P11-2002-274

Г. А. Ососков, А. В. Филимонов*

# ДИНАМИЧЕСКАЯ ОПТИМИЗАЦИЯ СТРУКТУРЫ ПЕРСЕПТРОНОВ

*Ивановский государственный университет

# Введение

В последнее время широкое распространение получило применение нейронных сетей для обработки данных [1]. Кратко напомним основные положения теории искусственных нейронных сетей (ИНС).

Нейронные сети состоят из большого количества однотипных элементов, имитирующих работу биологических нейронов. В дальнейшем эти элементы будем называть нейронами. Искусственный нейрон имеет несколько входов и один выход (аксон). Входы (синапсы), идущие от других нейронов, характеризуются своими весами, т.е. величинами, определяющими силу связи между нейронами (см. рис.1). Нейрон способен выполнять две операции: суммирование взвешенных входных сигналов и сжатие полученной суммы. Сжатие осуществляется обычно посредством логистической функции сигмоидального типа (см. рис. 2)

$$Y = \frac{1}{1 + e^{-\alpha \sum x_i \cdot w_i}}. \qquad (1)$$

Здесь $Y$ – выход нейрона, $X=(x_1, x_2, \ldots x_k)$ — входной вектор, $W$ – весовой вектор, $\alpha$ – кривизна сигмоиды.

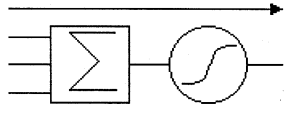

Рис.1. Функциональная схема нейрона

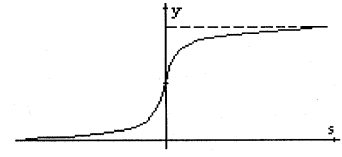


Рис.2. Внешний вид сигмоиды

Наиболее распространенная разновидность ИНС – это прямоточные многослойные нейронные сети, называемые персептронами [1,2]. В персептронах нейроны объединяются в слои, причем в каждом слое выходы нейронов являются входами нейронов следующего слоя. Персептроны с успехом применяют для решения таких задач, как классификация, прогнозирование, выявление аномальных данных и т.д. Однако чтобы персептрон можно было использовать в практических целях, его необходимо обучить.

Основная стратегия обучения персептронов – это «обучение с учителем». Суть такого обучения заключается в следующем. Перед обучением все веса в сети устанавливаются случайным образом. На вход сети подается входной вектор, а на выходе предъявляется целевой вектор (желаемый отклик). Обычно реальный выход сети (выходной вектор) не соответствует целевому вектору, поэтому необходимо так скорректировать веса в сети, чтобы уменьшить это несоответствие (ошибку).

## Метод обучения персептронов

Наиболее распространенным методом обучения персептронов является Back Propagation – метод обратного распространения ошибок (МОРО) [3]. Суть метода заключается в градиентном спуске по поверхности целевой функции с целью поиска глобального минимума, который соответствует минимальному различию между выходным и целевым вектором.

Согласно методу наименьших квадратов, минимизируемой целевой функцией ошибки ИНС является величина

$$E(w) = \frac{1}{2}\sum_{j,p}(y_{j,p}^{(N)} - d_{j,p})^2, \tag{2}$$

где $y_{j,p}^{(N)}$ – реальное выходное состояние нейрона $j$ выходного слоя $N$ нейронной сети при подаче на ее входы $p$-го образа; $d_{jp}$ – идеальное (желаемое) выходное состояние этого нейрона.

Суммирование ведется по всем нейронам выходного слоя и по всем обрабатываемым сетью образам. Минимизация ведется методом градиентного спуска, что означает подстройку весовых коэффициентов следующим образом:

$$\Delta w_{ij}^{(n)} = -\eta \frac{\partial E}{\partial w_{ij}}. \tag{3}$$

Здесь $w_{ij}$ – весовой коэффициент синаптической связи, соединяющей $i$-й нейрон слоя $n-1$ с $j$-м нейроном слоя $n$, $\eta$ – коэффициент скорости обучения, $0<\eta<1$;

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j}\frac{dy_j}{ds_j}\frac{\partial s_j}{\partial w_{ij}}. \tag{4}$$

Здесь под $y_j$, как и раньше, подразумевается выход нейрона $j$, а под $s_j$ – взвешенная сумма его входных сигналов, то есть аргумент активационной функции. Так как множитель $dy_j/ds_j$ является производной этой функции по ее аргументу, из этого следует, что производная активационной функции должна быть определена на всей оси абсцисс. Для сигмоидальной функции (1) производная равна

$$\frac{dy}{ds} = \alpha\, y\,(1-y). \tag{5}$$

Третий множитель $\partial s_j/\partial w_{ij}$, очевидно, равен выходу нейрона предыдущего слоя $y_i^{(n-1)}$.

