Цензурирующий фильтр в алгоритме обучения быстрых ортогональных нейронных сетей

А. Ю. Дорогов

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

vaksa2006@yandex.ru

Аннотация. **Быстрые ортогональные нейронные сети являются настраиваемым вариантом алгоритма Быстрого преобразования Фурье (БПФ). В статье описан способ обучения быстрых нейронных сетей к сигнальным функциям на основе метода фрактальной фильтрации. Показано, что при использовании данного метода обучения необходимо решать задачу цензурирования нулей при построении образующих функций. Предложен алгоритм реализации нормирующего фрактального фильтра с цензурированием нулей. Корректность работы алгоритма доказана на критических видах сигнальных функций. Приведён пример построения быстрой нейронной сети для одной из критических сигнальных функций.**

Ключевые слова: ортогональная нейронная сеть; быстрое преобразование; БПФ; фрактальная фильтрация; цензурироваание нулей

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 сигнала. Это преобразование не имеет быстрого алгоритма и поэтому не пригодно для обработки данных высокой размерности. На практике широкое распространение получили быстрые ортогональные преобразования, классическим примером которых является быстрое преобразование Фурье (БПФ).

В какой-то мере используемое название «ортогональная нейронная сеть» [1, 2] является данью моде, ранее употреблялся термин «обобщённое быстрое ортогональное преобразование». Первые предложения по построению обобщённого ортогонального преобразования на основе быстрых алгоритмов подобных БПФ были высказаны Эндрюсом и Каспари [3] в 70-х годах прошлого века. А первые подходы к обучению подобных преобразований были развиты в работах А.И. Солодовникова и его группы [4] в восьмидесятых годах прошлого века. В то время подобный класс преобразований называли также перестраиваемыми быстрыми преобразованиями. Предложенный в то время алгоритм обучения был основан на геометрической модели ортогонального ядра второго порядка (базовой операции типа «бабочка»), представленной в полярной системе координат. Алгоритм позволял приспособить преобразование к одной опорной функции, поэтому этот класс быстрых ортогональных преобразований тогда называли приспособленными.

На рис. 1 представлен 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 и настраивать, то мы приходим к варианту быстрых нейронных сетей (БНС) [4]. В этом контексте базовые операции уместно назвать нейронными ядрами.

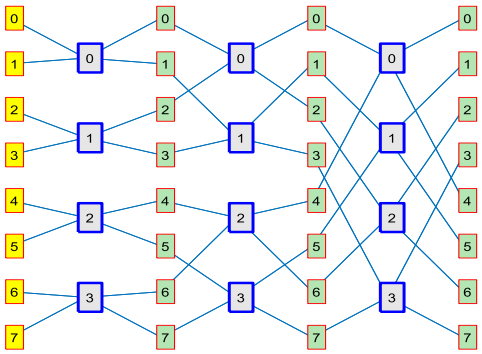

Рис. 1. Граф быстрого преобразования Фурье

Для ортогональных нейронных сетей нейронные ядра представляются ортогональными матрицами малой размерности.

Автором было доказано, что структура графов быстрых алгоритмов является самоподобной. Таким же свойством обладают фракталы. Это позволило разработать особые алгоритмы обучения на основе методов фрактальной фильтрации, которые пригодны для ортогональных ядер произвольных размерностей [5]. В [6] показано, что для самоподобных нейронных сетей элементы матрицы быстрого преобразования могут быть выражены через произведения элементов нейронных ядер:

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

где U,V – номер строки и номер столбца матрицы быстрого преобразования, $w_{z^m}^m(u_m,v_m)$ – элементы базовой операции с номером z^m в слое m, (где $m=0\cdots n-1$). Число слоёв определяется мультипликативным разложением значения размерности преобразования:

$$N = p_{n-1}\dots p_1 p_0,$$

а множители p_m – определяют размерности нейронных ядер по слоям. Для данной топологии номера строк, столбцов и базовых операций выражаются в позиционной системе счисления:

$$U=\langle u_{n-1}u_{n-2}\dots u_0\rangle,\qquad V=\langle v_{n-1}v_{n-2}\dots v_0\rangle,$$
$$z^m=\langle u_{n-1}u_{n-2}\dots u_{m+1}v_{m-1}v_{m-2}\dots v_0\rangle.$$

Эти выражения представляют собой топологическую модель и могут быть использованы для построения графа быстрого преобразования.

