

К.В. МАХОТИЛО, канд. техн. наук, НТУ «ХПИ» (г. Харьков),
Д.И. ВОРОНЕНКО

МОДИФИКАЦИЯ АЛГОРИТМА ЛЕВЕНБЕРГА-МАРКВАРДА ДЛЯ ПОВЫШЕНИЯ ТОЧНОСТИ ПРОГНОСТИЧЕСКИХ МОДЕЛЕЙ СВЯЗНОГО ПОТРЕБЛЕНИЯ ЭНЕРГОРЕСУРСОВ В БЫТУ

Для підвищення ефективності навчання неймережевих прогностичних моделей зв'язного споживання енергоресурсів в побуті запропоновано модифікацію алгоритму Левенберга-Марквардта. Виконані експериментальні дослідження та визначені граничні значення діапазонів налагоджувальних параметрів запропонованої модифікації алгоритму для даної прикладної галузі.

The modification of Levenberg-Marquardt algorithm for power consumption neural-network-based forecasting model learning is proposed. Numerical experimental study is implemented and limits for proposed algorithm tuning parameters range are estimated for such applications.

Введение. Прогнозирование электрической нагрузки в разных секторах потребления является одной из ключевых задач математического моделирования в энергетике. Одним из наиболее эффективных методов прогнозирования связного потребления электричества и тепла в быту является метод на основе искусственных нейронных сетей. Одним из способов улучшения качества прогнозирования является выбор наиболее подходящего алгоритма обучения нейронных сетей и адаптации алгоритма применительно к задачам прогнозирования потребления энергоресурсов. Целью настоящего исследования является выбор наиболее эффективного алгоритма обучения нейросетей и его эвристическое улучшение на основе особенностей прогнозирования энергопотребления.

Постановка задачи. Анализ зарубежной и отечественной литературы показывает, что нейронные сети эффективно использовались для прогнозирования энергопотребления в условиях зашумленных и неполных данных [1 – 3]. Большинство опубликованных научных исследований опираются на наиболее известный алгоритм обучения многослойных перцептронов – алгоритм градиентного спуска с обратным распространением ошибки [4, 5]. Однако, в [6, 7] убедительно доказано, что этот достаточно универсальный алгоритм не обеспечивает наилучшей сходимости, требует достаточно много времени для обучения нейросети, а также требует эмпирического или алгоритмического определения оптимальной скорости обучения, которая значительно влияет на результативность алгоритма. Ввиду популярности данного алгоритма, многочисленные исследования были направлены на его улучшение, что привело к появлению эвристик [6 – 9], улучшающих этот алгоритм. Наиболее известные эвристики, такие как «момент» [10], «изменяемая скорость обучения» [11] или «стохастическое обучение» [12], приводят к небольшим улучшениям, реализующимся при решении определенных классов задач.

Значительное улучшение результативности может быть достигнуто применением алгоритмов второго порядка, таких как алгоритмы Ньютона, алгоритм сопряженных градиентов или алгоритм Левенберга-Марквардта [13, 14]. Считается, что, несмотря на эвристический характер, алгоритм Левенберга-Марквардта позволяет достигать наименьшей ошибки нейросети, причем, нередко с наименьшими временными затратами [15]. Алгоритм обеспечивает приемлемый компромисс между скоростью сходимости, присущей алгоритмам Ньютона, и стабильностью, свойственной алгоритму градиентного спуска.

Основными недостатками алгоритма являются: большой объем компьютерной памяти, который требуется для операций с якобианами; необходимость инвертировать большие матрицы (ранг матрицы, которую необходимо инвертировать, равен количеству настраиваемых параметров системы, то есть количеству весов нейросети).

Таким образом, с увеличением размерности нейросети каждый цикл тренировки будет требовать все большего количества вычислительных ресурсов. Рост необходимых вычислительных ресурсов экспоненциален, что приводит к нивелированию достоинств алгоритма при применении его для нейросетей большой размерности.

Особенностью алгоритма является то, что он предназначен исключительно для минимизации среднеквадратической ошибки. Другие целевые функции, такие как энтропия или «softmax», не могут быть использованы.

