

РЕШАЮЩИЙ АЛГОРИТМ КОНКУРЕНЦИИ

В. С. Антюфеев

Институт вычислительной математики и математической геофизики СО РАН,
630090, Новосибирск, Россия

УДК 519.6

Предложен алгоритм решения задач, которые обычно решаются с помощью искусственных нейронных сетей (ИНС). Принципы работы нового алгоритма существенно отличаются от принципов работы ИНС (не используются понятия искусственного нейрона, нейронной сети). Обучающие сигналы реализуются методом Монте-Карло. Работа алгоритма может быть исследована математическими методами. Рассмотрен численный пример использования нового алгоритма.

Ключевые слова: искусственная нейронная сеть, нечеткий алгоритм, метод Монте-Карло.

The paper describes a fuzzy decision algorithm designed for solving problems that are usually solved with a help of artificial neural networks (ANN). But it uses quite different principles of its operating. It does not use concepts of artificial neuron, network. The learning signal are brought about by the Monte Carlo method. Mathematical methods can be applied to its investigation. A numerical example of its application is considered.

Key words: artificial neural network, fuzzy algorithm, Monte Carlo method.

Введение. Теория распознавания образов — раздел кибернетики, развивающий теорию и методы классификации и идентификации предметов, процессов, сигналов, которые характеризуются конечным набором числовых параметров (свойств, признаков). По этим параметрам определяется (распознается) класс, к которому относится объект. Иными словами, распознавание образов — это решение задачи соотнесения объекта с заданными характеристиками с определенным классом объектов.

Необходимость в таком распознавании возникает в различных областях начиная с военной промышленности и систем безопасности и кончая оцифровкой всевозможных аналоговых сигналов [1–3].

Задачи распознавания решаются с помощью искусственных нейронных сетей (ИНС), представляющих собой сложный логический алгоритм, который соотносится (возможно, необоснованно) с работой мозга [2, 3].

Основной элемент ИНС — искусственный нейрон. Для наглядности алгоритм изображается на специальной диаграмме (блок-схеме), напоминающей сеть. Поэтому алгоритм называется искусственной нейронной сетью. Основным свойством ИНС является способность к адаптации или обучению. Действия ИНС основаны на параллельных вычислениях [4].

ИНС имеют различные возможности: существует сеть, превращающая текст в фонетическое представление, которое затем с помощью уже иных методов превращается в речь [5]; создана сеть, способная распознавать рукописные буквы [6]; сконструирована система сжатия изображений, основанная на нейронной сети [7].

ИНС созданы в 50–60-х гг. XX в. и активно используются для решения разнообразных задач, например для предсказания погоды, анализа электрокардиограмм. Действительно, сравнительно простые задачи могут быть решены с помощью простейших однослойных сетей — перцептронов [2]. Сформулирована и доказана теорема об обучении перцептронов, т. е. о способности таких сетей решать задачи определенного вида [8].

Однако вскоре было обнаружено, что “сети не могли решать задачи, внешне весьма сходные с теми, которые они успешно решали” [2]. Исследования, проведенные в работе [9], показали, что используемые в то время перцептроны неспособны решить многие простые задачи. Это обусловило снижение интереса к исследованиям ИНС.

Исследования ИНС возобновились после того, как были разработаны так называемые многослойные ИНС [2] и методы их обучения, которые позволили решать задачи, не разрешимые для перцептронов. В настоящее время все ИНС для решения практических задач являются многослойными.

Следует отметить, что несмотря на огромное количество исследований в данной области, до сих пор имеются нерешенные теоретические и практические вопросы, связанные с ИНС. Например, не гарантировано обучение сети за конечное время. В то же время внутреннее представление информации, полученной в результате обучения, зачастую невозможно проанализировать, чтобы исправить ошибку обучения. Необходима мощная теорема сходимости (теорема обучения) для многослойных машин [2].

В настоящей работе предложен новый решающий алгоритм — алгоритм конкуренции. Эвристическая основа данного алгоритма не имеет отношения к предполагаемым свойствам биологических нейронов. Этот алгоритм основан на соображениях конкурентной борьбы между обучающими сигналами за влияние на остальные точки пространства сигналов, что отражено в названии алгоритма.

Предлагаемый алгоритм позволяет решать задачи, близкие к задачам, решаемым с помощью ИНС. Однако обучение и работа этого алгоритма основаны на других принципах, позволивших существенно упростить алгоритм, сделать его работу понятной. В частности, оказалось возможным применить при исследовании нового решающего алгоритма методы математического анализа, теории вероятностей и доказать соответствующую теорему обучения (ее доказательство не вошло в настоящую работу), которую не удалось доказать для многослойных ИНС.

1. Краткое описание алгоритма работы ИНС. Подробное описание ИНС приведено в различных учебниках. Ниже кратко перечислены принципы работы ИНС.

1. Сначала ИНС функционирует в режиме обучения, перевычисляя (настраивая) веса w_1, \dots, w_n (параметры сети). Начальные значения весов заданы заранее. На вход ИНС подается последовательность обучающих сигналов — числовые кортежи $z = (z_1, \dots, z_n)$, координаты которых описывают свойства исследуемого объекта (например, результаты обследования пациента — давление крови, температура тела и другие параметры). Для каждого обучающего сигнала заранее известен правильный ответ на поставленный вопрос, который будем называть основным вопросом. Верный ответ ИНС получает от “учителя”. Можно считать, что “учителем”, дающим правильные ответы, является специальный алгоритм.

2. Когда обучение закончено, ИНС продолжает функционировать в рабочем режиме. На вход поступают рабочие сигналы с неизвестными ответами, для которых ИНС должна получить ответы на основании совокупной информации, полученной от “учителя” в режиме обучения и сохраняемой весами сети.

3. Используя данную информацию, т. е. веса, ИНС обрабатывает сигналы $z = (z_1, \dots, z_n)$. Все слои сети [2] начиная с первого последовательно выполняют пару преобразований над вектором z :

1) $z \implies w \cdot z$ (w — матрица весов);

2) $z_k \implies F(z_k)$, где $F : R \rightarrow R$ — так называемая активационная функция [2]; и получают выходной сигнал.

