

# МЕТОДЫ ОБРАТНОГО ПРЕОБРАЗОВАНИЯ И РАСПРОСТРАНЕНИЯ В СИСТЕМАХ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

## *Back Propagation and Transformation Methods in Artificial Intelligence Systems*

Г. Москвин

Латвийский сельскохозяйственный университет

Liela iela 2, Jelgava LV- 3001, LLU

E-mail: logicor@llu.lv.lu

### **Abstract**

*Detailed description of methods of back propagation and back transformation also distributions for training of neural networks is given. A comparative estimation of a priority of methods of back transformation and back propagation for the decision of tasks of synthesis and training of neural networks, also for intelligent automatic measuring and AI systems for the first time is carried out.*

**Key words:** *neural network, intelligent automatic measuring systems, artificial intelligence.*

### **Введение**

В последние десятилетия в мире бурно развивается новая прикладная область математики, специализирующаяся на искусственных нейронных сетях (НС). Актуальность исследований в этом направлении подтверждается многообразием различных применений НС. Это автоматизация технологических процессов и распознавания образов, адаптивное управление, аппроксимация функционалов, прогнозирование, создание экспертных систем (ЭС) и систем искусственного интеллекта (ИИ), разработка ассоциативной памяти и многие другие приложения. Но широкий круг задач, решаемый НС, всё ещё не позволяет в настоящее время создавать универсальные, мощные интеллектуальные сети, вынуждая разрабатывать специализированные НС, функционирующие по различным алгоритмам. Модели НС в системах ИИ могут быть выполнены в виде программного и аппаратного исполнения [1-11].

Основу каждой НС составляют относительно простые, в большинстве случаев однотипные элементы (ячейки), имитирующие работу нейронов мозга. Под нейроном подразумевается искусственный нейрон, ячейка НС, рис 1.

Каждый нейрон характеризуется своим текущим состоянием по аналогии с нервными клетками головного мозга, которые могут быть возбуждены или заторможены. Нейрон обладает группой синапсов – однонаправленных входных связей, соединенных с выходами других нейронов, а также имеет аксон – выходную связь данного нейрона, с которой сигнал (возбуждения или торможения) поступает на синапсы следующих нейронов. Каждый синапс по физическому смыслу эквивалентен электрической проводимости и характеризуется величиной синаптической связи или ее весом  $W_i$ .

Текущее состояние нейрона определяется как взвешенная сумма его входов:

