

РЕАЛИЗАЦИЯ ПАРАЛЛЕЛЬНОГО АЛГОРИТМА ПОДБОРА АКТИВАЦИОННЫХ ФУНКЦИЙ ИСКУССТВЕННОЙ НЕЙРОННОЙ СЕТИ

Крючин О.В., Арзамасцев А.А.

Тамбовский государственный университет им. Г.Р. Державина

Аннотация. В данной работе описываются параллельные алгоритмы подбора активационных функций нейронов. Для оценки эффективности предложенных алгоритмов используется подсчет числа сделанных ими мультипликативных и аддитивных операций.

Ключевые слова: искусственные нейронные сети, параллельные алгоритмы, кластерные системы, активационные функции нейронов.

Введение

Известно, что задача обучения искусственных нейронных сетей (ИНС) сводится к минимизации значения невязки, для которого необходимо подобрать значения весовых коэффициентов и активационных функций нейронов. На сегодняшний день разработано множество алгоритмов подбора вектора весовых коэффициентов [1, 2, 3], но не существует эффективных методов подбора активационных функций нейронов. Одна из причин заключается в том, что комбинирование алгоритмов подбора активационных функций нейронов и значений весовых коэффициентов требует значительных временных затрат. Решением этой проблемы может стать использование кластерных систем — массивов однородных высокопроизводительных компьютеров, соединенных в вычислительную сеть. Целью данной работы является разработка алгоритмов подбора активационных функций нейронов и их реализация для кластерных систем.

Компоненты нейрона

Универсальный математический нейрон состоит из двух частей — сумматора и активационной функции (рис. 1). Сумматор получает на вход произведения выходных сигналов других нейронов (на рис. y_0, y_1, y_2) на значения весовых коэффициентов (на рис. w_0, w_1, w_2), а также величину импульса (добавочного весового коэффициента). В данном случае имеется два импульса, названных внутренним (p_i) и внешним (p_E), которые различаются тем, что только внешний импульс может подбираться наряду с весовыми коэффициентами. После того, как сумматор вычисляет значение, оно умножается на коэффициент и передается на вход активационной функции. Значение активационной функции является выходным значением нейрона (y_i). Таким образом, выходное значение i -ого нейрона можно выразить следующей формулой:

$$y_i = f(c_{fi} x_i) = f_i \left(c_{fi} \left(\sum_{j=0}^{N_{xi}-1} y_j w_{ji} + p_{Ei} + p_{Li} \right) \right) \quad (1)$$

где $w_{j,i}$ – значение весового коэффициента, соединяющего j -ый и i -ый нейроны, c_f – коэффициент активационной функции, f – активационная функция, N_x – количество входов нейрона [4].

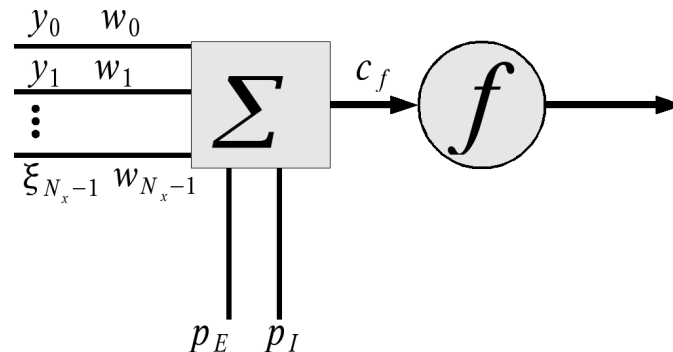


Рисунок 1. Модель математического нейрона.

Активационные функции обычно бывают следующих видов: линейная ($y = x$), квадратичная ($y = x^2$), кубическая ($y = x^3$), функция четвертой степени ($y = x^4$), синусоида ($y = \sin(x)$), косинусоида ($y = \cos(x)$), тангенсная ($y = \operatorname{tg}(x)$), котангенсная ($y = \operatorname{ctg}(x)$), сигмоида ($y = \frac{1}{1+x}$, $y = \frac{1}{1+e^x}$, $y = \frac{1}{1+e^{-x}}$, $y = \frac{1}{1+e^{|x|}}$), пороговая функция.