4. ИНС относит выходной кортеж к определенной группе по определенному правилу. На основании этого обследуемый пациент может быть отнесен к группе больных или, наоборот, к группе здоровых. Последний вывод и является целью работы ИНС с входным сигналом z .

На выходе сети получается одно окончательное число. Если это число не превышает $1/2$, то считается, что ИНС выдала 0, и исследуемый объект относится к одной группе, в противном случае — ко второй.

Данная схема работы ИНС является упрощенной, но в ней отмечены все принципиальные моменты. Сложная логика ИНС не позволяет понять до конца ее работу. Возможно, представление об ИНС как о логическом алгоритме, копирующем работу головного мозга, препятствует получению доказательства теоремы обучаемости.

Рассмотрим математическую интерпретацию работы ИНС, объясняющую смысл ее обучения и работы и позволяющую построить новый решающий алгоритм, основанный на другом принципе.

2. Геометрическое описание ИНС. Ниже изложена математическая интерпретация работы ИНС, позволившая создать новый решающий алгоритм.

Пусть ИНС прошла обучение с помощью k обучающих сигналов, и на вход ИНС поступил сигнал $z = (z_1, \dots, z_n)$. Используя веса w_{ij} , ИНС дает на выходе ответ 0 или 1. Рассмотрим процедуру получения ответа с геометрической точки зрения.

Согласно определению ИНС проверяет, удовлетворяют ли компоненты z_i следующей системе неравенств:

$$\{\text{неравенство 1}\} \& \{\text{неравенство 2}\} \& \{\text{неравенство 3}\} \vee \{\text{неравенство 4}\} \dots \quad (1)$$

Эта система составлена из элементарных линейных неравенств вида

$$\sum_j w_{mj} z_j < 1$$

или

$$\sum_j w_{mj} z_j > 1,$$

разделенных логическими знаками “и” и “или”. Порядок чередования логических знаков “&”, “ \vee ” в системе (1) определяет сложная топология графа нейронной сети.

На геометрическом языке каждое из приведенных элементарных неравенств задает полупространство в \mathbb{R}^n , ограниченное соответствующей гиперплоскостью, а система неравенств (1) определяет некоторое множество S_k (напомним, что k — число обучающих сигналов). Очевидно, S_k — многогранник или объединение нескольких разрозненных многогранников, возможно, невыпуклых. Отметим, что однослойная нейронная сеть позволяет образовывать лишь пересечения полупространств и как следствие лишь выпуклые односвязные множества. Поэтому однослойные сети не позволяли решать многие задачи. Лишь многослойные сети позволили конструировать сложные множества для решения сложных задач.

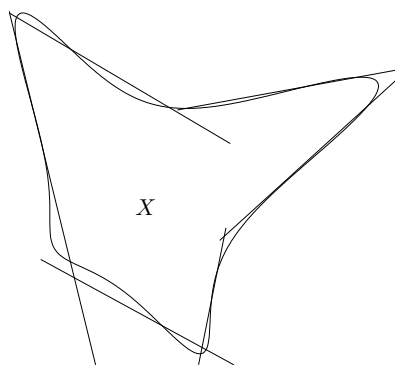


Рис. 1. Приближение искомого множества X многогранником S_k

Очевидно, сигнал z удовлетворяет системе неравенств (1) тогда и только тогда, когда точка z попадает в множество S_k . Таким образом, геометрическое событие {точка z принадлежит S_k } эквивалентно логическому событию {числа z_i , $i = 1, \dots, n$ удовлетворяют системе неравенств (1)}.

Обозначим через X множество тех точек-сигналов, которым соответствует положительный ответ (т. е. 1) на основной вопрос. Если множество X известно, то задачу для ИНС можно считать решенной. Поэтому множество $X \subset \mathbb{R}^n$ будем называть решающим множеством.

Заметим, что ИНС может допускать ошибки после k -го шага обучения, т. е. для некоторых сигналов ответ ИНС может быть неверным. Причиной ошибочных ответов является то, что множество S_k , заданное с помощью весов w_{ij} , не совпадает с решающим множеством X (рис. 1). Соответственно множество S_k можно назвать приближенным решающим множеством.

Теперь функционирование ИНС в обучающем режиме можно описать геометрически. На каждом шаге ИНС пересчитывает набор весов и находит соответствующее этому набору приближенное решающее множество S_k (k — номер шага обучения). Таким образом, в процессе обучения ИНС формирует последовательность S_1, S_2, \dots приближенных решающих множеств.

На языке теории множеств обучаемость алгоритма ИНС означает, что последовательность S_1, S_2, \dots в определенном смысле сходится к идеальному решающему множеству X (смысл этой сходимости уточняется ниже).

Итак, с геометрической точки зрения процесс обучения ИНС сводится к приближению идеального решающего множества X многогранниками S_k , каждый из которых является композицией (объединениями, пересечениями и дополнениями) полупространств.

Если множество X не является многогранником или объединением многогранников, то никакое приближенное решающее множество S_k , ограниченное гиперплоскостями, заведомо не может быть идеальным. Это значит, что конечное число обучающих шагов не позволяет получить точное решение задачи, т. е. до конца обучить систему.

Простейшие ИНС (персептроны) умеют формировать лишь связные выпуклые множества. Многослойные сети были разработаны именно для того, чтобы алгоритм ИНС мог формировать невыпуклые, несвязные множества. Однако это привело к существенному усложнению алгоритма ИНС.

Заметим, что невыпуклость и несвязность идеального решающего множества не влияют на качество работы нового решающего алгоритма, не являются препятствием для решения задач.

3. Описание ИНС в терминах теории функций. Опишем работу ИНС на языке теории функций.

После обработки каждой точки-сигнала алгоритм ИНС ставит ей в соответствие 0 или 1. Поэтому искусственную нейронную сеть в рабочем режиме можно отождествить с функцией ИНС : $\mathbb{R}^n \rightarrow \{0; 1\}$. Эту функцию задают логический алгоритм вычислений и набор параметров w — веса сети. Поэтому $\text{ИНС}(z) \equiv \text{ИНС}_w(z)$. Геометрический смысл параметров w определен в п. 2: веса w задают направления нормалей граничных гиперплоскостей.

