



УДК 004.032.26

© 2009

Е. В. Бодянский, С. В. Попов

Искусственная нейронная сеть для нелинейной декомпозиции функций

(Представлено членом-корреспондентом НАН Украины Ю. Г. Стояном)

Запропоновано нову архітектуру штучної нейронної мережі для розв'язання задачі нелинійної декомпозиції функцій. Використано спадний підхід, що не потребує апріорної інформації про властивості аналізованої функції. Можливості запропонованого методу продемонстровані на синтетичних тестових функціях і підтверджені розв'язанням реальної задачі.

1. При решении практических задач часто встречается ситуация, когда исследуемая переменная зависит сложным, заранее не известным образом от множества различных факторов x_1, x_2, \dots, x_n . Понимание природы влияния каждого из факторов может значительно облегчить решение многих задач анализа, проектирования и принятия решений и поэтому является критически важным для инженеров, проектировщиков, управляющего персонала и т. д. В достаточно простых ситуациях, когда процесс хорошо изучен, линеен или известны физические принципы функционирования системы, влияние различных факторов может быть выражено аналитически или легко идентифицировано с помощью линейных моделей, или вычислено на основе физических принципов. Однако для анализа более сложных систем эти подходы становятся неприменимыми, так как влияния факторов становятся нелинейными и взаимозависимыми. В общем случае зависимость между y и x_1, x_2, \dots, x_n имеет вид произвольной нелинейной функции

$$y = g(x_1, x_2, \dots, x_n) + \xi, \quad (1)$$

где ξ — сигнал помехи.

Если имеется достаточный объем совместных наблюдений x_1, x_2, \dots, x_n и y , проблема оценивания $g(\cdot)$ может быть решена с помощью методов нелинейной идентификации [1], среди которых искусственные нейронные сети [2] и нейрофаззи системы [3] являются наиболее подходящими вследствие своих универсальных аппроксимирующих возможностей [4]. Однако, если физические принципы функционирования системы неизвестны, т. е. структура модели не может быть выбрана однозначным образом, результатом моделирования

является “черный ящик”, параметры которого не имеют физического смысла. Кроме того, в нелинейном случае сложно выполнить независимый анализ чувствительности для входов x_i ($i = 1, \dots, n$) из-за их возможной взаимосвязи: форма влияния некоторого x_i может значительно зависеть от значений других входов.

В качестве примера рассмотрим зависимость такого вида:

$$y = \operatorname{tg}^{-1}(10x_1 + \sin(5x_2)) + \xi, \quad (2)$$

в которой влияние x_2 на y при $x_1 = 0$ является практически синусоидальным. Но, например, при $x_1 = 1$ влияние x_2 на y почти полностью отсутствует.

В общем случае допустим, что функция (1) представима композицией произвольных непрерывных функций, определенных на компактном множестве

$$y = f(f_1(x_1), f_2(x_2), \dots, f_n(x_n)) + \xi. \quad (3)$$

Доказано, что такое представление существует для любой непрерывной функции $g(\cdot)$, так как всегда существует частный случай (3), где $f(\cdot) \equiv g(\cdot)$, $f_1(x_1) \equiv x_1$, \dots , $f_n(x_n) \equiv x_n$. Декомпозиция функции (1) в виде (3) дает возможность разделения независимых влияний входных переменных, описываемых функциями $f_i(x_i)$, от их взаимозависимого влияния, описываемого функцией $f(\cdot)$.

Задача нелинейной декомпозиции функций уже привлекала внимание исследователей, однако она была решена лишь частично. Подавляющее большинство предложенных методов предназначено только для номинальнозначных (в частности, двоичных) функций [5] или имеют множество ограничений, делающих их малоприменимыми на практике [6]. Известен ряд нейросетевых методов декомпозиции сложных функций на множество более простых функций [7–9], однако они используют подход “снизу вверх”: сначала отдельно создаются и обучаются подсети, реализующие простые функции, а затем они объединяются вместе для реализации исходной, более сложной функции. Такой подход требует значительных априорных знаний и наличия выборок для обучения подсетей, которые зачастую недоступны на практике.

