

Е.В. Бодянский, А.А. Дейнеко, Н.А. Тесленко, М.А. Шаламов
**ЭВОЛЮЦИОННАЯ КАСКАДНАЯ НЕЙРОННАЯ СЕТЬ ДЛЯ
ПОСЛЕДОВАТЕЛЬНОГО АНАЛИЗА ГЛАВНЫХ
КОМПОНЕНТ И АЛГОРИТМ ЕЁ ОБУЧЕНИЯ**

Анотація. Розглянуто задачу аналізу головних компонент у реальному часі за допомогою нейронної мережі, вулами якої є нейрони Оя. Введено оптимальний за швидкодією алгоритм навчання синаптичних ваг та процедура еволюції її архітектури, що дозволяє знайти оптимальну кількість нейронів у мережі.

Введение

Во многих задачах, связанных с интеллектуальным анализом данных (Data Mining, Text Mining, Web Mining), распознаванием образов, обработкой изображений, кодированием информации и т.п., возникает проблема сжатия больших массивов данных, для решения которой широко используется метод главных компонент, состоящий в проецировании векторов данных $x(1), x(2), \dots, x(k), \dots, x(k) \in R^n$ из исходного n -мерного пространства в m -мерное пространство пониженной размерности ($m < n$), именуемое пространством главных компонент. С математической точки зрения метод сводится к поиску системы w_1, w_2, \dots, w_m n -мерных ортогональных собственных векторов корреляционной матрицы центрированных относительно среднего исходных данных, при этом $w_1 = (w_{11}, w_{12}, \dots, w_{1n})^T$ соответствует наибольшему собственному значению λ_1 корреляционной матрицы, w_2 - второму по величине собственному значению λ_2 и т.д. Иначе говоря, проблема сводится к поиску решений матричного уравнения:

$$(R - \lambda_j I) w_j = 0$$

таких, что $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m \geq 0$ и $\|w_j\| = 1$. Здесь

$$R = \frac{1}{N} \sum_{k=1}^N (x(k) - \bar{x})(x(k) - \bar{x})^T = \frac{1}{N} \sum_{k=1}^N \tilde{x}(k) \tilde{x}^T(k)$$

– $(n \times n)$ – корреляционная матрица данных,

$$\bar{x} = \frac{1}{N} \sum_{k=1}^N x(k)$$

– оценка среднего, I – $(n \times n)$ – единичная матрица.

Данная задача, известная также как алгебраическая проблема собственных значений, или преобразование Карунена-Лоэва, хорошо изучена и решена в предположении, что исходная выборка наблюдений $x(1), x(2), \dots, x(N)$, подлежащая обработке, задана априори и её характеристики не меняются с течением времени. Если же данные поступают на обработку последовательно в реальном времени, их объем заранее неизвестен, а сам объект, генерирующий данные, нестационарен, традиционные алгоритмы, реализующие метод главных компонент, становятся неэффективными и на первый план выходят адаптивные процедуры, основанные на нейросетевых технологиях [1,2].

Постановка задачи

Для нахождения первой главной компоненты Э. Оя [3, 4] было предложено правило самообучения адаптивного линейного ассоциатора

$$y(k) = w^T(k) \tilde{x}(k),$$

минимизирующее энергетическую функцию (критерий обучения) вида

$$E(k) = \frac{1}{2} \|\tilde{x}(k) - \hat{x}(k)\|^2, \quad (1)$$

где $w(k)$ – текущая оценка синаптических весов, представляющая собой, по сути, оценку собственного вектора w_1 , $y(k)$ – текущая оценка первой главной компоненты,

$$\hat{x}(k) = w(k) y(k)$$

– восстановленная оценка входного сигнала $\tilde{x}(k)$.

С учетом того, что градиент критерия (1) по настраиваемым весам имеет вид

$$\nabla_w E(k) = -(\tilde{x}(k) - \hat{x}(k))y(k) = -(x(k) - w^T(k)y(k))y(k),$$

правило Оя для первой главной компоненты может быть записано в виде

$$w_1(k+1) = w_1(k) + \eta(k)y_1(k)(\tilde{x}(k) - w_1(k)y_1(k)), \quad (2)$$

где $\eta(k)$ параметр шага, определяющий скорость сходимости алгоритма и выбираемый в соответствии с условиями А.Дворецкого. Таким образом, правило (2) представляет собой, по сути, процедуру стохастической аппроксимации, характеризующуюся низкой скоростью обучения и непригодную для работы в условиях нестационарности.