Что касается первого множителя в (4), то он легко раскладывается следующим образом:

$$\frac{\partial E}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k}\frac{dy_k}{ds_k}\frac{\partial s_k}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k}\frac{dy_k}{ds_k} w_{jk}^{(n+1)}. \tag{6}$$

Здесь суммирование по $k$ выполняется среди нейронов слоя $n+1$.

Введя новую переменную

$$\delta_j^{(n)} = \frac{\partial E}{\partial y_j}\frac{dy_j}{ds_j}, \tag{7}$$

мы получим рекурсивную формулу для расчетов величин $\delta_j^{(n)}$ слоя $n$ из величин $\delta_k^{(n+1)}$ более старшего слоя $n+1$:

$$\delta_j^{(n)} = \left[\sum_k \delta_k^{(n+1)}\, w_{jk}^{(n+1)}\right]\frac{dy_j}{ds_j}. \tag{8}$$

Для выходного же слоя

$$\delta_l^{(N)} = (y_l^{(N)} - d_l)\frac{dy_l}{ds_l}. \tag{9}$$

Теперь мы можем записать (3) в раскрытом виде:

$$\Delta w_{ij}^{(n)} = -\eta\, \delta_j^{(n)}\, y_i^{(n-1)}. \tag{10}$$

Полный алгоритм обучения ИНС с помощью процедуры обратного распространения строится так:

1. Подать на входы сети один из возможных образов и в режиме обычного функционирования ИНС, когда сигналы распространяются от входов к выходам, рассчитать значения последних. Напомним, что

$$s_j^{(n)} = \sum_{i=0}^{M} y_i^{(n-1)}\, w_{ij}^{(n)}, \tag{11}$$

где $M$ – число нейронов в слое $n-1$ с учетом нейрона с постоянным выходным состоянием +1, задающего смещение; $y_i^{(n-1)}=x_{ij}^{(n)}$ – $i$-й вход нейрона $j$ слоя $n$.

2. Рассчитать $\boldsymbol{\delta^{(N)}}$ для выходного слоя по формуле (9).

Рассчитать по формуле (10) изменения весов $\boldsymbol{\Delta w^{(N)}}$ слоя $N$.

3. Рассчитать по формулам (8) и (10) соответственно $\boldsymbol{\delta^{(n)}}$ и $\boldsymbol{\Delta w^{(n)}}$ для всех остальных слоев, $n=N-1,\ldots 1$.

4. Скорректировать все веса в ИНС

$$w_{ij}^{(n)}(t) = w_{ij}^{(n)}(t-1) + \Delta w_{ij}^{(n)}(t). \tag{12}$$

5. Если ошибка сети $E$ существенна, перейти на шаг 1. В противном случае – выход на конец обучения.

## Опыт практического применения ИНС

Как уже говорилось выше, персептроны успешно применяются для решения задач классификации и прогнозирования. Авторы давно сотрудничают с 7-й городской больницей г. Иваново и по заказу врачей создали программу, реализующую врачебную экспертную систему на базе нейросетевого анализа [4]. Однако, работая со статистическим материалом, полученным от сотрудников этой больницы, мы столкнулись с рядом проблем, касающихся структурирования нейронных сетей.

Перед нами была поставлена задача: создать экспертную систему для дифференцированной диагностики пневмонии и прогнозирования исходов.

В первоначальном варианте использовалось 9 клинико-лабораторных и рентгенологических показателей: возраст, температура, количество сегментов легкого с воспалительной инфильтрацией и т.д.

В системе использовались две независимые сети различной конфигурации. Это было необходимо для того, чтобы прогноз не зависел от диагноза. Для обучения сетей и их тестирования было обследовано 300 пациентов и 50 здоровых людей (контрольная группа). Данные о группе из 125 человек использовались для обучения сети, а данные об остальных – для тестирования. Основной проблемой была необходимость выбора оптимальной структуры ИНС. Для этого использовался прямой перебор различных конфигураций сетей. Всего было исследовано 20 различных конфигураций. Результаты представлены на рис. 3 и 4.

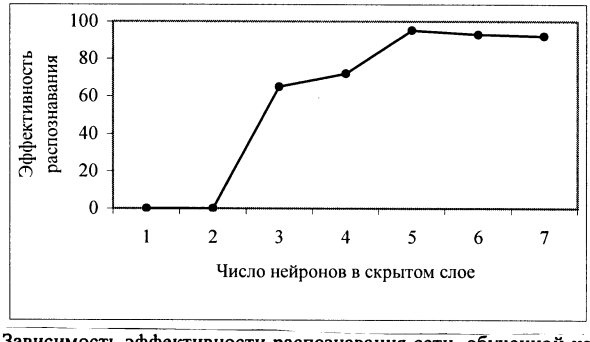


**Рис. 3. Зависимость эффективности распознавания сети, обученной на диагностику пневмонии (два исхода – два выходных нейрона), от числа нейронов в скрытом слое**