II. Обучение ортогональных нейронных сетей

Определяющее значение для построения алгоритма обучения БНС имело доказанное автором положение, что любая дискретная функция $f(U)$, заданная на интервале $N = p_{n-1}\dots p_1 p_0$ может быть представлена в виде произведения образующих функций [7]:

$$f(U)=\varphi_{i^0}(u_0)\varphi_{i^1}(u_1)\dots\varphi_{i^{n-1}}(u_{n-1}). \qquad (2)$$

Правило выбора образующих функций для каждого шага мультипликативного разложения подчиним топологии преобразования, положив $i^m=\langle u_{n-1}u_{n-2}\dots u_{m+1}\rangle$. Для нахождения мультипликативного разложения (2) введём последовательность вспомогательных функций, определив их рекуррентно следующим образом:

$$\begin{aligned}
&f(U)=f_0\langle u_{n-1}u_{n-2}\dots u_0\rangle,\\
&f_0\langle u_{n-1}u_{n-2}\dots u_0\rangle=f_1\langle u_{n-1}u_{n-2}\dots u_1\rangle\varphi_{i^0}(u_0),\\
&f_1\langle u_{n-1}u_{n-2}\dots u_1\rangle=f_2\langle u_{n-1}u_{n-2}\dots u_2\rangle\varphi_{i^1}(u_1),\\
&\vdots\\
&f_{n-2}\langle u_{n-1}u_{n-2}\rangle=f_{n-1}\langle u_{n-1}\rangle\varphi_{i^{n-2}}(u_{n-2}),\\
&f_{n-1}\langle u_{n-1}\rangle=\varphi_{i^{n-1}}(u_{n-1}).
\end{aligned} \qquad (3)$$

Из этих соотношений непосредственно следует:

$$\varphi_{i^m}(u_m)=\frac{f_m\langle u_{n-1}u_{n-2}\dots u_m\rangle}{f_{m+1}\langle u_{n-1}u_{n-2}\dots u_{m+1}\rangle},\qquad m=0,1,\dots,n-2. \qquad (4)$$

Правило построения вспомогательных функций, будет описано в следующем разделе, а сейчас будем полагать, что эти функции известны и множители в разложении (2) определены. Рассмотрим алгоритм обучения быстрой ортогональной нейронной сети к одной функции $f(U)$, т. е. построим приспособленное преобразование.

Определение. Ортогональное преобразование, заданное матрицей $\|h(U,V)\|$, приспособлено к функции $f(U)$ по столбцу S, если выполнены условия:

$$\begin{cases}\sum_U h(U,V)\,f(U)=1 & при\ \ V=S,\\ \sum_U h(U,V)\,f(U)=0 & при\ \ V\neq S.\end{cases} \qquad (5)$$

Предполагается, что опорная функция нормирована по энергии условием: $\sum_U f^2(U)=1$. Подставив в левую часть (5) выражения (1) и (2) после преобразований получим:

$$\sum_U h(U,V)\,f(U)=\sum_{u_0} w_{i^0}(u_0,v_0)\varphi_{i^0}(u_0)\sum_{u_1} w_{i^1}(u_1,v_1)\varphi_{i^1}(u_1)\cdots$$
$$\dots\sum_{u_{k-1}} w_{i^{n-1}}(u_{n-1},v_{n-1})\varphi_{i^{n-1}}(u_{n-1}).$$

Из данного выражения, следует, что приспособленность спектрального преобразования обеспечивается, когда ядра любого слоя приспособлены к соответствующим компонентам мультипликативного разложения опорной функции:

$$\begin{cases}\sum_{u_m} w_{i^m}(u_m,v_m)\varphi_{i^m}(u_m)=1 & при\ \ v_m=s_m,\\ \sum_{u_m} w_{i^m}(u_m,v_m)\varphi_{i^m}(u_m)=0 & при\ \ v_m\neq s_m,\end{cases}$$

где $S=\langle s_0 s_1\dots s_{m-1}s_{n-1}\rangle$. Таким образом, задача настройки быстрого приспособленного спектрального преобразования сводится к построению приспособленных ортогональных ядер.

