УДК 004.032.26:519.71

МЕТОДЫ ОБУЧЕНИЯ МНОГОСЛОЙНОГО ПЕРСЕПТРОНА. ПОПЫТКИ ОПТИМИЗАЦИИ ЗАДАЧИ ПОИСКА ГЛОБАЛЬНОГО МИНИМУМА ФУНКЦИИ ЭНЕРГИИ

О.А. Марьина, Д.А. Ладяев

Мордовский государственный университет им. Н.П. Огарева
Тел. 8-8342-290770. E-mail: dladyaev@mrsu.ru

Аннотация: статья посвящена обзору методик обучения нейронных сетей. Рассмотрены их достоинства и недостатки. Сделана попытка поиска глобального минимума функции энергии на основе применения вейвлет-преобразования.

Список ключевых слов: нейронная сеть, персептрон, алгоритм обучения, градиент, вейвлет.

Введение

Нейронные сети широко используются при решении самых разных задач, где обычные алгоритмические решения оказываются неэффективными или невозможными. Можно назвать следующие задачи: распознавание текстов, игра на бирже, фильтрация спама, проверка проведения подозрительных операций по банковским картам, системы безопасности и видеонаблюдения и другие.

При построении нейронных сетей обычно делается ряд допущений и значительных упрощений, но они демонстрируют такие свойства, как обучение на основе опыта, обобщение, извлечение существенных данных из избыточной информации. Нейронные сети могут менять свое поведение в зависимости от состояния окружающей их среды. После анализа входных сигналов (возможно, вместе с требуемыми выходными сигналами) они самонастраиваются и обучаются, чтобы обеспечить правильную реакцию. Обученная сеть может быть устойчивой к некоторым отклонениям входных данных, что позволяет ей правильно распознать образ, содержащий различные помехи и искажения.

Существует большое число разных конфигураций нейронных сетей с различными принципами функционирования, которые ориентированы на решение всевозможных задач. В качестве примера рассмотрим многослойную полносвязанную нейронную сеть прямого распространения (рис. 1), которая широко используется для поиска закономерностей и классификации образов. Полносвязанной нейронной сетью называется многослойная структура, в которой каждый нейрон произвольного слоя связан со всеми нейронами предыдущего слоя, а в случае первого слоя – со всеми входами нейронной сети. Прямое распространение сигнала означает, что такая нейронная сеть не содержит петель.

Математическая модель нейрона:

$$x_{(k)}^{i+1} = f\left(\sum_{j=1}^{N} w_j^{(k)} x_j^{(i)}\right)$$

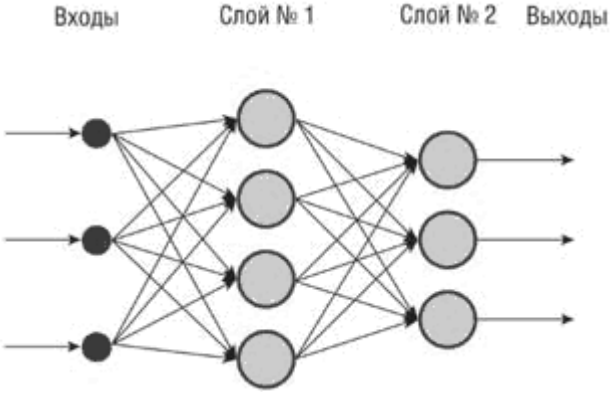

Рис. 1. Пример многослойной полносвязанной нейронной сети прямого распространения сигнала.

Обучение.

Под обучением искусственных нейронных сетей [1] понимается процесс настройки архитектуры сети (структуры связей между нейронами) и весов синаптических связей (влияющих на сигналы коэффициентов) для эффективного решения поставленной задачи. Обычно обучение нейронной сети осуществляется на некоторой выборке. По мере процесса обучения, который происходит по некоторому алгоритму, сеть должна все лучше и лучше (правильнее) реагировать на входные сигналы.