Аналогично можно определить нечеткую нейронную сеть [10, 11]. В этом случае областью значений функции f является весь промежуток $[0, 1]$, а не два числа 0 и 1:

$$\text{ИНС} : \mathbb{R}^n \rightarrow [0, 1].$$

Каждому множеству $S \subset \mathbb{R}^n$ взаимнооднозначно соответствует его характеристическая функция $\chi_S(z)$, $z \in \mathbb{R}^n$:

$$\chi_S(z) = \begin{cases} 1, & z \in S, \\ 0, & z \notin S. \end{cases}$$

При описании геометрической интерпретации ИНС определены решающие множества X , S_k — идеальное и приближенные. Теперь определим (идеальную) решающую функцию $f(z) \equiv \chi_X(z)$ и приближенные решающие функции $f_k(z) \equiv \chi_{S_k}(z)$. Очевидно, верный ответ на сигнал z в терминах 0-1 равен $\chi_X(z)$. С точки зрения теории функций процесс обучения ИНС представляет собой процесс приближения идеальной решающей функции $f(z)$ приближенными решающими функциями $f_k(z)$.

В конце п. 2 речь шла о сходимости последовательности множеств S_1, S_2, \dots к идеальному решающему множеству X :

$$\lim_{k \rightarrow \infty} S_k = X. \quad (2)$$

Дадим строгое определение этой сходимости. Выполнение предельного соотношения (2) означает, что почти всюду (за исключением границы множества X) имеет место поточечный предел последовательности характеристических функций соответствующих множеств:

$$\lim_{k \rightarrow \infty} \chi_{S_k} = \chi_X. \quad (3)$$

В процессе работы нейронная сеть формирует множества S_k , которые постепенно приближаются (в указанном выше смысле) к идеальному множеству X . Поэтому принцип работы стандартной нейронной сети можно рассматривать и с точки зрения теории функций: сначала ИНС формирует приближенные решающие множества в виде композиции полупространств, а затем в соответствии с этими множествами образует решающие функции в виде характеристических функций соответствующих решающих множеств.

4. Решающий алгоритм конкуренции. Определим новый решающий алгоритм.

Алгоритм ИНС разработан на основе представлений о функции биологического нейрона. В определении нового алгоритма не используется понятие нейронной сети, его работа основана на других естественных представлениях, описанных в данном пункте. Следствием этих представлений является новое понятие — функция влияния. Из этой функции нетрудно получить аналитическое представление для приближенных решающих функций нового

алгоритма, которые существенно отличаются от приближенных решающих функций для ИНС.

В новом алгоритме обучающие точки-сигналы являются случайными. Будем считать, что эти точки распределены равномерно на некотором ограниченном измеримом множестве $Z : X \subset Z \subset \mathbb{R}^n$. Как и выше, X — искомое множество, характеристическая функция $\chi_X(z)$ которого является идеальной решающей функцией алгоритма.

Моделируя случайные сигналы, получим случайные решающие функции $f_k(z)$ (k — количество обучающих сигналов). Найдем также формулы для приближенных решающих функций $f_k(z)$ нового алгоритма.

Приближенные решающие функции должны удовлетворять некоторым естественным требованиям. Перечислим эти требования, а затем сконструируем функции, удовлетворяющие им.

4.1. *Условия для приближенных решающих функций.* Рассмотрим естественные требования к приближенным решающим функциям. Соответствующие условия на решающие функции, следующие из этих требований, перечислим в конце подп. 4.1.

1. Предлагаемый алгоритм является нечетким, его действие в рабочем режиме можно отождествить с отображением $f : \mathbb{R}^n \rightarrow [0; 1]$. Отметим, что область значений этой функции — отрезок $[0; 1]$, а не пара чисел $\{0; 1\}$. Соответственно ответ алгоритма для произвольного сигнала z , т. е. число $f(z)$, — это эмпирическая вероятность события $z \in X$ после всего курса обучения алгоритма. Аналогично для каждой приближенной решающей функции f_k число $f_k(z)$ — эмпирическая вероятность события $z \in X$ после k -го шага обучения, т. е. число из промежутка $[0; 1]$. Следовательно, областью значений функции f_k является промежуток $[0; 1]$.

2. Пусть предлагаемый алгоритм находится в режиме обучения и $z^1 = (z_1^1, \dots, z_n^1)$ — первый обучающий сигнал, поступивший на вход алгоритма. Пусть также точка z^1 попадает в искомое множество X . Тогда решающая функция f_1 должна принять в точке z^1 значение 1: $f_1(z^1) = 1$. Если теперь некоторый сигнал z попадает в малую окрестность точки z^1 , то вероятность события $z \in X$ также должна быть близка к 1. Аналогичные правила с соответствующими исправлениями должны выполняться и в случае $z^1 \notin X$. В результате получаем требование непрерывности f_k .

3. Каждая функция f_k должна быть построена с учетом всех предыдущих обучающих сигналов z^1, z^2, \dots, z^k , а не только последнего сигнала z^k , поэтому при поступлении последующих обучающих сигналов ее значения в точках z^1, z^2, \dots, z^k должны оставаться равными 1 или 0.

4. Одни и те же обучающие сигналы z^1, z^2, \dots, z^N могут поступить на вход в различном порядке. Очевидно, результат этого обучения — функция f_N — не должен зависеть от порядка поступления сигналов.

Кратко сформулируем требования 1–4 в виде условий на функции f_k :

- 1) $0 \leq f_k(z) \leq 1$;
- 2) функции f_k должны быть непрерывными;
- 3) $f_k(z^i) = \begin{cases} 1, & z^i \in X, \\ 0, & z^i \notin X, \end{cases} \quad i = 1, \dots, k$;
- 4) функция f_k не зависит от порядка поступления обучающих сигналов.

Построение функций простого вида, удовлетворяющих перечисленным условиям, является основной целью настоящей работы. Такие функции не удалось сконструировать непосредственно, без промежуточных шагов, понятий.

С эвристической точки зрения выполнение перечисленных условий обеспечивает для функций f_k выполнение предельного соотношения, аналогичного (3). Однако строгое доказательство соответствующей теоремы требует проведения дополнительного исследования.

При решении данной задачи было введено новое понятие — функция влияния (см. подп. 4.2). Это позволило найти сравнительно простой способ построения приближенных решающих функций, не усложняя алгоритм.