Наиболее просто построить ортогональные ядра второго порядка. Для этого достаточно из образующей функции $\varphi(u_m)=[\varphi(0)\ \ \varphi(1)]$, заданной на интервале длиной 2, выделить постоянную и переменную составляющие и отнормировать их по энергии к единице. Две полученные компоненты, очевидно, будут ортогональны между собой. В общем случае для построения ортогональных ядер следует использовать процедуру ортогонализации Грамма–Шмидта [8].

III. Фрактальная фильтрация

В обобщённом представлении фильтрация – это операция в пространстве сигналов, реализующая проекцию сигнала в пространство сигналов меньшей размерности. Цель фрактальной фильтрации заключается в том, чтобы выявить пространственно-распределенные свойства изучаемого объекта в кратных временных масштабах.

Рассмотрим дискретный фрактальный фильтр. Пусть сигнал определён функцией $f(U)$, заданной на дискретном интервале длиной $N = p_{n-1}\dots p_1 p_0$. Представим аргумент функции в позиционной многоосновной системе счисления с основаниями $p_0, p_1,\dots, p_{n-1}$. Формулы перехода к поразрядному представлению, как известно, имеют вид:

$$U=\langle u_{n-1}u_{n-2}\dots u_0\rangle=u_{n-1}p_{n-2}p_{n-3}\dots p_0+u_{k-2}p_{k-3}p_{k-4}\dots p_0+\dots$$
$$\dots+u_1p_0+u_0,$$

где $u_i\in[0,1,\dots p_i-1]$ – разрядные переменные. В результате данного перехода сигнал представляется как

многомерная функция $f\langle u_{n-1}u_{n-2}\dots u_0\rangle$. Каждый аргумент функции определяет некоторый масштабный срез сигнала.

Зафиксируем все аргументы функции кроме u_m. Варьируя свободный аргумент u_m, получим выборку Q_m (с числом элементов p_m, которое назовём базой фильтра). Фрактальным фильтром масштабного уровня m будем называть произвольный функционал $F(Q_m)$, определённый на выборке Q_m. Операцию фрактальной фильтрации можно записать в виде:

$$f_{out}\langle u_{n-1}u_{n-2}\dots u_{m+1}u_{m-1}\dots u_0\rangle = \underset{u_m}{F}\left(f_{inp}\langle u_{n-1}u_{n-2}\dots u_0\rangle\right).$$

Наиболее простым вариантом является линейный фильтр вычисления среднего по выборке:

$$f_{out}\langle u_{n-1}u_{n-2}\dots u_{m+1}u_{m-1}\dots u_0\rangle = \frac{1}{p_m}\sum_{u_m} f_{inp}\langle u_{n-1}u_{n-2}\dots u_0\rangle.$$

Возвращаясь к предыдущему разделу, нетрудно заметить, что вспомогательные функции в выражении (3) могут быть определены цепочкой фрактальных фильтров, показанной на рис. 2.

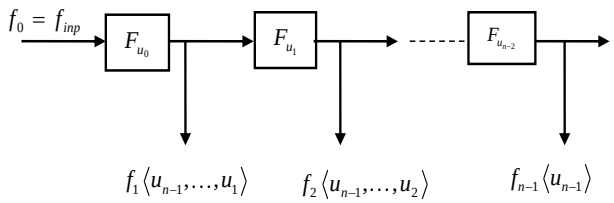

Рис. 2. Цепочка фрактальных фильтров

Схема последовательной фрактальной фильтрация, реализует принцип иерархической кратно-масштабной обработки данных.

