Т.к. $\dfrac{\partial E_s}{\partial z} = \dfrac{\partial\left(\dfrac{1}{L}\sum\limits_{k=1}^{L} E_s^{(k)}\right)}{\partial z} = \dfrac{1}{L}\sum\limits_{k=1}^{L}\dfrac{\partial E_s^{(k)}}{\partial z}$, то формулы настройки синаптических связей $w_{j_{n-1}j_n}^{(n)}(t+1) = w_{j_{n-1}j_n}^{(n)}(t) - \alpha^{(n)}\dfrac{\partial E_s}{\partial w_{j_{n-1}j_n}^{(n)}}$, $j_{n-1} = \overline{1, m_{n-1}}$, $j_n = \overline{1, m_n}$, принимают вид:

$$w_{j_{n-1}j_n}^{(n)}(t+1) = w_{j_{n-1}j_n}^{(n)}(t) - \alpha^{(n)} \cdot \frac{1}{L} \cdot \sum_{k=1}^{L} C^{(n)} \cdot M_{j_n j_{n-1}}^{(n)} \cdot Y^{(n-1),k}.$$

Аналогичным образом получаются формулы для настройки порогов нейронной сети.

Теорема доказана.

Данная теорема позволяет привести к алгоритму изменение весов и порогов сети в процессе обучения градиентным методом сведением их модификации к матричным операциям. Предложенный алгоритм обучения был реализован в среде MatLab 6.1.

Литература. 1. Осовский С. Нейронные сети для обработки информации / Пер. с польского Н.Д. Рудинского. – М.: Финансы и статистика, 2002. – 344 с.

АЛГОРИТМ ОБУЧЕНИЯ НЕЙРОННЫХ СЕТЕЙ С ИСПОЛЬЗОВАНИЕМ МЕТОДА СОПРЯЖЕННЫХ ГРАДИЕНТОВ

Кокош Н.В., БГУ, г.Минск

1. Введение

Рассмотрим нейронную сеть, состоящую из n нейронных элементов распределительного слоя и m - выходного слоя (рис. 1).

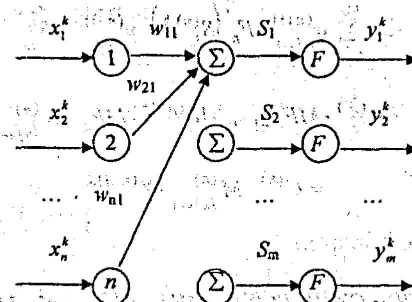

Рис. 1. Схема функционирования нейронной сети

Для данной сети каждый нейрон распределительного слоя имеет синаптические связи w_{ij}, $\left(i=\overline{1,n},\ j=\overline{1,m}\right)$ со всеми нейронами обрабатывающего слоя. В качестве нейронов выходного слоя используются элементы с некоторой функцией активации F [1, 2]. На вход сети подаются входные образы – векторы $\overline{x^k}=\left(x_1^k,\dots,x_n^k\right)$ $\left(k=\overline{1,L}\right)$.

Выходное значение j-ого нейрона сети для k-ого образа определяется выражением:

$$y_j^k=F\left(S_j^k\right),\text{ где } S_j^k=\sum_{i=1}^{n}w_{ij}x_i^k-T_j,\quad j=\overline{1,m},\quad k=\overline{1,L}. \qquad (1)$$

Задача обучения нейронной сети с фиксированной функцией активации F состоит в нахождении весовых коэффициентов w_{ij} $\left(i=\overline{1,n},\ j=\overline{1,m}\right)$ и порогов нейронных элементов T_j $\left(j=\overline{1,m}\right)$, которые минимизируют некоторую ошибку сети E_S, как отклонение выходных значений y_j^k от эталонных значений t_j^k – j-ого нейрона сети для k-ого образа. В качестве ошибки сети можно рассмотреть "квадратичное отклонение" $E_S=\frac{1}{2}\sum_{k=1}^{L}\sum_{j=1}^{m}\left(y_j^k-t_j^k\right)^2$, которое будем называть квадратичной ошибкой сети.

Столбец $\overline{W}=\left(w_{11},w_{21},\dots,w_{n1},T_1,w_{12},w_{22},\dots,w_{n2},T_2,\dots,w_{1m},w_{2m},\dots,w_{nm},T_m\right)^T$ будем называть приближенным решением или просто решением системы (по методу наименьших квадратов): $F\left(\sum_{i=1}^{n}w_{ij}x_i^k-T_j\right)=t_j^k$, $j=\overline{1,m}$, $k=\overline{1,L}$, если "квадратичное отклонение" $E_S=\frac{1}{2}\sum_{k=1}^{L}\sum_{j=1}^{m}\left(F\left(\sum_{i=1}^{n}w_{ij}x_i^k-T_j\right)-t_j^k\right)^2$ достигает своего наименьшего значения. Для нахождения такого решения можно применять различные градиентные методы [1, 2], например, метод сопряженных градиентов и его модификации, которые будут рассмотрены ниже.