Универсальная пороговая функция состоит из пяти элементов: значения порога (h_t), верхнего и нижнего значений по умолчанию (\hat{y}_1 , \hat{y}_0), а также верхнего и нижнего флага, показывающих, следует ли заменять входное значение на значение по умолчанию (b_{C1} , b_{C0}).

$$y = \begin{cases} \hat{y}_0, & b_{C0} = 1, \quad x \leq h_t; \\ x, & b_{C0} = 0, \quad x \leq h_t; \\ \hat{y}_1, & b_{C1} = 1, \quad x > h_t; \\ x, & b_{C1} = 0, \quad x > h_t; \end{cases} \quad (2)$$

Значение функции принимает входное значение в двух случаях: если входное значение меньше порога ($x < h_t$) и нижний флаг замещения равен нулю ($b_{C0} = 0$) и если входное значение не меньше порога ($x \geq h_t$) и верхний флаг замещения равен нулю ($b_{C1} = 0$). В противном случае выходное значение принимает либо верхнее, либо нижнее значение по умолчанию (\hat{y}_1 или \hat{y}_0) (в зависимости от того больше ли входное значение чем пороговое) [5].

Подбор компонентов активационных функций

Исходя из вышесказанного, нейрон, имеющий активационную функцию, отличную от ЭФТЖ, т.6, 2011

пороговой (2), имеет 3 подбираемых элемента (коэффициент активационной функции c_f , внутренний импульс p_I и тип активационной функции f), а при функции (3) – 8 (коэффициент активационной функции c_f , импульс p_I , значение порога h_t , а также верхнее и нижнее значения по умолчанию (\hat{y}_1 и \hat{y}_0) и флаги замены (b_{C0} и b_{C1})). Значения большинства этих элементов являются вещественными числами. Следовательно, могут быть заданы пределы (χ_d и χ_u) и шаг (χ_s) сканирования этих элементов, в результате чего значения могут быть выражены формулами (3)-(7).

$$c_f = \chi_d(c_f) + i\chi_s(c_f), \quad i = 0, 1, \dots, \left[\frac{\chi_u(c_f) - \chi_d(c_f)}{\chi_s(c_f)} \right] + 1 \quad (3)$$

$$p_I = \chi_d(p_I) + i\chi_s(p_I), \quad i = 0, 1, \dots, \left[\frac{\chi_u(p_I) - \chi_d(p_I)}{\chi_s(p_I)} \right] + 1 \quad (4)$$

$$\hat{y}_0 = \chi_d(\hat{y}_0) + i\chi_s(\hat{y}_0), \quad i = 0, 1, \dots, \left[\frac{\chi_u(\hat{y}_0) - \chi_d(\hat{y}_0)}{\chi_s(\hat{y}_0)} \right] + 1 \quad (5)$$

$$\hat{y}_1 = \chi_d(\hat{y}_1) + i\chi_s(\hat{y}_1), \quad i = 0, 1, \dots, \left[\frac{\chi_u(\hat{y}_1) - \chi_d(\hat{y}_1)}{\chi_s(\hat{y}_1)} \right] + 1 \quad (6)$$

$$\hat{h}_t = \chi_d(\hat{h}_t) + i\chi_s(\hat{h}_t), \quad i = 0, 1, \dots, \left[\frac{\chi_u(\hat{h}_t) - \chi_d(\hat{h}_t)}{\chi_s(\hat{h}_t)} \right] + 1 \quad (7)$$

Возможны случаи, когда в переборе участвуют не все элементы, а лишь некоторые, поэтому выразить количество состояний (вариантов активационных функций) нейрона затруднительно. Например, в случае, когда в переборе участвуют коэффициент и внутренний импульс, значения которых перебираются в пределах $[-1; 1]$ с шагом 0.5, нейрон имеет 25 состояний.