Выделяют три парадигмы обучения: с учителем, без учителя (или самообучение) и смешанная. В первом способе известны правильные ответы к каждому входному примеру, а веса подстраиваются так, чтобы минимизировать ошибку. Обучение без учителя позволяет распределить образцы по категориям за счёт раскрытия внутренней структуры и природы данных. При смешанном обучении комбинируются два вышеизложенных подхода.

Среди множества алгоритмов обучения с учителем наиболее успешным является алгоритм обратного распространения ошибки. Его основная идея заключается в том, что изменение весов синапсов происходит с учётом локального градиента функции ошибки. Разница между реальными и правильными ответами нейронной сети, определяемыми на выходном слое, распространяется в обратном направлении (рис. 2) – навстречу потоку сигналов. В итоге каждый нейрон способен определить вклад каждого своего веса в суммарную ошибку сети. Простейшее правило обучения соответствует методу наискорейшего спуска, то есть изменения синаптических весов пропорционально их вкладу в общую ошибку.

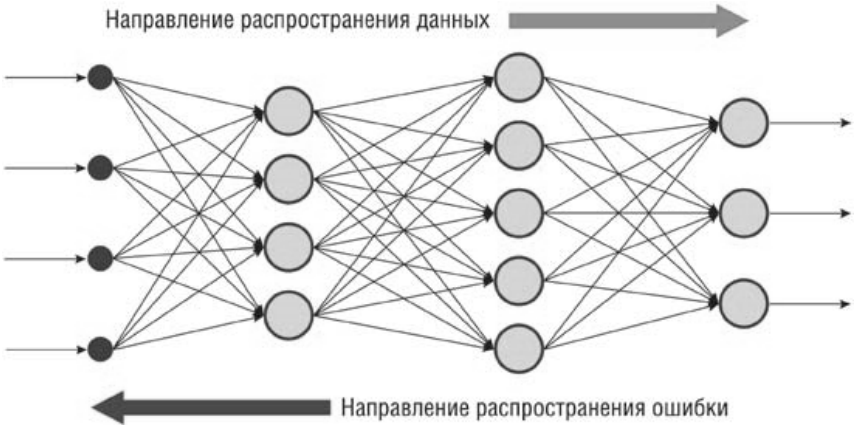

Рис. 2. Метод обратного распространения ошибки для многослойной полносвязанной нейронной сети.

При подобном обучении нейронной сети нет уверенности, что она обучилась наилучшим образом, поскольку всегда существует возможность попадания алгоритма в локальный минимум (рис. 3). Для этого используются специальные приёмы, позволяющие «выбить» найденное решение из локального экстремума. Если после нескольких таких действий нейронная сеть сходится к тому же решению, то можно сделать вывод о том, что найденное решение, скорее всего, оптимально.

Поправка к весовым коэффициентам:

$$\Delta w_{ij}^{(n)} = -\eta \cdot \frac{\partial E}{\partial w_{ij}^{(n)}},$$

где w – коэффициент синаптической связи, η – коэффициент скорости обучения сети, E – функция суммарной ошибки сети.

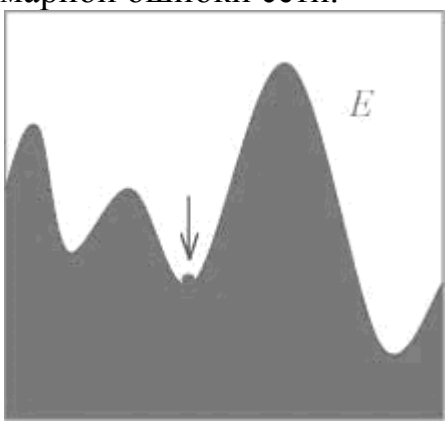

Рис. 3. Метод градиентного спуска при минимизации ошибки сети. Попадание в локальный минимум.

Методы обучения нейронной сети.

1. Метод обратного распространения ошибки. [2]

Метод обратного распространения ошибки – это итеративный градиентный алгоритм, который используется с целью минимизации ошибки работы многослойного перцептрона и получения желаемого выхода.

