

УДК 007.52

В.В. Сарапас

## ТЕСТОВАЯ ЗАДАЧА ДЛЯ ВИЗУАЛИЗАЦИИ ПРОЦЕССА ОБУЧЕНИЯ НЕЙРОННОЙ СЕТИ

*При разработке и реализации любого нейросетевого алгоритма возникает задача тестирования его работоспособности. В таких случаях используются различные тестовые задачи. При этом, как и в других областях научной деятельности, исследователю приходится использовать методы визуализации данных с целью более эффективного их анализа. Однако большинство тестовых задач не позволяют наглядно и объективно судить о ходе поиска решения в пространстве функции ошибки. В статье дан краткий обзор методов, применяемых для визуализации процесса обучения нейронной сети, а также описан подход к тестированию алгоритмов обучения нейронных сетей, основанный на использовании тестовых задач, обладающих известной функцией ошибки. В качестве иллюстрации к данному подходу предложена простая тестовая задача для нейронной сети минимального размера, которая позволяет наглядно визуализировать процесс обучения и объективно судить о его ходе.*

### Введение

Во многих областях научной деятельности исследователю приходится иметь дело с большим количеством числовых данных. Человек обладает высокоразвитыми способностями к распознаванию визуальных образов и способен получать информацию более эффективно из визуальных данных, чем путем анализа большого списка или таблицы значений.

Если данные содержат изменения только одной переменной, то для качественных показателей используют гистограммы или секторную диаграмму, а количественные показатели могут быть описаны частотной гистограммой. Одно- или двухмерные данные могут быть отображены непосредственно на графике в евклидовых координатах. При использовании этого метода получается простой и понятный график, дающий информацию о кластеризации, выбросах и общем распределении данных. Трехмерные данные могут быть легко представлены на графиках, использующих ортогональные оси. Сложнее обстоят дела с размерностями более трех, с которыми, как правило, имеют дело при работе с нейронными сетями.

В наиболее общем случае нейронная сеть, как показано на рис. 1, вычисляет функцию  $F(x^p, w)$ , где  $x^p$  –  $p$ -й входной паттерн,  $w$  – набор настраиваемых весовых коэффициентов. Функция потерь  $E^p = \|D^p - F(x^p, w)\|$  является мерой различия между желаемым выходным значением для данного входного паттерна  $D^p$  и выходным значением сети.

Суммарная ошибка  $E = \frac{1}{P} \sum_{p=1}^P E^p$  – это усредненная сумма ошибок для всего набора пар входных-выходных значений, называемого обучающей выборкой. Задача обучения нейронной сети заключается в поиске среди возможных значений весовых коэффициентов  $w$  такого набора, при котором значение суммарной ошибки минимально. В литературе по нейронным сетям пространство поиска обычно упоминается как пространство весов, а функция ошибки изображается как многомерная поверхность, расположенная над пространством весов.

Для проверки и сравнения работоспособности различных алгоритмов обучения их испытывают на тестовых задачах, которые являются примерами как реальных проблем, так и специально созданных для целей тестирования.