$$S = \sum_{i=1}^n X_i \cdot W_i$$

Выход нейрона есть функция его состояния:  $Y = f(S)$

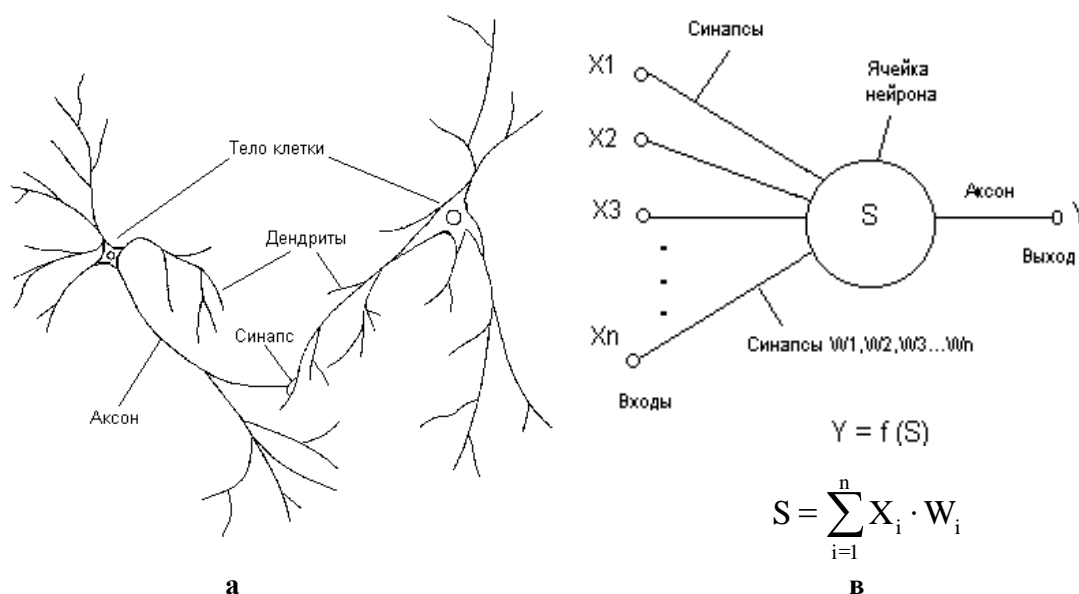


Рис.1 Нейрон мозга (а) и схема искусственного нейрона (в)

Вопрос о необходимых и достаточных свойствах сети для решения конкретных задач представляет собой отдельное направление нейрокомпьютерной науки. Так как синтез НС зависит от конкретно решаемой задачи, то в большинстве случаев оптимальный вариант получается на основе интуитивного подбора.

Процесс функционирования НС, то есть сущность действий, которые она способна выполнять, зависит от величин синаптических связей. Поэтому, задавшись определенной структурой НС, отвечающей какой-либо задаче, разработчик сети должен найти оптимальные значения всех переменных весовых коэффициентов. Этот этап называется обучением НС, и от того, насколько качественно он будет выполнен, зависит способность сети решать поставленные перед ней проблемы во время эксплуатации. На этапе обучения кроме параметра качества подбора весов важную роль играет время обучения. Как правило, эти два параметра связаны обратной зависимостью и их приходится выбирать на основе компромисса.

Обучение НС может вестись с учителем или без него. В первом случае сети предъявляются значения как входных, так и желательных выходных сигналов, и она по некоторому внутреннему алгоритму подстраивает веса своих синаптических связей. Во втором случае выходы НС формируются самостоятельно, а веса изменяются по алгоритму, учитывающему только входные и производные от них сигналы.

Существует множество различных алгоритмов обучения, которые делятся на два больших класса: детерминистские и стохастические. В первом из них подстройка весов представляет собой жесткую последовательность действий, во втором – она производится на основе действий, подчиняющихся некоторому случайному процессу [1-6,8,10]. Обычно формируют бинарные и аналоговые сети. Первые из них оперируют с двоичными сигналами, и выход каждого нейрона может принимать только два значения: логический ноль ("заторможенное" состояние) и логическая единица ("возбужденное" состояние). К этому классу сетей относится и перцептрон, так как выходы его нейронов, формируемые функцией единичного скачка, равны либо 0, либо 1. В аналоговых сетях выходные значения нейронов способны принимать непрерывные

значения. Алгоритмически ход времени в НС задается итерационным выполнением однотипных действий над нейронами.

### **Алгоритмы обратного распространения и преобразования**

Алгоритмы обратного преобразования и распространения проще всего понять, когда все элементы сети линейны. Алгоритм вычисляет каждую  $EW$ , сначала вычисляя  $EА$  - скорость, с которой изменяется ошибка при изменении уровня активности элемента. Для выходных элементов  $EА$  является просто разностью между действительным и желаемым выходом.

Чтобы вычислить  $EА$  для скрытого элемента в слое, непосредственно предшествующем выходному слою, сначала идентифицируются все веса между этим скрытым элементом и выходными элементами, с которыми соединен данный скрытый элемент. Затем эти веса умножаются на величины  $EА$  для этих выходных элементов и складываются полученные произведения. Эта сумма равна  $EА$  для данного скрытого элемента. Вычислив  $EА$  для всех элементов скрытого слоя, прилегающего к выходному, можно аналогичным образом рассчитать  $EА$  и для других слоев, перемещаясь в направлении, обратном тому направлению, в котором активность нейронов распространяется по сети. Отсюда и название алгоритма обратного «прослеживания» (обратного распространения или преобразования). После того, как значение  $EА$  для элемента вычислено, подсчитать  $EW$  для каждой входной связи элемента уже несложно. Величина  $EW$  является произведением  $EА$  и активности во входной цепи. Для нелинейных элементов алгоритм обратного распространения включает дополнительный шаг. Перед перемещением в обратном направлении  $EА$  необходимо преобразовать в  $EІ$  - скорость, с которой изменяется ошибка по мере изменения суммарного входа элемента.