Нулевая эффективность означает, что сеть с такой конфигурацией нельзя обучить до заданной погрешности. Данная ситуация возникает


**Рис. 4. Зависимость эффективности распознавания сети, обученной на прогнозирование исходов пневмонии (три выходных нейрона), от числа нейронов в скрытом слое**

при недостаточном количестве скрытых нейронов. При избыточном количестве скрытых нейронов сеть будет просто запоминать факты, а не обобщать их, что также приводит к уменьшению точности распознавания

На основании исследования была выбрана следующая оптимальная конфигурация диагностической сети: 9 входных, 6 скрытых и 3 выходных нейрона. Для сети, осуществляющей прогнозирование, оптимальной оказалась конфигурация: 9 входных, 6 скрытых и 1 выходной нейрон. При тестировании выяснилось, что эффективность работы сетей при выбранных конфигурациях составила 95,2%. Сеть даже с такой предварительной структурой оказалась полезной для использования в некоторых врачебных исследованиях [4].

## Другой метод обучения ИНС

Однако для создания практически применяемой экспертной системы, способной прогнозировать возможные осложнения, требуется учитывать 53 прогностических признака при 4 основных осложнениях. Для нейронной сети, реализующей модель для прогнозирования осложнений, путем выборочного тестирования была признана приемлемой следующая конфигурация: 53/30/1.

Сеть с такой конфигурацией содержит 1620 весов, и для ее обучения, т.е. минимизации функционала с таким количеством параметров, требуется слишком много времени. В поисках замены алгоритма МОРО более быстрым был найден и исследован алгоритм Resilient Propagation (resilient — эластичный) [5]. Суть этого метода, который мы будем называть методом эластичного распространения ошибок (МЭРО), заключается в том, что в нем при подсчете поправок к шагам по параметрам учитываются не сами производные функции ошибок, как в МОРО, а только их знаки. Для определения величины коррекции весов используется следующее правило:

$$\Delta^{(t)}{}_{ij} = \begin{cases} \eta^{+} \cdot \Delta_{ij}^{(t-1)}, & при \ \dfrac{\partial E^{(t)}}{\partial w_{ij}} \dfrac{\partial E^{(t-1)}}{\partial w_{ij}} > 0 \\ \eta^{-} \cdot \Delta_{ij}^{(t-1)}, & при \ \dfrac{\partial E^{(t)}}{\partial w_{ij}} \dfrac{\partial E^{(t-1)}}{\partial w_{ij}} < 0 \end{cases}, \qquad (13)$$

$$0 < \eta^{-} < 1 < \eta^{+}.$$

Если на текущем шаге частная производная по соответствующему весу поменяла свой знак, то это говорит о том, что последнее изменение было большим, и алгоритм проскочил локальный минимум, и, следовательно, величину изменения необходимо уменьшить на $\eta^{-}$ и вернуть предыдущее значение весового коэффициента. Если знак частной производной не изменился, то нужно увеличить величину коррекции на $\eta^{+}$ для достижения более быстрой сходимости. Начальные значения для всех $\Delta_{ij}$ устанавливаются равными *0,1*.

Для вычисления значения коррекции весов используется следующее правило:

$$\Delta w_{ij} = \begin{cases} -\Delta_{ij}^{(t)}, & при \ \dfrac{\partial E^{(t)}}{\partial w_{ij}} > 0 \\ +\Delta_{ij}^{(t)}, & при \ \dfrac{\partial E^{(t)}}{\partial w_{ij}} < 0 \\ 0, & при \ \dfrac{\partial E^{(t)}}{\partial w_{ij}} = 0 \end{cases}. \quad (14)$$

Если производная положительна, т.е. ошибка возрастает, то весовой коэффициент уменьшается на величину коррекции, в противном случае — увеличивается. Затем подстраиваются веса:

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + \Delta w_{ij}^{(t)}. \quad (15)$$

**В итоге алгоритм МЭРО состоит из следующих шагов:**

1. Проинициализировать величину коррекции.
2. Предъявить все примеры из выборки и вычислить частные производные.
3. Подсчитать величину коррекции.
4. Скорректировать веса.
5. Если условие останова не выполнено, то перейти к 2.