В настоящей работе предлагается специализированная архитектура искусственной нейронной сети, предназначенной для декомпозиции действительных функций, заданных их реализациями, на множество функций (3), с использованием нисходящего подхода, для которого требуется только обучающая выборка, сгенерированная реальной функцией $g(\cdot)$, и не требуются априорные знания о ее свойствах.

2. Архитектура нейронной сети для декомпозиции функций. Предложенная нейронная сеть имеет архитектуру с прямой передачей информации, представленную на рис. 1. В кружках, обозначающих нейроны, схематически показаны графики их активационных функций.

Сеть имеет три скрытых слоя: первый и третий скрытые слои состоят из нейронов с нелинейными активационными функциями $\psi^{[1]}$ и $\psi^{[3]}$ соответственно (мы использовали гиперболический тангенс в обоих слоях). Нейроны второго скрытого слоя и выходного слоя имеют линейные активационные функции. Первые два скрытых слоя формируют набор из n подсетей, обрабатывающих только сигналы соответствующих входов. Затем их выходные сигналы подаются на третий скрытый слой, который вместе с выходным слоем аппроксимирует функцию $f(\cdot)$. Таким образом, подсети играют роль универсальных аппроксиматоров независимых функций $f_i(x_i)$, обучаемых внутри общей сети. Задачей второго

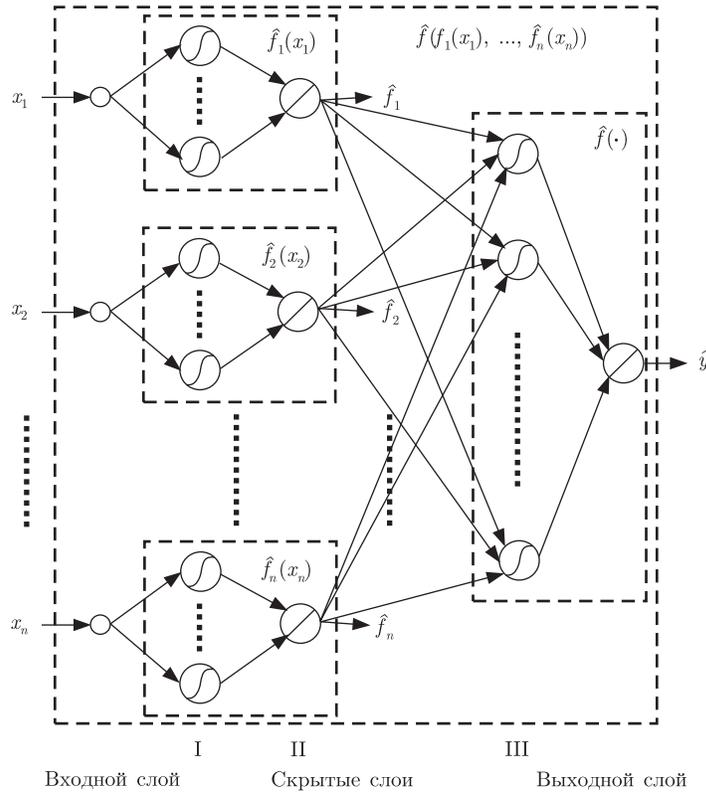


Рис. 1. Архитектура предлагаемой нейронной сети

скрытого слоя является выделение оценок \hat{f}_i истинных сигналов f_i , в связи с чем он был назван нами слоем наблюдения.

Отображение, реализуемое сетью, может быть выражено следующим образом:

$$\hat{y} = w_0^{[4]} + \sum_{j=1}^{n^{[3]}} \left(w_j^{[4]} \psi^{[3]} \left(w_{j0}^{[3]} + \sum_{i=1}^n w_{ji}^{[3]} \hat{f}_i(x_i) \right) \right), \quad (4)$$

где $n^{[3]}$ — количество нейронов в третьем скрытом слое; $w_j^{[4]}$ — синаптические веса выходного нейрона сети и $w_0^{[4]}$ — его параметр смещения; $w_{ji}^{[3]}$ — синаптические веса j -го нейрона третьего скрытого слоя; $w_{j0}^{[3]}$ — соответствующий параметр смещения.

