Факторизация преобразования Карунена–Лоэва

А. Ю. Дорогов

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)

vaksa2006@yandex.ru

Аннотация. **В работе представлен метод факторизации произвольных линейных преобразований, акцент сделан на построение факторизаций преобразования Карунена–Лоэва. Возможность факторизации позволяет организовать высокоскоростную конвейерную обработку на специализированных процессорах и таким образом снять ограничения по размерности при использовании данного преобразования. В статье факторизованная форма преобразования рассматривается как многослойная нейронная сеть, а вычисление коэффициентов факторизованной формы как обучение нейронных ядер сети. Алгоритм обучения не имеет обратных связей по ошибке, является абсолютно сходящимся и время его выполнения сопоставимо со временем обработки данных в сети.**

Ключевые слова: преобразование Карунена–Лоэва; факторизация; быстрая нейронная сеть; топологические модели

I. Введение

Для задач классификации и распознавания сигналов существенное значение имеют процедуры предварительной обработки, ориентированные на устранение избыточности и выделение информативных признаков. Использование ортогональных преобразований для этих целей позволяет представить информацию, содержащуюся в исходном сигнале в виде взаимно-независимых спектральных составляющих. Поскольку энергия сигнала определяется суммой квадратов спектральных коэффициентов, то по модулю спектрального коэффициента можно непосредственно судить о значимости информативного признака. Известно, что максимальное сокращение избыточности данных обеспечивается ортогональным преобразованием Карунена–Лоэва, которое образуется собственными векторами ковариационной матрицы сигналов. Преобразование впервые было предложено Хотеллингом [1] позже переоткрыто Каруненом и Лоэвом [2, 3]. Однако использование данного преобразования сопряжено со значительными вычислительными затратами. По этой причине метод Карунена–Лоэва в настоящее время не применяется для обработки данных высокой размерности.

Традиционно для обработки сигналов используются ортогональные преобразования, обладающие быстрыми алгоритмами выполнения. Быстрые алгоритмы основаны на возможности факторизации преобразования в произведение слабозаполненных матриц, каждую из которых можно интерпретировать как слой нейронной сети. Факторизация преобразований решает две задачи, во-первых сокращает общее число вычислительных операций и во-вторых позволяет организовать высокоскоростную конвейерную обработку на специализированных процессорах. Общеизвестно, что преобразование Карунена–Лоэва не имеет быстрого алгоритма. Это утверждение действительно справедливо для процессоров последовательного типа, но этот факт не связан с факторизацией. В этой статье будет показано, что факторизация преобразования Карунена–Лоэва возможна, и, следовательно, возможна его быстрое выполнение на специализированных процессорах конвейерного типа.

II. Базовая схема факторизации быстрых преобразований

В основе построения схемы факторизации лежат классические алгоритмы быстрых преобразований. На рис. 1 показан граф быстрого преобразования Фурье размерности 8 с топологией Кули–Тьюки «с прореживанием по времени». В каждом слое выделены четыре базовых операции типа «Бабочка». Для преобразования Фурье параметры базовых операций полностью определены, однако, если полагать, что их параметры можно изменять, то мы приходим к варианту быстрых нейронных сетей (БНС) [4]. В этом контексте базовые операции уместно назвать нейронными ядрами.

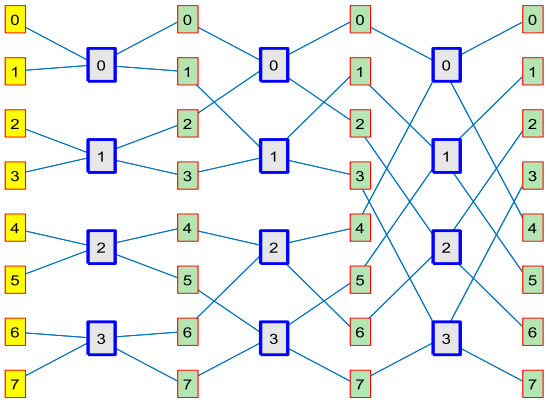

Рис. 1. Граф быстрого преобразования Фурье

Сетевая модель данной топологии и описывается набором кортежей [4]:

$$\begin{aligned} U^m &= \langle u_{n-1}u_{n-2}\cdots u_{m+1}u_m v_{m-1}v_{m-2}\cdots v_1 v_0\rangle, \\ V^m &= \langle u_{n-1}u_{n-2}\cdots u_{m+1}v_m v_{m-1}v_{m-2}\cdots v_1 v_0\rangle, \\ z^m &= \langle u_{n-1}u_{n-2}\cdots u_{m+1}v_{m-1}v_{m-2}\cdots v_1 v_0\rangle. \end{aligned} \qquad (1)$$