В нашем случае, применительно к математическим задачам энергетики, оба указанных недостатка не так существенны, так как информационно-управляющие комплексы, используемые в энергетике, наиболее эффективно работают с небольшими и средними нейросетями. Для данного класса задач большая размерность нейросетей приводит к снижению обобщающих способностей. Кроме того, минимизация среднеквадратической ошибки является идеальной целевой функцией для задач числового прогнозирования, поэтому особенность алгоритма никак не уменьшает возможностей его применения для задач прогнозирования энергопотребления.

Цель статьи. Статья посвящена исследованию и улучшению одного из наиболее эффективных алгоритмов обучения многослойных перцептронов – алгоритма Левенберга-Марквардта, для минимизации ошибки прогнозирования связного потребления электричества и тепла в коммунально-бытовом секторе. Во втором разделе статьи изложен алгоритм Левенберга-Марквардта. Третий раздел описывает два

эвристических метода улучшения поисковой способности алгоритма. Четвертый раздел представляет результаты краткого сравнительного анализа этих двух эвристик.

Алгоритм Левенберга-Марквардта. Алгоритм Левенберга-Марквардта является одним из приложений ньютоновской стратегии оптимизации. Основным выражением методов Ньютона является выражение:

$$p_k = -[H(w_k)]^{-1} g(w_k), \quad (1)$$

где p_k – направление, которое гарантирует достижение минимального для данного шага значения целевой функции;

$g(w_k)$ – значение градиента в точке последнего решения w_k ;

$H(w_k)$ – значение гессиана в точке последнего решения w_k .

При использовании алгоритма Левенберга-Марквардта точное значение гессиана $h(w)$ в (1) заменяется аппроксимированным значением $g(w)$, которое рассчитывается на основе содержащейся в градиенте информации с учетом некоторого регуляризационного фактора.

Для описания этого метода представим целевую функцию в виде, отвечающем существованию единственной обучающей выборки,

$$E(w) = \frac{1}{2} \sum_{i=1}^M d_i^2(w), \quad (2)$$

где $e_i = d_i(w) - d_i$.

При использовании обозначений

$$e(w) = \begin{bmatrix} e_1(w) \\ e_2(w) \\ \dots \\ e_M(w) \end{bmatrix}, \quad J(w) = \begin{bmatrix} \frac{\partial e_1}{\partial w_1} & \frac{\partial e_1}{\partial w_2} & \dots & \frac{\partial e_1}{\partial w_n} \\ \frac{\partial e_2}{\partial w_1} & \frac{\partial e_2}{\partial w_2} & \dots & \frac{\partial e_2}{\partial w_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial e_M}{\partial w_1} & \frac{\partial e_M}{\partial w_2} & \dots & \frac{\partial e_M}{\partial w_n} \end{bmatrix} \quad (3)$$

вектор градиента и аппроксимированная матрица гессиана, соответствующие целевой функции (2), определяются в виде

$$g(w) = J(w)^T e(w); \quad (4)$$

$$G(w) = J(w)^T J(w) + R(w), \quad (5)$$

где $R(w)$ – компоненты гессиана $H(w)$, содержащие высшие производные относительно w .

Сущность подхода Левенберга-Марквардта состоит в аппроксимации $R(w)$ с помощью регуляризационного фактора v_l , в котором переменная v , называемая параметром Левенберга-Марквардта, является скалярной величиной, изменяющейся в процессе оптимизации. Таким образом, аппроксимированная матрица гессиана на k -м шаге алгоритма приобретает вид

$$G(w_k) = J(w_k)^T J(w_k) + v_k I. \quad (6)$$

В начале процесса обучения, когда фактическое значение w_k еще далеко от искомого решения (велико значение вектора погрешности e), используется значение параметра v_k , намного превышающее собственное значение матрицы $J(w_k)^T J(w_k)$. В таком случае гессиан фактически подменяется регуляризационным фактором

$$G(w_k) \cong v_k I, \quad (7)$$

а направление минимизации выбирается по методу наискорейшего спуска

$$p_k = -\frac{g(w_k)}{v_k}. \quad (8)$$

По мере уменьшения погрешности и приближения к искомому решению величина параметра v_k понижается и первое слагаемое в формуле (5) начинает играть все более важную роль.