Функции $f_i(x_i)$ аппроксимируются соответствующими подсетями таким образом:

$$\hat{f}_i(x_i) = w_0^{[2,i]} + \sum_{l=1}^{n^{[1,i]}} (w_l^{[2,i]} \psi^{[1]}(w_{l0}^{[1,i]} + w_{l1}^{[1,i]} x_i)), \quad (5)$$

где $n^{[1,i]}$ — количество нейронов первого скрытого слоя i -й подсети; $w_l^{[2,i]}$ — синаптические веса выходного нейрона i -й подсети и $w_0^{[2,i]}$ — его параметр смещения; $w_{l1}^{[1,i]}$ — единственный синаптический вес l -го нейрона первого скрытого слоя i -й подсети; $w_{l0}^{[1,i]}$ — соответствующий параметр смещения.

Хотя соотношение (4) и сходно с формулировкой теоремы Колмогорова о представлении непрерывных функций нескольких переменных в виде суперпозиций непрерывных функций одной переменной и сложения [10], взаимосвязь которой с теорией искусственных нейронных сетей была отмечена в [11] и позже изучалась многими исследователями, наш подход имеет существенные отличия: аппроксимации обоих типов функций $\hat{f}(\cdot)$ и $\hat{f}_i(x_i)$ зависят от $g(\cdot)$, более того, $\hat{f}_i(x_i)$ не имеют ограничения на монотонность, а предназначены для описания реально существующих взаимосвязей x_i и y и, в принципе, могут иметь любую форму.

Нейронная сеть используется следующим образом. Сначала имеющаяся выборка наблюдений $\{(x_1(k), x_2(k), \dots, x_n(k)), y(k)\}$, $k = 1, \dots, N$ (здесь N — длина выборки) используется для обучения сети. Основное требование к обучающей выборке — быть репрезентативной, т. е. охватывать весь диапазон возможных значений x_i .

Обучение производится по стандартному квадратичному критерию

$$E = \sum_{k=1}^N e^2(k) = \sum_{k=1}^N (y(k) - \hat{y}(k))^2, \quad (6)$$

и поскольку сеть имеет архитектуру с прямой передачей информации, для вычисления локальных градиентов может быть использована стандартная процедура обратного распространения ошибки. Затем для уточнения синаптических весов сети применяется алгоритм Левенберга–Марквардта в пакетном режиме. Эта тема хорошо освещена в литературе [12], поэтому мы не будем здесь на ней останавливаться.

Наблюдательные выходы \hat{f}_i не используются в процессе обучения, так как для них отсутствуют обучающие сигналы. По этой же причине оценки функций $\hat{f}_i(x_i)$ могут подвергаться произвольным линейным преобразованиям (симметричному отражению, масштабированию и сдвигу). Поэтому для дальнейшего анализа может быть удобно масштабировать сигналы \hat{f}_i на интервал $[0, 1]$ путем применения преобразования

$$\tilde{f}_i = \frac{\hat{f}_i - \hat{f}_{i,\min}}{\hat{f}_{i,\max} - \hat{f}_{i,\min}}, \quad (7)$$

где $\hat{f}_{i,\min}$ и $\hat{f}_{i,\max}$ — минимальное и максимальное значения \hat{f}_i соответственно.

Затем обученные подсети выделяются из общей сети и могут использоваться самостоятельно в качестве аппроксиматоров соответствующих $f_i(x_i)$. С их помощью возможно, например, проводить визуальный анализ графика функции $\hat{f}_i(x_i)$, анализ первой и второй разности функции $\hat{f}_i(x_i)$ для нахождения ее особых точек (экстремумов и точек перегиба) и т. д.