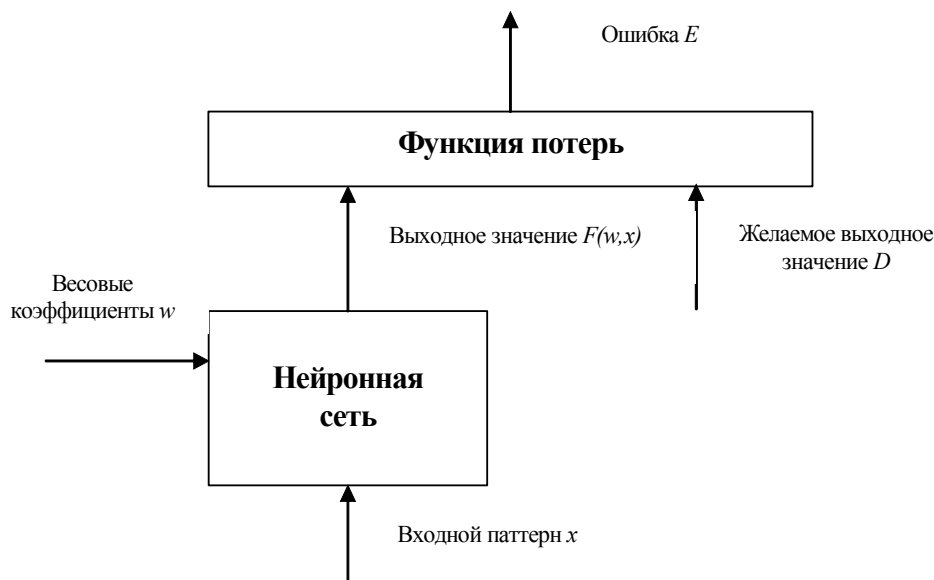


Рис. 1. Нейронная сеть

## 1. Методы визуализации многомерных данных

### 1.1. Рельефные диаграммы рассеивания

Простой способ представления трех- и четырехмерных данных заключается в отображении первых двух размерностей как координат точки на графике, а третьей – как радиуса окружности вокруг этой точки либо длин векторов, выходящих из нее [1] (рис. 2).

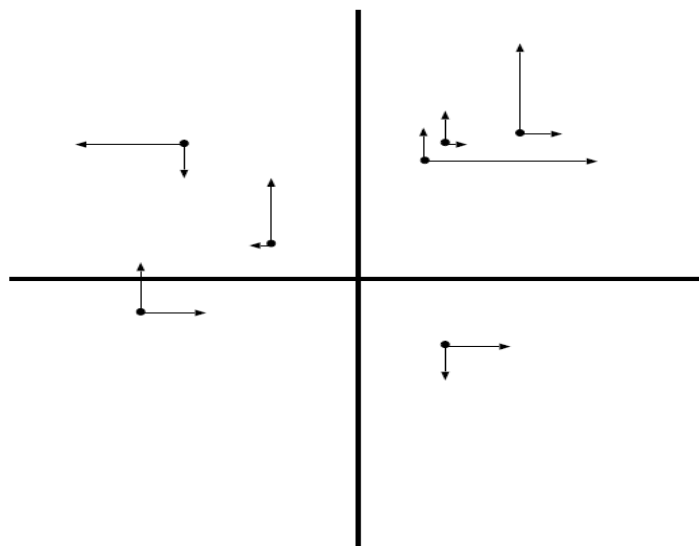


Рис. 2. Пример рельефной диаграммы рассеивания четырехмерного набора данных

Векторы, направленные на север и восток, отображают положительные значения, а на север и юг – отрицательные. Очевидно, что данный метод пригоден для размерностей вплоть до пятой или шестой из-за сложности интерпретации.

### 1.2. Кривые Эндрю и функциональное представление

Понятно, что многомерные данные могут быть представлены различными способами с целью получения полной картины внутри трехмерного пространства. Диаграммы рассеивания

трансформируют каждую величину в геометрический символ, параметры которого зависят от данных. Функциональное представление можно рассматривать как отображение многомерных данных в последовательность реальных чисел. Для каждого результата  $w$  функция определена как

$$f_w(t) = \frac{w_1}{\sqrt{2}} + w_2 \sin(t) + w_3 \cos(t) + w_4 \sin(2t) + w_5 \cos(2t) + \dots - \pi \leq t \leq \pi.$$

Функция Эндрю имеет следующее свойство: точки, находящиеся рядом в пространстве данных, образуют кривые, остающиеся рядом для всех  $t$ , а для удаленных данных кривые находятся на большем расстоянии, сохраняя при этом евклидовы расстояния, средние значения и дисперсию исходных данных.

На рис. 3, *a* и *б* показаны примеры визуализации весовых коэффициентов многослойного персептрона структуры 2-2-1 в процессе обучения [2]. Сеть обучалась методом обратного распространения ошибки в течение 1000 эпох, веса фиксировались через каждые 50 эпох. Коэффициенты обучения были равны 0,2 и 0,01 соответственно. Данные содержат 21 набор девятимерных величин. Обучение представляется изменением формы кривой от своего случайного начального вида к конечному состоянию.

Сравнение кривых Эндрю на рис. 3, *a* и *б* показывает, что сети достигли различных конечных состояний. Во втором случае обучение носило более плавный характер.