Таблица 1

| Конфигурация сети | Число корректируемых весов | Алгоритм обучения | Время обучения (мин) |
|---|---|---|---|
| 9/6/3 | 72 | Back Propagation | 10 |
| 9/6/1 | 60 | Back Propagation | 8 |
| 53/30/1 | 1620 | Resilient Propagation | 20 |

Как показали расчеты в нашей медицинской задаче, данный алгоритм в среднем сходится быстрее МОРО в 4-5 раз (см. Табл. 1). Тем не менее, было очевидно, что, несмотря на смену алгоритма, время обучения остается неприемлемым, особенно если учесть, что для выбора оптимальной структуры сети эти вычисления требовалось производить многократно. В медицинских задачах это особенно важно, т.к. для обучения и тестирования сети имеется сравнительно небольшой по объему статистический материал. Если мы обратимся к таблице 1 и проанализируем конфигурацию сети и число нейронов, которые должны корректироваться в ходе обучения, то можно заметить, что большая часть весов приходится на связи между первым и вторым слоем. Поэтому необходимо найти алгоритмы понижения размерности входного вектора.

## Методы понижения размерности входного вектора ИНС

В ряде случаев исходные данные в обучающей выборке в большой степени коррелируют между собой. Это, в частности, характерно для медицинских данных, связанных для конкретного пациента, поскольку содержащиеся в них симптомы болезней относятся к одному и тому же лицу. Зная взаимные корреляции входных данных, можно понизить их

размерность, используя метод главных компонент (МГК) [6]. Обычно используется следующая схема.

1. По всей исходной выборке рассчитывается матрица ковариаций для используемых входных параметров-симптомов.
2. Одним из известным методов ортогональных преобразований, например методом Якоби, производится диагонализация матрицы ковариаций. Таким образом, мы переходим в новый базис, в котором преобразованные признаки уже не кореллируют между собой.
3. На главной диагонали новой матрицы ковариаций располагаются по убыванию дисперсии признаков в новом базисе. Сумма всех диагональных элементов называется дисперсией системы. Теперь для того, чтобы корректно обучить сеть, можно использовать не все признаки, а только те, сумма дисперсий которых составляет, скажем, 95 % дисперсии системы (эта величина может меняться в зависимости от конкретной задачи). При этом происходит значительное понижение размерности входного вектора (в нашей задаче число признаков уменьшилось с 53 до 17).
4. С помощью полученного преобразования ортогонализации, представляющего линейную операцию, исходные данные сети пересчитываются в новый сокращенный базис и используются для ее более быстрого обучения.

Благодаря линейности преобразование МГК можно «встроить» в саму сеть, добавив к ней дополнительный слой нейронов, осуществляющий переход от старого к новому базису. Однако применение метода главных компонент имеет несколько недостатков:

- Если размерность входного вектора необходимо увеличить (если снова привести в качестве примера медицинскую задачу, то увеличение размерности входного вектора означает добавление дополнительных клинических признаков), то главные компоненты придется пересчитывать заново.
- При увеличении размерности входного вектора количество математических операций, необходимых для выделения главных компонент, растет по экспоненте.
- Наиболее существенным (и малоприемлемым с точки зрения нашей прикладной задачи) явилось принципиальное требование МГК, чтобы входные данные были распределены нормально, т.к. только в этом случае совместное распределение входных признаков будет описываться их ковариационной матрицей.
- Успехи применения МГК в физике, например, во многом объяснялись возможностью использовать метод Монте-Карло для вычисления ковариационной матрицы с высокой степенью точности, что фактически неосуществимо в прикладных медицинских задачах из-за существенной ограниченности статистического материала.

Нами были проведены проверки статистических гипотез о нормальности распределения основных применяемых симптомов,

показавшие неприемлемые отклонения многих из них от этой гипотезы. В качестве примера значения коэффициентов асимметрии и эксцесса для семи важнейших параметров приведены в табл.2. Более того, часть признаков вообще не являются непрерывными, являясь ответами на вопрос врача типа «да-нет».

Таблица 2

| Признак | A | E | Ua | Ue | Ua/\|A\| | Ue/\|E\| |
|---|---|---|---|---|---|---|
| Возраст | -0,4987 | -0,7194 | 0,2736 | 0,5261 | 0,5486 | 0,7313 |
| Температура | -0,0238 | -1,0021 | 0,2736 | 0,5261 | 11,4524 | 0,5249 |
| Стаж курения | 0,2788 | -1,3754 | 0,2736 | 0,5261 | 0,9813 | 0,3825 |
| Частота дыхания | 1,3158 | 2,3436 | 0,2736 | 0,5261 | 0,2079 | 0,2244 |
| СОЭ | 0,2225 | -0,7463 | 0,2736 | 0,5261 | 1,2296 | 0,7049 |
| Лейкоциты | 1,0206 | 1,9855 | 0,2736 | 0,5261 | 0,268 | 0,2649 |
| Алкоголизация | -0,0817 | -2,0487 | 0,2736 | 0,5261 | 3,3484 | 0,2567 |

A – асимметрия, E – эксцесс, Ua—теоретическое значение погрешности асимметрии, Ue – теоретическое значение погрешности эксцесса.

Подобное несоответствие гипотезе нормальности делает нерациональным применение метода главных компонент.

