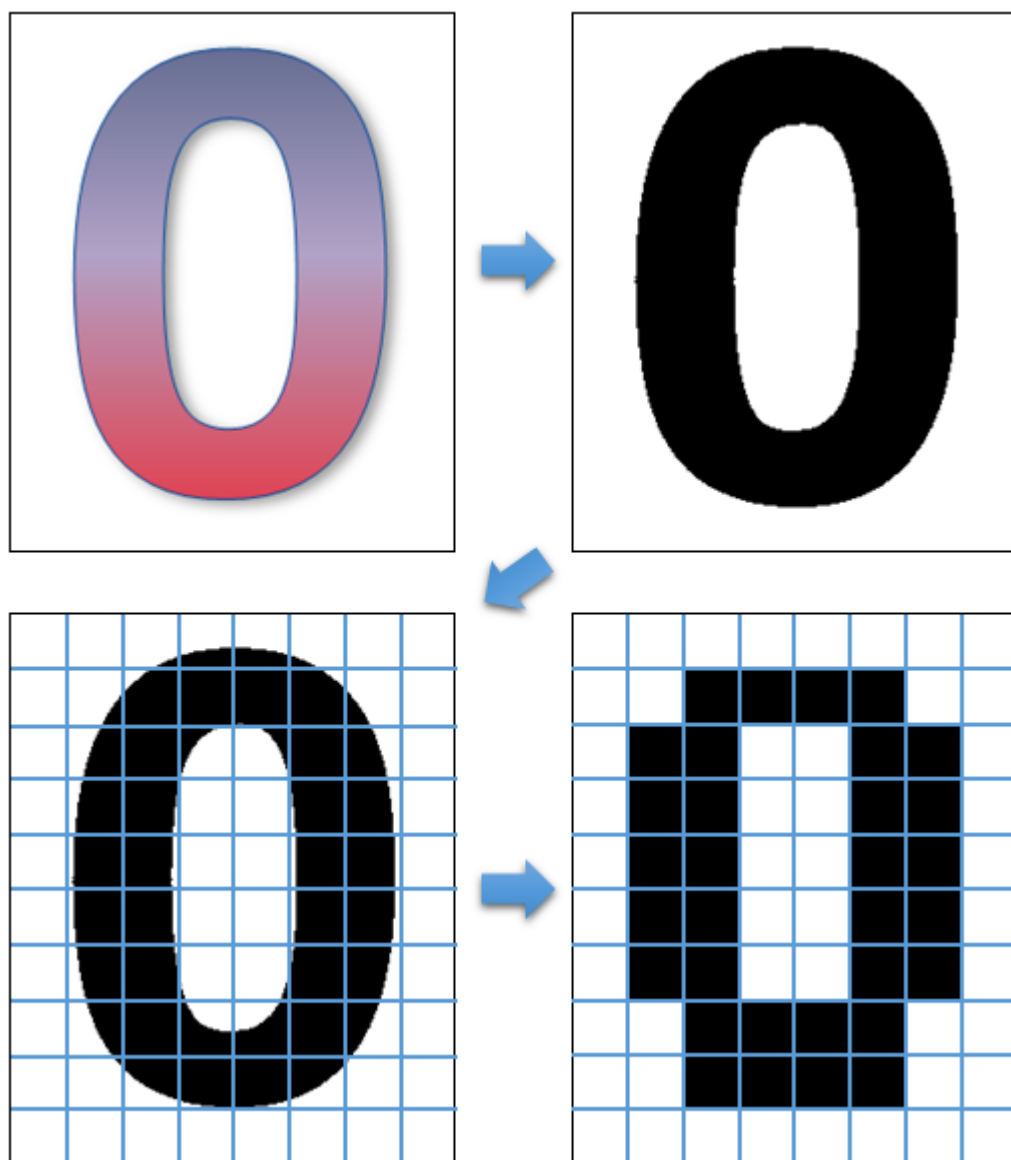


Рис.10. Преобразование исходного образа

Таким образом, вектор входных величин сети для отдельного распознаваемого образа (файла) будет состоять из одного и того же количества элементов ( $8 \times 10 = 80$  элементов), каждый из которых может принимать значения: 1 (цвет сегмента черный) или 0 (цвет сегмента белый).