Заметим также, что построение решающих функций для нейронных сетей — очень сложная и громоздкая процедура, которую выполняет логический алгоритм ИНС.

4.2. *Функция влияния.* В предлагаемом алгоритме обучающие точки-сигналы являются случайными и распределены равномерно на некотором ограниченном множестве $Z : X \subset Z \subset \mathbb{R}^n$. Множество Z — это пространство, в котором лежат все возможные сигналы. Разобьем множество Z на положительное подмножество X и отрицательное подмножество Y :

$$X \cup Y = Z, \quad X \cap Y = \emptyset.$$

Пусть $z^1 = (z_1^1, \dots, z_n^1)$ — первый обучающий сигнал, поступивший на вход алгоритма. Как подсказал “учитель”, точка z^1 попала в множество X .

Очевидно, что $z^1 \in X$, т. е. можно считать, что это утверждение имеет “бесконечное значение”. Если точка z лежит в малой окрестности z^1 , то она, по-видимому, также попадает в множество X . Наша уверенность в этом велика, хотя и не бесконечна. По мере удаления точки z от z^1 эта уверенность монотонно убывает до нуля, так как в удаленных точках система “забывает” то, что случилось в точке z^1 .

Аналогично, если точка z^1 попадает в отрицательное множество Y , мы присваиваем нашей уверенности значение $-\infty$. В малой окрестности z^1 уверенность вновь имеет отрицательный знак, велика по абсолютной величине и стремится к нулю по мере удаления точки z от z^1 .

Можно считать, что каждый обучающий сигнал создает вокруг себя положительное или отрицательное скалярное поле влияния. Представляет интерес суперпозиция этих элементарных полей.

Зададим аналитическое выражение для интенсивности скалярного поля влияния сигнала. Сначала определим вспомогательную функцию $\varepsilon(z)$ — индикатор или знак сигнала z :

$$\varepsilon(z) = \begin{cases} +1, & z \in X, \\ -1, & z \notin X. \end{cases}$$

Интенсивность поля влияния определим с помощью функции $h(z)$:

$$h(z) = \begin{cases} \frac{1}{|z|^m}, & z \neq \mathbf{0}, \\ +\infty, & z = \mathbf{0}. \end{cases}$$

Здесь $m \geq n$ — произвольное число; n — размерность пространства сигналов.

Пусть на вход алгоритма поступает очередной обучающий сигнал z^k (k — номер шага обучения). Число

$$\varepsilon(z^k)h(z - z^k) = \frac{\varepsilon(z^k)}{|z - z^k|^m}, \quad z \neq z^k \quad (4)$$

можно назвать эвристическим значением уверенности в том, что знак точки z совпадает со знаком обучающего сигнала z^k (напомним, что знак сигнала z^k сообщил “учитель”). Действительно, в малой окрестности z^k значения функции (4) велики, что соответствует большой

уверенности в знаке сигналов z из этой окрестности. При удалении точки z от z^k значения функции (4) убывают по абсолютной величине до нуля, что соответствует убыванию уверенности в знаке сигнала z .

Выражение (4) подобно формуле закона Кулона для случая взаимодействия точечных зарядов. Отличие состоит в том, что в (4) поле влияния — скалярное, а электрическое поле — векторное. Кроме того, показатель степени m в знаменателе формулы может превышать (и обычно превышает) 2.

В предлагаемом решающем алгоритме, как в электростатике, каждый обучающий сигнал создает вокруг себя поле влияния на окружающие “пробные точки”. Назовем функцию $\varepsilon(z^k)h(z - z^k)$ с аргументом z и параметром z^k элементарной функцией влияния сигнала z^k . Эта функция определяет поле влияния отдельного сигнала z^k , учитывая знак сигнала и точку его расположения. Далее для краткости будем использовать обозначение

$$h_k(z) \equiv \varepsilon(z^k)h(z - z^k).$$

Итак, каждый обучающий сигнал создает вокруг себя скалярное поле положительного или отрицательного влияния на все остальные точки. Элементарная функция влияния определяет величину интенсивности этого поля.

Совокупность $\{z^1, z^2, \dots, z^N\}$ нескольких обучающих сигналов также “индуцирует” поле влияния. Это поле является суперпозицией элементарных полей отдельных сигналов z^k , $k = 1, 2, \dots, N$. Будет естественным определить величину интенсивности этого поля в точке z суммой $h_1(z) + \dots + h_N(z)$ значений элементарных функций влияния обучающих сигналов z^k , $k = 1, 2, \dots, N$. Иными словами, суперпозицию полей влияния, как и суперпозицию физических полей, будем вычислять обычным сложением.

Итак, определено простое правило пересчета поля влияния на каждом шаге обучения. Напомним, что отыскание простого правила композиции решающих функций с выполнением перечисленных в подп. 4.1 требований оказалось трудной задачей.

Все обучающие сигналы — конечное множество точек — своими полями оказывают влияние на знаки остальных точек-сигналов. При этом в борьбе за влияние на произвольный сигнал $z \neq z^1, z^2, \dots$ конкурируют две группы обучающих сигналов — положительные и отрицательные, которые пытаются присвоить ему положительный или отрицательный знак. Поэтому описываемый решающий алгоритм можно назвать алгоритмом конкуренции. В основе этого алгоритма лежит идея конкуренции решений, так же как в основе алгоритма ИНС лежит идея искусственного нейрона. Что касается обучающих точек z^1, z^2, \dots , то для них ответы на основной вопрос, принадлежит ли рассматриваемая точка множеству X , заведомо известны. Поэтому обучающие точки не подвержены влиянию и не изменяют свой цвет.

На рис. 2 приведена диаграмма для четырех отрицательных и трех положительных обучающих сигналов, иллюстрирующая эффект влияния обучающих сигналов на остальные точки. Четко видны зоны влияния каждого сигнала. По мере того как цвет точек пространства непрерывно изменяется от черного к белому, значения функции влияния возрастают от $-\infty$ до $+\infty$. Удаленным от обучающих сигналов точкам, в которых значения функций влияния близки к 0, соответствует нейтральный серый цвет.