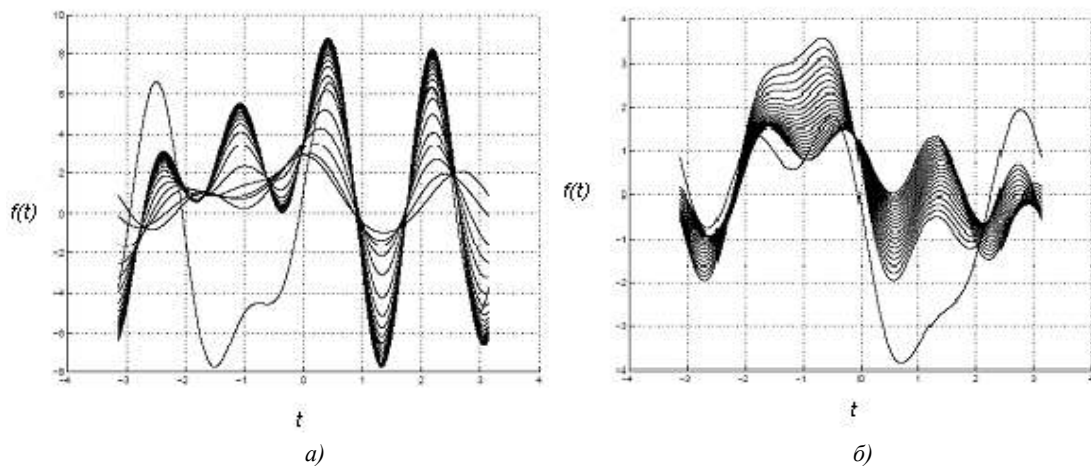


Рис. 3. Кривые Эндрю для данных, взятых с равномерным интервалом во время процесса обучения многослойного персептрона: *a*) с коэффициентом обучения 0,2; *б*) с коэффициентом обучения 0,01

Кривые Эндрю наиболее удобны для сравнительно небольших наборов данных (~50), так как график становится перегруженным в случае исследования больших массивов данных. Кроме того, этот метод приемлем для размерностей не более 20 – рост размерности увеличивает число высокочастотных составляющих в кривых и затрудняет определение свойств данных. Представление разных размерностей разными частотами также влечет трудности в интерпретации данных.

### 1.3. Сечения пространства функции ошибки

Наиболее часто применяемым способом визуализации многомерной функции ошибки является ее изображение в виде двухмерных срезов [3] (рис. 4). Так, Хаш и Хорн использовали данный способ для описания и демонстрации общих свойств поверхности ошибки [4]. Этот способ интуитивно понятен и дает пользователю много информации. Человек очень хорошо воспринимает трехмерные образы, изображенные на дисплее или листке бумаги, так как это умение основано на его жизненном опыте. Более того, он способен манипулировать подобными образами что называется «на лету», трансформируя и экстраполируя данные для невидимых

регионов. Также не представляется сложным извлечение необходимых характеристик поверхности или решение задач, связанных с их определением. Траектория алгоритма обучения легко налагается на такую поверхность и дает хорошее представление о динамике процесса и качестве решения задачи, что возможно даже в реальном времени.

Однако существует и ряд серьезных проблем при использовании двухмерных срезов. Самая главная – это катастрофическая потеря информации о непоказанных размерностях, что может ввести в заблуждение относительно реального вида многомерной поверхности функции ошибки [5]. Живя в трехмерном пространстве, люди экстраполируют ложным образом полученные срезы на остальные размерности. Кроме того, вопрос корректного выбора направления среза зависит от конкретной задачи и является нетривиальной задачей сам по себе. Таким образом, данный метод дает объективную визуализацию только для нейронных сетей с двумя весовыми коэффициентами.