Поэтому авторы предлагают иную схему уменьшения размерности входного вектора, в определенном смысле аналогичную этому методу. Идея нашего подхода была подсказана методом сжатия изображений с помощью рециркуляционной нейронной сети [7], где используется сеть со структурой, изображенной на рис. 5.

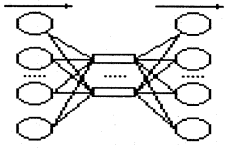

Рис.5. Рециркуляционная нейронная сеть

Фактически рециркуляционная сеть является трехслойным персептроном, но с числом выходных нейронов, равным числу входных. В такой сети изображение, подаваемое на вход, подвергается сжатию (выход скрытого слоя и есть сжатое изображение), а затем восстанавливается, чтобы по соотношению вход-выход можно было сравнить качество сжатия. Можно рассматривать сжатие изображения как некое извлечение главных компонент, число которых равно количеству нейронов скрытого слоя. Однако количество нейронов второго скрытого слоя устанавливается до обучения пользователем, а мы как раз и хотели бы это число узнать. Следовательно, применительно к нашей задаче мы должны заставить сеть в ходе ее обучения саму выполнять динамическую реконфигурацию и устанавливать оптимально необходимое количество нейронов скрытого слоя. Но прежде чем искать метод для динамического изменения количества нейронов скрытого слоя, надо определиться с критерием оценки эффективности работы этого метода. Очевидно, что лучшим критерием будет точность восстановления, в качестве которой мы выбрали среднюю погрешность, полученную путем усреднения разностей

входных и выходных нейронов рециркуляционной сети. На рис. 6 представлена зависимость средней погрешности на обучающем множестве от количества нейронов во втором слое, вычисленная методом перебора числа нейронов скрытого слоя.

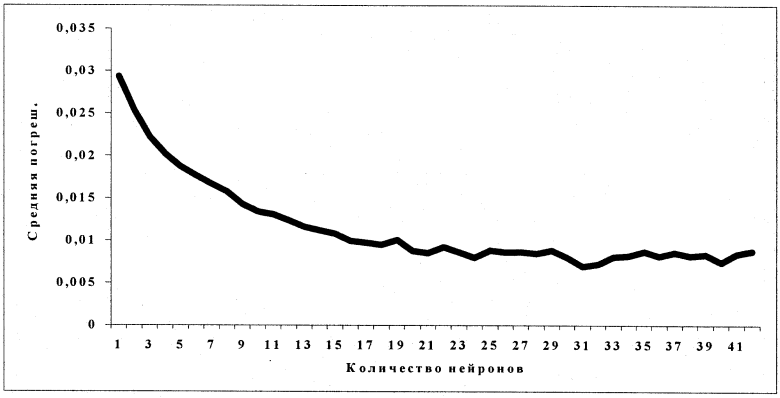


Рис. 6. Зависимость средней погрешности на обучающем множестве от количества нейронов в скрытом слое

На рис.6 видно, что при увеличении количества нейронов второго слоя погрешность сети вначале изменяется довольно сильно, а затем изменение становится очень незначительным. Это означает, что когда при увеличении количества нейронов погрешность сети изменяется мало, нет необходимости в дальнейшем увеличении количества нейронов, т.е. достигнут оптимум. Более того, если мы посмотрим на рис.7, на котором отображены процентные вклады дисперсий каждого из факторов в дисперсию системы, то можно заметить, что кривые на рис.6 и 7 подобны. Более того, точки оптимумов совпадают.

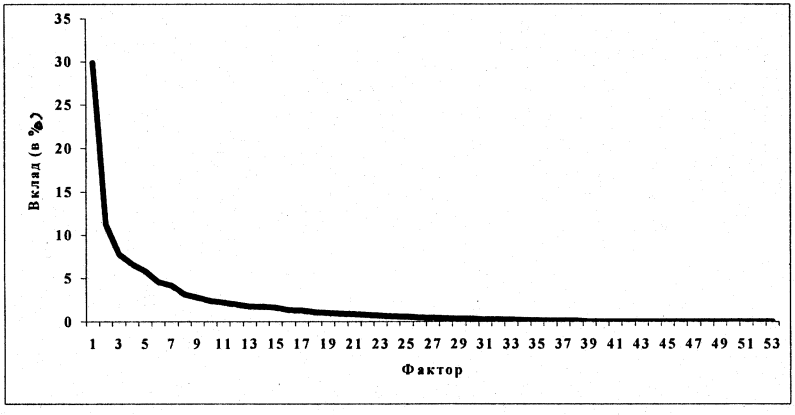


Рис. 7. Вклад в результат (дисперсию системы) каждого из факторов

Таким образом, мы понизили размерность входных данных, используя саму сеть. Однако чтобы построить зависимость погрешности от числа нейронов, мы использовали простой перебор сетей с разным числом нейронов в скрытом слое, а нам нужно научиться делать подбор количества нейронов прямо в ходе обучения сети. Из литературы была известна идея подбора оптимальной структуры сети путем динамического добавление нейронов в сеть (см. [2], например).