Чтобы обучить нейронную сеть (НС) решению какой-либо задачи, веса каждого элемента корректируются таким образом, чтобы уменьшалась ошибка - расхождение между действительным и желаемым выходом. Для этого нужно, чтобы нейронная сеть вычисляла производную от ошибки по весам ( $EW$ ). Другими словами, она должна вычислять, как изменяется ошибка при небольшом увеличении или уменьшении каждого веса. Чаще всего для вычисления  $EW$  применяется именно алгоритмы обратного распространения и преобразования [1-12, 14, 17, 42].

Чтобы реализовать эти алгоритмы, в качестве примера дадим математическое описание нейронной сети. Предположим, что элемент  $j$  - типичный элемент выходного слоя, а элемент  $i$  - типичный элемент слоя, который предшествует выходному. Активность элемента выходного слоя определяется двух шаговой процедурой. Сначала вычисляется суммарный взвешенный вход  $X_j$  с помощью формулы:

$$X_j = S_i (Y_i W_{ij}),$$

где  $Y_i$  - уровень активности  $i$ -го элемента в предшествующем слое и  $W_{ij}$  - вес связи между  $i$ -м и  $j$ -м элементами. Далее, элемент вычисляет активность  $Y_j$  с помощью некоторой функции от суммарного взвешенного входа. Обычно применяется сигма-функция:

$$Y_j = 1 / [1 + e^{(-X_j)}]$$

После того, как активности всех выходных элементов определены, сеть вычисляет ошибку, которая определяется выражением:

$$E = 1/2 \sum (Y_j - D_j)^2$$

где  $Y_j$  - уровень активности  $j$ -го элемента в верхнем слое, а  $D_j$  - желаемый выход  $j$ -го элемента.

Алгоритм обратного преобразования и распространения состоит из следующих основных шагов.

1) Вычислить, насколько быстро меняется ошибка при изменении выходного элемента. Эта производная ошибки (EA) есть разность между действительной и ожидаемой активностью.

$$EA_j = \frac{dE}{dY_j} = Y_j - D_j$$

2) Вычислить, насколько быстро изменяется ошибка по мере изменения суммарного входа, получаемого выходным элементом. Эта величина (EI) есть результат шага 1, умноженный на скорость изменения выходного элемента с изменением его суммарного входа.

$$EI_j = \frac{dE}{dX_j} = \frac{dE}{dY_j} \frac{dY_j}{dX_j} = EI_j Y_j (1 - Y_j)$$

3) Вычислить, как быстро изменяется ошибка по мере изменения веса на входной связи выходного элемента. Эта величина (EW) есть результат шага 2, умноженный на уровень активности элемента, из которого исходит связь.

$$EW_{ij} = \frac{dE}{dW_{ij}} = \frac{E}{dX_j} \frac{dX_j}{dX_{ij}} = EI_j Y_i$$

4) Вычислить, как быстро изменяется ошибка с изменением активности элемента из предыдущего слоя. Этот ключевой шаг позволяет применять обратное распространение к многослойным сетям. Когда активность элемента из предыдущего слоя изменяется, это влияет на активности всех выходных элементов, с которыми он связан. Поэтому, чтобы подсчитать суммарное воздействие на ошибку, мы складываем все эти воздействия на выходные элементы. Но эти воздействия нетрудно подсчитать. Этот результат шага 2, умноженный на вес связи к соответствующему выходному элементу.

$$EA_i = \frac{dE}{dY_{ij}} = S \left( \frac{dE}{dX_j} \frac{dX_j}{dY_{ij}} \right) = S (EI_j W_{ij})$$

Пользуясь шагами 2 и 4, мы можем преобразовать величины EA одного слоя элементов в EA предыдущего слоя. Эту процедуру можно повторять, чтобы вычислять EA столько предыдущих слоев, сколько их есть. Зная EA для элемента, мы можем воспользоваться шагами 2 и 3, чтобы вычислить EW на его выходных связях. Нейронную сеть можно научить решать определенную задачу, пользуясь следующей процедурой. Сначала следует предъявить сети серию тренировочных примеров, которые состоят из паттерна активностей входных элементов вместе с желаемым паттерном активностей выходных элементов [1-6].