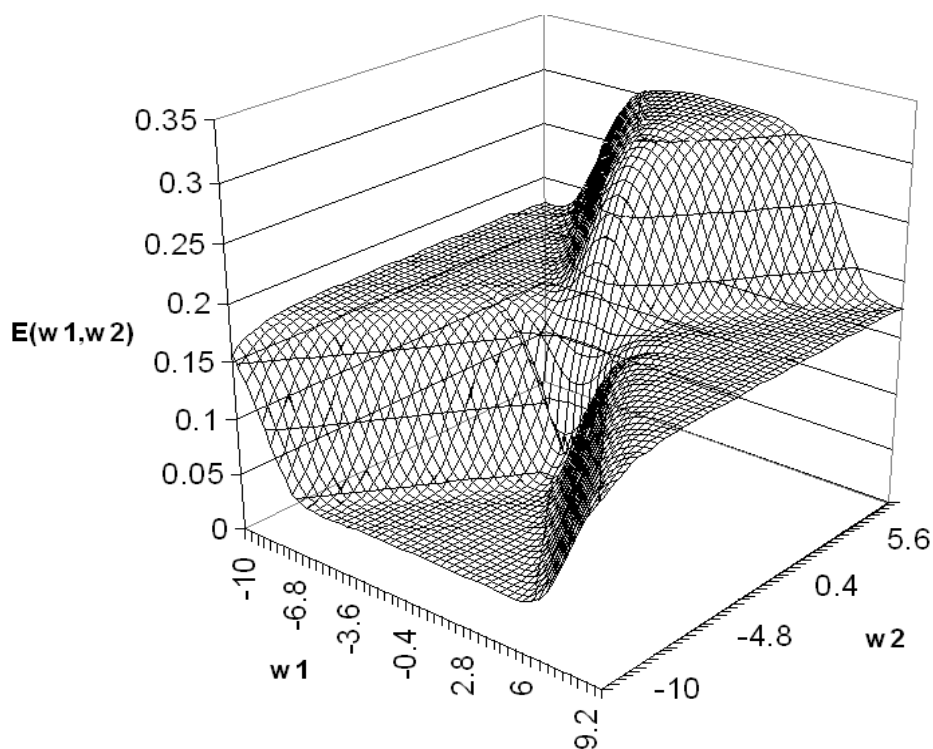


Рис. 4. Двухмерный срез функции ошибки

Некоторые исследователи предлагали модификации данного метода. Например, Андроулакис [6] использовал собственные векторы матрицы Хессеана, определенные в особых точках поверхности функции ошибки (например, вблизи локальных минимумов), для получения двухмерного среза. Он предлагал данную модификацию как эффективный инструмент сравнительности различных алгоритмов и степени их соответствия конкретной функции ошибки. Однако применение этого метода ограничено необходимостью предварительного определения особых точек поверхности ошибки и не гарантирует отсутствие потери важной информации.

#### **1.4. Диаграммы траекторий и метод главных компонент**

Вежчерт и Тесауро предложили использовать многомерные координатные системы в проекциях на двухмерную плоскость [7]. Это делает оси неортогональными и ведет к неоднозначности в изображении данных. Тем не менее, такой метод позволяет показать процесс обучения и траекторию движения в пространстве весовых коэффициентов, что дает возможность лучше понять природу функции ошибки в области траектории изменения весов. Несмотря на удобство, применение метода отображения траекторий невозможно для пространств больших размерностей, что характерно для используемых на практике нейронных сетей.

Дальнейшим развитием метода отображения траекторий стало использование анализа главных компонент (principal component analysis) для получения координат при построении траектории [8] (рис. 5). Метод анализа главных компонент широко применяется в статистике и служит эффективным инструментом уменьшения размерности данных при минимально возможных потерях информации.

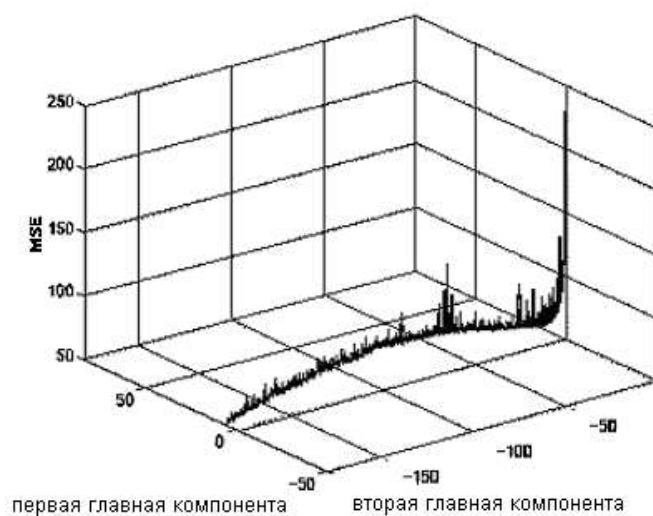


Рис. 5. Пример визуализации процесса обучения по двум главным компонентам

Данная методика показала себя весьма удобной и эффективной с точки зрения вычислительных затрат для большинства реальных приложений, но она не гарантирует того, что отброшенная при уменьшении размерности информация не является принципиально важной. Кроме того, ее использование ограничено визуализацией по двум главным компонентам, что не всегда достаточно.