IV. Нормирующий фильтр с цензурированием нулей

Предложенная схема формирования вспомогательных функций (3) и образующих функций (4) хорошо поясняет принцип мультипликативного разложения функций, но должна быть доработана при практической реализации по следующим причинам.

Во-первых, необходимо при выборе образующих функций обеспечить единичную норму по энергии для сигнальной функции $f(U)$. Норма определяется как корень квадратный из суммы квадратов значений функций по всем точкам дискретного интервала:

$$norm\left(f(U)\right) = \sqrt{\sum_U f^2(U)}.$$

Из (2) следует

$$\sum_U f^2(U) = \sum_{u_{n-1},u_{n-2},\dots,u_0} f_0\langle u_{n-1}u_{n-2}\dots u_0\rangle =$$

$$= \sum_{u_{n-1},u_{n-2},\dots,u_0}\left(\phi_{i^0}(u_0)\phi_{i^1}(u_1)\dots\phi_{i^{n-2}}(u_{n-2})\phi_{i^{n-1}}(u_{n-1})\right)^2 =$$

$$= \sum_{u_0}\left(\phi_{i^0}(u_0)\right)^2\sum_{u_1}\left(\phi_{i^1}(u_1)\right)^2\dots\sum_{u_{n-2}}\left(\phi_{i^{n-2}}(u_{n-2})\right)^2\sum_{u_{n-1}}\left(\phi_{i^{n-1}}(u_{n-1})\right)^2.$$

Таким образом, норма сигнальной функции $f(U)$ равна произведению норм образующих функций. В частности, если нормы образующих функций единичны, то и норма сигнальной функции также будет единична. Поэтому алгоритм мультипликативной декомпозиции должен предусматривать нормирование образующих функций.

Второе обстоятельство связано с формулой вычисления (4) для значений образующих функций. Может оказаться, что знаменатель в этом выражении равен нулю, тогда вычисление невыполнимо. Менее очевидно, что необходимо также проверять на нуль значения числителя. Если для всех значений разрядной переменной u_m числитель в выражении (4) равен нулю, то норма множителя $\varphi_{i^m}(u_m)$ также равна нулю, что недопустимо для ортогональных ядер. На рис. 3 представлен модифицированный алгоритм фрактального фильтра, обеспечивающего цензурирование нулей, и нормировку при вычислении образующих функций.

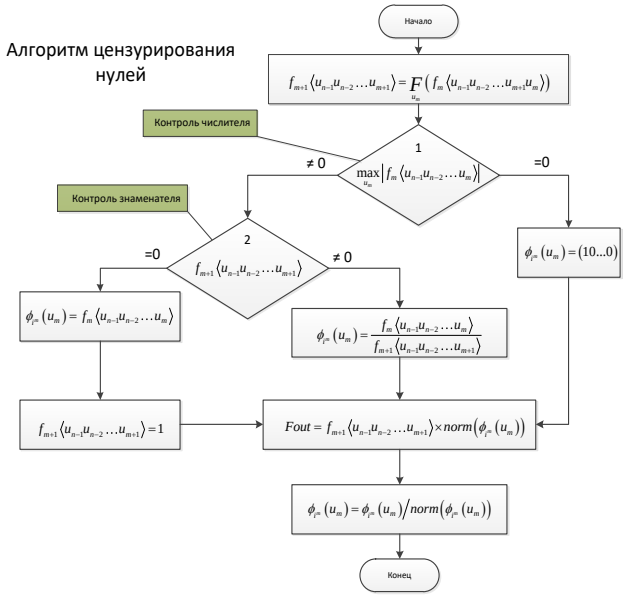

Рис. 3. Алгоритм цензурирования нулей

Первый блок алгоритма выполняет обычную фрактальную фильтрацию, например с помощью фильтра среднего. На выходе фильтра получим вспомогательную функцию $f_{m+1}\langle u_{n-1}u_{n-2}\dots u_{m+1}\rangle$, которая является знаменателем выражения (4). Логический блок 1 алгоритма выполняет контроль значений числителя выражения (4). Если все значения функции числителя при варьировании разрядной переменной u_m – нулевые, тогда и максимум модулей по всем значениям u_m также равен нулю. Для этой ветви алгоритма, выход фрактального фильтра средних также будет нулевым. В этом случае образующая функция выбирается в виде $\varphi_{i^m} = (100\dots0)$. Другие варианты также приемлемы. Образующая функция должна иметь хотя бы одно не нулевое значение.