Основная идея этого метода состоит в распространении сигналов ошибки от выходов сети к её входам, в направлении, обратном прямому распростране-

нию сигналов в обычном режиме работы. Для возможности применения этого метода передаточная функция нейронов должна быть дифференцируема.

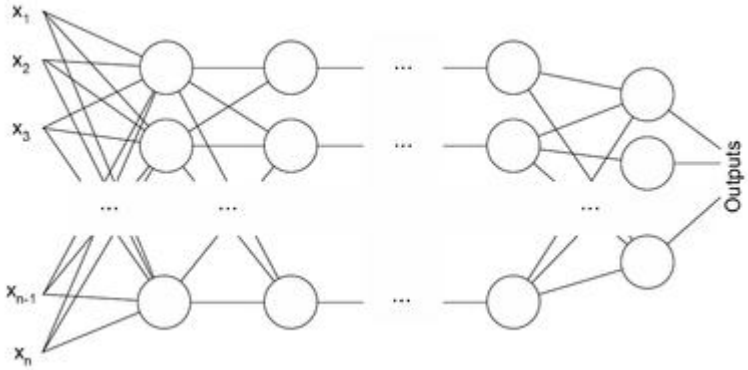

Рис. 4. Архитектура многослойного перцептрона

Алгоритм обратного распространения ошибки применяется для многослойного перцептрона. Из-за особенности вычисления поправок – вычислять поправку для узлов последнего уровня и выражать поправку для узла более низкого уровня через поправки более высокого – алгоритм называется алгоритмом обратного распространения ошибки.

На вход алгоритму, кроме указанных параметров, нужно также подавать в каком-нибудь формате структуру сети. На практике очень хорошие результаты показывают сети достаточно простой структуры, состоящие из двух уровней нейронов – скрытого уровня и нейронов-выходов; каждый вход сети соединён со всеми скрытыми нейронами, а результат работы каждого скрытого нейрона подается на вход каждому из нейронов-выходов. В таком случае достаточно подавать на вход количество нейронов скрытого уровня.

Несмотря на многочисленные успешные применения обратного распространения больше всего неприятностей приносит неопределённо долгий процесс обучения. В сложных задачах сеть может и вообще не обучиться. Причины могут быть следующие:

– *Паралич сети*. В процессе обучения сети значения весов могут в результате коррекции стать очень большими величинами. Большинство нейронов будут функционировать в области, где производная сжимающей функции очень мала. Процесс обучения может практически замереть.

– *Локальные минимумы*. Обратное распространение использует разновидность градиентного спуска – спуск вниз по поверхности ошибки – непрерывно подстраивая веса в направлении к минимуму. Сеть может попасть в локальный минимум, когда рядом имеется гораздо более глубокий минимум.

– *Размер шага*. Коррекции весов предполагаются бесконечно малыми. Это неосуществимо на практике, так как ведёт к бесконечному времени обучения. Однако размер шага должен браться конечным.

Следует также отметить возможность переобучения сети, что является скорее результатом ошибочного проектирования её топологии. При слишком большом количестве нейронов теряется свойство сети обобщать информацию. Весь набор образов, предоставленных к обучению, будет выучен сетью, но любые другие образы, даже очень похожие, могут быть классифицированы неверно.

Существуют современные алгоритмы второго порядка, такие как метод сопряжённых градиентов и метод Левенберга-Маркара, которые на многих задачах работают существенно быстрее (иногда на порядок). Разработаны также эвристические модификации этого алгоритма, хорошо работающие для определённых классов задач, – быстрое распространение и Дельта-дельта с чертой.

2. Метод сопряжённых градиентов. [3, 4]

Метод сопряжённых градиентов – метод нахождения локального минимума функции на основе информации о её значениях и её градиенте. В случае квадратичной функции в R^n минимум находится за n шагов.