Предположим, что мы хотим научить сеть распознавать образы в виде рукописных цифр. Можно воспользоваться матрицей, из 256 сенсоров, каждый из которых регистрирует присутствие или отсутствие чернильного пятнышка в пределах маленькой площадки - фрагмента одной цифры. Для сети, таким образом, потребуется 256 входных элементов (по одному на каждый сенсор), 10 выходных элементов (по одному на каждую возможную цифру) и некоторое количество скрытых элементов. Для

каждой цифры, регистрируемой сенсорами, сеть должна генерировать высокую активность в соответствующем выходном элементе и низкую в остальных выходных элементах.

Чтобы натренировать систему, следует предъявить ей изображение цифры и сравнить действительную активность на 10 выходных элементах с желаемой активностью. Затем надо подсчитать ошибку, определяемую как квадрат разности между действительным и желаемым выходом. После этого мы изменяем вес каждой связи, с тем, чтобы уменьшить ошибку. Описанный процесс тренировки следует повторять со многими различными представлениями каждой цифры, пока сеть не научится правильно распознавать все возможные изображения. Эта величина, называемая производной ошибки по весу и обозначаемая  $EW$ , вычисляется. Один из способов вычисления  $EW$  заключается в том, чтобы изменить вес на очень маленькую величину и посмотреть, как изменится ошибка. Чтобы реализовать эту процедуру, нужно изменять каждый вес на величину, пропорциональную скорости, с которой изменяется ошибка по мере изменения веса [2-5,12-45]. Этот метод требует отдельных вариаций для каждого из многих весов.

### **Оценка приоритетов алгоритмов обратного преобразования и распространения**

На протяжении нескольких лет после их опубликования, алгоритмы обратного распространения и преобразования оставались почти незамеченными и невостребованными. Вероятно, потому, что не были в должной мере оценены специалистами. В середине 70-х годов Г. Москвин (ЛСХА), в связи с разработкой автоматизированных микропроцессорных систем управления и созданием интеллектуальных измерительных приборов, которые были запатентованы в 26 ведущих странах мира, а затем в начале 80-х годов Д. Румельхарт, работавший в то время в Калифорнийском университете в Сан-Диего, и Д. Паркер из Стэнфордского университета независимо друг от друга впервые разработали свои оригинальные варианты применения алгоритмов обратного преобразования и распространения. В 1986 году Румельхарт, Р. Уильямс, также из Калифорнийского университета в Сан-Диего [9], и Джеффри Е. Хинтон [2] продемонстрировали уникальную способность алгоритмов обучать скрытые элементы и вырабатывать интересные представления для сложных паттернов на входе, что и обусловило их распространение.

В середине 70-х годов прошлого века, возможно, в 1974 году Поль Дж. Вербос и в 1977 году Геннадий Москвин (Латв.СХА) предложили эффективную процедуру для вычисления коэффициентов коррекции весовых функций различных статистических характеристик, например, производной ошибки по весу ( $EW$ ), когда работали над своей докторской диссертацией - Поль Дж. Вербос (back propagation algorithm) в Гарвардском университете, Г. Москвин в Латвийской сельскохозяйственной академии, г. Елгава, (back transformation algorithm) [12-17,42]. Процедуры, известные теперь как алгоритмы обратного «распространения» (BPA) у П. Вербоса (back propagation algorithm) и «преобразования» (back transformation algorithm, BTA) у Г. Москвина, практически аналогичны и стали одними из наиболее важных инструментов в обучении нейронных сетей, в повышении точности автоматизированных систем и интеллектуальных измерительных приборов.