Логический блок 2 выполняет контроль знаменателя выражения (4). Если знаменатель не равен нулю, то для вычислений значений образующей функции используется выражение (4), а если равен нулю, то

образующая функция устанавливается равной числителю выражения (4), а знаменатель – равным единице, который затем передаётся на выход фильтра, где домножается на норму образующей функции. Заключительный блок алгоритма выполняет нормирование образующей функции. Приведённый алгоритм заменяет фрактальные фильтры в цепочке, представленной на рис. 2.

Рассмотрим работу алгоритма на критичных вариантах сигнальных функции. На рис. 4 приведена нормированная сигнальная функция в виде меандра, заданная на интервале длиной 16. Будем полагать, что база фрактального фильтра равна двум. После первого шага фрактальной фильтрации выход фильтра средних будет нулевым и определённым на интервале длиной 8. На рисунке выход показан красной линией, которая условно растянута на исходный интервал длиной 16. Знаменатель выражения (4) равен нулю, но числитель нулю не равен. Образующая функция принимает значения $\phi_{u_0} = \left[0.25 \quad -0.25\right]$, а после нормирования $\hat{\phi}_{u_0} = \left[1/\sqrt{2} \quad -1/\sqrt{2}\right]$.

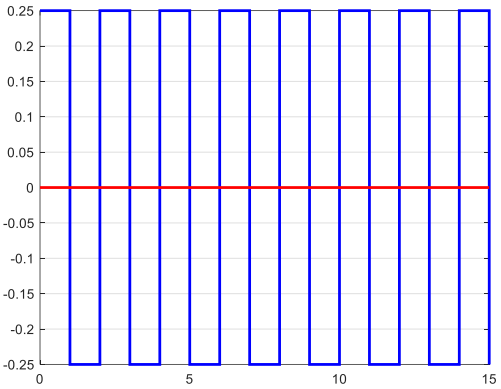

Рис. 4. Фрактальная фильтрация меандра

Алгоритм выполняется по левой ветви блок-схемы, знаменателю присваивается единичное значение, которое затем умножается на норму образующей функции ($norm\left(\phi_{u_0}\right) = 0.25\sqrt{2}$). Результат представлен на рис. 5 (красная линия).

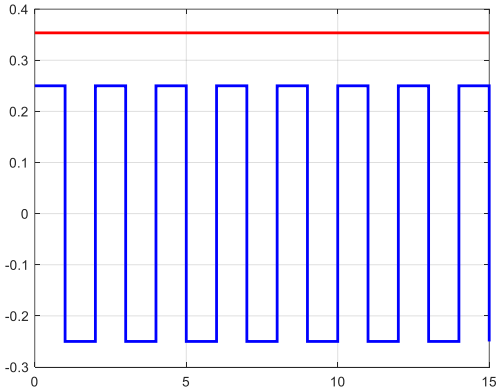

Рис. 5. Выход нормирующего фильтра с цензурированием нулей

Если двигаться по цепочке фильтров (рис. 2), то все последующие образующие функции после нормировки будут иметь значения $\hat{\phi}_{u_m} = \left[1/\sqrt{2} \quad 1/\sqrt{2}\right]$. Нетрудно проверить, что произведение (2) будет порождать исходную сигнальную функцию.

Другой критичный вариант сигнальной функции это одиночный импульс. На рис. 6 сигнальная функция показана синим цветом.