На рис. 2-3 изображена блок-схема алгоритма изменения активационной функции нейрона (под «стандартным значением» подразумевается значение по умолчанию). Ниже приводится краткое пояснение:

- Ввод параметров подбора подразумевает ввод указаний, какие из элементов нейрона следует подбирать, и ввод для них параметров сканирования ($\chi_d(c_f)$, $\chi_u(c_f)$, $\chi_s(c_f)$ — для коэффициента активационной функции, $\chi_d(p_I)$, $\chi_u(p_I)$, $\chi_s(p_I)$ — для импульса, $\chi_d(h_t)$, $\chi_u(h_t)$, $\chi_s(h_t)$ — для значения порога и т.д.).
- Условие проверки сканирования коэффициента активационной функции c_f выполняется в случае, когда этот коэффициент необходимо сканировать и его значение меньше верхнего предела ($c_f < \chi_u(c_f)$). В случае выполнения этого условия значение коэффициента увеличивается на величину шага ($c_f = c_f + \chi_s(c_f)$).
- Условие проверки сканирования импульса p_I выполняется в случае, когда значение

импульса необходимо сканировать и оно меньше верхнего предела ($p_l < \chi_u(p_l)$). В случае выполнения этого условия значение внутреннего импульса увеличивается на величину шага ($p_l = p_l + \chi_s(p_l)$), а значение коэффициента активационной функции (если его необходимо сканировать) устанавливается нижним значением ($c_f = \chi_d(c_f)$).

- Условие проверки, является ли тип функции f простым (непороговым). В этом случае осуществляется проверка, является ли тип функции последним из перечисляемых (в этом случае перебор завершается). Если текущий тип функции не последний, то устанавливается новый тип, а коэффициент и импульс (в случае перебора) устанавливаются нижними значениями ($c_f = \chi_d(c_f)$, $p_l = \chi_d(p_l)$).
- Условие проверки сканирования порогового значения h_l выполняется в случае, когда значение порога необходимо сканировать и оно меньше верхнего предела ($h_l < \chi_u(h_l)$). В случае выполнения этого условия значение порога увеличивается на величину шага ($h_l = h_l + \chi_s(h_l)$), а коэффициент активационной функции и импульс (в случае перебора) устанавливаются нижними значениями ($c_f = \chi_d(c_f)$, $p_l = \chi_d(p_l)$).
- Условие проверки флага нижнего стандартного значения b_{c0} выполняется в случае, когда этот флаг участвует в сканировании и равен нулю ($b_{c0} = 0$). В этом случае он устанавливается в 1, а коэффициент активационной функции, значение порога и импульс (в случае перебора) устанавливаются нижними значениями ($c_f = \chi_d(c_f)$, $h_l = \chi_d(h_l)$, $p_l = \chi_d(p_l)$).
- Условие проверки флага верхнего стандартного значения b_{c1} выполняется в случае, когда этот флаг участвует в сканировании и равен нулю ($b_{c1} = 0$). В этом случае он устанавливается в 1, нижний флаг (в случае перебора) в 0 ($b_{c1} = 1$, $b_{c0} = 0$), а коэффициент активационной функции, значение порога и импульс (в случае перебора) устанавливаются нижними значениями ($c_f = \chi_d(c_f)$, $h_l = \chi_d(h_l)$, $p_l = \chi_d(p_l)$).
- Условие проверки сканирования нижнего значения по умолчанию выполняется в случае, когда нижнее стандартное значение участвует в сканировании и меньше верхнего предела ($\hat{y}_0 < \chi_u(\hat{y}_0)$). В этом случае нижнее стандартное значение увеличивается на величину шага ($\hat{y}_0 = \hat{y}_0 + \chi_s(\hat{y}_0)$), флаги стандартных значений (в случае перебора) устанавливаются в 0 ($b_{c0} = 0$, $b_{c1} = 0$), а коэффициент активационной функции, значение порога и импульс (в случае перебора) устанавливаются нижними значениями ($c_f = \chi_d(c_f)$, $h_l = \chi_d(h_l)$, $p_l = \chi_d(p_l)$).
- Условие проверки сканирования верхнего значения по умолчанию выполняется в случае, если оно участвует в сканировании и меньше верхнего предела ($\hat{y}_1 < \chi_s(\hat{y}_1)$). В этом случае верхнее стандартное значение увеличивается на величину шага ($\hat{y}_1 = \hat{y}_1 + \chi_s(\hat{y}_1)$), флаги стандартных значений (в случае перебора) устанавливаются в