Метод сопряжённых градиентов является методом первого порядка, в то же время скорость его сходимости квадратична. Этим он выгодно отличается от обычных градиентных методов. Например, метод наискорейшего спуска и метод координатного спуска для квадратичной функции сходятся лишь в пределе, в то время как метод сопряженных градиентов оптимизирует квадратичную функцию за конечное число итераций. При оптимизации функций общего вида, метод сопряжённых направлений сходится в 4-5 раз быстрее метода наискорейшего спуска. При этом, в отличие от методов второго порядка, не требуется трудоёмких вычислений вторых частных производных.

3. Метод градиентного спуска с учётом моментов. [5]

Идея метода заключается в добавлении к величине коррекции веса значения пропорционального величине предыдущего изменения этого же весового коэффициента.

$$\Delta w(t) = -\eta \cdot \frac{\partial E}{\partial w} + \alpha \cdot \Delta w(t-1),$$

где $\Delta w(j)$ – значение коррекции веса на j-ом шаге, η – длина шага, $E(w)$ – функция ошибки, α – коэффициент инерции.

4. Метод Левенберга-Маркара. [6]

Данный метод считается самым быстрым и надёжным алгоритмом обучения. Однако его применение связано с определёнными ограничениями:

– Сети с одним выходом. Метод Левенберга-Маркара можно применять только для сетей с одним выходным элементом.

– Небольшие сети. Метод Левенберга-Маркара требует памяти, пропорциональной квадрату числа весов в сети. Фактически это ограничение не позволяет использовать метод для сетей большого размера (порядка тысячи и более весов).

– Среднеквадратичная функция ошибок. Метод Левенберга-Маркара применим только для среднеквадратичной функции ошибок. Если указан для сети другой вид функции ошибок, то этот выбор будет проигнорирован при обучении методом Левенберга-Маркара. Поэтому этот метод обычно подходит только для регрессионных сетей.

Метод Левенберга-Маркара предполагает, что функция, моделируемая нейронной сетью, является линейной. В таком предположении минимум определяется за один шаг вычислений. Затем найденный минимум проверяется, и если ошибка уменьшилась, весам присваиваются новые значения. Вся процедура по-

следовательно повторяется. Поскольку предположение о линейности, вообще говоря, не оправдано, могло бы получиться так, что пришлось бы проверять точки, лежащие далеко от текущей точки. В методе Левенберга-Маркара местоположение новой точки есть результат компромисса между продвижением в направлении наискорейшего спуска и описанного выше скачка. Успешные шаги принимаются, и баланс смещается в сторону предположения линейности (которое приблизительно верно в окрестности точки минимума). Неудачные шаги отвергаются, и алгоритм идет более осторожно вниз по склону. Таким образом, алгоритм Левенберга-Маркара все время меняет схему действия и может работать очень быстро.

Алгоритм Левенберга-Маркара специально разработан так, чтобы минимизировать среднеквадратичную функцию ошибок с помощью формулы, которая (частично) предполагает, что функция, которую моделирует сеть, является линейной. Вблизи точки минимума это предположение выполняется с большой точностью, так что алгоритм может продвигаться очень быстро. Вдали от минимума это предположение может быть неправильным. Поэтому метод Левенберга-Маркара находит компромисс между линейной моделью и градиентным спуском. Шаг делается только в том случае, если он уменьшает ошибку, и там, где это необходимо, для обеспечения продвижения используется градиентный спуск с достаточно малым шагом.

5. Быстрое распространение (метод градиентного спуска с адаптивным обучением). [7]

В методе быстрого распространения производится пакетная обработка данных. В то время как в методе обратного распространения веса сети корректируются после обработки каждого очередного наблюдения, в методе быстрого распространения вычисляется усреднённый градиент поверхности ошибок по всему обучающему множеству, и веса корректируются один раз в конце каждой эпохи.

Метод быстрого распространения действует в предположении, что поверхность ошибок является локально квадратичной. Если это так, то точка минимума на ней находится всего через одну-две эпохи. В общем случае такое предположение неверно, но даже если оно выполняется лишь приблизительно, алгоритм всё равно очень быстро сходится к минимуму.