3. Экспериментальные результаты. Возможности предложенного подхода продемонстрированы следующими результатами имитационного моделирования, основанными на искусственно составленных функциях, что позволяет вычислить ошибки аппроксимации для всех $f_i(x_i)$. Для удобства анализа все тестовые функции определены на гиперкубе $x_i \in [-1, +1]$. Выборки содержат по 10 000 точек, подчиняющихся n -мерному равномерному закону распределения на указанном гиперкубе, в каждой точке вычислено значение тестовой функции, 90% данных используется для обучения сети, 10% — для проверки. Помеха ξ во всех экспериментах является нормально распределенной случайной величиной с нулевым математическим ожиданием и среднеквадратичным отклонением 0,01. Количество нейронов в скрытых слоях сети установлено $n^{[1,i]} = 10$ и $n^{[3]} = 25$. После обучения

сети аппроксимации функций $\widehat{f}_i(x_i)$ извлекаются из нее путем подачи линейно растущего сигнала на соответствующие входы и наблюдения за выходными сигналами \widehat{f}_i . И тестовые функции $f_i(x_i)$, и их аппроксимации $\widehat{f}_i(x_i)$ масштабируются на интервал $[0, 1]$ для возможности их сравнения и вычисления ошибок аппроксимации.

Ошибки аппроксимации по каждой $f_i(x_i)$ для различных тестовых функций приведены в табл. 1. Полученные экспериментальные результаты показали широкие возможности по нелинейной декомпозиции функций с помощью предложенной сети и высокую точность аппроксимации независимых функций $f_i(x_i)$, что также подтверждается решением реальных задач. В [13] нами был предложен метод выявления зависимости энергопотребления от температуры воздуха на основе анализа архивных выборок данных энергопотребления и погодных условий с использованием модификации рассматриваемой здесь архитектуры. Метод был успешно применен для анализа двух энергосистем, расположенных в различных частях света (Северной Америке и Европе) со значительно отличающимися климатическими условиями. Выявленные зависимости согласовывались с теоретически обоснованной кривой температурной зависимости. Нами также было показано, что в ряде исследований других авторов использовались неверные аппроксимации температурных зависимостей, что привело к росту ошибок в задачах краткосрочного прогнозирования потребления электроэнергии. Более глубокий анализ выявленных зависимостей позволяет энергоснабжающим компаниям проводить более точное прогнозирование потребления и потерь электроэнергии, лучше планировать развитие энергосистем и более эффективно эксплуатировать имеющиеся энергетические сети и системы.

Таким образом, предложенную архитектуру можно рассматривать в качестве модульной нейронной сети, где n модулей образованы первыми двумя скрытыми слоями и еще один модуль образован двумя последними слоями сети. Однако в отличие от традиционного применения модульных сетей, где каждый модуль создается и обучается независимо для решения своей конкретной задачи, а затем они объединяются в общую сеть, в нашем подходе сеть работает как монолитная архитектура и только после обучения отдельные ее части могут быть использованы независимо как модули. Эти обученные подсети могут использоваться не только для анализа влияния отдельных входов, но и для предварительной обработки сигналов, прогнозирования, экстраполяции и решения других задач.

В настоящей работе мы описали наиболее общий случай задачи нелинейной декомпозиции, когда отсутствует априорная информация о структуре функции $g(\cdot)$. Однако во многих практических приложениях такая информация в какой-то мере уже имеется или может быть получена путем анализа обучающей выборки [14]. Такая информация может быть легко использована при построении архитектуры предлагаемой сети путем выбора соответствующей структуры подсетей и связей между. Следует обратить внимание, что подход остается нисходящим: сначала обучение всей сети на основе имеющейся реализации $g(\cdot)$, затем извлечение подсетей для независимого использования.