Применительно к нашей задаче обучение сети начинается с количеством нейронов, заведомо недостаточным для решения задачи. Для обучения используется метод МОРО. Обучение происходит до тех пор, пока ошибка не перестанет убывать и не выполнится условие

$$\frac{E(t) - E(t-\delta)}{E(t_0)} < \Delta, \tag{16}$$

$$t \geq t_0 + \delta, \tag{17}$$

где $t$ – время обучения, $\Delta$ — пороговое значение убыли ошибки, $\delta$ — минимальный интервал времени обучения между добавлениями новых нейронов, $t_0$ – момент последнего добавления. Когда выполняются оба условия, добавляется нейрон. Таким образом, мы добавляем дополнительный нейрон в том случае, когда относительная скорость изменения погрешности становиться меньше заданной величины. Обучение сети прекращается, когда выполняется какое-либо из условий остановки, определяемое пользователем, скажем, достижение критического уровня погрешности или заданной эпохи.

В нашей задаче критерием остановки служило достижение средней погрешности значения 0,01. Если посмотреть на рис. 6, то видно, что это значение соответствует точке оптимума. При использовании динамического добавления полученные значения количества нейронов колебались в пределах 15 – 18. На рис. 7 оптимальным значением было 16 нейронов. Таким образом, используя динамическое добавление нейронов для сжатия входных данных, мы получили результаты, близкие к тем, что были получены в результате простого перебора конфигураций сетей и МГК.

Возвращаясь к вышеприведенной схеме МОРО, мы увидим, что если использовать саму сеть для выделения главных компонент, то из описанной схемы следует выкинуть пункты 1-3 и 5. Теперь эта схема будет выглядеть так.

1. Создаем персептрон со структурой, подобной той, что изображена на рис. 5. Число нейронов скрытого слоя устанавливаем равным 1.
2. Начинаем обучать сеть, одновременно запуская алгоритм динамического добавления нейронов.
3. После достижения заданной погрешности, прекращаем обучение сети и удаляем третий слой.

В итоге мы получили два слоя нейронов, которые понижают размерность входного вектора с заданным уровнем ошибки восстановления. Их функция относится к задаче предобработки данных,

поэтому эти два слоя должны стать входными в обычный персептрон, обеспечивая экономичность его работы.

## Оптимизация структуры сети

Выбор оптимальной структуры нейронной сети является серьезной проблемой, которая до сих пор не решена полностью. Известны различные методики для выбора оптимальной структуры сети, однако в большинстве случаев их применимость к той или иной задаче сильно зависит от входных данных. Самым надежным средством для выбора структуры мог бы быть прямой перебор всех возможных конфигураций, неприемлемый для больших сетей, поскольку требует большого количества машинного времени. Для придания направленности процессу перебора вариантов в последнее время применяются так называемые генетические алгоритмы (ГА) (см., например, [8]). ГА основаны на эволюционных методах поиска, на принципе «выживает сильнейший» и с успехом применяются для выбора подходящей структуры сети. ГА – это самообучающаяся модель, которая для решения поставленных перед ней задач использует механизмы естественного отбора, т.е. такие понятия, как популяция, особь, наследование и мутация. Сначала алгоритм выдвигает возможные решения. Затем в каждом поколении, подобно биологическим организмам, решения «мутируют». «Слабые» решения «погибают», не удовлетворяя критерию эффективности, а «сильнейшие» — «скрещиваются», порождая новые решения.

Несмотря на то, что ГА эффективно решают задачи выбора оптимальной структуры, они имеют существенный недостаток: для функционирования алгоритма требуется много машинных ресурсов. В процессе тестирования генетического алгоритма в задаче выбора оптимальной структуры сети, решающей задачу классификации больных, мы обнаружили, что необходимо создавать популяции сетей, состоящие из 100 и более особей каждая. Все эти сети приходилось еще обучать и тестировать, а обучение сети – процесс долгий.

Поэтому мы предпочли использовать такие методы поиска наилучшей структуры сети, для которых не требуется создавать популяцию сетей, а достаточно работать с одной и той же сетью. К таким методам следует отнести методы динамического добавления нейронов, всевозможные виды пранинга (pruning — урезание, т.е. выбрасывание малозначимых нейронов) и т.д.

Метод динамического добавления нейронов уже рассматривался в предыдущей главе, поэтому просто посмотрим, что происходит с ошибкой при динамическом добавлении нейрона (см. рис. 8).