### 1.5. Гиперплоскости и другие методы

Некоторые исследователи пытаются использовать другие пространства, нежели пространство весов. Наиболее широко применяется концепция гиперплоскости, т. е. перенос возможных решений, которые может получить сеть, во входное пространство данных [9]. На рис. 6 две гиперплоскости изображены в виде линейных проекций. При изучении нейронных сетей также применяются диаграммы Хилтона [10] и бантовые диаграммы [7]. Каждый вес на них представлен как коробка, размер которой зависит от амплитуды, а цвет – от направления изменения величины, либо как «бант», который может быть различного размера и цвета.

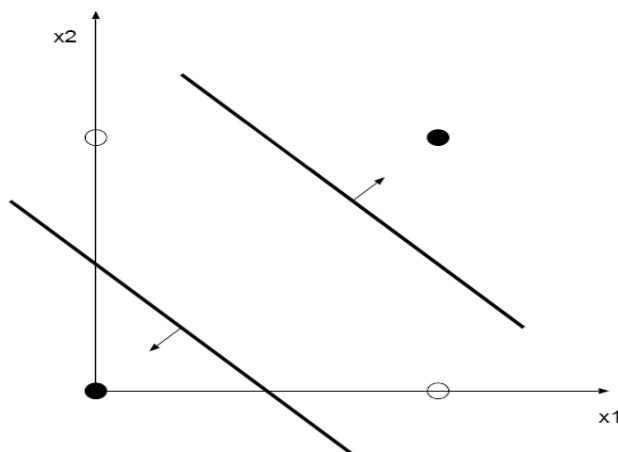


Рис. 6. Решение проблемы «исключающее ИЛИ» при помощи двух гиперплоскостей

## 2. Тестовые задачи и проблема «исключающее ИЛИ»

Во время реализации любого нейросетевого устройства возникает задача тестирования его работоспособности. В таких случаях используются различные тестовые задачи. В настоящий момент накоплено большое количество задач искусственного и естественного (взятого из реальной практики) происхождения, относящихся к различным областям. В работе [11–16] даны ссылки на архивы таких данных. На каких же проблемах тестируются нейронные сети? В первую очередь это реальные задачи, такие как распознавание речи, распознавание рукописных символов, диагностика заболеваний, прогнозирование различных временных рядов и т. д. К искусственным задачам относят сегментацию данных, порожденных каким-либо аналитически заданным процессом; распознавание искусственно синтезированных образов; аппроксимацию тригонометрических функций; воспроизведение функций булевой логики и т. д.

Одной из самых популярных контрольных задач является так называемое «исключающее ИЛИ» (XOR). Набор данных этой задачи содержит четыре паттерна (табл. 1)

Таблица 1

XOR-функция

Паттерн	X1	X2	Выходное значение	Нормализованное выходное значение
P <sub>00</sub>	0	0	0	0,1
P <sub>01</sub>	0	1	1	0,9
P <sub>10</sub>	1	0	1	0,9
P <sub>11</sub>	1	1	0	0,1

Несмотря на простоту самой задачи, функция ошибки не очевидна. Множество авторов посвятили свои работы ее исследованию. Так, Лисбоа и Перантонис [17] в 1991 г. выявили все критические точки и путем численного моделирования доказали, что часть из них – локальные минимумы, но в 1998 г. Хамни [18], Шпринкхузен-Купер и Боерс [19] показали, что найденные предыдущими авторами локальные минимумы соответствуют бесконечным весовым коэффициентам сети, а для конечных весов существует только глобальный минимум. К подобным выводам были близки и другие исследователи. Кроме того, Хамни [18] показал, что визуализация функции ошибки посредством срезов может привести к ошибочным выводам. Поэтому полной ясности в этом вопросе нет до сих пор.

К недостаткам используемых в настоящее время тестовых задач следует отнести:

- высокую сложность, что делает их применение малорезультативным на ранних этапах, когда необходимо убедиться в работоспособности системы и правильности реализации алгоритмов обучения;

- неопределенность функции ошибки – для этих задач не существует строго доказанного местоположения локальных и глобального минимумов и описания рельефа поверхности функции ошибки;

- большую размерность функции ошибки, не позволяющую наглядно визуализировать процесс обучения;

- тестирование алгоритмов, заключающееся, как правило, в многократном их пуске с разными начальными условиями и последующим усреднением результатов, что не позволяет судить о причинах того, почему один алгоритм оказался быстрее другого или в чем причина неудачи решения поставленной задачи.