Таблица 1

Функция $g(\cdot)$	NRMSE ₁	NRMSE ₂	NRMSE ₃
$y = \cos(x_1 + x_2 + x_3) + \xi$	0,006565	0,007777	0,008176
$y = \text{tg}^{-1}(10x_1 + \sin(5x_2)) + \xi$	0,110080	0,009599	—
$y = \sin(5x_1)x_2^2 + 5x_3 + \xi$	0,015760	0,362474	0,029165
$y = \sin(\text{tg}(1, 5x_1) + x_2^2 + x_3 + x_1x_3) + \xi$	0,396391	0,035662	0,056657

Несмотря на полученные положительные результаты, необходимо отметить, что декомпозиция (3) не уникальна для любой функции (1). Однако, когда в наших экспериментах с представленными и другими тестовыми функциями ошибка обучения была мала (NRMSE порядка 10^{-2} или ниже), подсети всегда правильно аппроксимировали истинные независимые функции. Этот вопрос все еще требует дополнительного изучения и является предметом наших дальнейших исследований.

1. *Nelles O.* Nonlinear system identification. – Berlin: Springer, 2001. – 785 p.
2. *Haykin S.* Neural networks. A comprehensive foundation. – Upper Saddle River: Prentice Hall, 1999. – 842 p.
3. *Jang J.-S.R., Sun C.-T., Mizutani E.* Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence. – Upper Saddle River: Prentice Hall, 1997. – 640 p.
4. *Tikk D., Koczy L. T., Gedeon T. D.* A survey on universal approximation and its limits in soft computing techniques // Intern. J. of Approximate Reasoning. – 2003. – **33**, No 2. – P. 185–202.
5. *Zupan B., Bratko I., Bohanec M., Demsar J.* Function decomposition in machine learning // Machine Learning and Its Applications / Ed. by G. Paliouras, V. Karkaletsis, C. D. Spyropoulos. – Berlin, Heidelberg: Springer, 2001. – P. 71–101.
6. *Demsar J., Zupan B., Bohanec M., Bratko I.* Constructing intermediate concepts by decomposition of real functions // Proc. Europ. Conf. on Machine Learning, ECML – **97**. – Prague, April 1997. – P. 93–107.
7. *Vaccari D. A., Wojciechowski E.* Neural networks as function approximators: teaching a neural network to multiply // Proc. IEEE World Congress on Computational Intelligence. – 1994. – **4**. – P. 2217–2221.
8. *Jansen W. J., Diepenhorst M., Nijhuis J. A. G., Spaanenburg L.* Assembling engineering knowledge in a modular multi-layer perception neural network // Proc. Int. Conf. on Neural Networks. – 1997. – **1**. – P. 232–237.
9. *Lu X.-J., Li H.-X.* Sub-domain intelligent modeling based on neural networks // Proc. IEEE World Congress on Computational Intelligence. – 2008. – P. 445–449.
10. *Колмогоров А. Н.* О представлении непрерывных функций нескольких переменных в виде суперпозиции непрерывных функций одного переменного и сложения // Докл. АН СССР. – 1957. – **114**, № 5. – С. 953–956.
11. *Hecht-Nielsen R.* Kolmogorov's mapping neural network existence theorem // Proc. of the Intern. Conf. on Neural Networks. – New York, 1987. – III. – P. 11–14.
12. *Shepherd A. J.* Second-order methods for neural networks (fast and reliable training methods for multi-layer perceptrons). – London: Springer, 1997. – 145 p.
13. *Bodyanskiy Ye., Popov S., Rybalchenko T.* Feedforward neural network with a specialized architecture for estimation of the temperature influence on the electric load // Proc. 4th Int. IEEE Conf. Intelligent Systems. – Varna, Bulgaria, September 6–8, 2008. – **1**. – P. 7–14–7–18.
14. *Chambless B., Lendaris G. G., Zwick M.* An information theoretic methodology for prestructuring neural networks // Proc. Intern. J. Conf. on Neural Networks. – 2001. – **1**. – P. 365–369.

Харьковский национальный
университет радиоэлектроники

Поступило в редакцию 15.04.2009

E. V. Bodyansky, S. V. Popov

The artificial neural network for a nonlinear decomposition of functions

A novel neural network architecture is proposed to solve the nonlinear function decomposition problem. The top-down approach that does not require an a priori knowledge about the function's properties is applied. Abilities of the proposed method are demonstrated using synthetic test functions and confirmed by solving a real problem.