В этом предположении алгоритм быстрого распространения работает так:

– На первой эпохе веса корректируются по тому же правилу, что и в методе обратного распространения, исходя из локального градиента и коэффициента скорости обучения.

– На последующих эпохах алгоритм использует предположение о квадратичности для более быстрого продвижения к точке минимума.

Исходные формулы метода быстрого распространения имеют ряд вычислительных недостатков. Во-первых, если поверхность ошибок не является вогнутой, алгоритм может уйти в ложном направлении. Далее, если вектор градиента не меняется или меняется мало, шаг алгоритма может оказаться очень большим и даже бесконечным. Наконец, если по ходу встретился нулевой градиент, изменение весов вообще прекратится.

Метод быстрого распространения обрабатывает данные в пакетном режиме: градиент ошибки вычисляется как сумма градиентов ошибок по всем обучающим наблюдениям.

На первой эпохе алгоритм быстрого распространения корректирует веса так же, как и алгоритм обратного распространения.

Затем изменения весов вычисляются по формуле быстрого распространения:

$$\Delta w(t) = \frac{s(t)}{s(t-1) - s(t)} \cdot \Delta w(t-1)$$

Данная формула численно неустойчива при $s(t)$ близком, равном, или большем, чем $s(t\text{-}1)$. Поскольку $s(t)$ находится после продвижения по направлению градиента, это может произойти, только если наклон поверхности стал постоянным или становится круче (т.е. поверхность не является вогнутой).

В таких случаях веса корректируются по формуле:

$$\Delta w(t) = a \cdot \Delta w(t-1),$$

где a - коэффициент ускорения.

Если градиент становится равным нулю, то приращение (дельта) веса также делается равным нулю и, по приведенной выше формуле, так нулем и остается, даже если градиент потом изменится. Обычный способ борьбы с такой трудностью состоит в добавлении к вычисленным выше изменениям весов маленького коэффициента. Однако это может привести к численной неустойчивости. Если градиент был нулем, а затем становится существенно отличным от нуля, то корректировка соответствующего веса вновь делается как для отрицательного градиента.

6. Метод Дельта-дельта с чертой (метод градиентного спуска с учётом моментов и адаптивным обучением). [8]

В некоторых случаях он оказывается эффективнее, хотя в большей степени, чем метод обратного распространения, склонен застревать в локальных минимумах. В отличие от метода обратного распространения, этот метод обычно довольно устойчив.

В методе Дельта-дельта с чертой скорости обучения для отдельных весов корректируются на каждой эпохе таким образом, чтобы соблюдались следующие важные эвристические требования.

– Если производная сохраняет знак на нескольких последовательных итерациях, то скорость обучения увеличивается (поверхность ошибок имеет малую кривизну, поэтому одинаковым продвижениям соответствуют примерно одинаковые понижения уровня);

– Если знак производной на протяжении нескольких последних итераций всякий раз менялся на противоположный, то скорость обучения значительно уменьшается (если этого не сделать, алгоритм может начать осциллировать вокруг точек с большой кривизной).

Чтобы удовлетворить этим условиям, в методе Дельта-дельта с чертой задаётся начальная скорость обучения, которая используется для всех весов на первой эпохе, коэффициент ускорения, который добавляется к скоростям обучения, когда производные не меняют знака, и коэффициент замедления, на кото-

рый умножаются скорости обучения в случае, когда производная меняет знак. Применение линейного роста и экспоненциального убывания для скоростей обучения придает алгоритму большую устойчивость.

Однако описанная схема может плохо работать на поверхностях ошибок, искажённых помехами, где при выраженном общем понижающемся рельефе производные могут резко менять знак. Поэтому при реализации алгоритма для увеличения или уменьшения скорости обучения берётся сглаженный вариант производной.

Веса корректируются по тем же формулам, что и в методе обратного распространения, с той разницей, что коэффициент инерции не используется, а каждый вес имеет свою собственную, зависящую от времени скорость обучения.