Устранить перечисленные выше недостатки можно путем тщательного исследования поверхности функции ошибки, но упомянутый ранее случай с относительно простой задачей воспроизведения функции «исключающее ИЛИ» показал, что сделать это не так-то просто. Более перспективен путь целенаправленного создания набора тестовых задач, содержащих специальные участки, например локальные минимумы и плато (области, в которых градиент очень мал), помогающие выявить проблемные места процесса обучения, а также сравнить и исследовать поведение различных алгоритмов в четко определенных условиях. Далее в качестве иллюстрации указанного подхода предлагается простая тестовая задача.

### 3. Простая тестовая задача

Исходя из возможностей современных методов визуализации, следует остановиться на трехмерном графике функции ошибки как наиболее информативном. Во избежание ложных толкований следует ограничить число весовых коэффициентов сети двумя. Таким образом, получаем сеть, доступную для исследования (рис. 7). Обычно на один из входов подается постоянный уровень – смещение. В данном случае это вход  $X_2$ .

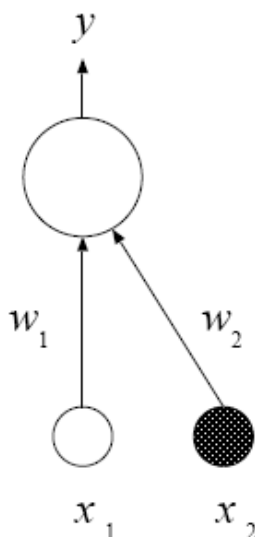


Рис. 7. Простая нейронная сеть

В качестве функции ошибки примем наиболее часто используемую среднеквадратичную функцию потерь

$$E^p = \frac{1}{2} (D^p - F(x^p, w))^2,$$

где  $E^p$  – ошибка для  $p$ -го паттерна;  $D^p$  – желаемое значение выходов сети;  $F(x^p, w)$  – выходное значение сети,  $w$  – весовые коэффициенты сети,  $x$  – значение входов сети.

В качестве функции активации наиболее часто применяются:

- сигмоидальная функция  $F(z) = 1/(1+e^{-z})$ ;
- симметричная сигмоидальная функция  $F(z) = 2/(1+e^{-z}) - 1$ ;
- функция гиперболического тангенса  $F(z) = \tanh(z)$ .

Для того чтобы наблюдать поведение алгоритма обучения в зависимости от текущего паттерна обучения, придется еще больше упростить сеть – пожертвовать смещением. Полученная сеть состоит из единственного нейрона с одним входом и выходом и имеет один весовой коэффициент. Выходное значение такой сети –  $y = F(x, w)$ . Данная сеть имеет одномерную функцию ошибки. Согласно [20] такая сеть может иметь  $[n/d]^d$  локальных минимумов, где  $n$  – количество паттернов в обучающей выборке, а  $d$  – число входов сети. Другими необходимыми условиями являются нереализуемость задачи, т. е. отсутствие значений весовых коэффициентов, при котором значение суммарной ошибки равно нулю, и ограниченность выходного значения функции активации, используемой совместно с квадратичной функцией потерь.

В табл. 2 приведены предлагаемые автором тестовые функции для сигмоидальной функции активации нейрона, а табл. 3 – значения для симметричной сигмоидальной и тангенциальной функций активации.

Таблица 2

Тестовая задача для сигмоидальной функции

Паттерн	X	Нормализованное выходное значение
P <sub>00</sub>	1	0,9
P <sub>01</sub>	8	0,1
P <sub>10</sub>	100	0,475

Таблица 3

Тестовая задача для симметричной сигмоидальной и тангенциальной функций

Паттерн	X	Нормализованное выходное значение
P <sub>00</sub>	1	- 0,9
P <sub>01</sub>	8	0,9
P <sub>10</sub>	100	-0,025

При необходимости получить более пологие склоны функции ошибки следует уменьшить значения X из-за большей крутизны первой производной тангенциальной функции.