где z^m – порядковый номер нейронного ядра, U^m, V^m – порядковые номера рецепторов и аксонов в слое m, u_m, v_m – локальные номера рецепторов и аксонов в пределах ядра слоя m. Каждый кортеж представляет собой поразрядную форму представления порядкового номера через разрядные переменные u_m, v_m. Основания

разрядных переменных u_m, v_m определяются целыми положительными числами, заданных соответствиями:

$$\begin{pmatrix} u_0 & u_1 & \cdots & u_{n-2} & u_{n-1} \\ p_0 & p_1 & \cdots & p_{n-2} & p_{n-1} \end{pmatrix}, \quad \begin{pmatrix} v_0 & v_1 & \cdots & v_{n-2} & v_{n-1} \\ g_0 & g_1 & \cdots & g_{n-2} & g_{n-1} \end{pmatrix}.$$

Для ядер размерности 2×2 все разрядные переменные принимают значения {0,1}. Координатные направления U^m и V^m в дальнейшем будем называть входной и выходной плоскостями нейронного слоя. Для терминальных плоскостей будем использовать обозначения U и V. Для построения быстрых алгоритмов размерность преобразования должна быть составным числом и чем больше множителей в разложении размерности, тем выше вычислительная эффективность быстрого алгоритма. Размерности быстрой нейронной сети по входу и выходу вычисляются через произведения оснований разрядных переменных:

$$N = p_{n-1} \ldots p_1 p_0, \quad M = g_{n-1} \ldots g_1 g_0.$$

Число слоёв в быстрых преобразованиях равно число сомножителей в этих произведениях. Несмотря на большое разнообразие быстрых алгоритмов, конфигурации их структур удовлетворяют системному инварианту самоподобия [4]. Как известно таким же свойством самоподобия обладают фракталы. Поэтому быстрые алгоритмы можно интерпретировать как квазифракталы. Свойство структурной фрактальности позволяет решить одновременно две задачи: реализовать быструю обработку данных и выполнить быстрое обучение преобразования. Обучение быстрого преобразования заключается в выборе значений элементов нейронных ядер, так чтобы в столбцах матрицы преобразования содержался заданный набор опорных функций, это могут быть, например, функции базиса Карунена–Лоэва. В работе [4] показано, что для самоподобных нейронных сетей элементы матрицы быстрого преобразования могут быть выражены через произведения элементов нейронных ядер:

$$h(U,V) = w^{n-1}_{z^{n-1}}(u_{n-1}, v_{n-1})\, w^{n-2}_{z^{n-2}}(u_{n-2}, v_{n-2}) \cdots w^{0}_{z^{0}}(u_0, v_0).$$

Там же доказано, что произвольная функция, заданная на дискретном интервале длиной $N = p_{n-1} \ldots p_1 p_0$, также может быть представлена в мультипликативной форме:

$$f(u) = \phi_{i^0}(u_0)\phi_{i^1}(u_1) \ldots \phi_{i^{n-2}}(u_{n-2})\phi_{i^{n-1}}(u_{n-1}).$$

где $i^m = \langle u_{n-1} u_{n-2} \ldots u_{m+1} \rangle$. Отсюда следует правило настройки нейронных ядер

$$w^m_{z^m}(u_m, v_m) = \phi^k_{i^m}(u_m) \qquad (2)$$

Здесь k – номер опорной функции. Зададим точку привязки функции в выходной плоскости числом, представленным в поразрядной форме:

$$x = \langle x_{n-1} x_{n-2} \cdots x_0 \rangle,$$

тогда номер настраиваемых ядер по слоям будет определяться выражением:

$$z^m = \langle u_{n-1} u_{n-2} \cdots u_{m+1} x_{m-1} x_{m-2} \cdots x_1 x_0 \rangle.$$