2. Выражения для нахождения параметров шага обучения нейронной сети с использованием метода сопряженных градиентов

Выражения для параметров шага обучения нейронной сети с использованием метода сопряженных градиентов:

$$\overline{W}(t+1)=\overline{W}(t)-\alpha(t)\cdot\nabla E_S(t)+\beta(t)\cdot\left(\overline{W}(t)-\overline{W}(t-1)\right) \tag{2}$$

после подачи на вход сети нескольких образов $\overline{x^k}=\left(x_1^k,\dots,x_n^k\right)$ $\left(k=\overline{1,L}\right)$ определяются следующим утверждением.

Утверждение. Величины параметров $\alpha(t)$ и $\beta(t)$ шага обучения нейронной сети с использованием метода сопряженных градиентов в момент времени t определяется соотношениями [3]:

$$\alpha(t)=\frac{\|\nabla E_S(t)\|^2\cdot\left(\nabla^2 E_S(t)\cdot\overline{\Delta W}(t),\ \overline{\Delta W}(t)\right)-\left(\nabla E_S(t),\ \overline{\Delta W}(t)\right)\cdot\left(\nabla^2 E_S(t)\cdot\nabla E_S(t),\ \overline{\Delta W}(t)\right)}{\left(\nabla^2 E_S(t)\cdot\nabla E_S(t),\ \nabla E_S(t)\right)\cdot\left(\nabla^2 E_S(t)\cdot\overline{\Delta W}(t),\ \overline{\Delta W}(t)\right)-\left(\nabla^2 E_S(t)\cdot\nabla E_S(t),\ \overline{\Delta W}(t)\right)^2}, \tag{3}$$

$$\beta(t)=\frac{\|\nabla E_S(t)\|^2\cdot\left(\nabla^2 E_S(t)\cdot\nabla E_S(t),\ \overline{\Delta W}(t)\right)-\left(\nabla E_S(t),\ \overline{\Delta W}(t)\right)\cdot\left(\nabla^2 E_S(t)\cdot\nabla E_S(t),\ \nabla E_S(t)\right)}{\left(\nabla^2 E_S(t)\cdot\nabla E_S(t),\ \nabla E_S(t)\right)\cdot\left(\nabla^2 E_S(t)\cdot\overline{\Delta W}(t),\ \overline{\Delta W}(t)\right)-\left(\nabla^2 E_S(t)\cdot\nabla E_S(t),\ \overline{\Delta W}(t)\right)^2}, \tag{4}$$

где функция квадратичной ошибки сети

$$E_S(t)=E_S\left(w_{11},w_{21},\dots,w_{n1},T_1,w_{12},w_{22},\dots,w_{n2},T_2,\dots,w_{1m},w_{2m},\dots,w_{nm},T_m\right)$$

– дважды непрерывно дифференцируемая функция нескольких переменных, $\overline{W}(t)=\left(w_{11},w_{21},\dots,w_{n1},T_1,w_{12},w_{22},\dots,w_{n2},T_2,\dots,w_{1m},w_{2m},\dots,w_{nm},T_m\right)^T$ – вектор переменных,

$$\overline{\Delta W}(t)=\overline{W}(t)-\overline{W}(t-1),$$

$$\nabla E_S(t)=\left(\frac{\partial E_S}{\partial w_{11}},\frac{\partial E_S}{\partial w_{21}},\dots,\frac{\partial E_S}{\partial w_{n1}},\frac{\partial E_S}{\partial T_1},\frac{\partial E_S}{\partial w_{12}},\frac{\partial E_S}{\partial w_{22}},\dots,\frac{\partial E_S}{\partial w_{n2}},\frac{\partial E_S}{\partial T_2},\dots,\frac{\partial E_S}{\partial w_{1m}},\frac{\partial E_S}{\partial w_{2m}},\dots,\frac{\partial E_S}{\partial w_{nm}},\frac{\partial E_S}{\partial T_m}\right)^T$$

– вектор градиента функции $E_S(t)$, $\nabla^2 E_S(t)$ – матрица Гессе вторых производных функции $E_S(t)$ в момент времени t.

Так как

$$\frac{\partial E_S}{\partial w_{ij}(t)}=\sum_p\left(y_j^p-t_j^p\right)\cdot F'\left(S_j^p\right)x_i^p,\ \left(i=\overline{1,n},\ j=\overline{1,m}\right),\ \frac{\partial E_S}{\partial T_j(t)}=-\sum_p\left(y_j^p-t_j^p\right)F'\left(S_j^p\right),\ \left(j=\overline{1,m}\right),$$

то, подставляя эти соотношения в (2), получим, что модификация синаптических связей с использованием квазиоптимальных параметров шага обучения определяется выражениями:

$$w_{ij}(t+1)=w_{ij}(t)-\alpha(t)\sum_{k=1}^{L}\left(y_j^k(t)-t_j^k\right)\cdot F'\left(S_j^k(t)\right)x_i^k+\beta(t)\left(w_{ij}(t)-w_{ij}(t-1)\right),\quad i=\overline{1,n},\quad j=\overline{1,m}, \quad (5)$$

$$T_j(t+1)=T_j(t)+\alpha(t)\sum_{k=1}^{L}\left(y_j^k(t)-t_j^k\right)F'\left(S_j^k(t)\right)+\beta(t)\left(T_j(t)-T_j(t-1)\right),\quad j=\overline{1,m}. \quad (6)$$

3. Алгоритм обучения нейронной сети

Приведем алгоритм обучения нейронной сети с использованием метода сопряженных градиентов, использующий соотношения (3)-(6):

1. Задается минимальная квадратичная ошибка сети ε_m, которой необходимо достичь в процессе обучения.