На эффективность алгоритма влияет грамотный подбор величины v_k . Слишком большое начальное значение v_k по мере прогресса оптимизации должно уменьшаться вплоть до нуля при достижении фактического решения, близкого к искомому. Известны различные способы подбора этого значения, но мы ограничимся описанием только одной оригинальной методики, предложенной Д. Марквардтом. Пусть значения целевой функции на k -м и $(k-1)$ -м шагах итерации обозначаются соответственно E_k и E_{k-1} , а значения параметра v на этих же шагах – v_k и v_{k-1} . Коэффициент уменьшения значения v обозначим r , причем $r > 1$. В соответствии с классическим алгоритмом Левенберга-Марквардта значение v изменяется по следующей схеме:

если $E\left(\frac{v_{k-1}}{r}\right) \leq E_k$, то принять $v_k = \frac{v_{k-1}}{r}$;

если $E\left(\frac{v_{k-1}}{r}\right) > E_k$ и $E(v_{k-1}) < E_k$, то принять $v_k = v_{k-1}$;

если $E\left(\frac{v_{k-1}}{r}\right) > E_k$ и $E(v_{k-1}) > E_k$, то увеличить последовательно m раз значение v до достижения $E(v_{k-1}r^m) \leq E_k$, одновременно принимая $v_k = v_{k-1}r^m$.

Такая процедура изменения значения v выполняется до момента, в котором так называемый коэффициент верности отображения q , рассчитываемый по формуле

$$q = \frac{E_k - E_{k-1}}{\Delta w_k^T g_k + 0,5 \Delta w_k^T G_k \Delta w_k}, \quad (9)$$

достигнет значения, близкого к единице. При этом квадратичная аппроксимация целевой функции имеет высокую степень совпадения с истинными значениями, что свидетельствует о близости оптимального решения. В такой ситуации регуляризационный фактор v_{kl} в формуле (6) может быть опущен ($v_k = 0$), процесс определения гессиана сводится к непосредственной аппроксимации первого порядка, а алгоритм Левенберга-Марквардта превращается в алгоритм Гаусса-Ньютона, характеризующийся квадратичной сходимостью к оптимальному решению.

Модификации алгоритма для прохождения локальных минимумов. Одной из особенностей алгоритма Левенберга-Марквардта является его чувствительность к локальным минимумам (как, впрочем, и всех алгоритмов локального поиска). Алгоритм Левенберга-Марквардта изменяет веса нейросети, используя квадратичную аппроксимацию поверхности ошибки. Данная аппроксимация помогает найти минимум достаточно быстро, но, с другой стороны, она увеличивает опасность нахождения локального экстремума на поверхности ошибки обучения.

В рассматриваемом классе задач поверхность ошибки часто включает в себя многочисленные плато и впадины, что делает проблему локальных минимумов одной из самых существенных преград для достижения максимальной эффективности. Для решения этой проблемы были разработаны и протестированы два эвристических подхода, которые позволяют избежать «залипания» поискового процесса в локальных минимумах.

Оригинальный алгоритм Левенберга-Марквардта не позволяет совершать шагов, которые увеличивают среднюю ошибку нейросети. Первая эвристика заставляет алгоритм совершать «рискованные» шаги по поверхности ошибки в случайном направлении с нарастающей величиной шага, пытаясь таким образом «выскочить» из локального минимума, чтобы затем продолжить продвижение к новому минимуму, используя правила оригинального алгоритма. После нескольких неудачных попыток, в нашем случае мы использовали 5 попыток, найденный минимум признается наименьшим, и алгоритм завершает свою работу.

Вышеуказанная эвристика была протестирована на пяти наборах данных. Наборы содержали от 3 до 5 входных числовых переменных и 1 выходную. Наборы представляли классы задач различного уровня сложности, которые требуется решать при прогнозировании энергопотребления. Все наборы данных были получены в результате предыдущих исследований [1]. Было разработано специальное программное обеспечение, реализующее вышеуказанную эвристику, а также обеспечивающее автоматизацию регистрации результатов экспериментов.

На первом этапе для каждого набора данных была найдена архитектура нейросети, а именно количество нейронов в скрытом слое, которая обеспечивает наименьшую ошибку. Архитектура была найдена путем полного перебора вариантов архитектур от 1 до 20 скрытых нейронов.

На втором этапе было проведено 50 тестовых прогонов: по 10 тренировок сетей с различной инициализацией весов для каждого набора данных без применения эвристики.

