

Литература.

1. Джеральд Фишбах. Психика и мозг// В мире науки, N11, 1992г.

МЕТОД ОБУЧЕНИЯ МНОГОСЛОЙНОЙ НЕЙРОННОЙ СЕТИ

Головко В.А., Дунец А.П., Левонюк Д.Н.

Брестский политехнический институт

Введение

В настоящее время наиболее распространенным методом обучения для нейронных сетей с прямыми связями является метод обратного распространения ошибки. Он характеризуется неадаптивной скоростью обучения, что значительно увеличивает временную сложность этого метода, а также неустойчивостью процесса обучения. Неустойчивость обучения заключается в том, что сходимость алгоритма обратного распространения ошибки зависит от начальной инициализации синаптических связей. Это происходит из-за наличия локальных минимумов в целевой функции.

В работе [1] был предложен алгоритм обратного распространения ошибки с адаптивным шагом обучения. Это позволило значительно сократить временную сложность алгоритма. Так на операциях распознавания рукописных цифр время обучения сократилось на порядок. Однако это не решило проблемы устойчивости алгоритма обучения, которая проявляется на некоторых задачах. Другой аспект этой проблемы состоит в том, что не существует теоретических методов архитектурного синтеза нейронных сетей с прямым распространением сигнала. Под архитектурным синтезом здесь понимается выбор количества слоев, количества нейронных элементов в слое и функции активации для каждого слоя. Как правило, существ-

вуют только эмпирические рекомендации. При этом все исследования ориентированы на гомогенные нейронные сети (с одной функцией активации нейронных элементов) и в лучшем случае на простейшие гетерогенные сети с линейной функцией активации на выходе.

В данной работе приводится новый подход к обучению многослойной нейронной сети, в котором заложены потенциальные возможности для архитектурного синтеза сети. Тестирование проводилось на задачах кодирования информации. Результаты экспериментов обсуждаются.

1. Теоретические основы метода

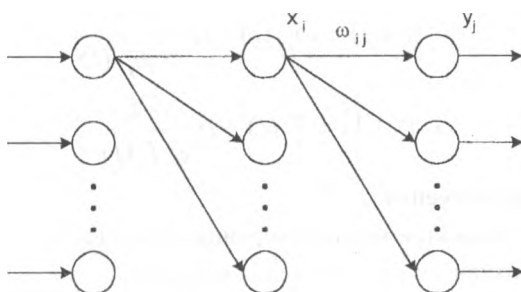


Рис. 1. Общая структура нейронной сети.

Дана многослойная нейронная сеть (рис. 1), в качестве функции активации нейронных элементов которой используется гиперболический тангенс

$$y_j = th(S_j) = \frac{e^{S_j} - e^{-S_j}}{e^{S_j} + e^{-S_j}}, \quad (1)$$

где S_j характеризует взвешенную сумму j -того нейронного элемента. Она определяется следующим образом

$$S_j = \sum_j x_i \omega_{ij} - T_j, \quad (2)$$

3. Искусственный интеллект и нейронные сети

где T_j - порог j -того нейронного элемента, x_i - выход нейронного элемента предыдущего слоя, ω_{ij} - весовой коэффициент между i -тым и j -тым нейронами.

Среднеквадратичная ошибка нейронной сети для одного образа определяется как

$$E = \frac{1}{2} \sum_j (y_j - t_j)^2, \quad (3)$$

где t_j - эталонное выходное значение для j -того нейрона.

Стандартный метод обратного распространения ошибки состоит в использовании алгоритма градиентного спуска в пространстве весовых коэффициентов и порогов нейронной сети:

$$\omega_{ij}(t+1) = \omega_{ij}(t) - \alpha \frac{\partial E}{\partial \omega_{ij}(t)}, \quad (4)$$

$$T_j(t+1) = T_j(t) - \alpha \frac{\partial E}{\partial T_j(t)}, \quad (5)$$

где α - скорость обучения.