В начале всем скоростям обучения присваиваются одинаковые стартовые значения; затем на каждой эпохе они корректируются.

Сравнение методов обучения

В программном пакете MATLAB была спроектирована нейронная сеть. Выбрана архитектура многослойного персептрона с одним промежуточным слоем. Входной слой состоит из 100 нейронов, выходной – из 3 нейронов, число нейронов в скрытом слое варьируется равно 8, 10 или 12. В качестве функции активации выбрана показательная функция, так как она легко дифференцируется. Обучение сети происходило разными методами:

– метод градиентного спуска,

– метод градиентного спуска с адаптивным обучением,

– метод градиентного спуска с учётом моментов,

– метод градиентного спуска с учётом моментов и с адаптивным обучением.

Анализ полученных результатов позволяет сделать следующие выводы.

– Сравнивая полученные результаты, можно говорить о том, что оптимальной для обучения нейронной сети представляется выборка из 100 групп точек на каждый тип сигнала. При увеличении количества обучающих групп точек наблюдается явление переобучения сети.

– Наилучшим образом обучаются сети, методом тренировки которых является метод градиентного спуска с адаптивным обучением или метод градиентного спуска с учётом моментов и адаптивным обучением (последний алгоритм предпочтительнее).

– Наименьшая погрешность классификации признака (порядка 10^{-6}) наблюдается у нейронных сетей с 10 нейронами в промежуточном слое.

Модель Хопфилда

Одним из вариантов, помогающим в решении задачи определения глобального минимума на поверхности ошибок является сглаживание функции энергии. Показательно проиллюстрировать данное предположение на модели Хопфилда.

Модель Хопфилда [9] представляет собой сеть связанных между собой нейронов, каждый из которых характеризуется своим состоянием - уровнем возбуждения x_i. Состояние любого нейрона в каждый момент времени t сигмоидально зависит от взвешенной суммы сигналов, поступающих к нему от других нейронов:

$$x_i(t) = \frac{1 - e^{-\frac{G}{T}\sum w_{ij} \cdot x_j(t-1)}}{1 + e^{-\frac{G}{T}\sum w_{ij} \cdot x_j(t-1)}}$$

Веса w_{ij} характеризуют силу синаптической связи между нейронами. Крутизна сигмоидной кривой, называемой функцией активации нейрона, определяется множителем G/T, стоящим перед взвешенной суммой входных сигналов.

Основной заслугой Дж. Хопфилда было использование при анализе работы нейронной сети связанной с ней так называемой функции энергии, или функции Ляпунова. Функция энергии зависит от состояния системы и при каждом его изменении может только уменьшиться. Из любого положения можно двигаться в направлении ближайшего локального минимума функции энергии. Локальные минимумы функции энергии называются аттракторами системы, а множество состояний, движение из которых приводит к данному аттрактору, – его бассейном притяжения. Состояние системы – это паттерн активностей всех её нейронов, а аттрактор – это состояние, соответствующее запомненному сетью образу.

Функция энергии в случае сигмоидальной функции активации может быть получена в явном виде. Она представляет собой сумму двух членов, первый из которых зависит от знаменателя T, а второй – нет:

$$E = -\frac{G}{2T}\sum_i \sum_j w_{ij} x_j^2 + \sum_i \ln\left((1 + x_i)(1 - x_i)\right) + \sum_i \ln\left(\frac{1 + x_i}{1 - x_i}\right)$$

Когда значение T очень мало, большую величину имеет множитель G/T в уравнении сигмоидной функции активации. Функция активации имеет большую крутизну (близка к пороговой), вклад второго члена в функцию энергии пренебрежимо мал и её форма определяется первым членом. Например, для сети их двух нейронов функция энергии в этом случае имеет два аттрактора, соответствующих активностям нейронов близким к наивысшим.