0 ($b_{c0} = 0, b_{c1} = 0$), а нижнее стандартное значение, коэффициент активационной функции, значение порога и импульс (в случае перебора) устанавливаются нижними значениями ($\hat{y}_0 = \chi_d(\hat{y}_0), c_f = \chi_d(c_f), h_t = \chi_d(h_t), p_l = \chi_d(p_l)$).

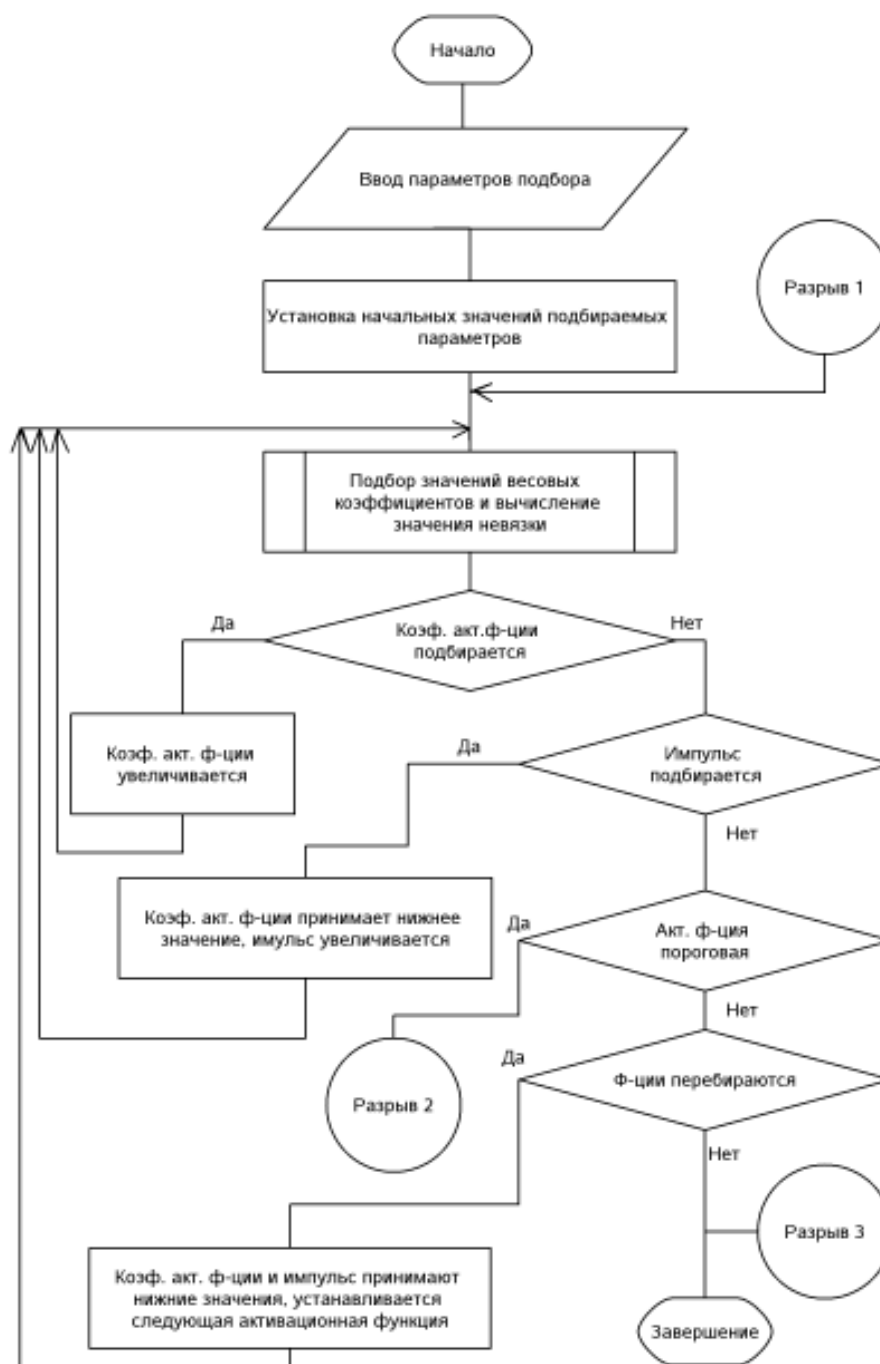


Рисунок 2. Блок-схема изменения одного из элементов нейрона (часть 1).