Для $m = 0$ имеем $z^0 = \langle u_{n-1} u_{n-2} \cdots u_1 \rangle$, это означает, что независимо от выбора точки привязки, все ядра слоя будут настраиваться, причём номер ядра определяется из условия: $z^0 = i^0$. Настройка элементов ядер этого слоя выполняется по правилу:

$$w^0_{z^0}(u_0, v_0) = \phi^k_{i^0}(u_0),$$

Очевидно, должно быть задано взаимно-однозначное соответствие $k \leftrightarrow v_0$ между номером опорной функции и разрядной переменной v_0. Эта разрядная переменная принимает значения $0,1,\cdots,g_0 - 1$. Отсюда следует вывод, что число опорных функций не может быть больше чем g_0, а если ещё потребовать выполнения условия ортогональности ядер, то матрица такого преобразования будет содержать только одну произвольную функцию в качестве столбца. Этого явно недостаточно для реализации произвольного ортогонального базиса Карунена–Лоэва, что и соответствует утверждению об отсутствии классического быстрого алгоритма для преобразования Карунена–Лоэва. Однако, как будет показано в следующем разделе, это не препятствует процедуре факторизации этого преобразования.

III. Самоподобные нейронные сети с коммутацией плоскостей

Дополним топологическую модель быстрого преобразования (1) дополнительными плоскостями нейронных ядер [5], номера которых определим кортежем π_m:

$$U^m = \langle u_{n-1} u_{n-2} \cdots u_{m+1} u_m v_{m-1} v_{m-2} \cdots v_1 v_0 \rangle,$$
$$V^m = \langle u_{n-1} u_{n-2} \cdots u_{m+1} v_m v_{m-1} v_{m-2} \cdots v_1 v_0 \rangle,$$
$$z^m = \langle u_{n-1} u_{n-2} \cdots u_{m+1} v_{m-1} v_{m-2} \cdots v_1 v_0 \rangle,$$
$$\pi_m = \langle v_{n-1} v_{n-2} \cdots v_{m+2} v_{m+1} \rangle.$$

Максимальное количество дополнительных плоскостей появится в нулевом слое. Номер плоскости в слое будет определяться кортежем $\pi_0 = \langle v_{n-1} v_{n-2} \cdots v_2 v_1 \rangle$. Плоскость с номером $\pi_0 = 0$ будем считать плоскостью базовой структуры. Число плоскостей в нулевом слое будет равно произведению оснований: $g_{n-1} g_{n-2} \cdots g_2 g_1$. По мере движения к выходному слою число дополнительных плоскостей будет уменьшаться и для последнего слоя $\pi_{n-1} = \langle\ \rangle$, т. е. их не будет совсем, останется только одна плоскость базовой топологии. Таким образом, в новой топологии плоскость последнего слоя останется прежней, а в младших слоях появятся дополнительные плоскости. Номер ядра, теперь следует уточнять его размещением в дополнительной плоскости. На рис. 2 показана новая топология, построенная на базе трёхслойной сети с основаниями 2.

В сети условно показаны коммутаторы (SW), которые служат для пояснения принципа работы сети. Коммутаторы управляются разрядными переменными точек выходной плоскости. Фактически коммутаторов нет, они реализуются в составе алгоритма и не препятствуют параллельной обработки входных данных. В контексте спектрального преобразования точки

выходной плоскости точки ассоциируются со спектральными коэффициентами, поэтому привязка спектрального коэффициента к координатам выходной плоскости предопределяет выбор дополнительных плоскостей, которые используются для обработки входных образов. При построении полного спектрального анализатора все векторные входы параллельно объединяются.

Поскольку правило порождения новых плоскостей не противоречит базовой топологической модели, то для настройки ядер преобразования к эталону можно использовать прежнее правило (2), расширив его аргументом для дополнительных плоскостей:

$$w_{z^m}^{m}\left\langle \pi_m \right\rangle (u_m, x_m) = \phi_{i^m}^{k}(u_m),$$