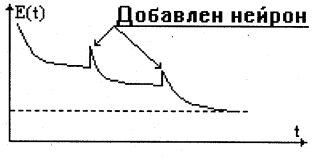


Рис. 8. Изменение погрешности в момент добавления нейрона

На рисунке видно, что в момент добавления нейрона ошибка кратковременно возрастает, но затем быстро достигает своего прежнего значения.

Pruning – это усечение малозначимых структурных элементов сети после ее обучения [9]. Можно выделить два вида усечения: весовое и структурное. При весовом усечении в обученной сети удаляются веса, которые имеют наименьшее значение. При структурном усечении удаляются отдельные нейроны или даже слои нейронов. Чтобы компенсировать удаленные элементы, поступают по-разному, например, корректируют веса у смещения или производят дополнительное обучение сети. В этом случае ошибка ведет себя несколько по-иному (см. рис. 9).


Рис. 9. Изменение погрешности в момент удаления нейрона

В момент удаления нейрона ошибка кратковременно возрастает, но потом быстро достигает своего прежнего значения. Однако с каждым последующим удалением нейрона сходимость ошибки замедляется, т.е. наблюдается вырождение сети.

Авторами предложено объединить усечение и метод динамического добавления нейронов, т.е. предпринята попытка создания **самоконфигурируемой сети.**

Суть метода в следующем.

Инициализация сети производится случайным образом, т.е. число нейронов в слоях определяется произвольным образом. Как будет показано ниже, это хорошо работает на простых задачах, в случае же более сложных задач количество нейронов выбирается близким к оптимальному варианту. Затем сеть начинает обучаться. Если нейронов в сети недостаточно, то относительная скорость изменения ошибки становится меньше критической величины и нужно добавить нейрон. Если сеть имеет только один скрытый слой, то нейрон добавляется именно в него. В случае нескольких скрытых слоев использовалось понятие **среднего веса нейрона в слое**, который определялся следующим образом.

Пусть мы имеем некоторый скрытый слой $X$. Тогда суммируем абсолютные значения всех нейронов следующего слоя $Y$ и делим на количество нейронов слоя $X$. Это и будет средним весом нейронов слоя $X$.

Подобным образом мы определяем средние веса для нейронов всех скрытых слоев нашей сети и выбираем слой с наибольшим весом. Именно туда мы и добавляем нейрон. Это делается потому, что нейроны этого слоя наиболее близки к насыщению, т.е. наблюдается нехватка нейронов в этом слое.

Теперь допустим, что мы сумели обучить сеть до нужной погрешности. Но это можно сделать и при избыточном количестве нейронов. Поэтому

предполагаем, что в сети избыточное количество нейронов и можно выкинуть самый малозначимый нейрон из сети. Чтобы определить его, вводится понятие **удельного веса нейрона**.

Удельный вес нейрона в сети определяется следующим образом. Пусть мы имеем какой-то скрытый слой $X$, а в нем нейрон $x$. Тогда следующий за ним слой обозначим через $У$. Просуммируем абсолютные значения весов тех синапсов, принадлежащих нейронам слоя $У$, которые взаимодействуют с аксоном нейрона $x$. Полученную сумму разделим на количество нейронов в слое $У$. Это и будет удельный вес нейрона $x$. Нейрон с наименьшим удельным весом в сети удаляется.

Чтобы было понятнее, представим себе крайнюю ситуацию, когда удельный вес нейрона равен нулю. Что это означает? Это означает, что нейрон с таким весом участвует в вычислениях, т.е. занимает ресурсы машины, а его вклад в состояние нейронов следующего слоя нулевой. Поэтому от такого нейрона надо избавиться. В реальной ситуации удельный вес редко равен нулю, поэтому удаление нейрона, пусть даже и с наименьшим удельным весом, обязательно отзовется на состояниях нейронов следующего слоя. Вот почему происходит увеличение ошибки (см. рис. 9). Если мы продолжим обучение сети, то можем компенсировать удаление нейрона, однако, если мы повторим процедуру удаления, то удаляемый нейрон будет иметь больший удельный вес, а следовательно, погрешность возрастет. Таким образом, можно заметить, что если использовать для оптимизации структуры сети только динамическое удаление нейронов, то это приведет к вырождению структуры и сеть не сможет обучиться. Вот почему необходимо скомпенсировать вырождение динамическим добавлением нейронов.

Однако выяснилось, что сеть все равно имеет тенденцию к вырождению, хотя это и проявляется значительно медленнее. Это связано с тем, что в ходе обучения часть нейронов переходит в насыщенное состояние и сеть слабо реагирует на введение дополнительных нейронов.

Чтобы избежать вырождения структуры, было предложено удалять **насыщенные нейроны** и одновременно добавлять новые.

Как выявить насыщенный нейрон? Для его выявления введено понятие стрессового порога. Он определяется эмпирически. В наших экспериментах он равен 0,9999. Если в ходе работы выход нейрона станет больше этой величины, то будем считать, что он «пережил» стресс и должен погибнуть.