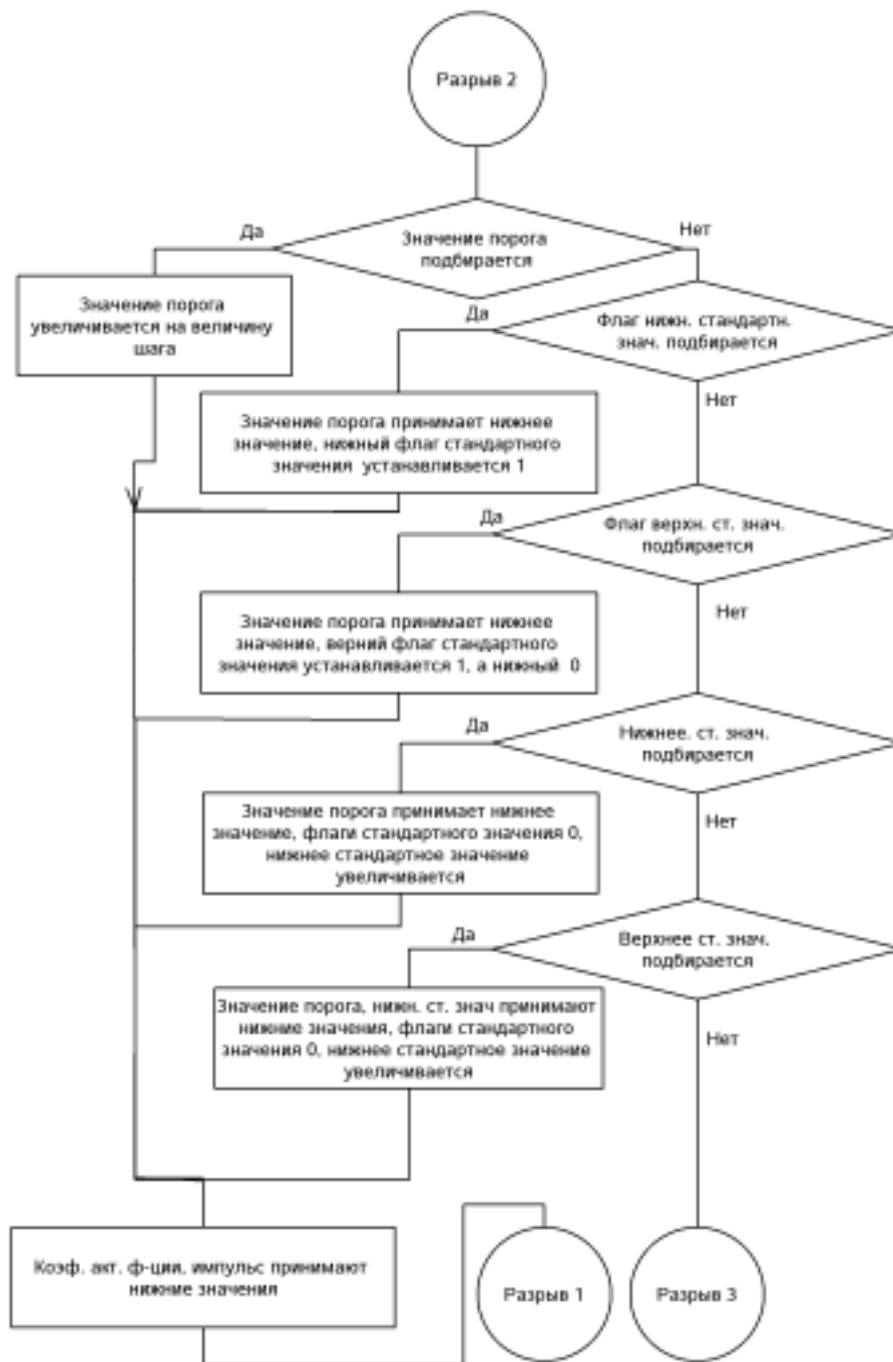


Рисунок 3. Блок-схема изменения одного из элементов нейрона (часть 2).

Подбор состояний всех нейронов

Для успешного обучения необходимо подобрать значения $\bar{\mu}$ и \bar{w} таким образом, чтобы минимизировать невязку. Сделать это можно следующим образом: перебрать все комбинации активационных функций, подбирая для каждой из них весовые коэффициенты. При этом следует учитывать, что каждый нейрон может иметь уникальный набор активационных функций. Обозначим набор вариантов для i -го нейрона как \vec{v}_i , а размерность этого вектора, характеризующую количество комбинаций активационных функций нейрона — $N_{\mu}(\vec{v}_i)$.

Следовательно, активационную функцию i -го нейрона на I -ой итерации можно вычислить

по формуле (8).

$$\mu_i^{(l)} = v_{i,0} + \left(\left[\frac{I}{\nu(i)} \right] \bmod N_\mu(\vec{v}_i) \right) \quad (8)$$

где $\nu(i)$ вычисляется по формуле (9).

$$\nu(i) = \begin{cases} 1, & i = l_\mu - 1; \\ \prod_{j=i+1}^{l_\mu-1} N_\mu(\vec{v}_j), & i < l_\mu - 1; \end{cases} \quad (9)$$

Для каждого набора активационных функций подбираются значения весовых коэффициентов и вычисляется значение целевой функции $\varepsilon(\vec{\mu}_i, \vec{w})$, и, таким образом, после перебора всех вариантов получается набор значений погрешности. Минимальная из них соответствует наилучшему состоянию сети.