Процесс обучения можно описать следующим образом. До начала обучения всем точкам пространства соответствует нейтральный серый цвет. Это значит, что сначала какая-либо информация отсутствует и с вероятностью $1/2$ каждую точку можно отнести в искомое множество X или в его дополнение. Когда начинается обучение, каждый обучающий сигнал

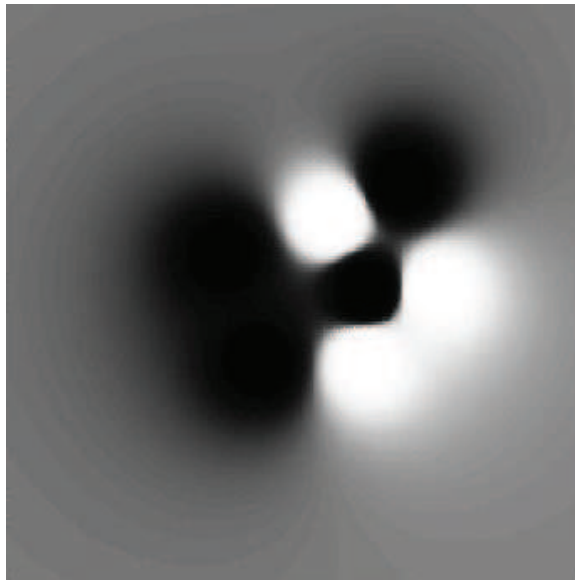


Рис. 2. Диаграмма функции влияния:
 черный цвет — отрицательные обучающие сигналы;
 белый — положительные обучающие сигналы

окрашивает окружающее его пространство в черный или белый цвет, причем интенсивность окраски окружающих точек зависит от расстояний между ними и обучающими сигналами.

Формулу (4) для функции влияния — зависимости интенсивности окраски от расстояния до обучающего сигнала — можно задавать различными способами.

В конце достаточно длительного процесса обучения почти вся искомая область X будет окрашена белым цветом, а почти вся область Y (дополнение к X) — черным цветом. Лишь точки, находящиеся вблизи границы $\Gamma(X)$ искомой области, будут закрасены неопределенным цветом.

Следует отметить, что граница $\Gamma(X)$ имеет меру 0, и соответственно область неопределенности имеет пренебрежимо малую меру, если искомая область не устроена экзотически, как, например, ковер Серпинского.

Итак, определена функция влияния, выяснен ее смысл. Осталось найти выражения для приближенных решающих функций, удовлетворяющие условиям, приведенным в подп. 4.1. Теперь решение этой задачи вызывает незначительные технические трудности.

4.3. *Построение приближенных решающих функций.* Введем несколько вспомогательных функций, затем рассмотрим определение решающих функций f_k , которое соответствует описанной функции влияния и удовлетворяет перечисленным выше требованиям.

1. $F(t) : (-\infty, +\infty) \rightarrow (0, 1)$ — функция одной переменной с характерным s -образным графиком (рис. 3), которую можно задать, например, формулой

$$F(t) = \frac{1}{2} \left(\frac{t}{\sqrt{t^2 + 1}} + 1 \right).$$

Очевидно, что

$$\lim_{t \rightarrow -\infty} F(t) = 0, \quad \lim_{t \rightarrow +\infty} F(t) = 1.$$

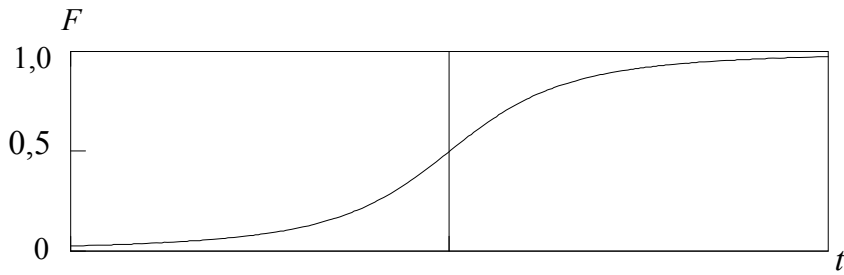


Рис. 3. График функции $F(t)$

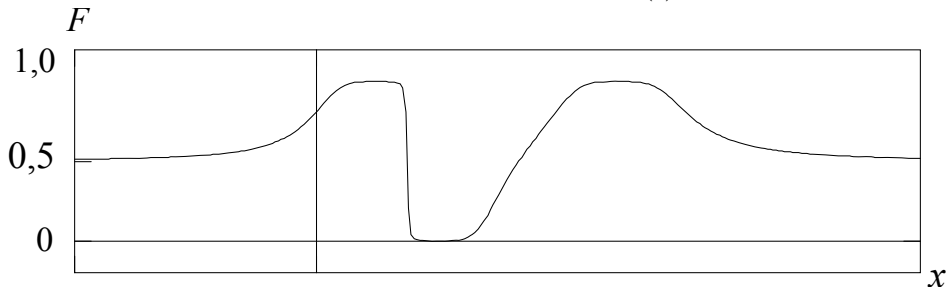


Рис. 4. График решающей функции ($n = 1$)

С помощью этой функции функцию $R^1 \rightarrow R^1$ с неограниченной областью значений можно преобразовать в функцию с требуемой областью значений $[0, 1]$.

2. Функция влияния множества $\{z^1, z^2, \dots, z^N\}$ имеет вид

$$H_N(z) = \sum_{k=1}^N h_k(z).$$

3. Определим решающие функции f_k . Пусть $f_0(z) \equiv 1/2$, а при $N > 0$

$$f_N(z) = F(H_N(z)), \quad z \neq z^k,$$

$$f_N(z^k) = \lim_{z \rightarrow z^k} F(H_N(z)) = \begin{cases} 1, & z^k \in X, \\ 0, & z^k \notin X. \end{cases}$$

Функция H_N не определена в точках $z = z^k, k = 1, \dots, N$, поэтому была доопределена функция f_N в этих точках.

На рис. 4, 5 приведены графики решающих функций при $n = 1$ и $n = 2$. По оси z отложены значения решающей функции, на горизонтальных осях — параметры, характеризующие исследуемый объект.

Осталось доказать, что для построенных решающих функций имеют место свойства, указанные в подп. 4.1.