На третьем этапе было проведено еще 50 тестовых прогонов: по 10 тренировок сетей с различной инициализацией весов для каждого набора данных с применением эвристики.

Результаты тестовых прогонов приведены в табл. 1.

Таблица 1

Результаты тестовых прогонов эвристики «случайного шага».

	СКО оригинального алгоритма			СКО с применением эвристики «случайного шага»		
	Мин	Макс	Сред.	Мин	Макс	Сред.
Набор данных 1	5,16	13,54	6,12	3,93	12,25	6,23
Набор данных 2	0,13	0,28	0,16	0,11	0,25	0,13
Набор данных 3	184,26	308,58	231,14	208,12	321,43	240,85

Набор данных 4	13,97	18,30	15,81	11,60	18,47	16,11
Набор данных 5	6,51	10,83	9,01	4,48	11,6	8,63

В результате, очевидно, что эвристика повышает вероятность нахождения удачных решений. Однако, следует отметить, что в некоторых случаях она может приводить к утере уже найденного наилучшего решения с последующим «застреванием» в позже найденном локальном минимуме. Борьба с этим недостатком можно увеличением количества случайных шагов, что, в свою очередь, приводит к увеличению вычислительных ресурсов. Как было указано ранее, рекомендуемые нами 5 попыток обеспечивают оптимальный компромисс для нашей предметной области.

Вторая эвристика состоит в том, чтобы при достижении локального минимума изменять веса нейросети следующим образом:

$$w_{i,j} = w_{i,j} + \theta w_{i,j}, \quad (10)$$

где $w_{i,j}$ – весовой коэффициент соединения от i -го нейрона к j -му;

θ – случайное число из диапазона $\underline{\theta} \leq \theta \leq \bar{\theta}$.

Таким образом, при достижении локального минимума веса сети изменяются случайным образом на величину, меньшую или равную θ , в большую или меньшую сторону. Величины $\underline{\theta}$ и $\bar{\theta}$ были подобраны эмпирически в результате экспериментов. Были опробованы различные значения в интервале $[-0,5; 0,5]$ с шагом 0,01. В результате оптимальное значение $\underline{\theta}$ было установлено равным $-0,05$, а оптимальное значение $\bar{\theta}$ было установлено равным $0,05$.

Большие значения часто существенно нивелируют предыдущий процесс тренировки, и нейросеть вынуждена как бы тренироваться заново, то есть эвристика вырождается в набор тестовых прогонов с различной начальной инициализацией весов. Меньшие значения не позволяют выйти нейросети из локального минимума, и она снова и снова возвращается к первоначально найденному минимуму, не обеспечивая выигрыша от применения эвристики.

Для других предметных областей, предложенные нами границы диапазона могут быть слишком большими или слишком маленькими, что зависит от типичной поверхности ошибки для конкретной предметной области.

После «встряски» весов обучение продолжается по правилам оригинального алгоритма. В случае, если новый минимум не достигнут, проводится повторное случайное изменение. Эмпирически установлено, что число случайных изменений не следует устанавливать больше 3.

Для проверки эффективности второй эвристики были использованы те же наборы данных, что и для первой. Также были использованы те же оптимальные архитектуры нейросетей, которые были найдены на предыдущем этапе.

Аналогично, было проведено 50 тестовых прогонов: по 10 тренировок сетей с различной инициализацией весов для каждого набора данных с применением эвристики.

Результаты тестовых прогонов приведены в табл. 2.

Таблица 2

Результаты тестовых прогонов эвристики «встряски весов».

	СКО оригинального алгоритма			СКО с применением эвристики «встряски весов»		
	Мин	Макс	Сред.	Мин	Макс	Сред.
Набор данных 1	5,16	13,54	6,12	4,02	13,12	6,54
Набор данных 2	0,13	0,28	0,16	0,27	0,28	0,30
Набор данных 3	184,26	308,58	231,14	186,25	338,30	228,56
Набор данных 4	13,97	18,30	15,81	10,98	17,01	14,77
Набор данных 5	6,51	10,83	9,01	5,18	10,69	8,33

В результате, очевидно, что вторая эвристика, как и первая, с одной стороны, значительно повышает вероятность нахождения глобального минимума, а с другой, обладает тем же недостатком, что и первая – в некоторых случаях может приводить к утере найденного наилучшего решения. Кроме того, она требует подбора эмпирическим образом границ диапазона изменения параметра θ .