Общее количество итераций вычисляется по формуле (10).

$$I_\mu = \prod_{i=0}^{l_\mu-1} N_\mu(\vec{v}_i), \quad (10)$$

Параллельные алгоритмы подбора активационных функций

При использовании параллельных алгоритмов и n процессоров каждый процессор может выполнить часть итераций подбора активационных функций. Таким образом, ведущий процессор будет воспроизводить \hat{J}_μ итераций, а неведущий — J_μ (J_μ и \hat{J}_μ вычисляются по формулам (11)-(12))

$$J_\mu = \begin{cases} \left[\frac{I_\mu}{n} \right], & I_\mu \bmod n = 0; \\ \left[\frac{I_\mu}{n-1} \right], & I_\mu \bmod n \neq 0; \end{cases} \quad (11)$$

$$\hat{J}_\mu = \begin{cases} J_\mu, & I_\mu \bmod n = 0; \\ I_\mu - J_\mu(n-1), & I_\mu \bmod n \neq 0; \end{cases} \quad (12)$$

Таким образом, на управляющем процессоре алгоритм следующий:

- формирование $n-1$ структур для неуправляющих процессоров;
- рассылка структур на процессоры;
- перебор всех возможных вариантов из имеющихся на данном процессоре и выбор оптимального набора активационных функций и связанной с ним погрешности;
- получение с процессоров наилучшего вектора активационных функций и соответствующего ему значения невязки;

- выбор наименьшего значения невязки.

