УДК 681.322

АЛГОРИТМ ОБРАТНОГО РАСПРОСТРАНЕНИЯ ОШИБКИ ДЛЯ НЕЙРОННЫХ СЕТЕЙ С НЕЛИНЕЙНОСТЬЮ В СИНАПТИЧЕСКИХ СВЯЗЯХ, ЗАДАННОЙ В ВИДЕ КВАДРАТИЧНОЙ ФОРМЫ

Р.А. Воронкин, Н.П. Микула

Рассматривается алгоритм обратного распространения для нейронных сетей с нелинейностью в синаптических связях, заданной в виде квадратичной формы. Введение подобного рода нелинейности повышает вычислительную мощь сети, что особенно важно при решении задач критичных к числу используемых нейронов.

In this article is considered back propagation training algorithm for neural networks with non-linear functions in synaptic relationships given in the manner of square-law forms. Entering a similar sort non-linear function raises a computing power network that particularly it is important when deciding the problems critical to the number of used neurons.

В общем случае для нахождения взвешенной суммы входов нейрона не обязательно должна использоваться линейная функция вида:

$$s=\sum_{i=1}^{n} x_i \cdot w_i \ , \tag{1}$$

в некоторых приложениях нейронных сетей (НС) полезно ввести другую запись[1], например

$$s=\sum_{i=1}^{n} x_i^2 \cdot w_i \tag{2}$$

или даже

$$s=\sum_{i=1}^{n} x_i \cdot x_{((i+1)\bmod n)} \cdot w_i \ . \tag{3}$$

Рассмотрим более общий случай нелинейности, заданной в виде квадратичной формы, в этом случае взвешенная сумма входов нейрона дается формулой:

$$s=\sum_{i=1}^{n-1}\sum_{j=i}^{n} x_i \cdot x_j \cdot \omega_{ij} + \sum_{i=1}^{n} x_i \cdot w_i \ . \tag{4}$$

Одним из преимуществ использования данного подхода является существенное повышение вычислительной мощности нейросети. Уже для однослойной нейросети снимается проблема функции ИСКЛЮЧАЮЩЕЕ ИЛИ. Действительно, эта функция двух аргументов принимает значение равное единице, когда один из аргументов равен единице (но не оба). Обозначим один вход через *x*, а другой через *y*, тогда все их возможные комбинации будут состоять из четырех точек на плоскости *x*-*y*.

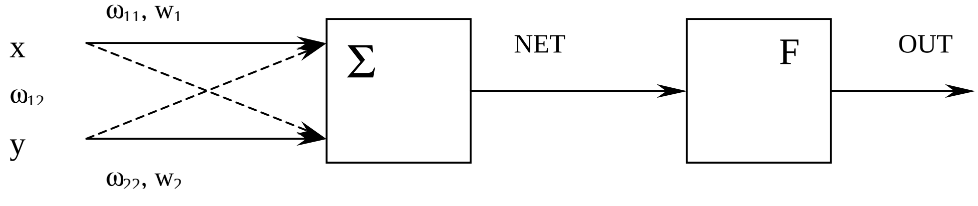

Рисунок 1 – Однослойная нейронная система

В сети (рис. 1) функция F является обычным порогом, так что OUT принимает значение ноль, когда NET меньше 0,5. Нейрон выполняет следующее вычисление:

$$NET = x^2\omega_{11} + xy\omega_{12} + y^2\omega_{22} + xw_1 + yw_2 \ . \tag{5}$$

Легко доказать, что существует бесконечное множество значений весов, которые могут дать соотношения между входом и выходом, отвечающие заданной функции. Чтобы понять снятие ограничения для функции ИСКЛЮЧАЮЩЕЕ ИЛИ для нейронных сетей с нелинейностью в синаптических связях, заданной формулой (4), зафиксируем значение NET на величине порога 0,5. Сеть в этом случае описывается уравнением (6). Это уравнение второго порядка по x и y, т.е. все значения по x и y, удовлетворяющие этому уравнению, будут лежать на некоторой кривой второго порядка в плоскости x-y.

$$x^2\omega_{11} + xy\omega_{12} + y^2\omega_{22} + xw_1 + yw_2 = 0{,}5\,. \qquad (6)$$

Любые входные значения для x и y на этой кривой будут давать пороговое значение 0,5 для NET. Входные значения с одной стороны кривой обеспечат значения NET больше порога, следовательно, OUT=1. Входные значения по другую сторону кривой обеспечат значения NET меньше порогового значения, делая OUT равным 0. Изменения значений ω_{11}, ω_{12}, ω_{22}, w_1, w_2 и порога будут менять вид и положение кривой. Можно подобрать множество значений коэффициентов, удовлетворяющих данному условию. Следовательно, уже для однослойной нейронной сети с подобными характеристиками ограничение для функции ИСКЛЮЧАЮЩЕЕ ИЛИ снимается, что существенно повышает ее вычислительную мощь, при меньшем количестве нейронов и слоев.