2. Записывается число t=0 в счетчик числа итераций алгоритма.

3. Случайным образом инициализируются весовые коэффициенты сети $w_{ij}(t)$ $\left(i=\overline{1,n},\ j=\overline{1,m}\right)$, и пороговые значения нейронных элементов $T_j(t)$ $\left(j=\overline{1,m}\right)$.

4. Подаются входные образы $\overline{x^k}=\left(x_1^k,\ldots,x_n^k\right)$ $\left(k=\overline{1,L}\right)$ на нейронную сеть и вычисляются векторы $\overline{y^k(t)}=\left(y_1^k(t),\ldots,y_m^k(t)\right)$ $\left(k=\overline{1,L}\right)$ выходной активности сети, определяемые соотношениями (1).

5. Если $t\neq0$, то величины квазиоптимальных параметров $\alpha(t)$, $\beta(t)$ шага обучения с использованием метода сопряженных градиентов вычисляются в соответствии с соотношениями (3)-(4), в противном случае параметр $\alpha(t)$ определяется выражением (1) [2], а $\beta(t)$ полагается равным нулю.

6. Производится изменение весовых коэффициентов $w_{ij}(t+1)$ $\left(i=\overline{1,n},\ j=\overline{1,m}\right)$ и порогов нейронной сети $T_j(t+1)$ $\left(j=\overline{1,m}\right)$ согласно выражениям (5) и (6), соответственно.

7. Полагается t=t+1.

8. Алгоритм завершает свою работу, если суммарная квадратичная ошибка сети $E_S(t)=\frac{1}{2}\sum_{j=1}^{m}\sum_{k=1}^{L}\left(y_j^k(t)-t_j^k\right)^2$ или норма вектора $\overline{\Delta W}(t)=\overline{W}(t)-\overline{W}(t-1)$ не превосходят заданной величины ε_m, т. е. $E_S(t)\le\varepsilon_m$ или

$$\left\|\overline{\Delta W}(t)\right\|=\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m}\left(w_{ij}(t)-w_{ij}(t-1)\right)^2+\sum_{j=1}^{m}\left(T_j(t)-T_j(t-1)\right)^2}<\varepsilon_m,$$

в противном случае выполняется п. 4.

Литература. **1.** Головко В.А. Нейронные сети: обучение, организация и применение. Кн. 4: Учебное пособие для вузов / Общая ред. А.И. Галушкина. – М.: ИПРЖР, 2001. – 256 с.: ил. (Нейрокомпьютеры и их применение). **2.** Гладкий И.И., Головко В.А., Махнист Л.П. Обучение нейронных сетей с использованием метода наискорейшего спуска // Вестник Брестского государственного технического университета. Физика, математика, химия. – Брест: БГТУ, 2001. – 5(11):– С. 47-55. **3.** Кокош Н.В., Махнист Л.П. О решении одной системы разностно-дифференциальных уравнений и ее применении // Тезисы докладов международной математической конференции "Еругинские чтения – VIII". – Брест: БрГУ, 2002.– С. 88-89.

NEURAL NETWORK SIMULATION OF DETERMINISTIC AND STOCHASTIC PROCESSES

Nazarov P.V, Popleteev A.M., Belarusian State University, Minsk

1. Introduction

The methods of computer simulation have been proved as very powerful tools for the exploration of different complex processes [1]. They gain a considerable attention in recent years, when being used for adequate forecasting of the behaviour of explored systems under different external or internal conditions. Classical approximation methods are generally used for the analysis of well-known analytical expressions, which are far too simple to describe the real physical processes. For the correct interpretation of the experimental data computer simulation must be included in the process of data analysis. One of the forms of such application is a simulation-based fitting (SBF) [1]. The idea of SBF is the approximation of experimental data by synthetic data obtained *via* simulation modeling. In comparison to standard analytical data fitting techniques, SBF has the advantage that it fits natural physical parameters of the system itself and gives a direct insight in how they affect the experimental characteristics of the system.

However, in some cases it is not necessary to operate with a simulation model (or a "white box" model), which gives precise results but is far more computationally expensive than analytical approximation. For example, in SBF only parameters of the model are modified, when its structure holds constant. In such a case, it may be useful to perform a "black box" modeling, which still operates with real physical parameters but can be performed much faster. In the current work, it is proposed to use artificial neural networks (ANNs) [2] as "black box" simulators of physical processes.