Г. Москвин [17, 42] (ЛСХА), а также Р. Андерсен из Массачусетского технологического института и Д. Зипсер из Калифорнийского университета в Сан-Диего [9] теоретически и практически доказали, что алгоритмы обратного преобразования и распространения представляют собой весьма эффективный инструмент не только для совершенствования технических систем, но и для понимания функций некоторых нейронов в коре головного мозга. Предложенные алгоритмы

позволяют научить нейронную сеть реагировать на различные сенсорные стимулы – звуковые сигналы, визуальные образы, импульсные и тактильные сигналы и так далее. Было обнаружено, что реакция скрытых элементов удивительно схожа с реакцией реальных нейронов, выполняющих преобразование визуальной информации, поступающей от сетчатки, в форму, необходимую для более глубоких областей мозга, перерабатывающих сигналы визуальных образов. При этом выяснилось, что восхождение абстракций перцептивных образов в системах ИИ происходит аналогично работе НС: от *абстрактных* ощущений, восприятий, представлений, понятий, суждений к *конкретным* выводам - умозаключениям. Этот результат сегодня имеет исключительно большое значение при разработке и создании локальных систем разделенного интеллекта, интеллектуальных адаптивных систем управления, при создании интеллектуальных высокоточных измерительных систем и систем искусственного интеллекта (ИИ). Алгоритмы обратного преобразования и распространения оказались на удивление эффективными в обучении сетей со многими слоями для решения широкого класса задач [1-5,12-45]. Они оказались наиболее эффективны в ситуациях, когда отношения между входом и выходом нелинейны, а количество обучающих данных достаточно велико.

Применяя обсуждаемые алгоритмы, были созданы нейронные сети, способные распознавать графические образы и рукописные символы [3], предсказывать изменения валютного курса, оптимизировать технологические процессы [12-16], обеспечивать высочайшую точность автоматических измерений в системах искусственного интеллекта, например, на основе парадигмы построения интеллектуальных измерительных систем с использованием фракталов метрических образов, рис 2. [42-45].

Видимая простота использования нейронных сетей весьма обманчива. Подобно настоящим живым организмам, нейронные сети обладают различными потенциальными возможностями. И хотя их можно контролировать, изменяя параметры создаваемых сетей, такая задача вовсе не является тривиальной. Настройка нейронных сетей и сейчас, после многих лет их успешного применения, остается в большой степени искусством, чем наукой. Поэтому не случайно сегодня на помощь НС пришли так называемые генетические алгоритмы (ГА). Они проводят естественный отбор, в ходе которого выживает «сильнейший», то есть, лучшая НС сеть, наиболее подходящая для решения поставленной задачи.