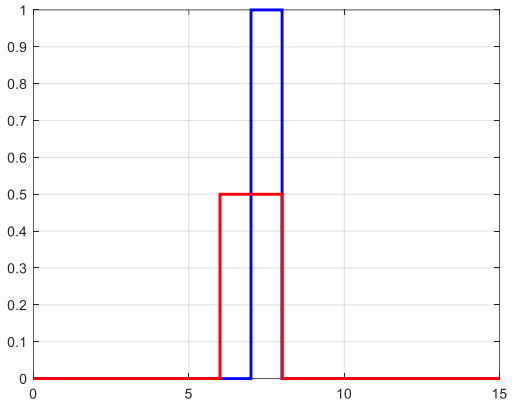

Рис. 6. Фрактальная фильтрация одиночного импульса

После шага фильтрации вспомогательная функция на выходе, определённая на интервале длиной 8, также будем иметь одиночный импульс (красная линия – кривая условно растянута на исходный интервал длиной 16).

В начальной области сигнала на выходе фрактального фильтра ноль появляется как в числителе, так и в знаменателе выражения (4). В этом случае выполняется правая ветвь алгоритма (рис. 3), образующая функция (при базе равной 2) устанавливается равной $\phi_{i^0} = \left[1 \quad 0\right]$, а на выход передается нулевой знаменатель. Там, где существует импульс, выход фильтра (т. е. знаменатель выражения (4)) отличен от нуля. И числитель также будет отличен от нуля, в этом случае для получения образующей функции используется выражение (4). На выходе алгоритма вновь получим импульс с единичной амплитудой (рис. 7).

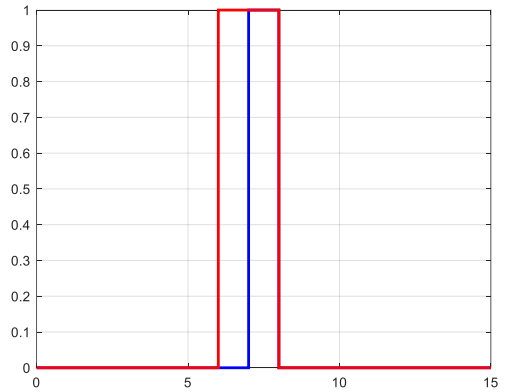

Рис. 7. Выход нормирующего фильтра с цензурированием нулей

Пусть импульс сигнальной функции размещён в позиции:

$$x = \left\langle x_{n-1} x_{n-2} \ldots x_0 \right\rangle.$$

В результате первого шага фильтрации по схеме рис. 2 мы получим набор нормированных множителей $\hat{\phi}_{i^0}(u_0)$ где

$$i^0 = \langle u_{n-1} u_{n-2} \dots u_1 \rangle.$$

Позиции импульса будет соответствовать множитель $\hat{\phi}_{i^0}(u_0)$ для которого $i^0 = \langle x_{n-1} x_{n-2} \dots x_1 \rangle$. Причём $\hat{\phi}_{i^0}(x_0) = 1$. Если аргумент множителя не равен x_0 то произведение (2) будет равно нулю.

В результате второго шага фильтрации по схеме рис. 2 мы получим набор множителей $\hat{\phi}_{i^1}(u_1)$, где $i^1 = \langle x_{n-1} x_{n-2} \dots x_2 \rangle$. Причём $\hat{\phi}_{i^1}(x_1) = 1$. Если аргумент фрактального множителя не равен x_1 то произведение (2) будет равно нулю.

Продолжая подобные рассуждения дальше, мы приходим к результату, что произведение (2) будет равно 1 только для значения аргумента $x = \langle x_{n-1} x_{n-2} \dots x_0 \rangle$. Для всех остальных значениях аргумента сигнальной функции произведение будет равно нулю. Таким образом, произведение (2) будет порождать исходную сигнальную функцию.

В заключение приведём пример построения ортогонального преобразования, приспособленного к функции меандра по второму столбцу. На рис. 8 в виде функций показаны столбцы построенной матрицы быстрого ортогонального преобразования.