Для неуправляющего процессора алгоритм другой:

- получение структуры;
- перебор всех возможных вариантов и выбор оптимальных значений активационных функций и связанной с ними погрешности;
- отправка на ведущий процессор значений активационных функций и погрешности.

Количество мультипликативных операций, которые производит последовательный алгоритм подбора активационных функций можно записать следующим образом:

$$z_{\mu} = z_w I_{\mu} + 2\sigma I_{\mu} \quad (13)$$

где σ — коэффициент, равный отношению времени, затрачиваемого на одну аддитивную операцию к времени, затрачиваемому на одну мультипликативную, z_w — количество мультипликативных операций, необходимых для подбора весовых коэффициентов.

Для вычисления количества мультипликативных операций, производимых параллельным алгоритмом необходимо рассмотреть этапы, из которых он состоит:

1. инициализация;
2. передача данных с ведущего процессора на прочие;
3. перебор значений весовых коэффициентов, принадлежащих данному процессору;
4. передача данных со всех процессоров на ведущий;
5. выбор ведущим процессором оптимальной конфигурации;

Если обозначить количество элементов i -го нейрона как $N_s(\mu_i)$, то по формуле (17) можно выразить общее количество элементов в сети:

$$\hat{l}_{\mu} = \sum_{i=0}^{l_{\mu}-1} N_s(\mu_i) + l_w \quad (14)$$

где l_w — количество весов в ИНС.

На первом этапе на ведущем процессоре происходит $2\hat{l}_{\mu}$ аддитивных операции (\hat{l}_{μ} для организации цикла и \hat{l}_{μ} для присвоения), на втором — $\hat{l}_{\mu}(n-1)$ мультипликативных и $2\hat{l}_{\mu}(n-1)$ аддитивных. На неведущих на втором этапе происходит \hat{l}_{μ} мультипликативных и $2\hat{l}_{\mu}$ аддитивных операций, но начаться они могут лишь после того, как ведущий процессор отправит данные, поэтому на неведущем процессоре происходит $C_{\mu k}^{(2)}$ мультипликативных операций.

$$C_{\mu k}^{(2)} = k\hat{l}_{\mu} + 2k\sigma\hat{l}_{\mu} + \gamma(\hat{l}_{\mu}, v) + \hat{l}_{\mu} + 2\sigma\hat{l}_{\mu} = \hat{l}_{\mu}(k + 2k\sigma + 2\sigma + 1) + \gamma(\hat{l}_{\mu}, v) \quad (15)$$

Количество операций на третьем этапе аналогично количеству операций, выполняемых

последовательной версией алгоритма.

На четвертом этапе ведущему процессору необходимо получить лучшие весовые коэффициенты и соответствующее им значение невязки с каждого процессора, поэтому он делает $(n-1)(\hat{l}_\mu + 1)$ мультипликативных и $2(n-1)(\hat{l}_\mu + 1)$ аддитивных операций, а неведущий — $\hat{l}_\mu + 1$ мультипликативных и $2\hat{l}_\mu + 2$ аддитивных. Кроме того, ведущий процессор должен ожидать отправку.

На пятом ведущий процессор делает $2n$ аддитивных операций. Если привести аддитивные операции к мультипликативным, то можно получить выражения, приведенные в табл. 1.