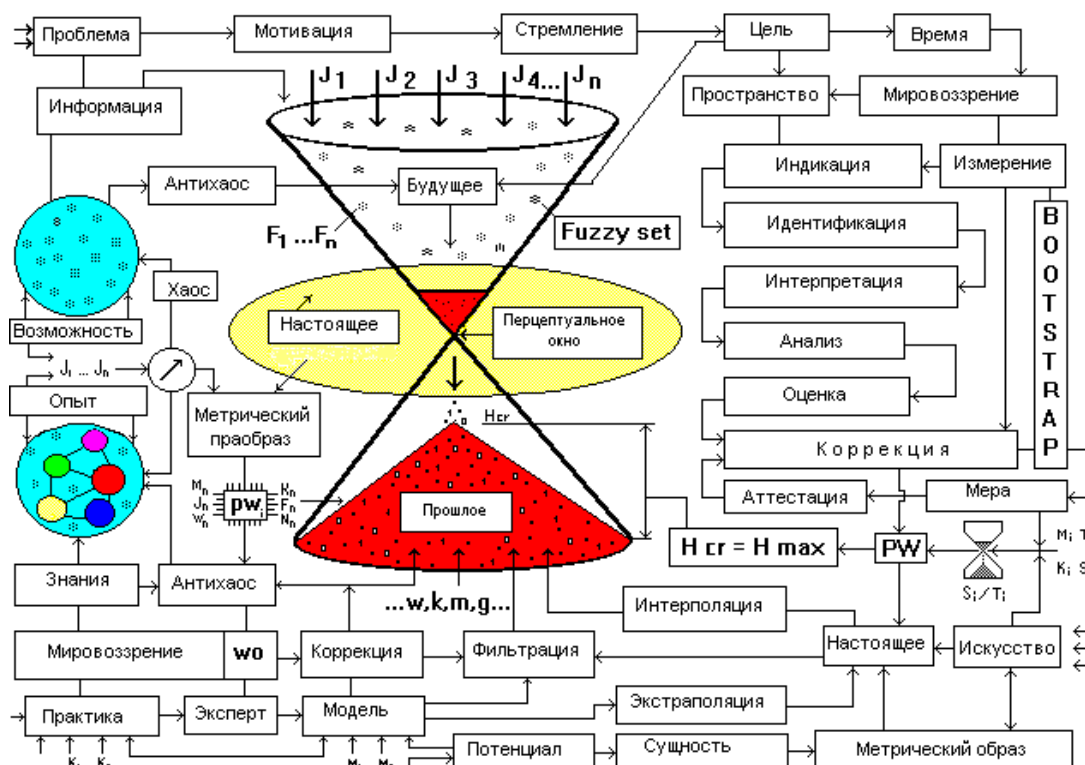


Рис.2 Парадигма интеллектуальной измерительной системы ИИ

Генетические алгоритмы – это самообучающаяся модель, которая для решения поставленных перед ней задач использует механизмы естественного отбора. Здесь используются такие понятия, как популяция, особь, наследование и мутация. Сначала алгоритм выдвигает возможные решения. Затем, в каждом поколении, подобно биологическим организмам, "сильнейшие" решения "скрещиваются", порождая новые решения, а "слабые" погибают. Генетические алгоритмы – это один из новых методов оптимизации НС. Возможности ГА не ограничиваются настройкой нейронных сетей. ГА используются для решения самых различных задач. Использование генетических алгоритмов совместно с нейронными сетями для подбора входных данных и выбора оптимальных конфигураций сетей и параметров обучения уже показало свои сильные стороны и получает все большее распространение в системах автоматического управления с ограниченным интеллектом. Когда мы, наконец, узнаем, каким образом обучаются живые нейроны мозга, тогда и системы ИИ, использующие НС и ГА смогут конкурировать с естественным интеллектом (ЕИ) человека. Разумеется, что пока искусственные НС и ГА с мозгом человека конкурировать не могут. Проблема заключается вовсе не в способе обучения системы, а в наличии великой тайны исходной модели устройства и функционирования живого нейрона. Методы обратного распространения и трансформации не имеют аналогов в живой природе, поэтому применение методов обратного распространения и преобразования сегодня не являются универсальным средством решения всех проблем. И, тем не менее, это достаточно эффективный метод интеллектуализации технологий, разумный дополнительный инструмент преобразования будущего на основе широкого применения нанотехнологий и систем искусственного интеллекта.