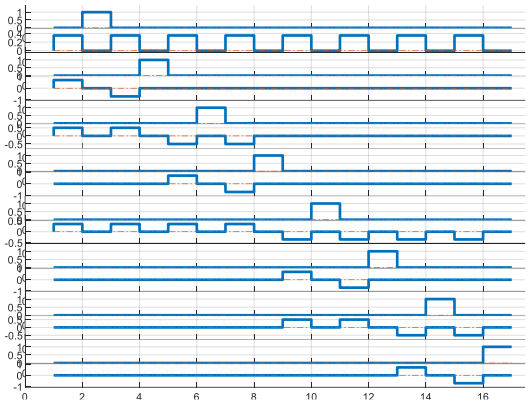

Рис. 8. Ортогональное преобразование, приспособленное к функции меандра

Преобразование реализуется быстрой ортогональной нейронной сетью, ортогональные ядра получены с помощью нормирующего ортогонального фильтра с цензурированием нулей и процедуры Грамма–Шмидта. При формировании ядер реализована стратегия минимизации числа вычислительных операций, в этом случае ортогональное преобразование обладает максимальной степенью пространственной локализации базисных функций.

V. Заключение

Быстрые преобразования имеют многослойную структуру, но в отличие от многослойных нейронных сетей, межслойные связи разрежены, а нелинейные функции активации и смещения отсутствуют. Быстрые нейронные сети – это быстрые линейные преобразования, в которых число слоёв и топология межслойных связей определяется правилом построения быстрых алгоритмов, а коэффициенты преобразования настраиваются, т. е. существует процедура обучения.

Предложенный метод обучения быстрых ортогональных нейронных сетей позволяет точно настроиться на сигнальную функцию за конечное число шагов. Проблемы сходимости здесь не существует, алгоритм абсолютно сходится, но существует необходимость контроля нулей в базовом выражении алгоритма обучения. Рассмотренный способ цензурирования нулей решает эту задачу. Данный способ может быть в равной мере использован и для не ортогональных вариантов БНС. Предложенный метод настройки позволяет обучить быструю ортогональную нейронную сеть только к одной произвольной функции. Но заметим также, что БНС относятся к классу самоподобных сетей, и в работах [5, 9] показано, что потенциал их обучения может быть кардинально увеличен расширением топологической схемы построения самоподобных сетей.

Список литературы

[1] Bartłomiej Stasiak and Mykhaylo Yatsymirsky. Fast Orthogonal Neural Networks // Lecture Notes in Computer Science, 2006, Volume 4029/2006, 142–149. DOI:10.1007/11785231_16

[2] Дорогов А.Ю. Реализация спектральных преобразований в классе быстрых нейронных сетей // Программирование. 2003. №4. С.13-26.

[3] Andrews H.C., Caspari K.L. A General Techniques for Spectral Analysis // IEEE. Tr. Computer. 1970. Vol. C. 19.-Jan, No 1. pp. 16–25. DOI:10.1109/TC.1970.5008895

[4] Солодовников А.И., Спиваковский А.М. Основы теории и методы спектральной обработки информации. Л., 1986. 272 с.

[5] Дорогов А.Ю. Самоподобные нейронные сети быстрого обучения. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2024. 188 с. www.dorogov.su.

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

[7] Дорогов А.Ю. Методы настройки быстрых перестраиваемых преобразований // Нейрокомпьютеры: разработка и применение. 2004. № 7–8. С. 17–32.

[8] Белман Р. Введение в теорию матриц. М.: Наука, 1976. 352 с.

[9] Дорогов А.Ю. Реализация преобразования Карунена–Лоэва в классе многослойных самоподобных нейронных сетей // XXVII Международная конференция по мягким вычислениям и измерениям (SCM'2024). Санкт-Петербург, СПбГЭТУ «ЛЭТИ», 2024. С. 235-238. https://elibrary.ru/download/elibrary_67959657_90909860.pdf