В связи с этим в [5,6] было введено модифицированное правило Оя вида

$$\begin{cases} w_1(k+1) = w_1(k) + r^{-1}(k)y_1(k)(\tilde{x}(k) - w_1(k)y_1(k)), \\ r(k) = \alpha r(k-1) + y_1^2(k), 0 \leq \alpha \leq 1, \end{cases} \quad (3)$$

обладающее повышенным быстродействием и дополнительными сглаживающими свойствами, обеспечиваемыми с помощью варьируемого параметра α .

Так, при $\alpha=0$, процедура (3) совпадает по своим свойствам с оптимальным по быстродействию алгоритмом Качмажа–Уидроу–Хоффа [7], при $\alpha=1$ приходим к адаптивному алгоритму идентификации на основе стохастической аппроксимации [8]. Таким образом, обработку информации следует начинать с малых значений α , обеспечивая тем самым высокое быстродействие, далее увеличением α можно добиться сходимости к оптимальным значениям оцениваемых характеристик.

На рис. 1 приведена схема обучения адаптивного линейного ассоциатора с помощью правила (3), а на рис. 2 – условное обозначение модифицированного нейрона Оя.

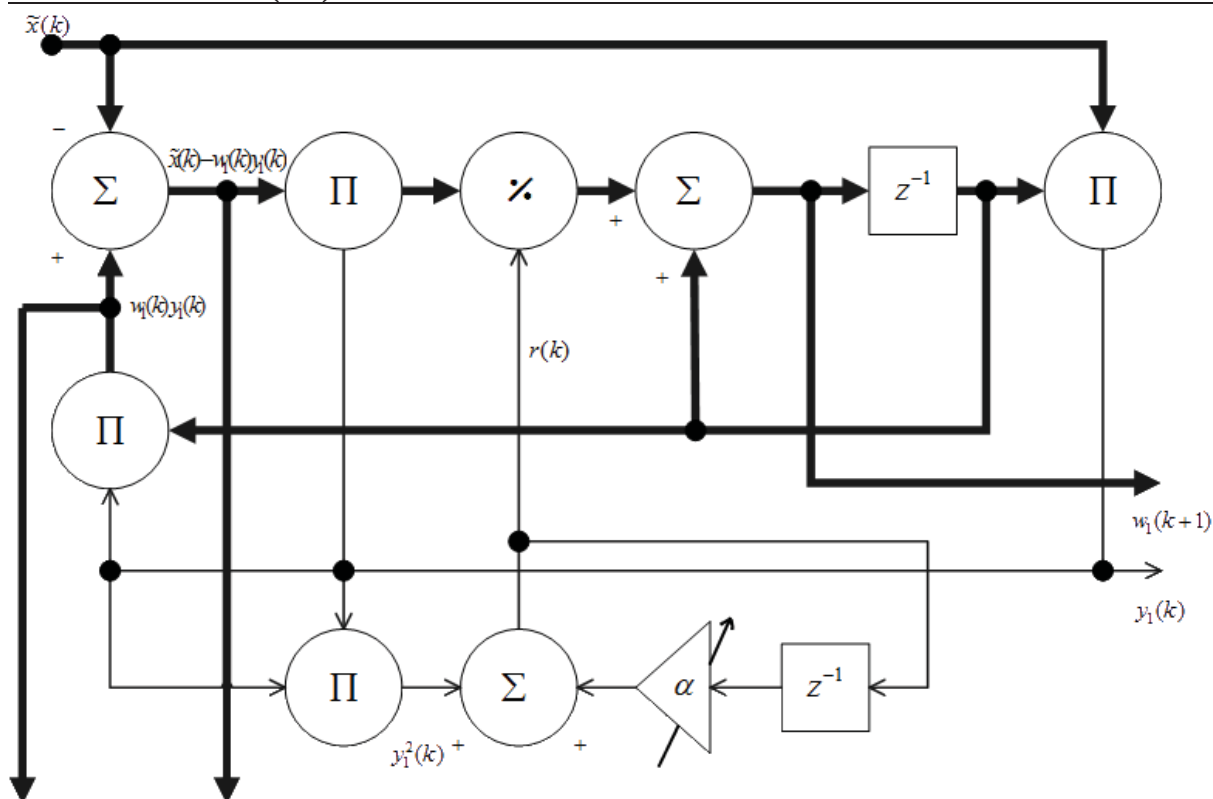


Рисунок 1 – Модифицированный нейрон Оя с контуром обучения

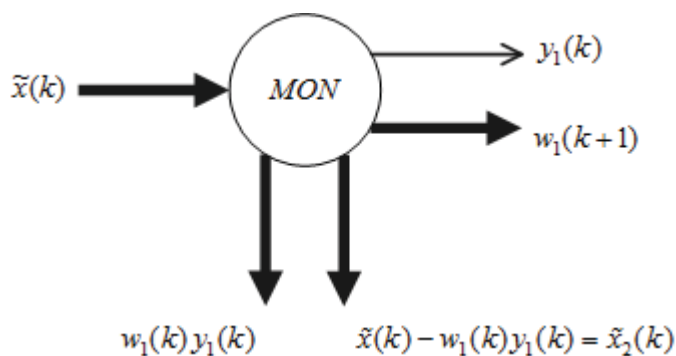


Рисунок 2 – Модифицированный нейрон Оя

Поскольку применение правила Оя подразумевает, что исходные данные центрированы относительно среднего, следует предусмотреть предварительную обработку данных, которая может быть реализована с помощью соотношения

$$\begin{cases} \bar{x}(k) = \bar{x}(k-1) + \frac{1}{k}(x(k) - \bar{x}(k-1)), \\ \tilde{x}(k) = x(k) - \bar{x}(k), \end{cases} \quad (4)$$

где первое выражение, рекуррентно вычисляющее среднее, является в то же время оптимальным алгоритмом самообучения Я.З. Цыпкина[9].