При использовании метода скорейшего спуска можно получить выражение для адаптивного шага обучения нейронной сети:

$$\alpha(t) = \frac{\sum_j \gamma_j^2 / ch^2 S_j}{(1 + \sum_i x_i^2) \sum_j \gamma_j^2 (1/ch^2 S_j)^2}, \quad (6)$$

где γ_j - ошибка j -того нейронного элемента. Для выходного слоя

$$\gamma_j = y_j - t_j, \quad (7)$$

а для остальных слоев

$$\gamma_j = \sum_i \gamma_i \frac{1}{ch^2 S_j} \omega_{ji}, \quad (8)$$

где i - количество нейронов следующего слоя по отношению к слою j . С учетом этого

$$\omega_{ij}(t+1) = \omega_{ij}(t) - \alpha(t) \gamma_j y_i / ch^2 S_j, \quad (9)$$

$$T_j(t+1) = T_j(t) - \alpha(t) \gamma_j / ch^2 S_j, \quad (10)$$

Как уже отмечалось, выражение (6) позволяет значительно повысить скорость обучения. Однако это не решает проблемы устойчивости алгоритма. Для нейтрализации этого недостатка предлагается наряду с настройкой весовых коэффициентов и порогов нейронных элементов проводить также настройку выходов нейронных элементов промежуточных слоев. Так для последнего слоя выходы нейронных элементов будут изменяться следующим образом:

$$x_i(t+1) = x_i(t) - \alpha_i(t) \frac{\partial E}{\partial x_i(t)}. \quad (11)$$

Найдем производную функции ошибки по x_i ,

$$\frac{\partial E}{\partial x_i} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial S_j} \frac{\partial S_j}{\partial x_i} = \sum_j (y_j - t_j) \frac{1}{ch^2 S_j} \omega_{ij} = \gamma_i. \quad (12)$$

Для нахождения адаптивного шага обучения $\alpha_i(t)$ будем использовать метод скорейшего спуска. Тогда

$$\alpha_i(t) = \min \left\{ E \left(x_i - \alpha_i(t) \frac{\partial E}{\partial x_i(t)} \right) \right\}. \quad (13)$$

Определим взвешенную сумму как

$$S_j' = \omega_{ij} (x_i(t) - \alpha_i \gamma_i) + \sum_{k \neq i} \omega_{kj} x_k(t) - T_j. \quad (14)$$

После преобразования этого выражения получаем

$$S_j' = S_j - \alpha_i \gamma_i \omega_{ij}. \quad (15)$$

Раскладывая функцию $y_j' = th(S_j')$ в ряд Тейлора и преобразуя полученное выражение имеем

3. Искусственный интеллект и нейронные сети

$$y_j' = y_j - \alpha_i \gamma_i \omega_{ij} . \quad (16)$$

Отсюда среднеквадратичная ошибка сети

$$E = \frac{1}{2} \sum_j (y_j' - t_j)^2 . \quad (17)$$

Находим такое α_i , при котором среднеквадратичная ошибка сети минимальна

$$\frac{\partial E}{\partial \alpha_i} = \sum_j (y_j - \alpha_i \gamma_i \omega_{ij} - t_j)(-\gamma_i \omega_{ij}) = 0 . \quad (18)$$

Преобразуя последнее выражение, находим значение адаптивного шага обучения для i -того нейрона промежуточного слоя

$$\alpha_i = \frac{\sum_j (y_j - t_j) \omega_{ij}}{\sum_j \omega_{ij}^2 \sum_j (y_j - t_j) \frac{1}{ch^2 S_j} \omega_{ij}} . \quad (19)$$

Исходя из независимости слоев предполагаем, что $y_j - t_j = \gamma_j$. Тогда

$$\alpha_i = \frac{\sum_j \gamma_j \omega_{ij}}{\gamma_i \sum_j \omega_{ij}^2} , \quad (20)$$

где γ_j, γ_i - соответственно ошибка j -того нейрона следующего слоя и i -того нейрона предыдущего слоя.

Таким образом предполагается проводить обучение нейронной сети по трем параметрам: весовые коэффициенты, пороги и выходы нейронных элементов. Выходы нейронных элементов изменяются с целью минимизации среднеквадратичной ошибки нейронной сети следующим образом

$$x_i(t+1) = x_i(t) - \alpha_i(t) \gamma_i , \quad (21)$$

где γ_i - ошибка i -того нейронного элемента.

2. Тестирование

Экспериментальная проверка полученных результатов проводилась на задаче кодирования информации. При этом нейронная сеть должна на основе циклического кодирования осуществлять преобразование информационного полинома в избыточный код.