## Выводы

1. Методы обратного преобразования и распространения сегодня довольно широко используются, так как они достаточно хорошо работают при создании представлений о распознаваемом зрительном или метрическом образе в скрытых элементах сети. В данной работе впервые расставлены акценты авторских приоритетов на алгоритмы методов обратного преобразования и распространения, которые показали высокую эффективность процедур обучения НС, уменьшения ошибки и коррекции точности интеллектуальных систем, в которых веса постепенно изменяются.
2. Следует признать заблуждением мнение многих ученых, которые около 25 лет назад полагали, что подобные методы окажутся совершенно бесполезными, поскольку должны были, по их мнению, неизбежно приводить к локальным оптимумам, и в более широком масштабе – к неверным решениям. Несостоятельным также оказалось распространенное убеждение, что процедура обучения представляет интерес только в том случае, если она гарантирует сходимость к глобально оптимальному решению. Методы обратного преобразования и распространения показали, что для многих задач глобальная сходимость не является необходимым условием для того, чтобы достичь приемлемых результатов.
3. С биологической точки зрения сегодня следует признать, что методы обратного преобразования и распространения, как принцип подобия работы головного мозга, сегодня выглядят не очень убедительно. Наиболее очевидная трудность однозначного утверждения заключается в том, что информация должна проходить по тем же самым связям в обратном направлении, от каждого последующего уровня к предыдущему. В реальных нейронах мозга существует множество путей, ведущих от последующих слоев нервных клеток к предыдущим. Эти нейронные связи могут использоваться для передачи информации, необходимой для обучения. Осуществляется ли такая обратная связь в нейронных сетях живого мозга на самом деле? Наука о мозге пока не дала нам однозначного ответа.
4. Более трудную проблему представляет собой быстрое действие алгоритмов обратного преобразования и распространения. Здесь центральным является вопрос о том, как быстро растет время, необходимое для обучения, по мере возрастания размеров сети. Время, требующееся для вычисления производных от ошибки по весам на заданном тренировочном примере, пропорционально размерам сети, поскольку объем вычислений пропорционален количеству весов. Однако более крупные сети требуют большего количества тренировочных примеров, и им приходится модифицировать веса большее число раз. Следовательно, время обучения растет значительно быстрее, чем размеры сети.
5. С точки зрения создания систем искусственного интеллекта (ИИ) самая серьезная проблема применения методов обратного преобразования и распространения сегодня заключается в том, что такая НС требует учителя, предоставляющего желаемый выход для каждого тренировочного примера. Но, во - первых, в естественной среде обитания человек способен обучаться и без помощи учителя. Во - вторых, никто не дает нам заранее детальное описание и готовые к употреблению идеальные глобальные представления картины мира, по которым мы могли бы научиться извлекать пользу при использовании нашего собственного, локального сенсорного входа. В третьих, мы учимся понимать речь, зрительные сцены и визуальные образы без каких-либо прямых инструкций, то есть, мы всё же способны воспринимать готовые к употреблению знания непосредственно, эзотерически, без участия интеллектуальных посредников.



6. Если нейронная сеть сталкивается с большим набором сочетаний сигналов, но не имеет никакой информации о том, что с ними следует делать, то, очевидно, что задача для нейронной сети просто нечетко сформулирована. Исследователи разработали несколько универсальных, неконтролируемых процедур, которые могут правильно регулировать весовые параметры сети. Все эти процедуры имеют два общих качества: они оперируют, явно или неявно, с некоторым понятием качества представления и работают, изменяя веса, чтобы повысить качество представления, вырабатываемого скрытыми элементами.
7. Несмотря на отмеченные недостатки, применение методов обратного преобразования и распространения и сегодня весьма эффективно, так как при этом снижается степень неопределенности системы и поэтому не возникает ситуации неопределенности действий, которые необходимо проделать с информацией, поступающей на вход отдельных нейронных сетей или системы ИИ в целом.