Таблица 1. Количество мультипликативных операций на различных этапах параллельного подбора активационных функций.

| Этап | Ведущий процессор | Неведущий (k -ый) процессор |
|------|---|--|
| 1 | $2\sigma\hat{l}_\mu$ | |
| 2 | $\hat{l}_\mu(n-1)(1+2\sigma)$ | $\hat{l}_\mu(k+2k\sigma+2\sigma+1)+\gamma(\hat{l}_\mu, \nu)$ |
| 3 | $z_w\hat{J}_\mu + 2\sigma\hat{J}_\mu$ | $z_wJ_\mu + 2\sigma J_\mu$ |
| 4 | $n(l_w+1)(1+2\sigma)+\gamma(\hat{l}_\mu+1)$ | $(\hat{l}_\mu+1)(1+2\sigma)$ |
| 5 | $2\sigma n$ | |

До начала получения ведущим процессором весовых коэффициентов и значений невязки он осуществляет 3 этапа (обозначим общее количество операций $\hat{C}_{\mu 0}$), а прочие — 4 (обозначим общее количество операций $\hat{C}_{\mu k}$).

Таким образом, параллельный подбор активационных функций требует Z_μ операций:

$$Z_\mu = \max_{k=1..n-1} \left(\max(\hat{C}_{\mu 0} + k(\hat{l}_\mu + 1)(2\sigma + 1), \hat{C}_{\mu k} + \gamma(\hat{l}_\mu + 1, \nu)) \right) + (\hat{l}_\mu + 1)(1 + 2\sigma) + 2\sigma n \quad (16)$$

где $\hat{C}_{\mu 0}$, $\hat{C}_{\mu k}$ вычисляются по формулам (17)-(18).

$$\hat{C}_{\mu 0} = 2\sigma\hat{l}_\mu + \hat{l}_\mu(n-1)(1+2\sigma) + z_w\hat{J}_\mu + 2\sigma\hat{J}_\mu \quad (17)$$

$$\hat{C}_{\mu k} = \hat{l}_\mu(k+2k\sigma+\sigma+1) + \gamma(\hat{l}_\mu, \nu) + z_wJ_\mu + 2\sigma J_\mu + (\hat{l}_\mu+1)(1+2\sigma) \quad (18)$$

Таким образом, эффективность параллельного алгоритма можно выразить формулой (19).

$$\alpha_\nu(Z) = \frac{z_\mu}{nZ_\mu} = \frac{z_w I_\mu + 2\sigma I_\mu}{nZ_\mu} \quad (19)$$

Выводы

Таким образом, разработан параллельный алгоритм обучения ИНС, включающий в себя подбор активационных функций нейронов. Поскольку данный алгоритм подразумевает невысокую нагрузку на интерконект, то в качестве кластерной системы может быть использована обычная локальная сеть.

Используемая литература

1. Крючин О.В., Арзамасцев А.А., Королев А.Н., Горбачев С.И., Семенов Н.О. Универсальный симулятор, базирующийся на технологии искусственных нейронных сетей, способный работать на параллельных машинах. // Вестн. Тамб. Ун-та. Сер. Естеств. и техн. науки. – Тамбов, 2008. Т.13. Вып. 5. с. 372 – 375.
2. Крючин О.В. Параллельные алгоритмы обучения искусственных нейронных сетей с использованием градиентных методов. // Актуальные вопросы современной науки, техники и технологий. Материалы II Всероссийской научно-практической (заочной) конференции. — М.: Издательско-полиграфический комплекс НИИРРР, 2010 — 160 с., с. 81-86.
3. Крючин О.В. Разработка алгоритмов обучения искусственных нейронных сетей, использующих параллельное вычисление целевой функции // Известия Смоленского государственного университета. Ежеквартальный журнал №2 (10). Смоленск, 2010. с. 126-134.
4. Крючин О.В., Арзамасцев А.А. Сравнение эффективности последовательных и параллельных алгоритмов обучения искусственных нейронных сетей на кластерных вычислительных системах // Вестн. Тамб. ун-та. Сер. Естеств. и техн. науки. – Тамбов, 2010, Т. 15. Вып. 6. с. 372 – 375
5. Крючин О.В. Нейросетевой симулятор с автоматическим подбором активационных функций // Информационные технологии №7. Москва. 2010. с.47-52.