Количество элементов вектора выходных величин будет соответствовать количеству классов распознаваемых образов. Значения элементов вектора нормализованы и равны 1 (признак принадлежности к классу) или 0 (в противном случае).

**Этап 3.** Проектирование архитектуры искусственной нейронной сети:

- количество слоев – 3;
- количество нейронов в слое:
  - входном – 80;
  - промежуточном – 40;



- выходном – 10;
- тип связей между нейронами – прямые;

- функция активации – сигмоидальная  $f(P) = \frac{1}{1+e^{-P}}$ .
- первая производная функции активации –  $f'(P) = P(1 - P)$ .

**Этап 4.** Обучение сети выполняется с помощью алгоритма с обратным распространением ошибки.

Для обучения экспертом приготовлено 20 обучающих образов – по два на каждый класс (цифру), один из которых эталонный (идеальный) и один с незначительной модификацией. Каждый образ характеризуется строго определенными векторами входных и выходных величин. Вектор входных величин соответствует преобразованному обучающему образу (файлу) и состоит из 80 элементов, значения которых 0 или 1 согласно цвету сегмента. Вектор выходных величин соответствует классу, которому принадлежит образ, и состоит из 10 элементов, значения которых равны 1 или 0. Так для обучающего образа цифры «0» вектор выходных величин – [1, 0, 0, 0, 0, 0, 0, 0, 0, 0].

Перед процедурой обучения весовые коэффициенты всех связей инициализируются малыми случайными значениями в диапазоне от [-0.3, 0.3].

При обучении все 20 образов последовательно прогоняются через сеть с корректировкой весовых коэффициентов после прогона каждого образа. Данная процедура повторяется 500 раз (количество эпох – 500). Норма обучения принята равной 0.1.

**Этап 5.** Тестирование сети выполняется с помощью контрольных образов. Например, если на вход сети был подан образ цифры «2», а на выходе получен вектор [0 0.11 **0.76** 0.23 0.01 0.05 0 0 0 0.25], то распознавание прошло успешно.

## ПРАКТИЧЕСКАЯ ЧАСТЬ

Разработать программу «Распознавание образов», отвечающую следующим требованиям.

А) Распознавание образов должно выполняться искусственной нейронной сетью, обучаемой по алгоритму с обратным распространением ошибки. Общие требования к сети и программе:

- связи между нейронами – прямые;
- количество скрытых слоев – 1;
- количество нейронов выходного слоя (классов образов) – не менее 4;
- количество обучающих образов – не менее 1 на каждый класс образов;
- функция активации – сигмоидальная  $f(P) = \frac{1}{1+e^{-P}}$ .
- распознаваемые (тестовые) образы – формируются пользователем;

- норма обучения и количество эпох обучения – задаются пользователем;
- режим обучения или распознавания – задается пользователем.

Примерный интерфейс программы показан на следующем рисунке.