Функция ошибки для каждого паттерна и суммарная ошибка показаны на рис. 8 и 9 соответственно. Данная функция ошибки содержит глобальный минимум с координатами (-0,00213; 0,05297), локальный минимум (-0,18145; 0,07199) и плато начиная от точки (2,1972; 0,18094). Небольшой холм (0,59888; 0,18928) может стать серьезным препятствием при выходе с плато в случае движения в сторону уменьшения веса.

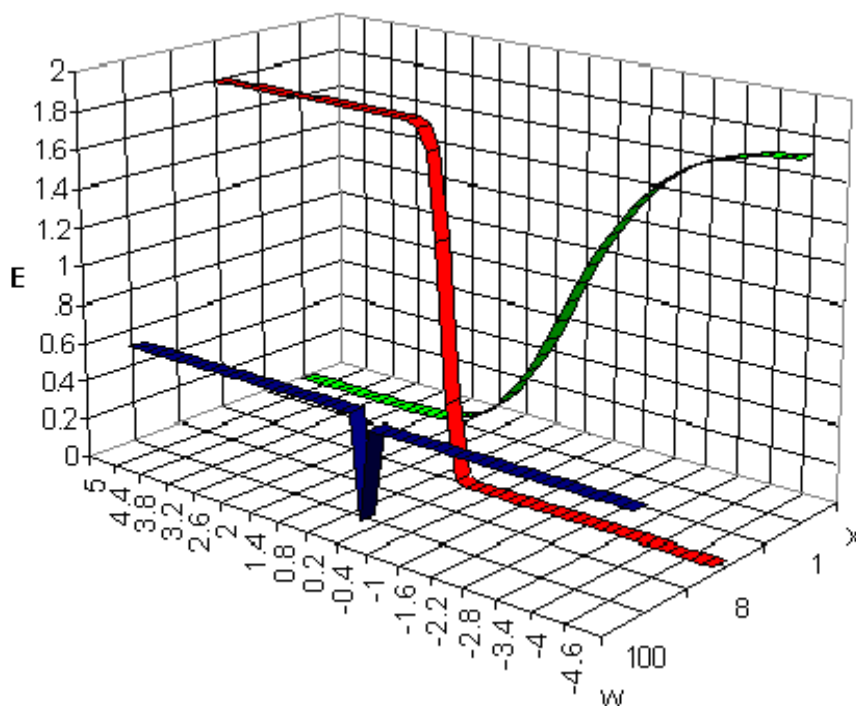


Рис. 8. Функции ошибки отдельных паттернов



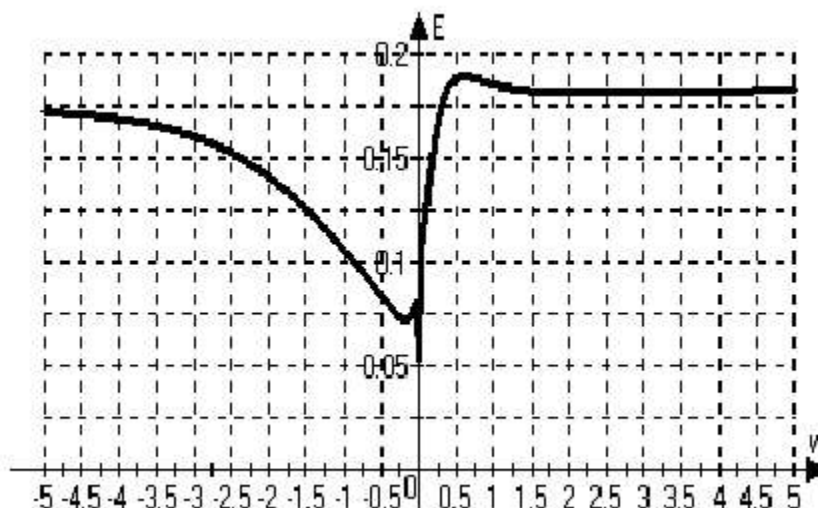


Рис. 9. Суммарная функция ошибки

Таким образом, кроме проверки работоспособности сети, на данной задаче возможна проверка алгоритмов обучения, исключающих попадание в локальные минимумы, а также поведения алгоритмов в зоне плато. Благодаря тому, что «расщелина», ведущая к глобальному минимуму, достаточно узкая, возможна оценка правильности выбора параметров обучения, например коэффициента скорости обучения.