На рис. 3 приведена схема центрирования исходных данных с помощью соотношений (4)

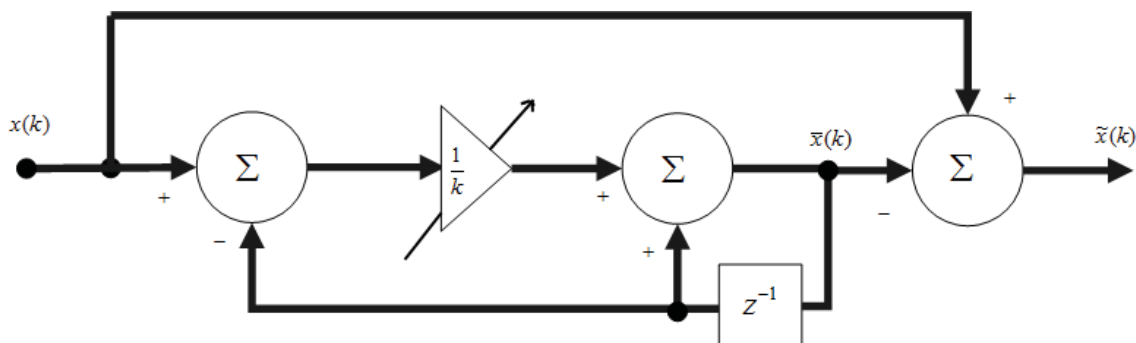


Рисунок 3 – Блок центрирования данных

Для вычисления m главных компонент можно воспользоваться нейронной сетью Т. Сэнгера [10], образованной нейронами Оя, последовательно вычисляющими первую, вторую и последующие компоненты и собственные векторы. В [11] в качестве узлов этой сети было предложено использовать модифицированные нейроны Оя, приведенные на рис. 1,2. При этом алгоритм обучения нейронной сети в целом может быть записан в виде

$$\begin{cases} w_j(k+1) = w_j(k) + r_j^{-1}(k) y_j(k) (\tilde{x}_j(k) - w_j(k) y_j(k)), j = 1, 2, \dots, m, \\ r_j(k) = \alpha r_j(k-1) + y_j^2(k), 0 \leq \alpha \leq 1, \\ \tilde{x}_j(k) = \tilde{x}_{j-1}(k) - w_{j-1}(k) y_{j-1}(k), \tilde{x}_1(k) = \tilde{x}(k), \\ y_j(k) = w_j^T(k) \tilde{x}_j. \end{cases} \quad (5)$$

Из (5) следует, что первый нейрон сети вычисляет первую главную компоненту $y_1(k)$. Далее из $\tilde{x}(k)$ вычитается его проекция на первый собственный вектор $w_1(k)$

$$\tilde{x}(k) - w_1(k) y_1(k) = \tilde{x}_2(k)$$

и вычисляется $y_2(k)$, являющаяся первой главной компонентой $\tilde{x}_2(k)$ или, что то же самое, второй главной компонентой исходного сигнала $\tilde{x}(k)$. Третья главная компонента вычисляется путем проекции каждого исходного вектора $\tilde{x}(k)$ на первые два собственных вектора, вычитания этой проекции из $\tilde{x}(k)$ и нахождения первой главной компоненты $\tilde{x}_3(k)$, являющейся третьей главной компонентой $\tilde{x}(k)$. Ос-

тальные главные компоненты по m -ую включительно вычисляются рекурсивно согласно описанной стратегии.