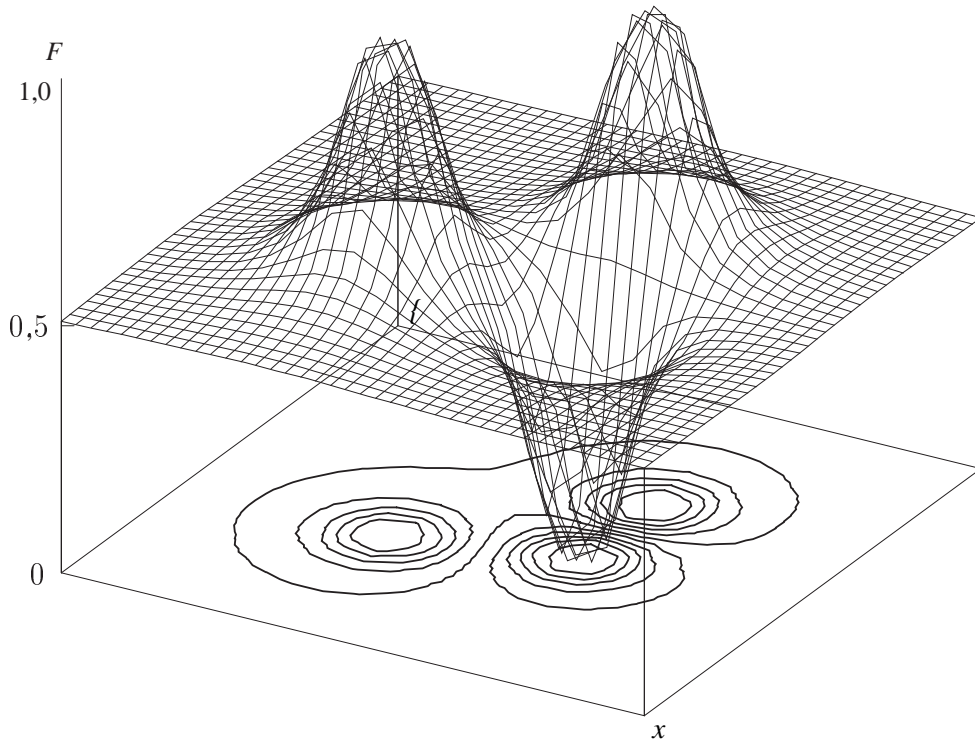
Утверждение. Решающие функции f_k удовлетворяют условиям 1–4 (п. 4).

Доказательство. Перечислим данные условия.

1. $0 \leq f_k(z) \leq 1$. Согласно определению

$$f_N(z) = \begin{cases} F(H_N(z)), & z \neq z^k, \\ 0 \text{ или } 1, & z = z^k. \end{cases}$$

Следовательно, значения функции $f_N(z)$ лежат в области $(0, 1)$ значений функции F либо равны 0 или 1.

Рис. 5. График решающей функции ($n = 2$)

2. Приближенные решающие функции f_k непрерывны. Очевидно, функция $H_N(z)$ непрерывна в своей области определения $z \neq z^1, z^2, \dots, z^N$. Поэтому $f_N(z)$ непрерывна при $z \neq z^1, z^2, \dots, z^N$ как суперпозиция непрерывных (в области определения) функций F и H . Согласно определению в особых точках z^k

$$\lim_{z \rightarrow z^k} f_N(z) = \lim_{z \rightarrow z^k} F(H_N(z)) = f_N(z^k).$$

$$3. f_k(z^i) = \begin{cases} 1, & z^i \in X, \\ 0, & z^i \notin X, \end{cases} \quad i = 1, \dots, k.$$

Значения функции f_k в особых точках $z = z^i$, $i = 1, \dots, k$ определены отдельно, и это определение совпадает с условием 3.

4. Функция f_k не зависит от порядка поступления обучающих сигналов. Значения функции $H_k(z) = \sum_{i=1}^k h_i(z)$ не зависят от порядка слагаемых, а следовательно, и от порядка поступления обучающих сигналов z^i , но полностью определяют значения $f_k(z)$ (см. определение функции f_k).

Итак, построенные приближенные решающие функции удовлетворяют условиям, перечисленным в подп. 4.1. В подп. 4.4 представлен алгоритм конкуренции, основанный на использовании этих функций.

Сделаем замечание относительно термина “решающая функция”. В математической статистике [12] решающую функцию выбирает исследователь (статистик). Аргументами этой функции являются выборочные точки некоторого пространства, а ее значением (т. е. результатом решения задачи) — вектор, близкий к искомому вектору-параметру. В рассматриваемой задаче также моделируются случайные точки-сигналы (т. е. выполняется N шагов обучения решающего алгоритма), в результате чего получается функция f_N . Таким образом, в данном случае решающий алгоритм выполняет такую же роль, как и решающая функция в

математической статистике. Аргументы алгоритма — выборочные точки-сигналы z_1, \dots, z_N , но результатом является не вектор-параметр, а функция f_N , близкая к искомой характеристической функции χ_X .

4.4. *Функционирование алгоритма конкуренции.* Напомним формальную постановку задачи: имеется неизвестное множество $X \subset \mathbb{R}^n$, требуется найти его характеристическую функцию.

Запишем последовательность действий при решении задачи, т. е. решающий алгоритм конкуренции.

1. Задаем функцию влияния $H_0(z) \equiv 0, z \in Z$.
2. Равномерно в Z ($X \subset Z \subset \mathbb{R}^n$) моделируем последовательность обучающих сигналов z^1, z^2, \dots .
3. При поступлении очередного сигнала z^k изменяем функцию влияния: $H_{k-1}(z) \rightarrow H_k(z) = H_{k-1}(z) + h_k(z)$. До появления сигнала z^k точка z уже имела некоторый знак, совпадающий со знаком числа $H_{k-1}(z)$ и полученный под влиянием сигналов z^1, z^2, \dots, z^{k-1} . Появление сигнала z^k может изменить знак точки z , сделать его таким же, как знак сигнала z^k . В этом заключается свойство конкуренции обучающих сигналов. Каждый из них “стремится” изменить знаки окружающих точек (см. также замечание в подп. 4.2).
4. Строим решающую функцию f_k .
5. Выбираем номер N шага, на котором заканчиваем обучение. Информацию, полученную при обучении, сохраняют координаты обучающих сигналов.
6. Система переходит в рабочий режим. В рабочем режиме для каждой точки $z \in Z$ вычисляем значение $f_N(z)$.