Когда T очень велико, то вклад первого члена в функцию энергии пренебрежимо мал, и она определяется вторым членом. В этом случае она имеет только один минимум, соответствующий нулевой активности нейрона. В промежуточных случаях каждый из членов вносит свой вклад в функцию энергии. С увеличением значения T два первоначальных аттрактора становятся все менее

выраженными и, в конце концов, полностью исчезают, т. е. со временем рельеф функции энергии сглаживается.

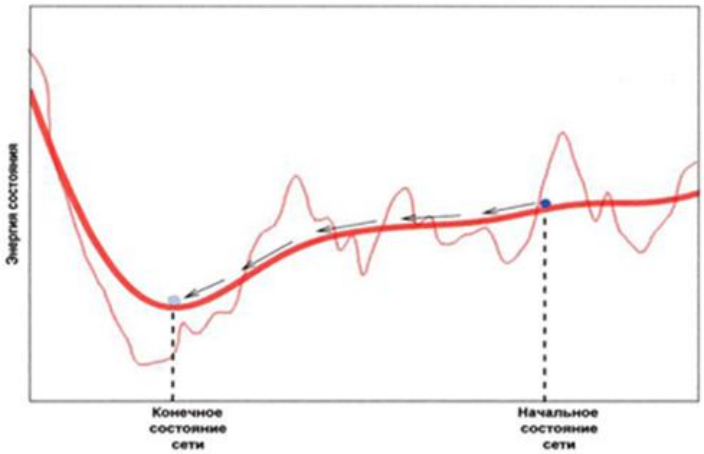

Рис. 5. Эффект сглаживания нейронной сети

На рисунке тонкой линией показана исходная несглаженная функция энергии, а толстой – сглаженная функция. Глобальный аттрактор, соответствующий стратегически более правильному решению проблемы, находится довольно далеко от исходного состояния сети и не может быть достигнут в случае несглаженной функции энергии, так как сеть остановится в состоянии, соответствующем ближайшему аттрактору. И наоборот, в случае сглаженной функции энергии глобальный аттрактор может быть легко достигнут.

При сглаживании теряются мелкие детали, которые в определённых ситуациях могут быть очень важны, и это следует учитывать.

Применение вейвлет-преобразования.

Для определения расположения глобального минимума функции ошибки представляется возможным подвергнуть её быстрому вейвлет-преобразованию. [10] Во-первых, преобразование (один уровень) можно выполнить за $O(n)$ операций. Во-вторых, оно не только раскладывает сигнал на некоторое подобие частотных полос (путём анализа его в различных масштабах), но и представляет временную область, то есть моменты возникновения тех или иных частот в сигнале.

Данное предположение является целью дальнейшего исследования оптимизации работы нейронных сетей.

На рисунках ниже представлено разложение функции энергии под действием различных вейвлет-преобразований. О местоположении глобального минимума можно судить по большим светлым пятнам, которые отчётливо видны при разложении на верхних уровнях.

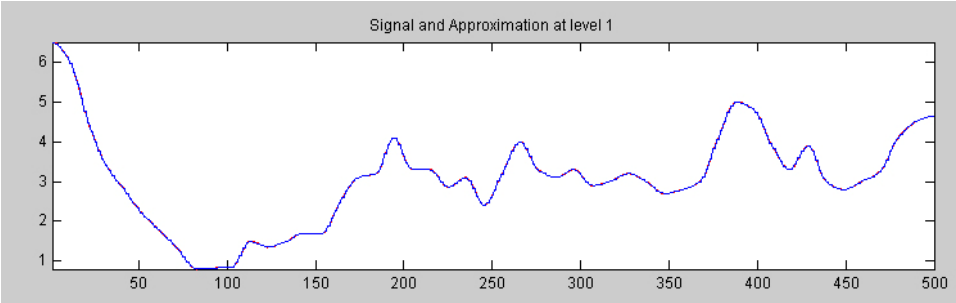

Рис. 6. Сигнал и его аппроксимация на уровне 1.