При этом открытым остается вопрос: сколько же компонент необходимо вычислять в реальном времени, чтобы обеспечить приемлемый уровень компрессии исходного сигнала при минимальной потере информации? Для ответа на этот вопрос целесообразно воспользоваться концепцией эволюционных коннекционистских систем [12], подразумевающей не только настройку синаптических весов нейронной сети, но и её архитектуры непосредственно в процессе обучения.

Эволюционная каскадная нейронная сеть

Архитектура каскадной нейронной сети для последовательного анализа главных компонент приведена на рис.4. Её узлами являются модифицированные нейроны Оя (рис. 1,2), кроме того, она содержит дополнительный сумматор, предназначенный для декомпрессии входного сигнала и блок принятия решений (DM), в котором оценивается необходимость добавления дополнительного нейрона в сеть.

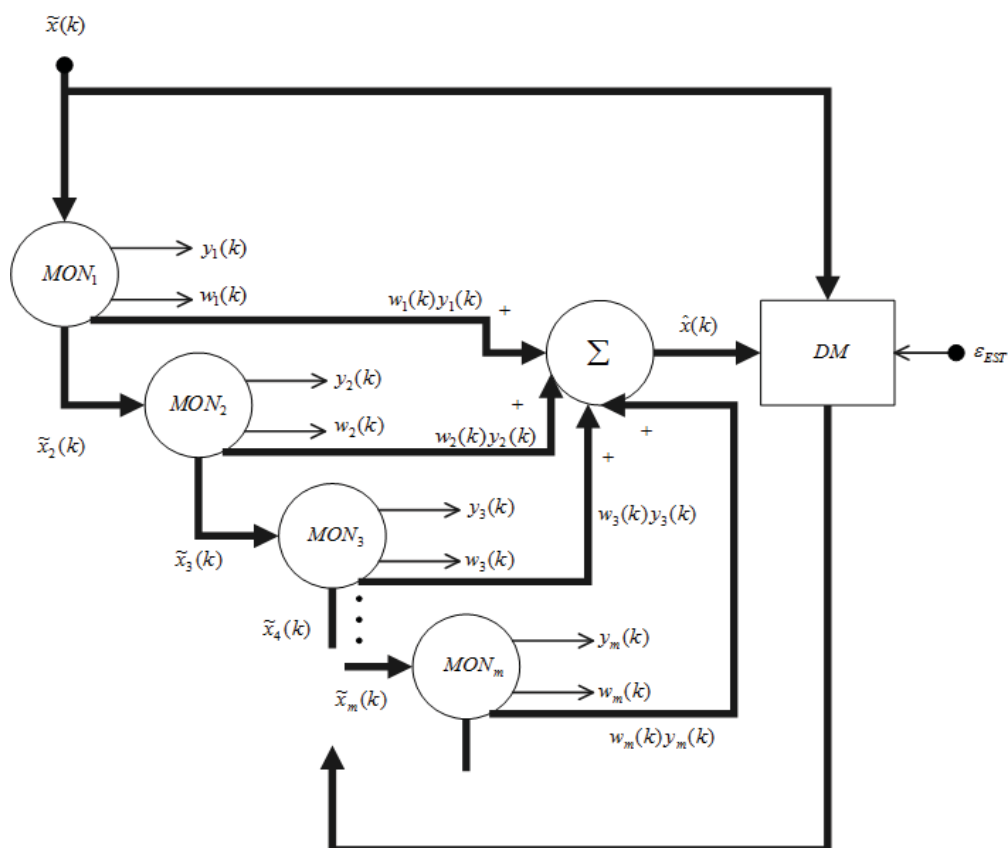


Рисунок 4 – Эволюционная каскадная нейронная сеть для последовательного анализа главных компонент

