

Быстрые нейронные сети глубокого обучения

А. Ю. Дорогов

Санкт-Петербургский государственный электротехнический университет
«ЛЭТИ» им. В.И. Ульянова (Ленина)
vaksa2006@yandex.ru

Аннотация. Рассматривается алгоритм построения сетевой топологии и обучения двумерных быстрых нейронных сетей с дополнительными коммутируемыми плоскостями. Построенная топология идеологически близка к топологиям свёрточных нейронных сетей глубокого обучения, но является регулярной с числом слоев, устанавливаемым факторным представлением размерностей изображения и выходной плоскости классов. Алгоритм обучения имеет аналитическое представление, является абсолютно устойчивым и сходится за конечное число шагов. Дополнительные плоскости расширяют информационную емкость перестраиваемого преобразования до максимальной. Управление плоскостями в режиме обучения и обработки реализуется числовыми кодами координат выходной плоскости

Ключевые слова: быстрое перестраиваемое преобразование; нейронная сеть; быстрое обучение; свёрточная нейронная сеть; растровые изображения; плоскости нейронных слоев

I. ВВЕДЕНИЕ

Технология глубокого обучения предполагает наличие процесса конфигурационного усложнения информативных признаков в последовательности нейронных слоев. Начиная с неокогнитрона К. Фукусима [1] предложено множество вариантов реализации этой идеи. Одним из удачных решений являются архитектура свёрточных нейронных сетей [2], которая показала высокую эффективность при решении различных задач. Отличительной особенностью этой архитектуры является наличие в свёрточных слоях нескольких каналов обработки данных (называемых картами или плоскостями). В каждой плоскости выполняется свертка выходного образа предшествующего слоя с фиксированным ядром небольшой размерности. Свёрточные слои чередуются со слоями пулинга, кратно снижающих размерность пространства признаков. Слои пулинга не являются обязательными и существуют варианты их полного устранения из архитектуры сети [3]. Второй отличительной особенностью архитектуры является использование полулинейные функции активации, выполняющих роль коммутационных ключей, управляемых значениями переменных скрытых слоев.

Недостатком архитектуры свёрточных сетей является отсутствие теоретически-обоснованных методов выбора структуры сети и параметров ядер свертки. Существует несколько прекрасно работающих конфигураций сетей для

решения конкретных задач, но не понятно, как нужно строить сеть для новой задачи. До сих пор выбор структуры свёрточной сети является предметом искусства. Второй существенный недостаток связан с временем обучения свёрточных сетей. На типовом процессоре время может варьироваться от нескольких часов до нескольких суток, поэтому для обучения сетей часто используют высокопроизводительные графические процессоры. В работе [4] авторами была предложена идея использовать алгоритмы быстрых преобразований для построения структуры и топологии многослойных персептронных нейронных сетей. Покажем, что этот подход с некоторой модификацией можно использовать также для построения структуры и топологии свёрточных нейронных сетей.

В настоящее время широко известны быстрые алгоритмы линейных преобразований Фурье, Уолша, Хаара и других им подобных. При использовании быстрых алгоритмов выигрыш по вычислительным операциям экспоненциально растет при увеличении размерности преобразования. В конце 20 века возникло направление быстрых перестраиваемых преобразований [5], которые по существу являются нейронными сетями с ограниченными связями и линейными функциями активации. Для обучения подобных нейронных сетей были разработаны методы, абсолютно сходящиеся за конечное число шагов. Число слоев в быстрых преобразованиях и их конфигурация определяются размерностью обрабатываемых образов. Для построения быстрых алгоритмов размерность преобразования должна быть составным числом и чем больше множителей в разложении размерности, тем выше вычислительная эффективность быстрого алгоритма. Несмотря на большое разнообразие быстрых алгоритмов, конфигурации их структур удовлетворяют системному инварианту самоподобия [6]. Как известно таким же свойством обладают фракталы. Поэтому быстрые алгоритмы можно интерпретировать как квазифракталы. Свойство структурной фрактальности позволяет решить одновременно две задачи: реализовать быструю обработку данных и быстрое обучение преобразования.

В настоящей работе будет показано, что небольшая модификация системного инварианта быстрых алгоритмов приводит к структурам свёрточных нейронных сетей. При этом удастся сохранить алгоритм быстрого обучения и расширить информационную емкость сети по распознаваемым образом вплоть до максимально-возможной, определяемой числом нейронов выходного

слоя сети. Предлагаемый вариант архитектуры нельзя назвать сверточными сетями, поскольку в плоскостях нейронных слоев используется более общие преобразования, чем свертка, слои пулинга в преобразовании отсутствуют, но принцип пулинга используется при обучении сети. Все нейроны имеют линейные функции активации, нелинейная обработка существует, но реализуется не за счет функций активации, а за счет коммутации плоскостей нейронных сетей. В какой-то мере это подобно коммутирующим полулинейным функциям активации сверточных нейронных сетей. Управление коммутацией плоскостей осуществляется по координатам нейронов в выходной плоскости сети. Назовем данный класс сетей нейронными сетями с управляемой коммутацией плоскостей (Control Switching of Planes Neural Networks – CSPNN).

II. ТОПОЛОГИЯ ДВУМЕРНЫХ БЫСТРЫХ ПЕРЕСТРАИВАЕМЫХ ПРЕОБРАЗОВАНИЙ

Обозначим через $F(U_y, U_x)$ матрицу изображения размерностью $N_y \times N_x$. При воздействии на изображение линейного преобразования $H(U_y, U_x; V_y, V_x)$ получается массив из $M_y \times M_x$ коэффициентов. Двумерное преобразование выполняется по правилу:

$$S(V_y, V_x) = \sum_{U_y=0}^{N_y-1} \sum_{U_x=0}^{N_x-1} F(U_y, U_x) H(U_y, U_x; V_y, V_x). \quad (1)$$

Необходимым условием существования быстрого алгоритма является возможность мультипликативной декомпозиции значений входных и выходных размерностей преобразования в равное число сомножителей:

$$\begin{aligned} N_y &= p_0^y p_1^y \dots p_{n-1}^y, & M_y &= g_0^y g_1^y \dots g_{n-1}^y, \\ N_x &= p_0^x p_1^x \dots p_{n-1}^x, & M_x &= g_0^x g_1^x \dots g_{n-1}^x. \end{aligned} \quad (2)$$

Индексы x, y здесь означают принадлежность к осям координат исходного изображения, а значение n определяет число слоёв в графе быстрого алгоритма. Используя сомножители декомпозиций (2), координаты точек изображения представим в позиционной системе счисление со смешанными основаниями:

$$\begin{aligned} U_y &= \langle u_{n-1}^y u_{n-2}^y \dots u_1^y u_0^y \rangle, \\ U_x &= \langle u_{n-1}^x u_{n-2}^x \dots u_1^x u_0^x \rangle, \end{aligned} \quad (3)$$

где вес m -го разряда определяется выражением $p_{m-1}^* p_{m-2}^* \dots p_1^* p_0^*$, а u_m^* – является разрядной переменной, принимающей значения $[0, p_m^* - 1]$ (звездочка здесь заменяет индексы x, y). Аналогично можно представить координаты спектральных коэффициентов в плоскости $[V_y, V_x]$:

$$\begin{aligned} V_y &= \langle v_{n-1}^y v_{n-2}^y \dots v_1^y v_0^y \rangle, \\ V_x &= \langle v_{n-1}^x v_{n-2}^x \dots v_1^x v_0^x \rangle, \end{aligned} \quad (4)$$

где вес m -го разряда определяется выражением $g_{m-1}^* g_{m-2}^* \dots g_1^* g_0^*$, а v_m^* является разрядной переменной, принимающей значения $[0, g_m^* - 1]$.

Алгоритм быстрого преобразования обычно представляется в виде графа с топологией различного вида. Поразрядную форму удобно использовать для аналитического описания графа топологии быстрого алгоритма. Например, для топологии «Кули-Тьюки с прореживанием по времени» граф может быть описан в виде лингвистического предложения [6] (топологической модели):

$$\left[\langle u_{n-1}^* u_{n-2}^* \dots u_1^* u_0^* \rangle \langle u_{n-1}^* u_{n-2}^* \dots u_1^* v_0^* \rangle \dots \dots \langle u_{n-1}^* u_{n-2}^* \dots u_{m+1}^* v_m^* v_{m-1}^* v_{m-2}^* \dots v_0^* \rangle \dots \langle v_{n-1}^* v_{n-2}^* \dots v_1^* v_0^* \rangle \right],$$

где словами являются поразрядные представления координатных чисел, а буквами – имена разрядных переменных. Число слов в предложении равно $n+1$. Первое и последнее слова в предложении соответствуют координатам точек терминальных плоскостей представленных выражениями (3) и (4). Промежуточные слова определяют входные U_y, U_x и выходные координаты V_y, V_x в плоскостях внутренних слоев быстрого алгоритма. Для алгоритма с замещением значений дополнительно выполняется условие:

$$U_y^{m+1} = V_y^m, \quad U_x^{m+1} = V_x^m. \quad (5)$$

Граф топологии в слое m содержит базовые операции (нейронные ядра) $W_{i_x^m, i_y^m}^m(u_m^x u_m^y; v_m^x v_m^y)$, представляющие собой четырехмерные матрицы размерности $[p_m^y, p_m^x; g_m^y, g_m^x]$. Где поразрядные выражения индексов ядер слоя m для выбранной топологии имеют вид:

$$\begin{aligned} i_x^m &= \langle u_{n-1}^x u_{n-2}^x \dots u_{m+1}^x v_{m-1}^x v_{m-2}^x \dots v_0^x \rangle, \\ i_y^m &= \langle u_{n-1}^y u_{n-2}^y \dots u_{m+1}^y v_{m-1}^y v_{m-2}^y \dots v_0^y \rangle. \end{aligned} \quad (6)$$

Выражение (6) является аналитическим представлением системного инварианта быстрых алгоритмов [6]. В общем случае топологии для направлений x и y могут быть различными. Взаимосвязь между базовыми операциями определяется структурной моделью быстрого преобразования, где каждой вершине соответствует нейронное ядро.

Для выбранной топологии граф структурной модели описывается лингвистическим предложением:

$$\left[\langle u_{n-1}^* u_{n-2}^* \dots u_1^* \rangle \langle u_{n-1}^* u_{n-2}^* \dots u_2^* v_0^* \rangle \dots \dots \langle u_{n-1}^* u_{n-2}^* \dots u_{m+1}^* v_{m-1}^* v_{m-2}^* \dots v_0^* \rangle \dots \langle v_{n-1}^* v_{n-2}^* \dots v_1^* v_0^* \rangle \right]$$

Каждое слово в этом предложении определяет номер базовой операции i_m^* в слое m . Число слов в предложении равно n .

На рис. 1 показана структурная модель быстрого двумерного преобразования для размерности изображения 8×8 . Входное изображение поступает на нижний слой, а спектральные коэффициенты получаются в верхнем слое. Вершинам модели соответствуют базовые операции (иначе называемые нейронными ядрами) размерности $[2,2;2,2]$. Ядро в слое m выполняет двумерное преобразование над пространственным блоком размером $p_m^y \times p_m^x$:

$$S^m(V_y^m, V_x^m) = \sum_{u_m^y} \sum_{u_m^x} F^m(U_y^m, U_x^m) W_{i_x^m, i_y^m}^m(u_m^y u_m^x; v_m^y v_m^x) \quad (7)$$

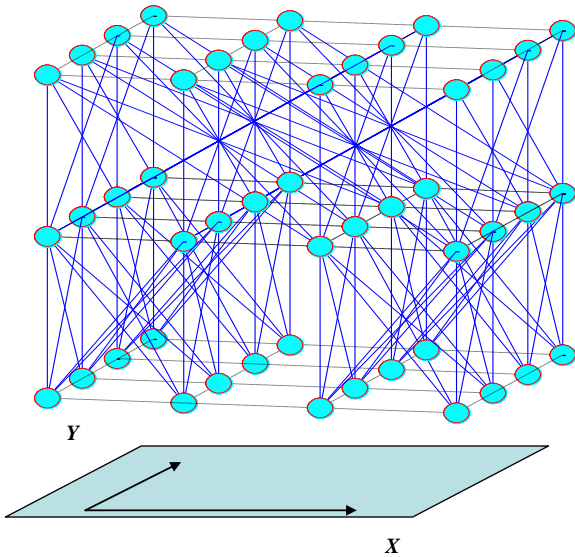


Рис. 1. Структурная модель двумерного быстрого преобразования

III. МУЛЬТИПЛИКАТИВНАЯ ДЕКОМПОЗИЦИЯ ЭЛЕМЕНТОВ МАТРИЦЫ БЫСТРОГО ПРЕОБРАЗОВАНИЯ

Задание конкретных значений для всех разрядных переменных u_m^*, v_m^* (где m пробегает значения $0, 1, \dots, n-1$) определяет некоторый путь в топологическом графе между парой вершин начального и конечного слоя. Из однозначности поразрядного представления координатных чисел следует, что такой путь единственен для каждого парного сочетания пространственных точек входной и выходной плоскости. Это обстоятельство позволяет получить удобное аналитическое выражение, связывающее элементы матрицы быстрого преобразования с элементами ядер. Из выражения (1) следует что

$$H(U_y, U_x; V_y, V_x) = \frac{\partial S(V_y, V_x)}{\partial F(U_y, U_x)}. \quad (8)$$

Дифференцируя (8) по правилу дифференцирования сложной функции получим:

$$H(U_y, U_x; V_y, V_x) = \frac{\partial S^{n-1}}{\partial F^{n-1}} \frac{\partial F^{n-1}}{\partial S^{n-2}} \frac{\partial S^{n-2}}{\partial F^{n-2}} \dots \frac{\partial F^1}{\partial S^0} \frac{\partial S^0}{\partial F^0}.$$

Из условия (5) следует, что для всех m имеет место $\frac{\partial F^m}{\partial S^{m-1}} = 1$, а из (7), – что $\frac{\partial S^m}{\partial F^m} = W_{i_x^m, i_y^m}^m(u_m^y u_m^x; v_m^y v_m^x)$. Таким образом получим, что каждый элемент четырехмерной матрицы преобразования H выражается через элементы ядер в виде произведения:

$$H(U_y, U_x; V_y, V_x) = W_{i_x^{n-1}, i_y^{n-1}}^{n-1}(u_{n-1}^y u_{n-1}^x; v_{n-1}^y v_{n-1}^x) \dots W_{i_x^{n-2}, i_y^{n-2}}^{n-2}(u_{n-2}^y u_{n-2}^x; v_{n-2}^y v_{n-2}^x) \dots W_{i_x^0, i_y^0}^0(u_0^y u_0^x; v_0^y v_0^x), \quad (9)$$

где поразрядные выражения индексов ядер слоя m для выбранной топологии определяются выражением (6).

IV. МУЛЬТИПЛИКАТИВНАЯ ДЕКОМПОЗИЦИЯ ДВУМЕРНЫХ ИЗОБРАЖЕНИЙ

Алгоритм мультипликативной декомпозиции основан на идеях фрактальной фильтрации [6] (в терминологии свёрточных нейронных сетей эта операция называется пулингом). Для двумерного случая фрактальная фильтрация представляет собой кратно-масштабную обработку изображений, последовательно сжимающую его размеры вплоть до единственной точки. Схему фрактальной фильтрации можно представить в виде пирамиды показанной на рис. 2. Основанием пирамиды является исходное изображение, $F(U_y, U_x)$ для которого аргументы U_y и U_x представлены в позиционной системе счисления (выражение (3)). В этом позиционном представлении зафиксируем все разряды кроме двух младших u_0^y и u_0^x . Если варьировать эти разряды по всем возможным значениям, то получим двумерную выборку размером $p_0^y \times p_0^x$. Под фрактальным фильтром понимается произвольный функционал Φ , действующий на этой выборке. Формально это можно записать в виде выражения:

$$F_1(\langle u_{n-1}^y u_{n-2}^y \dots u_1^y \rangle, \langle u_{n-1}^x u_{n-2}^x \dots u_1^x \rangle) = \Phi_{(u_0^y, u_0^x)} \left[F(\langle u_{n-1}^y u_{n-2}^y \dots u_1^y u_0^y \rangle, \langle u_{n-1}^x u_{n-2}^x \dots u_1^x u_0^x \rangle) \right]$$

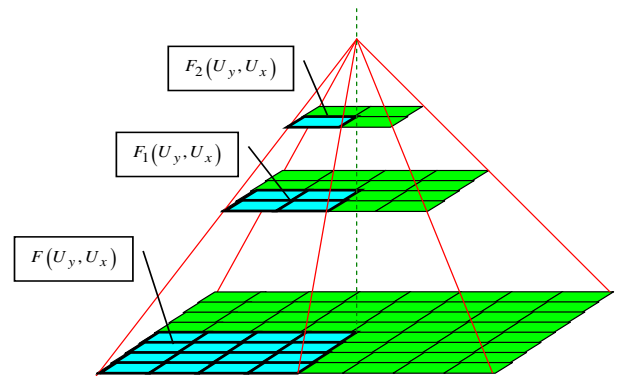


Рис. 2. Схема фрактальной фильтрации изображения

Очевидно, что изображение F_1 будет кратно уменьшенным по размерам по отношению к исходному изображению. Функционалом, например, может быть правило вычисления среднего значения выборки или ее медианы. Исходное изображение теперь можно формально представить в виде произведения:

$$F\left(\langle u_{n-1}^y u_{n-2}^y \cdots u_1^y u_0^y \rangle, \langle u_{n-1}^x u_{n-2}^x \cdots u_1^x u_0^x \rangle\right) = F_1\left(\langle u_{n-1}^y u_{n-2}^y \cdots u_1^y \rangle, \langle u_{n-1}^x u_{n-2}^x \cdots u_1^x \rangle\right) f_{j_y^0 j_x^0}(u_0^y, u_0^x),$$

где $f_{j_y^0 j_x^0}(u_0^y, u_0^x)$ – набор двумерных функций-множителей, зависящих от разрядных переменных u_0^y и u_0^x , а индексы j_y^0, j_x^0 выделяют двумерную функцию из этого набора. Значение этих индексов устанавливается равным значениям аргументов изображения F_1 , т.е. $j_y^0 = \langle u_{n-1}^y u_{n-2}^y \cdots u_1^y \rangle$ и $j_x^0 = \langle u_{n-1}^x u_{n-2}^x \cdots u_1^x \rangle$. Для получения функций множителей достаточно выполнить скалярное деление изображения F на изображение F_1 при варьировании всех разрядных переменных. Изображение F_1 в свою очередь также может быть представлено как произведение изображения F_2 на множители из набора $f_{j_y^1 j_x^1}(u_1^y, u_1^x)$. Повторяя многократно операцию фрактальной фильтрации и разложения, достигнем вершины пирамиды изображений и получим мультипликативную декомпозицию:

$$F\left(\langle u_{n-1}^y u_{n-2}^y \cdots u_1^y u_0^y \rangle, \langle u_{n-1}^x u_{n-2}^x \cdots u_1^x u_0^x \rangle\right) = f_{j_y^{n-1} j_x^{n-1}}(u_{n-1}^y, u_{n-1}^x) f_{j_y^{n-2} j_x^{n-2}}(u_{n-2}^y, u_{n-2}^x) \cdots f_{j_y^1 j_x^1}(u_1^y, u_1^x) f_{j_y^0 j_x^0}(u_0^y, u_0^x), \quad (10)$$

где индексы сомножителей определяются выражениями:

$$j_x^m = \langle u_{n-1}^x u_{n-2}^x \cdots u_{m+1}^x \rangle, \quad j_y^m = \langle u_{n-1}^y u_{n-2}^y \cdots u_{m+1}^y \rangle. \quad (11)$$

V. НАСТРОЙКА ПРИСПОСОБЛЕННЫХ ПРЕОБРАЗОВАНИЙ

Преобразование назовем приспособленным к изображению, если одна из функций преобразования с координатами (V_y, V_x) совпадает с данным изображением. Очевидно, что значение скалярного произведения изображения с этой функцией будет максимальным среди прочих коэффициентов спектральной области преобразования. В этом заключается цель настройки преобразования. Значение координат (V_y, V_x) функции в спектральной плоскости назовем точкой приспособления.

Приспособление может быть осуществлено и к нескольким изображениям. Если сравнить полученную мультипликативную декомпозицию изображения с декомпозицией быстрого преобразования (9), то нетрудно заметить, что они подобны. Причем множество индексов ядер в каждом слое покрывает множество индексов функций-сомножителей. Отсюда следует конструктивный вывод, что быстрое преобразование будет приспособлено,

к изображению, когда ядра преобразования будут приспособлены к функциям-сомножителям разложения (10). Настройка ядер преобразования определяется правилом:

$$W_{i_x^m i_y^m}^m(u_m^y, u_m^x; v_m^y, v_m^x) = f_{j_x^m j_y^m}(u_m^y, u_m^x). \quad (12)$$

Сравнивая выражения (6) для i_x^m, i_y^m и (11) для j_x^m, j_y^m , можно сделать вывод, что количество компонентов в мультипликативном разложении изображения (10) и количество ядер преобразования совпадают для слоя $m=0$ (т.е. имеют место равенства $i_x^0 = j_x^0$ и $i_y^0 = j_y^0$), и меньше числа ядер для всех остальных слоев. Поэтому при настройке часть степеней свободы преобразования не используется. Свободно варьируемыми переменными являются разрядные переменные v_0^y, v_0^x , и, следовательно, ядро может быть приспособлено одновременно к $D = g_0^y g_0^x$ изображениям. Остальные слои имеют большее число степеней свободы и не могут ухудшить это значение. Таким образом, можно заключить, что быстрое преобразование можно приспособить не более чем к D различным изображениям. На этом возможности данного алгоритма настройки исчерпываются. Значение D назовём уровнем приспособленности преобразования.

VI. РЕГУЛЯРНЫЕ ПРЕОБРАЗОВАНИЯ С ДОПОЛНИТЕЛЬНЫМИ ПЛОСКОСТЯМИ

Оставаясь в рамках рассмотренной топологии, не использованные при настройке степени свободы можно доопределить несколькими способами (более детально см. [6]). В этом случае уровень приспособленности не меняется, но при этом видоизменяются остальные функции преобразования.

Рассмотрим альтернативное решение, которое заключается в расширении топологии дополнительными плоскостями, так чтобы в новой топологии использовать оставшиеся степени свободы для увеличения уровня приспособленности. В этом случае число вычислительных операций возрастает, но структура сети остается регулярной.

Уточним вначале правило выбора настраиваемых ядер для прежней топологии. Выразим координаты точки приспособления в позиционной системе счисления, обозначив разрядные переменные через y и x :

$$V_y = \langle y_{n-1}, y_{n-2} \cdots y_0 \rangle, \quad V_x = \langle x_{n-1}, x_{n-2} \cdots x_0 \rangle \quad (13)$$

Фиксированные значения разрядных переменных y_m, x_m соответствуют переменным v_m^y, v_m^x , поэтому (как это следует из (6)) при выборе данной точки приспособления, согласно правилу (12) должны настраиваться только ядра с номерами:

$$i_x^m = \langle u_{n-1}^x u_{n-2}^x \cdots u_{m+1}^x x_{m-1} x_{m-2} \cdots x_0 \rangle, \quad (14)$$

$$i_y^m = \langle u_{n-1}^y u_{n-2}^y \cdots u_{m+1}^y y_{m-1} y_{m-2} \cdots y_0 \rangle.$$

В частности для $m = 0$ имеем:

$$i_x^0 = \langle u_{n-1}^x u_{n-2}^x \dots u_1^x \rangle,$$

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

Т.е. независимо от выбора точек приспособления все ядра нулевого слоя всегда должны настраиваться. При этом уровень приспособленности преобразования ограничен значением D .

Для увеличения уровня приспособленности введем в каждом слое дополнительные плоскости, структурно копирующие основную плоскость слоя. Номера дополнительных плоскостей в пределах слоя определим правилом:

$$\pi_m = \langle x_{n-1} x_{n-2} \dots x_{m+1}, y_{n-1} y_{n-2} \dots y_{m+1} \rangle \quad (15)$$

Очевидно, что максимальное количество дополнительных плоскостей будет в нулевом слое, а по мере увеличения номера слоя количество дополнительных плоскостей будет уменьшаться, и в последнем слое получим $\pi_{n-1} = \langle \rangle$, т.е. дополнительных плоскостей не будет совсем. Таким образом, в новой топологии плоскость последнего слоя останется прежней, а в младших слоях появятся дополнительные плоскости.

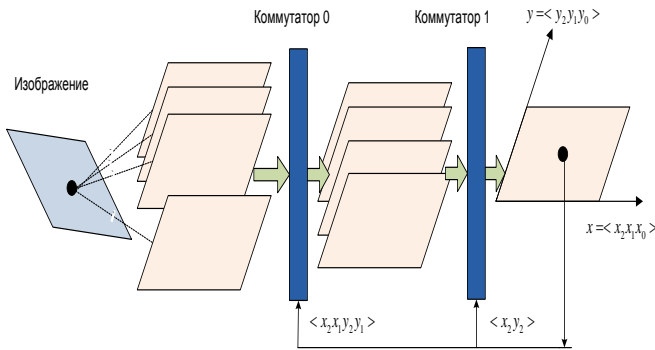


Рис. 3. Архитектура регулярной нейронной сети с дополнительными плоскостями

Архитектура сети с дополнительными плоскостями показана на рис. 3. Входное изображение подается одновременно на все плоскости входного слоя, слои разделены коммутаторами, которые управляются разрядными переменными координатных чисел выходного класса.

Поскольку правило порождения новых плоскостей не противоречит условию (14), то для настройки ядер преобразования можно использовать прежнее правило (12), расширив его для дополнительных плоскостей:

$$W_{i_x^m, i_y^m}^m \langle \pi_m \rangle (u_m^x u_m^y v_m^x v_m^y) = f_{j_x^m, j_y^m}^k (u_m^y, u_m^x). \quad (16)$$

Индекс k в правой части нумерует точку приспособления. Для $m = 0$ имеем $i_x^0 = j_x^0$ и $i_y^0 = j_y^0$, а варьируемыми переменными в левой части являются

номер плоскости $\pi_0 = \langle x_{n-1} x_{n-2} \dots x_1, y_{n-1} y_{n-2} \dots y_1 \rangle$ и разрядные переменные $v_0^y v_0^x$. Вместе они покрывают весь диапазон координат выходной плоскости. Этому диапазону отвечают возможные значения индекса k в правой части (16). Остальные слои не ухудшают степень приспособленности. Таким образом, преобразование с дополнительными плоскостями может быть приспособлено к $D_\pi = M_y \times M_x$ изображениям, т.е. каждой точке выходной плоскости будет соответствовать один опорный образ.

На рис. 4 показаны двумерные функции преобразования приспособленного к усреднённым образам рукописных цифр [2]. Для данного преобразования выбрано:

$$N_y = N_x = p_0 p_1 p_2 = 4 \cdot 3 \cdot 2 = 24, \quad M_y = M_x = g_0 g_1 g_2 = 2 \cdot 2 \cdot 2 = 8$$

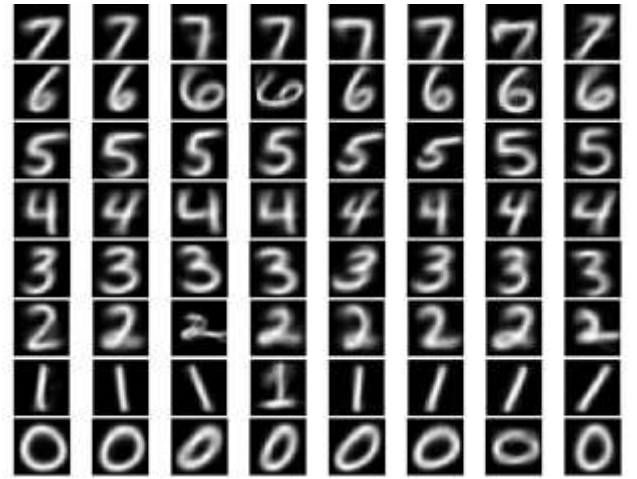


Рис. 4. Двумерные функции приспособленного преобразования

При выполнении преобразования входное изображение поступает одновременно на все входные плоскости. Если изображение отвечает одной из точек приспособления, то значение этого коэффициента спектральной плоскости будет максимально. Результат выполнения преобразования над образом цифры «0» показан на рис. 5.

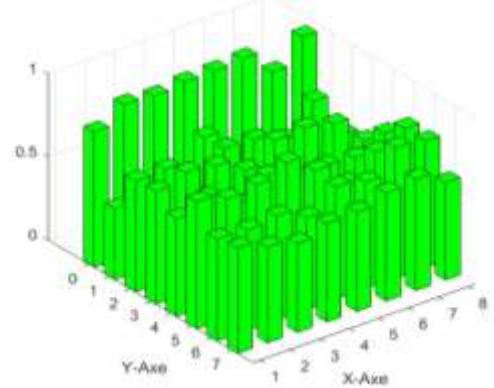


Рис. 5. Результат выполнения двумерного преобразования

Наглядно видно, что коэффициенты отвечающие подклассам цифры «0» имеют максимальные значения.

VII. Выводы

Регулярные перестраиваемые преобразования обладают уникальной возможностью аналитического представления топологии реализующей сети, что позволяет разработать алгоритмы обучения, абсолютно-сходящиеся за конечное число шагов. В работе показано, что реализующая топология легко расширяется дополнительными плоскостями, при этом число распознаваемых образов кардинально возрастает и покрывает все элементы выходной плоскости. Причем расширение топологии не нарушает принципа построения обучающего алгоритма. Построенная топология идеологически близка к топологиям свёрточных сетей глубокого обучения [2], но является регулярной. Представленное решение позволяет дать конструктивный ответ на принципиальные вопросы нейронных сетей

глубокого обучения: как выбрать топологию и как сократить время обучения сети.

СПИСОК ЛИТЕРАТУРЫ

- [1] Fukushima K., Miyake S., Takayuki I. Neocognitron: A neural network model for a mechanism of visual pattern recognition. *IEEE Transaction on Systems, Man and Cybernetics* SMC-13(5):826-34. 1983.
- [2] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel: Backpropagation Applied to Handwritten Zip Code Recognition, *Neural Computation*, 1(4):541-551, Winter 1989.
- [3] Springenberg, Jost Tobias; Dosovitskiy, Alexey; Brox, Thomas & Riedmiller, Martin (2014-12-21), "Striving for Simplicity: The All Convolutional Net", arXiv:1412.6806 <https://arxiv.org/pdf/1412.6806>
- [4] Дорогов А.Ю., Алексеев А.А. // Математические модели быстрых нейронных сетей. В сб. научн. Тр. СПбГЭТУ "Системы управления и обработки информации". Вып.490, 1996, с.79-84.
- [5] Солодовников А.И., Спиваковский А.М. Основы теории и методы спектральной обработки информации: Учеб. пособие. Л.: Изд-во ЛГУ, 1986. 272с.
- [6] Дорогов А.Ю. Теория и проектирование быстрых перестраиваемых преобразований и слабосвязанных нейронных сетей. СПб.: «Политехника», 2014. 328 с. <http://dorogov.su>