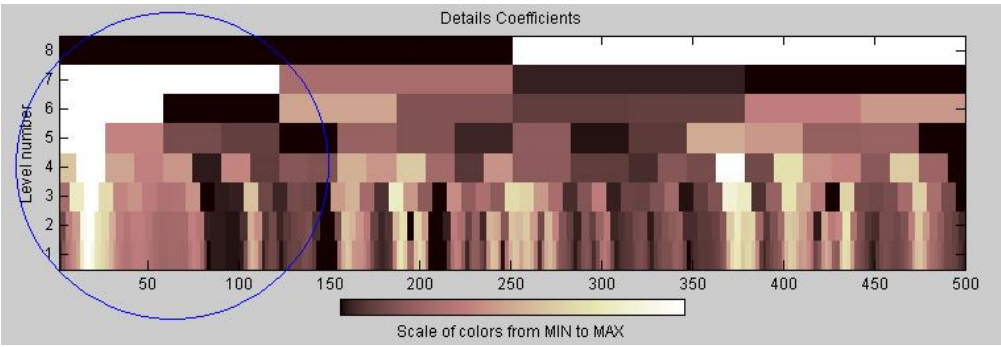

Рис. 7. Разложение функции энергии под действием вейвлет-функции Хаара (коэффициент равен 2, уровень 8).

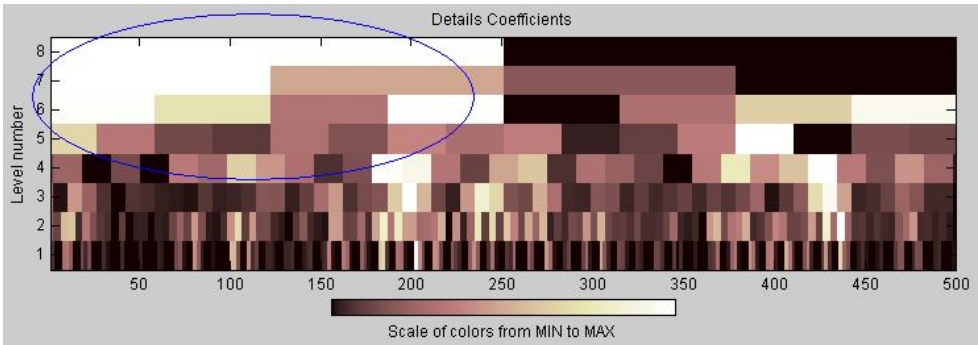

Рис. 8. Разложение функции энергии под действием вейвлет-функции Добеши (коэффициент равен 4, уровень 8).

Список использованной литературы

1. Даниил Кальченко, «КомпьютерПресс» №1, январь 2005: http://www.neuroproject.ru/articles_dak_nn.php
2. http://ru.wikipedia.org/wiki/Метод_обратного_распространения_ошибки
3. http://ru.wikipedia.org/wiki/Метод_сопряжённых_градиентов
4. http://www.basegroup.ru/library/analysis/neural/conjugate/
5. Ежов А.А., Шумский С.А. “Нейрокомпьютинг и его применение в экономике и бизнесе”.1998.
6. http://www.statsoft.ru/home/portal/applications/NeuralNetworksAdvisor/Adv-new/LevenbergMarquardt.htm
7. http://www.statsoft.ru/home/portal/applications/neuralnetworksadvisor/adv-new/QuickPropagation.htm
8. http://www.statsoft.ru/home/portal/applications/neuralnetworksadvisor/adv-new/DeltaBarDelta.htm
9. http://www.conf.muh.ru/080130/thesis_Terehin.htm
10. http://ru.wikipedia.org/wiki/Дискретное_вейвлет-преобразование

Сведения об авторах

Марьина Оксана Александровна, *аспирант кафедры «Автоматизированные системы обработки информации и управления»,* e-mail: fd_oxy@mail.ru
Ладяев Дмитрий Александрович, *кандидат технических наук, старший преподаватель кафедры «Автоматизированные системы обработки информации и управления»,* тел. (8-8342)-290770, e-mail: dladyaev@mrsu.ru