Сеть начинает свою работу с единственного нейрона MON_1 , при этом, если точность восстановленного сигнала $\hat{x}_1(k) = w_1(k)y_1(k)$ ниже некоторого заданного порога ε_{EST} , принимается решение о добавлении второго нейрона MON_2 так, как это показано на рис. 4. При этом следует заметить, что добавление второго и последующих нейронов (каскадов) никак не влияет на работу предыдущих узлов. Процесс наращивания каскадов продолжается до тех пор, пока восстановленный сигнал

$$\hat{x}_1(k) = \sum_{j=1}^m y_j(k) w_j(k)$$

по точности не будет удовлетворять критерию оценивания

$$E_{EST}^k = \frac{1}{k} \sum_k \frac{\|\tilde{x}(k) - \hat{x}(k)\|^2}{\|\tilde{x}(k)\|^2} \leq \varepsilon_{EST}. \quad (6)$$

Как только значение E_{EST}^k становится меньше уровня порога ε_{EST} , блок DM прекращает наращивание числа каскадов. В случае существенного изменения свойств сигнала $\tilde{x}(k)$ и нарушения неравенства (6), процесс роста сети может быть возобновлен.

Следует отметить, что архитектура на рис. 4 строится в рамках конструктивного подхода, т.е. может наращивать число своих нейронов от одного до $n-1$. Для решения задачи последовательного анализа главных компонент может быть использован и деструктивный подход, когда исходная сеть создается из максимально возможного числа нейронов $n-1$, которое сокращается в процессе самообучения. При этом, однако, вместо нейронов Э. Оя, вычисляющих главные компоненты, должны использоваться нейроны Д. Фенга [13,14], предназначенные для анализа минимальных компонент.

Выводы

Предложенная эволюционная каскадная нейронная сеть позволяет в режиме реального времени выделять произвольное число главных компонент многомерного стохастического сигнала, обеспечивая при этом требуемую точность и высокое быстродействие. Результат

вычислительного эксперимента подтверждают эффективность развиваемого подхода.

ЛИТЕРАТУРА

1. Haykin S. Neural Networks. A Comprehensive Foundation. – Upper Saddle River, N.J. : Preutice Hall, Inc., 1999 – 842 p.
2. Ham F.M., Kostanic I., Principle of Neurocompting for Science and Engineering. – N.Y. McGraw-Hill, Inc., 2001 – 642 p.
3. Oja E. A simplified neuron model as a principal component analyzer // Journal of Mathematical Biology, – 1982. – 15. – p. 267-273.
4. Oja E. Principal components, minor components, and linear neural networks // Neural Networks. – 1992. – 5. – p. 927-935.
5. Бодянский Е.В., Плисс И.П., Тесленко Н.А. Оптимальный алгоритм обучения нейрона Оя.//Теорія прийняття рішень. III Міжнародна школа-семінар: тези допов. – Ужгород, 2006.–с. 10-11.
6. Бодянский Е.В., Плисс И.П., Тесленко Н.А. Модифицированный нейрон Оя для анализа нестационарных данных // Автоматизация: проблемы, идеи, решения. : Междунар. науч.-тех. : тезисы докл. – Севастополь. – 2006. – с. 17-21.
7. Kaczmarz S. Approximate solution of systems of linear equations // Int. J. Control. – 1993. – 53. – 6. – p. 1269-1271.
8. Goodwin G.C., Ramadge P.J., Caines P.E. A globally convergent adaptive predictor// Automatica. – 1981. – 1. – p. 135-140.
9. Цыпкин Я.З. Основы теории обучающихся систем. – Москва : Наука, 1970. –252 с.
10. Sanger T. Optimal unsupervised learning in a single-layer linear feedforward neural Network// Neural Networks. – 1989. – 2. – p. 459-473.
11. Бодянский Е.В., Плисс И.П., Тесленко Н.А. Нестационарный нейросетевой анализ главных компонент // Інтелектуальні системи прийняття рішень та прикладні аспекти інформаційних технологій. – Тези доп. наук.-практ. конф. – Херсон, 2006. – Т.2. – с. 188-191.
12. Kasabov N. Evolving Connectionist Systems. – London : Springer-Verlag. 2003. –307 p.
13. Feng D.Z., Bao Z., Jiao L.C. Total least mean squares algorithm // IEEE Trans. Signal Process. – 1998, – 46. – №8. – p. 2122–2130.
14. Бодянский Е.В., Плисс И.П., Тесленко Н.А. Скоростной адаптивный алгоритм обучения нейрона для вычисления минимальной компоненты // Информационные технологии и информационная безопасность в науке, технике и образовании “Инфотех-2007” : Междунар. науч.-практ. конф.: тезисы докл. – Севастополь, – 2007. – Ч. I. – с. 8-11.