Напомним, что значение $f_N(z)$ в каждой точке $z \in Z$ представляет собой эвристическую вероятность события $\{z \in X\}$. Если $f_N(z) \leq 1/2$, то можно считать, что событие не произошло (точка z не попала в X); в противном случае считаем, что событие произошло (точка z попала в X).

5. Численный эксперимент. В данном пункте рассматривается тестовый пример для проверки эффективности работы предложенного алгоритма.

Пусть A — квадратная числовая матрица с известными коэффициентами, $\det(A)$ — ее неизвестный определитель. Основным вопросом является следующий: верно ли, что $\det(A) > 0$? Иными словами, требуется отделить матрицы $A: \det(A) < 0$ от матриц $A: \det(A) > 0$.

Далее для определенности будем рассматривать матрицы размером 3×3 . Пусть A — некоторый обучающий сигнал. Используя решающий алгоритм, получаем информацию о матрицах в ограниченной окрестности матрицы A .

Однако множество всех матриц данного размера является неограниченным в естественной метрике. Поэтому, используя конечное число обучающих сигналов, невозможно получить существенную информацию обо всех матрицах.

Запишем матрицу A в виде набора вектор-столбцов $A = (a_{.1}, a_{.2}, a_{.3})$:

$$a_{.k} = \begin{bmatrix} a_{1k} \\ a_{2k} \\ a_{3k} \end{bmatrix}.$$

Если умножить любой столбец на число, определитель также будет умножен на это число. Обозначим через $\bar{a}_{.k}$ нормированный вектор-столбец, а через \bar{A} — матрицу, составленную из таких столбцов:

$$\bar{a}_{.k} = \frac{1}{\sqrt{a_{1k}^2 + a_{2k}^2 + a_{3k}^2}} a_{.k}, \quad \bar{A} = (\bar{a}_{.1}, \bar{a}_{.2}, \bar{a}_{.3}).$$

Тогда согласно приведенному выше замечанию знак $\det(\bar{A})$ совпадает со знаком $\det(A)$. Поэтому, решая задачу, можно рассматривать не множество всех матриц $\{A\}$, а лишь подмножество $\{\bar{A}\}$, которое в отличие от $\{A\}$ является ограниченным: норма каждого столбца каждой матрицы \bar{A} равна 1, поэтому матрицы \bar{A} равномерно ограничены в естественной метрике. Осталось записать точное выражение для какой-либо естественной метрики и получить соответствующее выражение для решающей функции $f(A)$.

Пусть \bar{A}, \bar{B} — матрицы с нормированными столбцами. Определим расстояние $\rho(\bar{A}, \bar{B})$, полагая

$$\rho(\bar{A}, \bar{B}) = \sqrt{\sum_{k=1}^3 d^2(\bar{a}_{.k}, \bar{b}_{.k})}, \quad (5)$$

где $d(\bar{a}_{.k}, \bar{b}_{.k})$ — некоторое расстояние между k -ми столбцами матриц \bar{A}, \bar{B} . Этим столбцам соответствуют точки \bar{a}, \bar{b} на единичной сфере $S = \{x : |x| = 1\}$. Введем естественное расстояние между этими точками — длину большой дуги на единичной сфере с концами в точках \bar{a}, \bar{b} :

$$d(\bar{a}, \bar{b}) = C \arccos \frac{\bar{a} \cdot \bar{b}}{|\bar{a}| |\bar{b}|} = C \arccos(\bar{a} \cdot \bar{b}).$$

Итак, определено расстояние между нормированными вектор-столбцами. По формуле (5) определим расстояние между матрицами с нормированными столбцами.

Как отмечено выше, в процедуре обучения достаточно рассматривать матрицы, состоящие из трех нормированных вектор-столбцов. Числа в этом столбце являются координатами точки на двумерной сфере S_2 . Значит, вся нормированная числовая матрица является элементом 6-мерного пространства \mathbb{R}^6 . В свою очередь \mathbb{R}^6 можно представить в виде произведения $\mathbb{R}^6 = S_2 \times S_2 \times S_2$.

Для успешного обучения необходимо подать на вход большое количество обучающих матриц, чтобы обучающие сигналы плотно заполнили множество Z возможных сигналов.

Моделирование равномерно распределенных точек в многомерных областях удобно выполнять, используя формулы метода Монте-Карло. Чтобы смоделировать очередную нормированную матрицу, смоделируем три независимые точки, равномерно распределенные на единичной сфере S_2 . Моделирующие формулы для координат x_1, x_2, x_3 такой точки имеют следующий вид [13]:

$$\begin{aligned} x_1 &= 1 - 2\alpha_1, \\ x_2 &= \sqrt{1 - (2\alpha_1 - 1)^2} \cos(2\pi\alpha_2), \\ x_3 &= \sqrt{1 - (2\alpha_1 - 1)^2} \sin(2\pi\alpha_2). \end{aligned}$$

Здесь α_1, α_2 — независимые случайные числа, равномерно распределенные на отрезке $[0, 1]$. При моделировании случайных чисел α_1, α_2 использовался датчик псевдослучайных чисел, равномерно распределенных на $[0, 1]$.

Алгоритм является нечетким, поэтому для каждого сигнала A он выдает не одно из пары чисел 0 и 1, а число из промежутка $[0, 1]$. Если это число $f(A) > 1/2$, ответ на основной вопрос считаем положительным, в противном случае — отрицательным.

Как отмечено выше, обычное функционирование решающего алгоритма (ИНС, алгоритма конкуренции) включает два этапа — обучающий и рабочий. Однако целью тестового расчета является не только проверка функционирования алгоритма конкуренции, но и изучение динамики эффективности его работы. Поэтому тестовый расчет построен таким образом, чтобы совместить эти этапы, и включает одновременно и процесс обучения, и проверку работы обученной системы.

Перечислим последовательные шаги расчета. Итак, пусть в процессе обучения на вход уже были поданы $k - 1$ обучающих сигналов, т. е. матрицы A_1, \dots, A_{k-1} .

1. Подаем сигнал A_k .