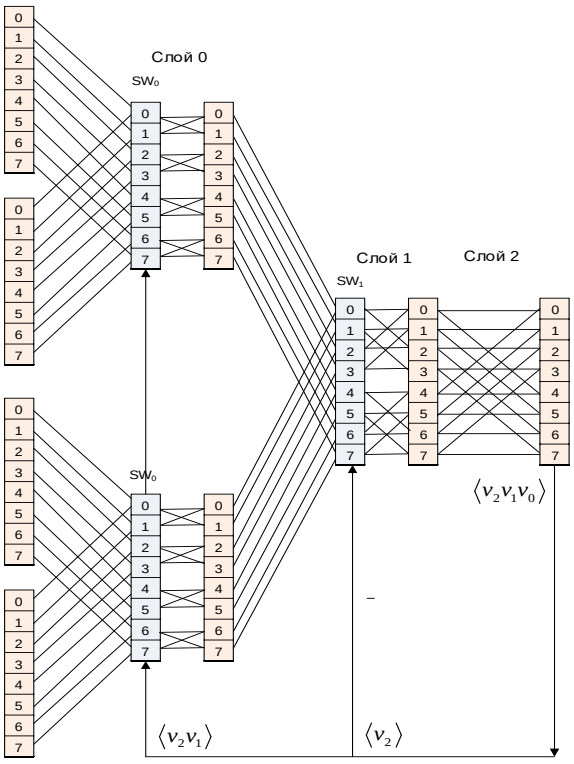

Рис. 2. Топология самоподобной нейронной сети с дополнительными плоскостями

здесь k – номер эталонной функции, $x = \left\langle x_{n-1}x_{n-2}\cdots x_0 \right\rangle$, точка привязки, $z^m = \left\langle u_{n-1}u_{n-2}\cdots u_{m+1}x_{m-1}x_{m-2}\cdots x_1x_0 \right\rangle$ номер настраиваемых ядер по слоям, $\pi_m = \left\langle x_{n-1}x_{n-2}\cdots x_{m-1} \right\rangle$ – номер плоскости размещения ядер. Индекс k в правой части нумерует точку приспособления. Для $m = 0$ имеем $z^0 = i^0 = \left\langle u_{n-1}u_{n-2}\cdots u_1 \right\rangle$, а варьируемыми переменными в левой части являются номер плоскости $\pi_0 = \left\langle x_{n-1}x_{n-2}\cdots x_1 \right\rangle$ и разряд x_0. Вместе они покрывают весь диапазон координат выходной плоскости. Этому диапазону отвечают возможные значения индекса k в правой части, отсюда следует, что каждая точка выходной плоскости может быть связана с собственной опорной функцией. Т. е. построенная сеть обладает максимально возможным числом точек привязки, покрывающих всю выходную плоскость и, таким образом может быть использована для реализации произвольного линейного преобразования размерности $N \times M$. В частности, при реализации ортогонального преобразования Карунена–Лоэва $N = M$

IV. Заключение

В статье показано, что топология БНС легко расширяется дополнительными плоскостями, при этом число реализуемых опорных функций кардинально возрастает и покрывает все элементы выходной плоскости. Причём расширение топологии не нарушает принципа построения обучающего алгоритма. Полученные результаты позволяют построить факторизованное представление для любого линейного преобразования, в том числе и для ортогонального преобразования Карунена–Лоэва. Вычислительный эффект от факторизации достигается при использовании специализированных конвейерных вычислителей. Предложенная архитектура реализуют сети с глубокой степенью обучения. Можно показать, что они сегментируются в лес независимых пирамидальных сетей [6]. Это порождает уникальное качество – возможность дообучения сети к новым образам без изменения или потери ранее накопленных знаний. Сети могут работать в системах реального времени, поскольку время обучения не превышает времени обработки данных в сети. Алгоритмы обучения являются абсолютно устойчивыми и завершаются за конечное число шагов. Выбор структуры сети не вызывает проблем и определяется системными инвариантами самоподобных сетей. Возможны различные структурные решения, обладающие одинаковыми качествами.

Список литературы

[1] Hotelling H. (1933). Analysis of a complex of statistical variables into principal components. // Journal of Educational Psychology, 24, 417-441, and 498-520.

[2] K. Karhunen, Kari, Uber lineare Methoden in der Wahrscheinlichkeitsrechnung // Ann. Acad. Sci. Fennicae. Ser. A. I. Math.-Phys., 1947, No. 37, 1-79.

[3] Лоев М. Теория вероятностей, М.: ИЛ, 1962.

[4] Дорогов А.Ю. Теория и проектирование быстрых перестраиваемых преобразований и слабосвязанных нейронных сетей. СПб.: Политехника, 2014. 328 с.

[5] Дорогов А.Ю. Быстрые нейронные сети глубокого обучения // III Международная научная конференция по проблемам управления в технических системах (CTS'2019): Сб. докладов. Санкт-Петербург. 30 октября – 1 ноября 2019 г. СПб.: СПбГЭТУ «ЛЭТИ». С. 275-280.

[6] Дорогов А.Ю. Быстрые пирамидальные нейронные сети глубокого обучения для адаптивной цифровой обработки сигналов // 78-я Научно-техническая конференция Санкт-Петербургского НТО РЭС им. А.С. Попова, посвящённая Дню радио: Сб. материалов. 2023. СПб.: СПбГЭТУ «ЛЭТИ», с. 107-111. Электронное издание. ISBN 978-5-7629-3183-0.