#### Литература

1. Барцев С.И. Некоторые свойства адаптивных сетей //Красноярск: Институт физики СО АН СССР,1987.
2. Джеффри Е. Хинтон. Как обучаются нейронные сети.// В мире науки , 1992 , N 11, N 12, с. 103-107.
3. Куусуль В.М., Байдык Т.Н. Разработка архитектуры нейроподобной сети для распознавания формы объектов на изображении.//Автоматика , 1990, N 5, с. 56-61.
4. Суворов С.В., Матихина Н.Ю. Программное моделирование нейроподобных структур.//Распределенная обработка информации. Улан-Уде, 1989, с. 28.
5. Трикоз Д.В. Нейронные сети: как это делается?// Компьютеры + программы,1993, N 4(5), с. 14-20.
6. Bardcev S.I., Okhonin V.A. The algorithm of dual functioning (back-propagation): general approach, versions and applications. Krasnojarsk: Inst. of biophysics SB AS USSA - 1989.
7. Computing with neural circuits: a model.//Science, 1986. V. 233. p. 625-633.
8. Fogelman Soulie F. Neural networks, state of the art, neural computing.// London: IBC Technical Services, 1991.
9. Kuzewski Robert M., Myers Michael H., Grawford William J. Exploration of fourword error propagation as self organization structure.//IEEE Ist. Int. Conf. Neural Networks, San Diego, Calif., June 21-24, 1987. V. 2. - San Diego, Calif., 1987. - p. 89-95.
10. Rumelhart B.E., Minton G.E., Williams R.J. Learning representations by back propagating error.// Wature, 1986. V. 323. p. 1016-1028.
11. Takefuji D.Y. A new model of neural networks for error correction.//Proc. 9th Annu Conf. IEEE Eng. Med. and Biol. Soc., Boston, Mass., Nov. 13-16, 1987. V. 3, New York, N.Y., 1987 - p. 1709-1710.
12. Москвин Г. Некоторые вопросы точности автоматических измерительных систем// Труды ЛСХА, 1979, вып. 159, стр.51-58;
13. Москвин Г. и др. Распознавание животных при создании АСУТП.//Труды ЛСХА, 1979, вып.159, стр. 159-165.
14. Москвин Г., Бернис А. Пути повышения точности автоматизированного учета молока на молочных фермах с конвейерной технологией // 5-ый Всесоюзный симпозиум по машинному доению , часть 2, стр.17-22.
15. Москвин Г. Метод повышения точности учета молока // Труды ЛСХА, 1981, вып. 193, стр.35-43.
16. Москвин Г. Исследование метрологических аспектов повышения точности счетчиков молока // Труды ЛСХА, 1984, вып.224, стр. 113-124.
17. Москвин Г. Методы и технические средства учета молока на основе применения микропроцессорной техники. Автореферат к.т.н., Минск, 1988 , 26 с.
18. Москвин Г. А.с. СССР №1109092.
19. Москвин Г. А.с. СССР №1175403.
20. Москвин Г. А.с. СССР №1345059.
21. Москвин Г. А.с. СССР №1432825.
22. Москвин Г. А.с. СССР №1720599.
23. Москвин Г. А.с. СССР №4413033.
24. Москвин Г. А.с. СССР №1720601.
25. Москвин Г. А.с. СССР №1720600.
26. Москвин Г. А.с. СССР №1424150
27. Москвин Г. А.с. СССР №1731107.
28. Москвин Г. Европатент №0381762 A1

29. Москвин Г. Европатент №0406426 A1
30. Москвин Г. Европатент №0382852 A1
31. Москвин Г. Европатент №0372089 A1
32. Москвин Г. Европатент №0471076 A1
33. Москвин Г. Патент США №4989445
34. Москвин Г. Патент США №5012762
35. Москвин Г. Патент США №5161483
36. Москвин Г. Патент США №5016569
37. Москвин Г. Патент Новой Зеландии №231781
38. Москвин Г. Патент Новой Зеландии №228985
39. Москвин Г. Патент Новой Зеландии №228983
40. Москвин Г. Патент Новой Зеландии №228982
41. Москвин Г. Патент Венгрии № 205529
42. Moskvins G. Intelektualizētas automātiskās mērīšanas, dozēšanas un uzskaites sistēmas// Dr.hab.sc.ing. zin. darba kopsavilkums, Jelgava, 1996, 92 lpp.
43. Mākslīgā intelekta aktualitātes.// Zinātnes filozofija. LLU, Jelgava, 2002, lpp. 76-89;
44. Moskvin G.A. Artificial Intelligence Measuring, Automatic Control and Expert Systems in Agriculture,3-rd IFAC-CIGR Workshop on Artificial Intelligence in Agriculture, Makuhari, Japan, April 24-26,1998, Preprints, p.176-181.
45. New Method and Low-Cost Intelligent Instruments for the Fraud Detection and Conformity Control of Agricultural Products. 2002 ASAE Annual Meeting and XV CIGR WORLD Congress . July 28 -July 31, 2002 Hyatt Regency, Chicago, IL, USA, 14 pag., ASAE Paper 023077.