2. К моменту k решающая функция f_{k-1} уже обучена сигналами A_1, \dots, A_{k-1} . Вычисляем значение $f_{k-1}(A_k)$. Используя это значение, находим ответ на основной вопрос для матрицы A_k .

3. Находим истинный ответ на основной вопрос, т. е. находим знак определителя $\det(A_k)$.

4. Если истинный ответ и ответ из п. 3 совпадают, увеличиваем количество правильных ответов m на единицу:

$$m \rightarrow m + 1.$$

5. Считаем, что A_k — обучающий сигнал, и с помощью этого сигнала с известным правильным ответом изменяем обучающую функцию:

$$f_{k-1} \rightarrow f_k.$$

6. Изменяем значение счетчика:

$$k \rightarrow k + 1$$

и переходим к п. 1.

Итак, вычислив эффективность (т. е. долю верных ответов), с помощью очередной группы рабочих сигналов используем эти сигналы для последующего обучения. Используемый алгоритм расчета позволяет сделать это (см. п. 5).

Таким образом, расчет позволяет выяснить, каким образом частота появления верных ответов зависит от количества обучающих примеров. Расчет частоты проводим следующим образом. Последовательность матриц A_1, \dots, A_{k-1} разбиваем на группы по 100 матриц:

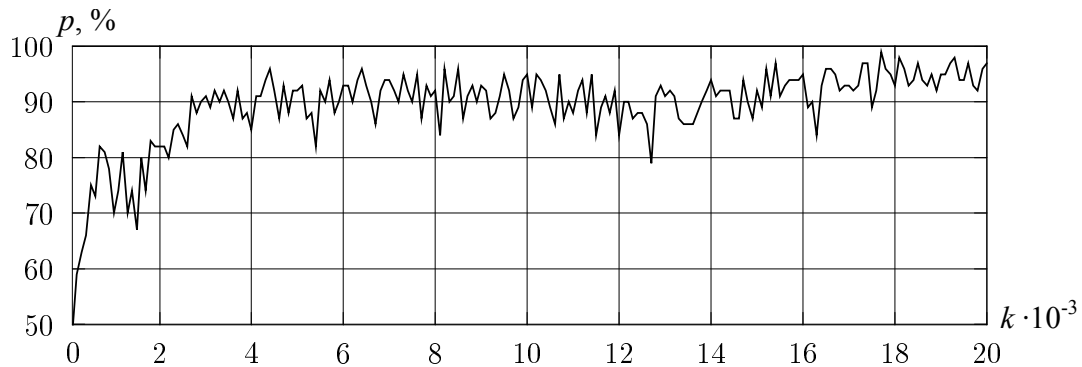
$$G_1 = \{A_1, \dots, A_{100}\}, G_2 = \{A_{101}, \dots, A_{200}\}, \dots$$

Для каждой группы G_k подсчитываем число n_k верных ответов и вычисляем долю верных ответов в этой группе (в процентах):

$$p(k) = \frac{n_k}{100} \cdot 100 \%$$

На рис. 6 показана зависимость частоты правильных ответов на основной вопрос от количества обучающих сигналов. Обучающие точки-сигналы являются случайными, следовательно, весь процесс обучения является случайным (см. замечания во введении к п. 4 и в конце подп. 4.3). Поэтому с ростом числа обучающих сигналов доля p правильных ответов изменяется немонотонно, претерпевая случайные скачки.

Из рис. 6 следует, что в начале обучения величина $p = 1/2$ (это соответствует отсутствию информации о решении) и по мере поступления обучающих сигналов быстро растет.

Рис. 6. Зависимость $p(k)$

Затем скорость роста резко уменьшается. Однако тенденция роста сохраняется, несмотря на случайные скачки. В конце процедуры обучения-расчета, когда число обучающих сигналов превышает 18 000, величина p близка к единице. Следовательно, для рассмотренного примера удалось обучить алгоритм и получить удовлетворительное решение поставленной задачи.

Вопросы, связанные с увеличением скорости сходимости предложенного алгоритма, требуют специального исследования.

Список литературы

1. БАРСКИЙ А. Б. Нейронные сети: распознавание, управление, принятие решений. М.: Финансы и статистика, 2004.
2. УОСЕРМЕН Ф. Нейрокомпьютерная техника: Теория и практика. М.: Мир, 1992.
3. PARKER D. B. Learning logic: Invent. Report / Stanford Univ. N S81-64. Stanford, 1982.
4. RUMELHART D. E., HINTON G. E., WILLIAMS R. J. Learning internal representations by error propagation // Parallel distributed processing. Cambridge: MIT Press, 1986. V. 1. P. 318–362.
5. SEJNOWSKI T. J., ROSENBERG C. R. Parallel networks that learn to pronounce english text. Complex Systems 3:145–168. 1987.
6. BURR D. J. Experiments with a connectionist text reader // Proc. of the 1st Intern. conf. on neural networks / Ed. by M. Caudill, C. Butler. San Diego: SOS Printing, 1987. V. 4. P. 717–724.
7. COTTRELL G. W., MUNRO P., ZIPSER D. Learning internal representations from gray-scale images: An example of extensional programming: Proc. of the 9th Annual cognitive sci. soc. conf., Seattle (USA), July. 16–18, 1987. Norwood: Ablex, 1987. P. 461–473.
8. РОЗЕНБЛАТТ Ф. Принципы нейродинамики. М.: Мир, 1965.
9. МИНСКИЙ М. Л. Перцептроны / М. Л. Минский, С. Пейперт. М.: Мир, 1971.
10. КОФМАН А. Введение в теорию нечетких множеств. М.: Радио и связь, 1982.
11. ОРЛОВСКИЙ С. А. Проблемы принятия решений при нечеткой исходной информации. М.: Наука, 1981.
12. БОРОВКОВ А. А. Математическая статистика. Новосибирск: Наука. Сиб. издат. фирма, 1997.
13. ЕРМАКОВ С. М. Курс статистического моделирования / С. М. Ермаков, Г. А. Михайлов. М.: Наука, 1976.

*Антюфеев Виктор Степанович — д-р физ.-мат. наук, проф.
Института вычислительной математики и математической геофизики СО РАН;
тел.: (383) 330-77-21; e-mail: ant@osmf.sccc.ru*

Дата поступления — 14.11.11 г.