Рассмотрим алгоритм обратного распространения ошибки для сетей подобного рода.

Алгоритм обратного распространения – это итеративный градиентный алгоритм обучения, который используется с целью минимизации среднеквадратичного отклонения текущих от требуемых выходов многослойных нейронных сетей с последовательными связями.

Согласно методу наименьших квадратов, минимизируемой целевой функцией ошибки нейронной сети является величина [2]:

$$E(\omega, w) = \frac{1}{2}\sum_{j,k}(y_{j,k}^{(Q)} - d_{j,k})^2\,, \qquad (7)$$

где $y^{(Q)}_{j,k}$ – реальное выходное состояние нейрона j-го выходного слоя нейронной сети при подаче на ее входы k-го образа; $d_{j,k}$ – требуемое состояние этого нейрона.

Суммирование ведется по всем нейронам выходного слоя и по всем обрабатываемым сетью образам. Минимизация методом градиентного спуска обеспечивает подстройку весовых коэффициентов следующим образом:

$$\Delta w_{ij}^{(q)} = -\eta\frac{\partial E}{\partial w_{ij}}\,, \qquad (8)$$

$$\Delta \omega_{i_1 i_2 j}^{(q)} = -\eta\frac{\partial E}{\partial \omega_{i_1 i_2 j}}\,, \qquad (9)$$

где $w_{ij}^{(q)}$ – весовой коэффициент синаптической связи, соединяющей i-й нейрон слоя (q-1) с j-м нейроном слоя q; $\omega_{i_1 i_2 j}^{(q)}$ – комплексный весовой коэффициент синаптической связи соотносящей i_1-й и i_2-й нейроны слоя (q-1) с j-м нейроном слоя q; η - коэффициент скорости обучения,
$0 < \eta < 1$.

В соответствие с правилом дифференцирования сложной функции:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial s_j}\frac{\partial s_j}{\partial w_{ij}}\,, \qquad (10)$$

$$\frac{\partial E}{\partial \omega_{i_1 i_2 j}} = \frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial s_j}\frac{\partial s_j}{\partial \omega_{i_1 i_2 j}}\,, \qquad (11)$$

где s_j – взвешенная сумма входных сигналов нейрона j, т.е. аргумент активационной функции, следует напомнить, что s_j дается формулой:

$$s_j^{(q)} = \sum_{i_1=1}^{n-1} \sum_{i_2=i_1}^{n} y_{i_1}^{(q-1)} \cdot y_{i_2}^{(q-1)} \cdot \omega_{i_1 i_2 j}^{(q)} + \sum_{i=1}^{n} y_i^{(q-1)} \cdot w_{ij}^{(q)} . \quad (12)$$

Так как производная активационной функции должна быть определена на всей оси абсцисс, то функция единичного скачка и прочие активационные функции с неоднородностями не подходят для рассматриваемых нейронных сетей. В них применяются такие гладкие функции, как гиперболический тангенс или классический сигмоид с экспонентой. Например, в случае гиперболического тангенса:

$$\frac{\partial y}{\partial s} = 1 - s^2 . \quad (13)$$

Третий множитель $\partial s_j / \partial w_{ij}$ в формуле (10) равен выходу нейрона предыдущего слоя $y_i^{(q-1)}$, тогда как в формуле (11) $\partial s_j / \partial \omega_{i_1 i_2 j}$ равен произведению выходов предыдущего слоя $y_{i_1}^{(q-1)} \cdot y_{i_2}^{(q-1)}$.

Множитель $\partial E / \partial y_j$ в обоих случаях легко раскладывается следующим образом:

$$\frac{\partial E}{\partial y_j} = \sum_r \frac{\partial E}{\partial y_r} \frac{\partial y_r}{\partial s_r} \frac{\partial s_r}{\partial y_j} = \sum_r \frac{\partial E}{\partial y_r} \frac{\partial y_r}{\partial s_r} \left(w_{jr}^{(q+1)} + y_j \omega_{jjr}^{(q+1)} + \sum_p y_p \omega_{pjr}^{(q+1)} \right) . \quad (14)$$

Здесь суммирование по r выполняется среди нейронов слоя (q+1).