### Заключение

В данной статье дан краткий обзор методов графического представления параметров нейронной сети и поверхности функции ошибки и указаны основные недостатки, присущие наиболее широко используемым тестовым задачам, применяемым при работе с нейронными сетями. В качестве возможной альтернативы предложено использовать специально созданные тестовые задачи с известной поверхностью функции ошибки. Для иллюстрации указанного подхода предложена простая тестовая задача, которая может быть использована для проверки работоспособности нейронной сети на начальном этапе ее создания и как средство для визуализации процесса обучения сети.

### Список литературы

1. Everitt B.S. Graphical techniques for multivariate data. – London: Heinemann educational books, 1978.
2. Manly B.F.J. Multivariate statistical methods: A primer. – London: Chapman and Hall, 1994.
3. Phillips S. The effect of representation on error surface // Fourth Australian conference on neural networks (ACNN'93). – Australia: University of Sydney, 1993. – P. 86–89.
4. Hush D.R., Horne B., Salas J. M. Error surfaces for multi-layer perceptrons // IEEE Transactions on systems, man, and cybernetics. – V. 22. – № 5. – 1992. – P. 1152–1161.
5. Hamey L.G.C. The structure of neural network error surfaces // Proc. Sixth Australian conference on neural networks. – University of Sydney, 1995. – P.197–200.
6. Androulakis G.S., Magoulas G.D., Vrahatis M. N. Geometry of learning: Visualizing the performance of neural network supervised training methods // Nonlinear analysis, theory, methods & applications. – V. 30. – № 7. – 1997. – P. 4359–4544.
7. Jolliffe T. Principal component analysis. – New-York: Springer-Verlag, 1986.
8. Wejchert J. and Tesauro G. Neural network visualization // Advances in neural information processing systems. – V. 2. – 1990. – P. 465–472.
9. Nilsson N.J. Learning machines. – New-York: McGraw-Hill, 1965.

10. Hinton G.E., McClelland J.L. and Rumelhart D.E. Distributed representations // Parallel distributed processing. – V.1. – Chapter 3. – 1986. – P. 77–109.
11. UCI Machine learning repository // <http://www.ics.uci.edu/~mlearn/MLRepository.html>
12. UCI Knowledge discovery in databases // <http://kdd.ics.uci.edu/>
13. Delve: Data for Evaluating Learning in Valid Experiments // <http://www.cs.toronto.edu/~delve/>
14. Bilkent university function approximation repository // <http://funapp.cs.bilkent.edu.tr/funapp/>
15. CEDAR: Database of handwritten cities, states, ZIP codes, digits, and alphabetic characters // <http://www.cedar.buffalo.edu/Databases/>
16. Otago speech corpus // [http://divcom.otago.ac.nz/infosci/kel/software/RICBIS/hyspeech\\_main.html](http://divcom.otago.ac.nz/infosci/kel/software/RICBIS/hyspeech_main.html)
17. Lisboa P.J.G., Perantonis S.J. Complete solution of the local minima in the XOR problem // Network. – V. 2. –1991. – P. 119–124.
18. Hamney L.G.C. XOR has no local minima: a case study in neural network error surface analysis // Neural Networks. – V. 11 – № 4. – 1998. – P. 669–682.
19. Sprinkhuizen-Kuyper I.G., Boers E.J.W. The error surface of the 2-2-1 XOR network: the finite stationary points // Neural Networks. – V. 11 – № 4. – 1998. – P. 683–690.
20. Auer P., Herbster M. and Warmuth M.K. Exponentially many local minima for single neurons // Technical Report UCSC-CRL-96-1. – Univ. of Calif. computer research lab., 1996.

Поступила 29.01.05

*Объединенный институт проблем  
информатики НАН Беларуси,  
Минск, Сурганова, 6  
e-mail: valera\_by@yahoo.com*

**V.V. Sarapas**

## **A TEST PROBLEM FOR NEURAL NETWORK TRAINING PROCESS VISUALIZATION**

Verification is very important in the development of neural network based devices. There is a great number of training problems on this matter. Visualization has proven to be a powerful tool for research data analysis in many fields of science. The most classical test problem doesn't allow obvious visualization of the weight trajectory on the error surface during the training process. This article gives a short overview of visualization techniques used in the domain of neural networks and offers a simple training problem for neural network of minimal size that can be used for training process visualization.