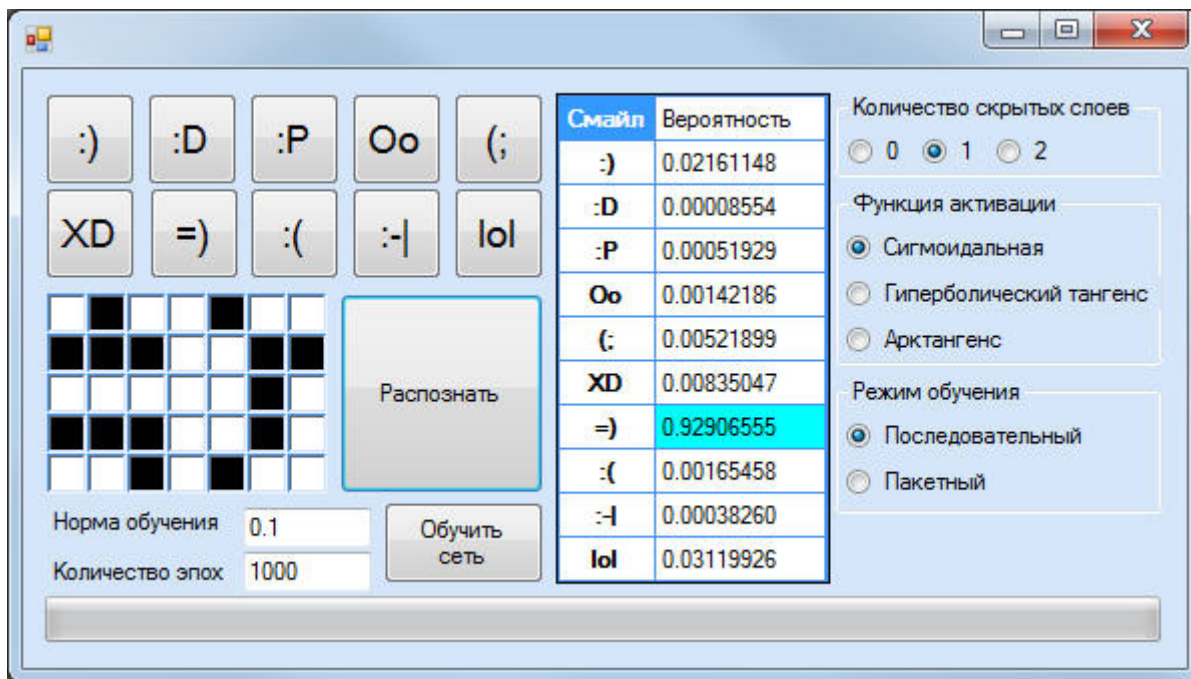


Рис. 11. Программа «Искусственная нейронная сеть»

**Б) Индивидуальный вариант выбрать согласно таблице 1.**

Варианты заданий на выполнение лабораторной работы

| № варианта | Тип образа  | Кол-во нейронов скрытого слоя |
|------------|---|-------------------------------|
| 1          | Арабские цифры  | 4                             |
| 2          | Римские цифры   | 5                             |
| 3          | Буквы русского алфавита   | 6                             |
| 4          | Буквы латинского алфавита   | 7                             |
| 5          | Арифметические операции<br>(+, -, /, *, $\sqrt{\quad}$ , % и т.п.)                      | 8                             |
| 6          | Логические операции<br>(&, $\wedge$ , $\vee$ , $\leftrightarrow$ , $\downarrow$ и т.п.) | 5                             |
| 7          | Операции сравнения<br>(<, =, $\geq$ , $\equiv$ , $\approx$ и т.п.)                      | 7                             |

|    |   |   |
|----|---|---|
| 8  | Операции с множествами<br>( $\cap$ , $\cup$ , $\emptyset$ , $\in$ и т.п.) | 6 |
| 9  | Цветочки  | 4 |
| 10 | Виды транспорта   | 5 |
| 11 | Машинки   | 6 |
| 12 | Карточные масти   | 7 |
| 13 | Птицы   | 8 |
| 14 | Рыбы  | 4 |
| 15 | Фрукты  | 5 |
| 16 | Смайлики  | 6 |
| 17 | Геометрические фигуры   | 7 |
| 18 | Дорожные знаки  | 8 |
| 19 | Виды заливки  | 4 |
| 20 | Логотипы компаний   | 6 |
| 21 | Знаки зодиака   | 5 |
| 22 | Домино  | 4 |

## СОДЕРЖАНИЕ ОТЧЕТА

1. Наименование и цель работы.
2. Исходный код модулей программного обеспечения.
3. Выводы по работе.