Введя новую переменную

$$\delta_j^{(q)} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial s_j} , \quad (15)$$

получим рекурсивную формулу для расчета величин $\delta_j^{(q)}$ слоя q из величин $\delta_l^{(q+1)}$ более старшего слоя (q+1):

$$\delta_j^{(q)} = \left[\sum_r \delta_r^{(q+1)} \left(w_{jr}^{(q+1)} + y_j \omega_{jjr}^{(q+1)} + \sum_p y_p \omega_{pjr}^{(q+1)} \right) \right] \frac{\partial y_j}{\partial s_j} . \quad (16)$$

Для выходного слоя:

$$\delta_j^{(Q)} = (y_j^{(Q)} - d_j) \frac{\partial y_j}{\partial s_j} . \quad (17)$$

Теперь можно записать (8) и (9) в раскрытом виде:

$$\Delta w_{ij}^{(q)} = -\eta \cdot \delta_j^{(q)} \cdot y_i^{(q-1)} , \quad (18)$$

$$\Delta \omega_{i_1 i_2 j}^{(q)} = -\eta \cdot \delta_j^{(q)} \cdot y_{i_1}^{(q-1)} \cdot y_{i_2}^{(q-1)} . \quad (19)$$

Таким образом, полный алгоритм обучения нейронной сети с нелинейностью в синаптических связях, заданной в виде квадратичной формы, с помощью процедуры обратного распространения строится следующим образом:

ШАГ 1. Подать на входы сети один из возможных образов и в режиме обычного функционирования нейронной сети, когда сигналы распространяются от входов к выходам, рассчитать значения последних.

ШАГ 2. Рассчитать $\delta^{(Q)}$ для выходного слоя по формуле (17).

Рассчитать по формулам (18) и (19) изменения весов $\Delta w^{(Q)}$ и $\Delta \omega^{(Q)}$ для слоя Q.

ШАГ 3. Рассчитать по формулам (16), (18) и (19) соответственно $\delta^{(q)}$, $\Delta w^{(q)}$ и $\Delta \omega^{(q)}$ для всех остальных слоев q = (Q-1)…1.

ШАГ 4. Скорректировать веса в нейронной сети:

$$w_{ij}^{(q)}(t) = w_{ij}^{(q)}(t-1) + \Delta w_{ij}^{(q)}(t) , \quad (20)$$

$$\omega_{i_1 i_2 j}^{(q)}(t) = \omega_{i_1 i_2 j}^{(q)}(t-1) + \Delta \omega_{i_1 i_2 j}^{(q)}(t) . \quad (21)$$

ШАГ 5. Если ошибка сети существенна, перейти на шаг 1. В противном случае – конец.

Сети на шаге 1 попеременно в случайном порядке предъявляются все тренировочные образы, чтобы сеть, образно говоря, не забывала одни по мере запоминания других.

Из выражений (18) и (19) следует, что когда выходные значения $y_i^{(q-1)}$, $y_{i_1}^{(q-1)}$ и $y_{i_2}^{(q-1)}$ стремятся к нулю, эффективность обучения заметно снижается. При двоичных входных векторах более половины весовых коэффициентов не будет корректироваться, поэтому область значений выходов нейронов (0, 1) желательно сдвинуть в пределы (-0,5, 0,5), что достигается простыми модификациями логистических функций. Например, сигмоид с экспонентой преобразуется к виду:

$$f(s) = -\frac{1}{2} + \frac{1}{1+e^{-\alpha s}}. \tag{22}$$

При проектировании нейронной сети с нелинейностью в синаптических связях важен вопрос, чтобы разработчик нейронной сети четко понимал какими ценными свойствами он наделяет нейрон и каких лишает. Введение нелинейности, заданной формулой (4) увеличивает вычислительную мощь сети, то есть позволяет из меньшего числа нейронов с «нелинейными» синапсами сконструировать нейронную сеть, выполняющую работу обычной нейросети с большим числом стандартных нейронов и более сложной конфигурации. Это особенно важно при решении задач критичных к числу используемых нейронов [3].

Помимо указанных достоинств, нейронные сети с нелинейностью в синаптических связях, заданной в виде квадратичной формы имеют и свои недостатки. Наиболее существенным из них, как расплата за повышение вычислительной мощности сети, является склонность подобного рода нейронных сетей к переобучению.

Литература

1. Richard P. Lippmann. An Introduction to Computing with Neural Nets, IEEE Acoustics. Speech, and Signal Processing Magazine. April 1987.
2. Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика. – М.: Горячая линия – Телеком, 2001. – 382 с.
3. Уоссермен Ф. Нейрокомпьютерная техника. – М.: Мир, 1992. – 278 с.