Пусть разрядность информационного полинома $n = 4$, а избыточного полинома $m = 7$. Тогда архитектура многослойной нейронной сети содержит 4 входных и 7 выходных нейронов. В промежуточном слое будем использовать 8 нейронных элементов. Тогда имеем трехслойную сеть с объемом обучающей выборки $L = 16$.

Для обучения такой сети используется следующий алгоритм:

1. Случайная инициализация весовых коэффициентов нейронной сети и задание минимальной среднеквадратичной ошибки сети E_m .
2. Для L входных образов производится настройка весовых коэффициентов и порогов по выражениям (9), (10) только для последнего слоя. Одновременно для каждого образа в соответствии с (21) определяются желаемые выходы X_i нейронных элементов предпоследнего слоя.
3. Рассматриваются только последний и предпоследний слои сети: в качестве входной информации используются желаемые выходы X_i нейронных элементов, в качестве выходной - эталонные выходы
 - 3.1. Для L входных образов производится настройка весовых коэффициентов, порогов и желаемых выходов по выражениям (9), (10) и (21) соответственно.
 - 3.2. Пункт 3.1 повторяется, пока суммарная среднеквадратичная ошибка рассматриваемого фрагмента нейронной сети не станет меньше E_m .

3. Искусственный интеллект и нейронные сети

4. Производится настройка на L образах весовых коэффициентов и порогов следующего слоя сети. При этом ошибка i -того нейронного элемента равна $\gamma_i = X_i - \bar{X}_i$.
5. Процедура повторяется с пункта 2, пока суммарная среднеквадратичная ошибка нейронной сети не станет меньше E_m .

Эксперименты показали, что в отличие от других алгоритмов данный обладает 100 % устойчивостью. Так при любой начальной инициализации весовых коэффициентов нейронная сеть обучалась до минимальной ошибки. При использовании стандартного алгоритма обратного распространения ошибки только в 90 % всех попыток обучить сеть достигался приемлемый результат.

Однако при использовании описанного выше алгоритма увеличилось время обучения нейронной сети в среднем в 3 раза по сравнению с алгоритмом обратного распространения ошибки, который использует адаптивный шаг обучения. По сравнению со стандартным алгоритмом обратного распространения ошибки, когда $\alpha = const$, предложенный алгоритм обладает в 2 раза меньшей временной сложностью.

Заключение

На основе разработанного метода проведена экспериментальная проверка алгоритма, который характеризуется независимым обучением каждого из слоев нейронной сети. Эксперименты показали, что предложенный алгоритм обладает большей устойчивостью по отношению к стандартному алгоритму обратного распространения ошибки. В предложенном методе заложены также потенциальные возможности для автоматической генерации архитектуры нейронной сети. В настоящее время в этом направлении проводятся исследования.

Литература

1. Golovko V., Savitsky Ju., Gladischuk V. A neural net for predicting problem. Timisoara: University of Timisoara, Romania, 1996.

ИНТЕЛЛЕКТУАЛЬНАЯ НЕЙРОННАЯ СИСТЕМА ДЛЯ АВТОНОМНОГО УПРАВЛЕНИЯ МОБИЛЬНЫМ РОБОТОМ

Головко В.А.

Брестский политехнический институт

Введение

В работе описывается интеллектуальная нейронная система для автономного управления мобильным роботом. Она позволяет интегрировать преимущества биологических существ и вычислительной техники для ориентации робота не знакомой местности. Входной информацией является информация от сенсорных датчиков, в качестве которых используется 7 ультразвуковых датчиков и 2D инфракрасный сканер. Ультразвуковые датчики расположены по периметру робота согласно рис.1. Задача нейронной системы состоит в том, чтобы на основе информации от разнородных сенсорных устройств и координат конечной точки формировать оптимальное направление движения в пространстве с препятствиями. Это эквивалентно обеспечению минимального угла между направлением на цель и текущим направлением робота. Нейронная система должна на основе неточной информации от сенсорных датчиков обеспечить робастное управление мобильным роботом. Архитектура нейронной сети является многослойной и состоит из различных типов нейронных сетей. В процессе функционирования она решает следующие задачи: