

УДК 519.9

О.И. Нечаева

СГЛАЖИВАНИЕ АДАПТИВНЫХ СЕТОК, ПОСТРОЕННЫХ МЕТОДОМ САМООРГАНИЗУЮЩИХСЯ КАРТ

Статья посвящена проблеме сглаживания адаптивных сеток, полученных с помощью нейросетевой модели самоорганизующихся карт Кохонена (SOM). Чтобы добиться достаточного уровня гладкости сеток необходимо увеличить радиус обучения в модели SOM, что в свою очередь приводит к нежелательному граничному эффекту. В статье предлагается методика сглаживания, позволяющая избежать граничный эффект при любом радиусе обучения.

Ключевые слова: *Адаптивные сетки, сглаживание, самоорганизующиеся карты Кохонена, граничный эффект.*

Построение адаптивных сеток является одной из актуальных областей научных исследований, которая привлекает в настоящее время многих исследователей. Адаптивные сетки, являясь дискретной моделью пространственной области, получили важнейшие применения при решении сложных задач численного моделирования [1]. Кроме того, они широко используются при обработке изображений, визуализации данных, в графических приложениях и т.д. [2].

Использование адаптивных сеток в задачах численного моделирования позволяет повысить точность приближенного решения задачи без существенного увеличения числа узлов. Среди всех типов сеток можно выделить важный класс, в котором адаптивные сетки получаются в результате отображения некоторой вычислительной области на физическую, переводящего зафиксированную, обычно равномерную, сетку в адаптивную с заданным распределением плотности. К методам построения сеток этого класса относятся такие традиционные методы [3], как метод эквираспределения, метод Томпсона, эллиптический метод и т.д. Для получения качественных адаптивных сеток все эти методы, а также алгебраические методы и методы конформных отображений [4], в конечном счете, требуют решения сложных систем нелинейных дифференциальных уравнений (ДУ) с частными производными. Необходимость решения ДУ накладывает ряд ограничений на возможности этих методов, к числу которых относятся жесткий контроль над способами задания физической области и функции плотности сетки, контроль над свойствами граничных и начальных данных, проблемы при попытках автоматизации работы с этими методами и их эффективного распараллеливания.

Одним из альтернативных и многообещающих подходов [5] к построению адаптивных сеток (не только рассматриваемого класса) является использование стохастических нейросетевых моделей самоорганизации, таких как самоорганизующиеся карты Кохонена (Self-Organizing Maps, SOM), растущий нейронный газ (Growing Neural Gas, GNG), растущие клеточные структуры (Growing Cell Structures, GCS) [6]. Принципы самоорганизации, заложенные в этих моделях, позволяют строить сетки с произвольными начальными данными и автоматической расстановкой узлов по границе физической области; стохастическая природа моделей снимает требования с функции плотности сетки, так как плотность контро-

лируется вероятностным распределением; внутренний параллелизм, присущий нейросетевым моделям, делает возможным эффективно распараллеливать алгоритмы построения адаптивных сеток [5].

Однако стохастический подход имеет и свои проблемы, к числу которых, например, относится недостаточная гладкость результирующих сеток. Целью данной работы является разработка методики сглаживания адаптивных сеток, получаемых с помощью нейросетевых моделей. Эта методика иллюстрируется на примере построения адаптивных сеток упомянутого выше класса с помощью модели SOM.

Было установлено, что гладкость сеток напрямую зависит от радиуса обучения нейронной сети SOM. Чем больше радиус, тем выше гладкость адаптивной сетки. Но при увеличении радиуса обучения возникает граничный эффект, который проявляется в том, что сетка перестает аппроксимировать границы физической области, сосредотачиваясь вблизи центра области. Причиной такого эффекта является несимметричность расположения узлов сетки, находящихся в окрестности граничных узлов. Таким образом, основной задачей при достижении требуемой гладкости сеток является преодоление граничного эффекта за счет балансирования упомянутой несимметричности, чтобы обеспечить возможность использования большого радиуса обучения.

Граничный эффект – это известная проблема нейросетевой модели SOM. Несколько подходов по его преодолению было предложено ранее, к их числу, например, относятся правило весов Кохонена [6] и реализация модели SOM на сферической решетке Риттера [8]. Однако первый подход не учитывает размер радиуса обучения, который напрямую влияет на граничный эффект и является ключевым параметром при сглаживании сетки. Второй подход неприменим при построении адаптивных сеток рассматриваемого класса, так как требует изменения структуры зафиксированной сетки. Методика, предлагаемая в данной статье, не требует изменения структуры зафиксированной сетки и воздействует только на алгоритм обучения для модели SOM, сохраняя все перечисленные ее свойства. Автор надеется, что предложенная в данной статье методика окажется полезной не только при построении адаптивных сеток, но и в других многочисленных применениях модели SOM, таких, как обработка изображений, сжатие данных, реконструкция поверхностей и др.

1. Использование модели SOM для построения адаптивных сеток

Пусть в евклидовом пространстве R_G^n с системой координат $(x^1, \dots, x^n)$ задана физическая область G , на которой требуется построить адаптивную сетку. Пусть Q – это вычислительная область в пространстве R_Q^k , $k \leq n$, с системой координат $(q^1, \dots, q^k)$. В вычислительной области Q зафиксирована некоторая сетка, множество узлов которой обозначается $Q_N = \{q_1, \dots, q_N\}$, где N – количество узлов сетки, $q_i \in Q$, $q_i = (q_i^1, \dots, q_i^k)$, $i = 1, \dots, N$. В данной статье предполагается, что зафиксированная сетка является равномерной прямоугольной, что, вообще говоря, необязательно при построении сетки с использованием модели SOM. Пусть d_Q – это шаг зафиксированной сетки Q_N по горизонтальному и вертикальному направлениям. Требуемая плотность адаптивной сетки задается функцией плотности сетки $w: G \rightarrow R^+$.

Согласно классической постановке задачи построения адаптивных сеток рассматриваемого класса, требуется найти отображение Q на G , которое переводит сетку Q_N , трансформируя ее, в искомую адаптивную сетку G_N с заданной функцией плотности. При отображении граничные узлы сетки Q_N должны переводиться в граничные узлы сетки G_N , распределенные по границе G . Пусть N_b – это число граничных узлов сетки, а N_{int} – число внутренних узлов.

Для использования нейросетевой модели SOM при построении адаптивных сеток нейроны ставятся в соответствие узлам сетки. При этом количество нейронов совпадает с числом узлов N , положения нейронов в нейронном слое модели SOM задаются координатами узлов q_i зафиксированной сетки Q_N , а их веса соответствуют координатам x_i искомой адаптивной сетки G_N , которые требуется найти в процессе обучения SOM. Таким образом, нейрон в модели SOM рассматривается как пара $e_i = (q_i, x_i)$. Каждые два нейрона e_i и e_j , $i, j = 1, \dots, N$, в нейронном слое связаны латеральной связью, вес которой тем меньше, чем больше расстояние между этими нейронами $\|q_i - q_j\|$. В качестве входных данных для модели SOM используются случайные точки области G , сгенерированные с заданным вероятностным распределением $p(x)$. Это распределение пропорционально функции плотности сетки $w(x)$ и имеет вид $p(x) = w(x) / \int_G w(z) dz$.

Алгоритм обучения нейронной сети SOM, предложенный Кохоненм [6], заключается в следующем. На каждой итерации t алгоритма обучения модели SOM, генерируется случайная точка y из области G , выбирается ближайший к ней узел сетки $x_m(t)$, и все нейроны корректируют свои веса в соответствии с формулой

$$x_i(t+1) = x_i(t) + \delta(t) \eta_{q_m}(t, q_i)(y - x_i(t)), \quad (1)$$

где $\delta(t) \in [0, 1]$ отвечает за шаг обучения, а $\eta_{q_m}(t, q_i) \in [0, 1]$ задает вес латеральной связи между нейронами e_m и e_i . Эти две функции контролируют величины сдвигов узлов x_i в области G по направлению к случайной точке y , существенно влияют на качество результирующих сеток и скорость процесса построения.

Следуя [5], функция шага обучения в данной работе выбиралась в виде $\delta(t) = t^{-0.2} \chi(t)$, где $\chi(t) = 1 - e^{5(t-T)/T}$ и T – это максимальное количество итераций, которое фиксируется до начала итерационного процесса пропорционально N (например, $T = 10N$). Пусть $B_\gamma(q)$ – это замкнутая окрестность точки q в вычислительной области радиуса γ , т.е.

$$B_\gamma(q) = \{p \in R_Q^k \mid d(p, q) \leq \gamma\},$$

где $d(\cdot, \cdot)$ – это евклидово расстояние. Функция, задающая веса латеральных связей, в данной работе имеет следующий вид:

$$\eta_{q_m}(t, q_i) = s \left(\frac{d(q_m, q_i)}{r(t)} \right)^2,$$

где $s \in (0, 1)$ – заранее зафиксированная, близкая к нулю константа (например, $s = 10^{-5}$), а $r(t)$ – это радиус обучения на шаге t , смысл которого заключается в том, что веса латеральных связей между нейроном e_m и всеми нейронами e_i , расположенными в пространстве R_Q^k внутри сферы $B_{r(t)}(q_m)$ радиуса $r(t)$, не меньше s , т.е. $\forall q_i \in B_{r(t)}(q_m) \Rightarrow \eta_{q_m}(t, q_i) > s$. Функция $\eta_{q_m}(t, q_i)$ удовлетворяет также

следующим условиям:

- (1) если $d(q_m, q_i) = r(t)$, то $\eta_{q_m}(t, q_i) = s$;
- (2) $\eta_{q_m}(t, q_i) = \eta_{q_i}(t, q_m)$, т.е. латеральная связь между q_m и q_i симметрична;
- (3) $\eta_{q_m}(t, q_m) = 1$.

Таким образом, на каждой итерации наибольшее приращение получают веса нейрона победителя, тогда как остальные нейроны меняют веса тем меньше, чем дальше они расположены от победителя в нейронном слое. Радиус обучения меняется в течение итерационного процесса по следующему правилу:

$$r(t) = r(T) + \chi(t)(r(1)0,05^{t/T} - r(T))t^{-0,25}.$$

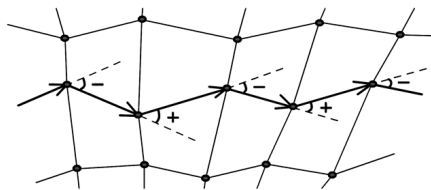
Здесь $r(1)$ и $r(T)$ – начальный и конечный радиусы, $r(1) > r(T)$.

Как показано в [5], приведенный выше алгоритм обучения модели SOM в чистом виде неприменим для построения сеток. Для того чтобы получать приемлемые адаптивные сетки, в [5] предлагается композиционный алгоритм построения сеток, основанный на чередовании применения алгоритма SOM к граничным и внутренним узлам. В результате граничные узлы автоматически растягиваются по границе, а внутренние узлы, согласуясь с граничными, заполняют внутренность области. При использовании такой композиции можно не только исключить нарушения топологии сетки и значительно снизить количество узлов, вышедших за границу области в случае сложных невыпуклых областей, но и преодолеть граничный эффект при малом радиусе обучения. Так, в конце итерационного процесса рекомендуется использовать такие значения радиуса $r(T)$, чтобы окрестность $B_{r(t)}(q_i)$ захватывала только ближайших соседей узла q_i . Однако такой маленький радиус обучения ведет к негладким адаптивным сеткам, что может явиться причиной снижения точности последующих расчетов на этих сетках. Связь гладкости результирующих сеток, радиуса обучения и граничного эффекта подробно рассматривается в следующем пункте.

2. Измерение гладкости и демонстрация граничного эффекта

Для измерения гладкости четырехугольных адаптивных сеток можно рассматривать ломаные, являющиеся образом вертикальных и горизонтальных линий зафиксированной сетки Q_N при отображении, которое осуществляет SOM. Сегменты этих ломаных связывают соответствующие узлы сетки G_N . В двумерном случае гладкость ломаной может быть измерена значениями синусов углов между сегментами вдоль ломаной. Чем меньше число перемен знаков и амплитуда этих синусов, тем ломаная является более гладкой. В трехмерном случае, поскольку сегменты ломаной у трехмерной сетки, как правило, не лежат в одной плоскости, удобно измерять абсолютное значение синусов без знака.

Для демонстрации влияния радиуса обучения на гладкость сеток и на граничный эффект был поставлен следующий эксперимент. Сначала с помощью композиционного алгоритма была построена сетка с искусственно маленьким финальным радиусом обучения $r(T) = 0,5d_Q$ (т.е. к концу итерационного процесса окрестность $B_{r(t)}(q_m)$ даже не захватывала ближайших соседей). Далее, внутренние узлы



сетки в течение некоторого количества итераций обрабатываются процедурой обучения SOM с постоянным коэффициентом обучения, в котором шаг был зафиксирован как $\delta = 0,005$, а радиус – в виде $r(T) = 10d_0$. При этом граничные узлы не меняли свое положение, но имели право становиться победителями, чтобы обеспечить согласованность внутренних и граничных узлов (такой же механизм согласования используется и в композиционном алгоритме). Описанный эксперимент проводился в двумерном и трехмерном случаях.

Сетки, построенные композиционным алгоритмом с маленьким радиусом (рис. 1, *a* и *d*), негладкие даже зрительно, и на графиках рис. 1, *c* и *f* серые линии показывают большие значения синусов с многочисленными переменами знаков вдоль одной из ломаных сетки. Гладкость обработанных сеток (рис. 1, *b* и *e*) намного лучше, что показывают графики. Однако полученные сетки неприменимы для вычислений из-за сильного граничного эффекта, проявившегося из-за большого радиуса обучения. Итак, главной задачей при получении достаточного уровня гладкости сеток является борьба с граничным эффектом.

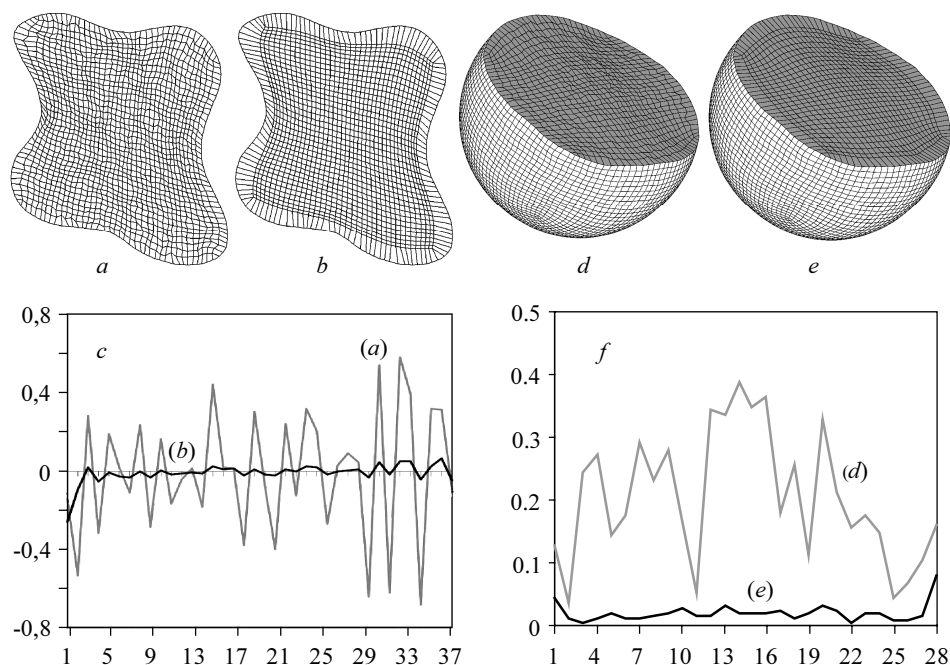


Рис. 1. Измерение гладкости сеток; *a* и *d* – сетки, полученные композиционным алгоритмом с искусственно маленьким конечным радиусом обучения $r(T) = 0,5d_0$; *b* и *e* – сетки, полученные применением процедуры SOM с постоянным коэффициентом обучения при $r(T) = 10d_0$; *c* и *f* – измерения гладкости сеток

3. Балансировка граничного эффекта

После работы композиционного алгоритма все узлы сетки распределены по физической области в соответствии с заданным распределением плотности. Следовательно, предполагается, что для этой сетки выполняется условие равноправия узлов [8]: каждый узел имеет одинаковые шансы стать победителем при случайно сгенерированной входной точке, и соответствующая вероятность равна $1/N$. В

теории нейронных сетей, это условие играет роль критерия качества построенного модели SOM отображения.

Граничный эффект возникает из-за несимметричности латеральных связей у нейронов, которые расположены близко к границе вычислительной области. Пусть q_i – это внутренний нейрон, для которого расстояние до границы вычислительной области больше радиуса обучения r . Структура сетки Q_N такова, что все нейроны q_j из $B_r(q_i)$ располагаются симметрично относительно q_i . Согласно выбору коэффициента обучения, значения весов латеральных связей $\eta_{q_j}(q_i)$ также располагаются

симметрично относительно q_i . Если при этом учитывать условие равноправия узлов, то на нейрон q_i с одинаковой вероятностью может быть оказано воздействие любым другим нейроном q_j . В физической области это означает, что узел x_i имеет одинаковую вероятность сдвинуться симметрично во всех направлениях под влиянием нейронов из $B_r(q_i)$. Так как s близко к нулю, то предполагается, что взаимное влияние между нейронами q_i и $q_j \notin B_r(q_i)$ пренебрежительно мало.

Если расстояние от q_i до границы вычислительной области меньше r , то в окрестности $B_r(q_i)$ расположено недостаточно нейронов для симметрии. В этом случае большинство нейронов в $B_r(q_i)$ заставляют узел x_i двигаться в основном к центру физической области. Поэтому для балансировки асимметрии узел x_i вынужден отодвигаться от границы G .

Для оценки асимметрии предлагается рассмотреть следующую характеристику нейрона e_i :

$$\alpha_i = \sum_{j \in B_r(q_i)} \eta_{q_j}(q_i). \quad (2)$$

Для каждого нейрона эта характеристика равна сумме весов латеральных связей со всеми другими узлами из окрестности $B_r(q_i)$. Если q_i расположен близко к границе Q , то в сумме недостаточно слагаемых. Поэтому α_i убывает вблизи границы Q . На рис. 2. показан график величины α_i в случае равномерной прямоугольной зафиксированной сетки.

Все узлы, расположенные на расстоянии от границы, большем r , имеют равные значения характеристики α_i . Для балансировки граничного эффекта необходимо, чтобы эта характеристика была одинаковой для всех нейронов. Предлагается методика, которая позволяет использовать граничные узлы в качестве представителей недостающих нейронов за пределами вычислительной области Q .

Можно представить, что с каждым граничным нейроном связано K виртуальных нейронов, расположенных за пределами вычислительной области. Эти виртуальные нейроны не существуют в алгоритме и только помогут понять идею, лежащую в основе предлагаемой методики. Точные положения виртуальных нейронов неизвестны. Известно только то, что расстояние между k -м виртуальным нейроном и соответствующим ему граничным нейроном с положением q_m равно kd_Q , $k = 1, \dots, K$, где $K = \lceil r/d_Q \rceil$ и $\lceil a \rceil = \arg \min_{n \in \mathbb{Z}} (a < n)$ – это наименьшее целое число, которое больше a .

Чтобы встроить виртуальные нейроны в процесс обучения, требуется ответить на следующие вопросы: (1) при каких условиях виртуальный нейрон может стать победителем? (2) чему равны веса латеральных связей между нейронами q_i , $i = 1, \dots, N$, и виртуальными нейронами? (3) каковы направление и величина сдвигов узлов сетки в физической области, когда победителем является виртуальный нейрон?

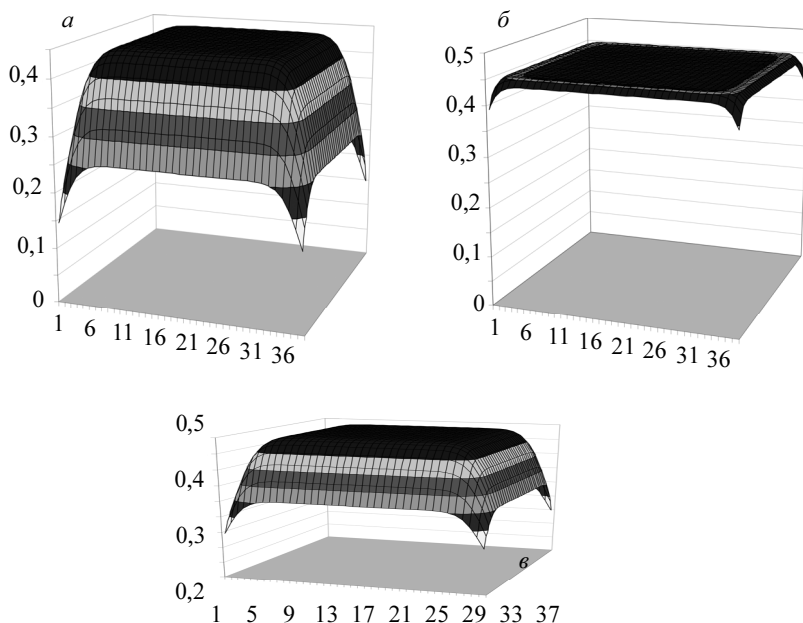


Рис. 2. Характеристика симметричности латеральных связей; a – значения α_i без виртуальных нейронов; $б$ – значения $\bar{\alpha}_i$ с виртуальными нейронами; $в$ – разрез поверхностных графиков (a) и ($б$), где пунктирная серая линия соответствует (a), а сплошная черная линия ($б$)

Ответ на вопрос (1). В случае наличия виртуальных нейронов, выбор победителя не может быть основан только на близости к случайной точке, так как за пределами физической области нет точек. Поэтому на каждой итерации, во-первых, необходимо решить, из какого типа нейронов будет выбираться победитель. Так как предполагается, что выполняется условие равноправия нейронов, виртуальные нейроны имеют ту же вероятность стать победителем, что и все остальные нейроны. Вероятность того, что станет победителем виртуальный нейрон, равна $N_b K / (N_b K + N_{int})$, и таким образом внутренний нейрон может стать победителем с вероятностью $1 - N_b K / (N_b K + N_{int})$. Чтобы выбрать победителя среди виртуальных нейронов, генерируется случайная точка y на границе G , выбирается ближайший к y граничный узел, после чего среди виртуальных нейронов, соответствующих выбранному граничному узлу, равновероятно выбирается k -й виртуальный нейрон и назначается победителем.

Ответ на вопрос (2). Для задания весов латеральных связей между виртуальными и внутренними нейронами необходимо знать расстояние между ними в вычислительной области. Предполагается, что расстояние между k -м виртуальным нейроном и внутренним нейроном с положением q_i равно $d(q_m, q_i) + kd_Q$, где q_m – это положение граничного нейрона, соответствующего виртуальному нейрону. Это расстояние приближенное, так как точное положение виртуального нейрона в вычислительной области не задается. Вес латеральной связи между k -м виртуальным нейроном, соответствующим граничному нейрону e_m , и нейроном e_i равен

$$\eta_{q_m,k}(q_i) = s^{\left(\frac{d(q_m,q_i)+kd_Q}{r}\right)^2}.$$

Ответ на вопрос (3). Для задания направлений и величин сдвигов узлов сетки в физической области при виртуальном победителе предлагается использовать случайную точку y на границе G , которая была сгенерирована для выбора победителя среди виртуальных узлов. Еще раз стоит подчеркнуть, что на стадии сглаживания могут двигаться только внутренние узлы сетки. Для каждого внутреннего узла x_i направление его сдвига задается вектором $y - x_i(t)$, т.е. узел x_i сдвигается в направлении точки y , расположенной на границе G . Величина сдвига равна $\delta \eta_{q_m,k}(q_i) \cdot v_i(t) \cdot d(y, x_i(t))$, где $v_i(t) = 1 + kd_Q / d(q_m, q_i)$ и q_m – это граничный узел, который соответствует виртуальному победителю. Это значение было найдено на основе предположения, что отношение между $d(q_m, q_i)$ и $d(y, x_i(t))$ равно отношению между $d(q_m, q_i) + kd_Q$ и $v_i(t) \cdot d(y, x_i(t))$. Так как правило применяется только ко внутренним узлам, то $d(q_m, q_i) \neq 0$.

Учитывая наличие виртуальных нейронов, характеристика (2) изменяется следующим образом:

$$\bar{\alpha}_i = \sum_{j=1}^N \eta_{q_j}(q_i) + \sum_{k=1}^K \sum_{m=1}^{N_b} \eta_{q_m,k}(q_i), \quad (3)$$

где $m = 1, \dots, N_b$ – это индекс граничного нейрона. На рис. 2, в показаны графики α_i и $\bar{\alpha}_i$ для одной из линий сетки. Можно видеть, что $\bar{\alpha}_i$ почти постоянна для всех нейронов по сравнению с α_i . Таким образом, предложенная методика балансирует асимметрию латеральных связей вблизи границы.

4. Алгоритм сглаживания

В алгоритме сглаживания применяется алгоритм SOM с постоянным коэффициентом обучения, не зависящим от номера итерации. Радиус обучения r выбирается сравнительно большим, однако его значение может быть выбрано в зависимости от кривизны границы G . Шаг обучения δ должен быть маленьким, так как при сглаживании нужна только тонкая подстройка сетки.

Кроме того, когда победителем является виртуальный нейрон, воображаемая случайная точка находится за пределами физической области, что может привести к тому, что узлы сетки пересекут границу G . Для исключения такой ситуации шаг обучения должен удовлетворять следующему условию: $\delta(1 + kd_Q / d(q_m, q_i)) < 1$ для каждого граничного нейрона e_m и внутреннего нейрона e_i . Это значит, что

$$\delta < \min_{m,i,k} \left(\frac{d(q_m, q_i)}{d(q_m, q_i) + kd_Q} \right) = \frac{d_Q}{d_Q + Kd_Q} = \frac{1}{K+1}.$$

Алгоритм сглаживания

В течение фиксированного числа итераций повторяются следующие операции.

1. Реализуется случайная величина α , равномерно распределённая на $[0, 1]$.
2. Если $\alpha \in [0, N_{\text{int}} / (N_b K + N_{\text{int}})]$, то выполняются следующие операции:

а) Генерируется случайная точка y из G в соответствии с вероятностным распределением $p(x)$.

б) Среди всех узлов сетки выбирается узел-победитель $x_m(t)$. Если $x_m(t)$ является граничным узлом, то случайная точка заменяется весами нейрона-победителя $y := x_m(t)$.

в) Проводится корректировка весов только внутренних нейронов по следующему правилу:

$$x_i(t+1) = x_i(t) + \delta \eta_{q_m}(q_i)(y - x_i(t)).$$

3. Если $\alpha \in (N_{\text{int}} / (N_b K + N_{\text{int}}), 1]$, то выполняются следующие операции:

а) Генерируется случайная точка y с границы G в соответствии с вероятностным распределением $p(x)|_{\partial G}$.

б) Среди граничных узлов выбирается узел $x_m(t)$, ближайший к точке y .

в) Среди чисел $\{1, \dots, K\}$ равновероятно выбирается число k .

г) Проводится корректировка весов только внутренних нейронов по следующему правилу:

$$x_i(t+1) = x_i(t) + \delta \eta_{q_m, k}(q_i) \left(1 + k d_Q / d(q_m, q_i) \right) (x_m(t) - x_i(t)).$$

Следует подчеркнуть, что виртуальные нейроны не существуют в алгоритме, поэтому нет необходимости изменять структуру зафиксированной сетки при преодолении граничного эффекта. Кроме того, усилия были направлены на то, чтобы сделать правило обучения как можно более простым по следующим причинам: (1) чтобы сохранить внутренний параллелизм SOM-алгоритма, который состоит в том, что все нейроны обрабатываются по одному и тому же правилу независимо друг от друга; (2) чтобы избежать проблем при построении сеток на сложных многосвязных областях, т.е. на областях с одной или с несколькими дырами, так как граничный эффект должен контролироваться на каждой из границ области.

На рис. 3 приведены примеры адаптивных сеток, построенных композиционным алгоритмом и сглаженным предложенным выше алгоритмом сглаживания.

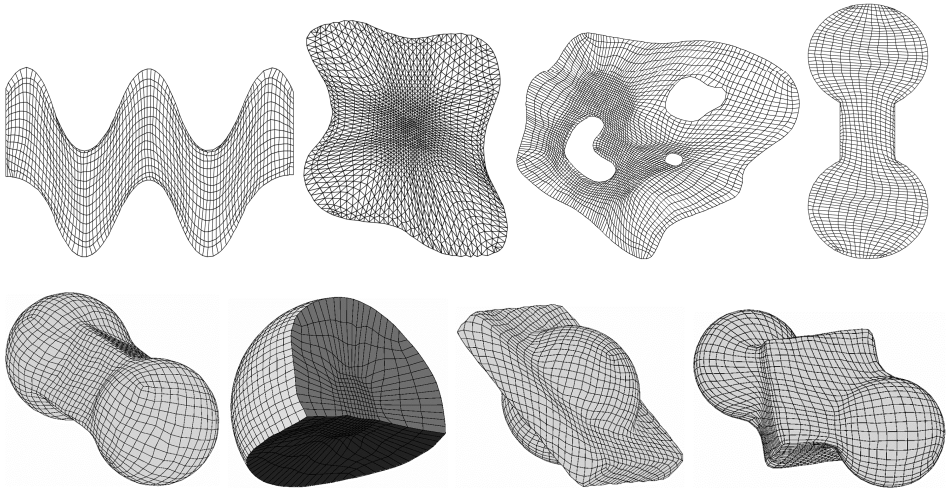


Рис. 3. Примеры адаптивных сеток, построенных с помощью композиционного алгоритма и обработанных предложенным алгоритмом сглаживания

ЛИТЕРАТУРА

1. Лебедев А.С., Лисейкин В.Д., Хакимзянов Г.С. Разработка методов построения адаптивных сеток. // Вычислительные технологии. Том 7, №3, 2002. С. 29.
2. Brankov J.G., Yang Y., Galatsanos N.P. Image restoration using content-adaptive mesh modeling. // ICIP 03, Vol. II, 2003, P. 997 – 1000.
3. Хакимзянов Г.С., Шокин Ю.И., Барахнин В.Б., Шокина Н.Ю. Численное моделирование течений жидкости с поверхностными волнами. Новосибирск: Изд-во СО РАН, 2001.
4. Gordon W.J., Thiel L.C. Transfinite mappings and their applications to grid generation. // Numerical Grid Generation, Appl. Math. and Comp. V. 2/3, 1982. P. 171 – 192.
5. Nechaeva O. Composition of Self Organizing Maps for Adaptive Mesh Construction on Complex-shaped Domains // Proc. of 6th Int. Workshop on Self-Organizing Maps (WSOM 2007), Bielefeld, Germany, September 3 – 6, 2007.
6. Kohonen T. Self-organizing Maps // Springer Series in Information Sciences, V. 30, Springer, Berlin, Heidelberg, New York, 2001.
7. Ritter, H. Self-Organizing Maps on Non-Euclidean Spaces, In: Oja E, Kaski S, editors. Kohonen maps. Amsterdam. The Netherlands: Elsevier BV; 1999. P. 97 – 109.
8. Fritzke B. Some competitive learning methods. Technical report, Systems Biophysics, Inst. for Neural Comp., Ruhr-Universität Bochum, April 1997.

Статья представлена кафедрой программирования факультета прикладной математики и кибернетики Томского государственного университета и оргкомитетом 7-й Российской конференции с международным участием «Новые информационные технологии в исследовании сложных структур», поступила в научную редакцию 3 октября 2008 г.