Выводы. Предложены две эвристики, позволяющие предотвратить «залипание» поискового процесса в локальных минимумах на поверхности ошибки в процессе тренировки нейросети с использованием алгоритма Левенберга-Марквардта, и, таким образом, улучшающие поисковую способность алгоритма.

Приведены результаты экспериментов, сравнивающих результативность предложенных эвристик. Оба подхода приводят к сравнимым результатам, однако второй подход требует больше вычислительных ресурсов и немного сложнее в программной реализации. Для второй эвристики экспериментальным путем найдены граничные значения диапазонов настроечных параметров для данной предметной области.

В виду универсальности подходов обе эвристики могут использоваться для улучшения других алгоритмов тренировки нейросетей не только для задач прогнозирования энергоресурсов, но и в других предметных областях.

Список литературы: 1. *Вороновский Г.К.* Усовершенствование практики оперативного управления крупными теплофикационными системами в новых экономических условиях. – Х.: Изд-во «Харьков», 2002. – 240 с. 2. *Вороновский Г.К., Махотило К.В.* Совершенствование алгоритмической базы эволюционного синтеза нейросетевых моделей потребления электрической энергии в коммунально-бытовом секторе // Вестник Национального технического университета "ХПИ". – Х.: НТУ "ХПИ", 2001. – Вып. 10. – С. 427–431. 3. *Khotanzad A., Afkhami-Rohani R., Maratukulam D.* ANNSTLF – Artificial Neural Network Short-Term Load Forecaster – Generation Three // IEEE Transactions on Power Systems – 1998. – Vol. 13. – № 4. – P. 1413 – 1422. 4. *Rumelhart D.E., Hinton G.E., Williams R.J.* Learning Internal Representations by Error Propagation, In Parallel Distributed Processing. – Cambridge, MA: MIT Press. – Vol. 1. – P. 318–362. 5. *Werbos P. J.* Back-Propagation: Past and Future / Proceeding of International Conference on Neural networks. – San Diego, CA. – 1988. – Vol. 1. – P. 343–354. 6. *Samad T.* Back-Propagation Improvements Based on Heuristic Arguments / Proceedings of International Joint Conference on Neural networks, Washington – 1990. – Vol. 1. – P. 565–568. 7. *Solla S. A., Levin E., Fleisher M.* Accelerated Learning in Layered Neural Networks // Complex Systems, 1988. – Vol. 2. – P. 625–639. 8. *Sperdui A., Starita A.* Speed Up Learning and Network Optimization with Extended Back-Propagation // Neural Networks. – 1993. – Vol. 6. – P. 365–383. 9. *Van Ooten A., Nienhuis B.* Improving the Convergence of the Back-Propagation Algorithm // Neural Networks. – 1992. – Vol. 5. – P. 465–471. 10. *Mimiani A.A., Williams R.D.* Acceleration of Back-Propagation Through Learning Rate and Momentum Adaptation / Proceedings of International Joint Conference on Neural Networks. – San Diego, CA, 1990. – Vol. 1. – P. 676–679. 11. *Jacobs R.A.* Increased Rates of Convergence Through Learning Rate Adaptation // Neural Networks. – 1988. – Vol. 1. – № 4. – P. 295–308. 12. *Salvetti A., Wilamowski B.M.* Introducing Stochastic Process Within the Backpropagation Algorithm for Improved Convergence / Presented at Annie'94 – Artificial Neural Networks in Engineering, St. Louis, Missouri, USA, 1994. 13. *Battiti R.* First- and Second-Order Methods for Learning: Between Steepest Descent and Newton's Method // Neural Computation. – 1992. – Vol. 4. – № 2. – P. 141–166. 14. *Charalambous C.* Conjugate Gradient Algorithm for Efficient Training of Artificial Neural Networks // IEEE Proceedings. – 1992. – Vol. 139. – № 3. – P. 301–310. 15. *Hagan M.T., Menhaj M.* Training Feedforward Networks with the Marquardt Algorithm // IEEE Transactions on Neural Networks. – 1994. – Vol. 5. – № 6. – P. 989–993.

Поступила в редакцию 23.09.2005