Эта стратегия прекрасно работает на тривиальных задачах, типа проблемы исключающего ИЛИ (XOR), но на более сложных задачах, вроде обработки медицинской информации, она не работает: сеть просто не успевает обучиться до необходимого уровня.

После того как конфигурация сети начнет колебаться около какого-либо значения, это значение можно считать оптимальной конфигурацией сети. После достижения оптимума следует остановить алгоритм, т.к. сеть имеет тенденцию к вырождению.

В таблице 3 представлены сравнительные результаты работы различных способов поиска оптимальной структуры НС.

Таблица 3

| Задача | Метод | Время, мин | Конфигурация |
|---|---|---|---|
| XOR - проблема | ГА | 8 | 2/6/1-2/10/1 |
| XOR - проблема | Динамическая оптимизация | 5 | 2/8/1 |
| Медицинская задача | Выборочное тестирование | 35 | 53/30/1 |
| Медицинская задача | Динамическая оптимизация | 20 | 53/22/1 |

Если посмотреть на результаты, представленные в таблице 4, то видно, что предложенный авторами комплексный метод оптимизации структуры ИНС, а именно динамическое сжатие входных данных с использованием нейронных сетей и динамический выбор конфигурации сети, позволяет сократить время обучения сети в 7 раз, что доказывает эффективность данного метода.

Таблица 4

| Сеть | Конфигурация | Время обучения, мин | Число корректируемых весов |
|---|---|---|---|
| До сжатия входных данных | 53/30/1 | 35 | 1620 |
| | 53/22/1 | 20 | 1188 |
| После сжатия | 15/22/1 | 5 | 352 |

В последнее время одним из авторов (А.В.Ф.) была предложена удобная для врачей реализация нейронной сети, использующая все вышеизложенные методы, встроенная в широко известную систему Ms-Excel со значительным расширением ее функциональных возможностей. Такая схема организации анализа медицинских данных существенно облегчает работу с этим программным приложением и упрощает обучение врачебного персонала деятельности такого рода.

## Литература


1. Уоссермен Ф. Нейрокомпьютерная техника. М., Мир, 1992.
2. Заенцев И.В. Нейронные сети: основные модели. http://iiss.vit.narod.ru
3. Короткий С. Нейронные сети: алгоритм обратного распространения. http://www.orc.ru/~stasson/n2.zip
4. Карманова И.В. и др. Применение нейронных сетей для дифференцированной диагностики тяжести течения пневмонии. Труды конференции «Физика и радиоэлектроника в медицине и экологии (ФРЭМЭ - 2000)». Владимир, 2000.
5. Шахиди Акобир. Алгоритм обучения RProp - математический аппарат. http://www.basegroup.ru/neural/rprop.htm
6. Лоули Д., Максвелл А. Факторный анализ как статистический метод. М.:Мир, 1967.
7. Bryliuk D., Starovoitov V. Application of recirculation neural network and principal component analysis for face recognition. http://metalwarrior.narod.ru
8. Стариков А. Генетические алгоритмы – математический аппарат. http://www.basegroup.ru/genetic/math.htm
9. Thimm G. and Fiesler E. Evaluating Pruning Methods. http://www.idiap.ch/nn-papers/pruning/


Получено 4 декабря 2002 г.

Ососков Г. А., Филимонов А. В.
Р11-2002-274
Динамическая оптимизация структуры персептронов

Описывается комплексный метод динамической оптимизации структуры персептронов. Данный метод состоит из двух частей: понижение размерности входных данных с помощью рециркуляционной нейронной сети и динамического добавления нейронов, а также оптимизация структуры персептронов в ходе их обучения путем объединения методов динамического добавления и удаления нейронов из сети. Данная методика была апробирована на конкретной медицинской задаче и показала приемлемые результаты.

Работа выполнена в Лаборатории информационных технологий ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна, 2002

Перевод авторов

Ososkov G. A., Filimonov A. V.
P11-2002-274
Dynamical Optimization of the Perceptron Structure

A complex method of the perceptron structure dynamic optimization is proposed. This method consists of two parts: the reduction of the input data dimension by means of a recirculative neural network and dynamic restructuring of the perceptron, as well as its structure optimization. The latter is realized by applying two dynamic procedures: adding new and removing faint neurons from network. This method was approved by applying it to a medical problem and demonstrated acceptable results.

The investigation has been performed at the Laboratory of Information Technologies, JINR.

Communication of the Joint Institute for Nuclear Research. Dubna, 2002

Редактор *Е. К. Аксенова*
Макет *Е. В. Сабаевой*

Подписано в печать 19.12.2002.
Формат 60 × 90/16. Бумага офсетная. Печать офсетная.
Усл. печ. л. 1,06. Уч.-изд. л. 1,28. Тираж 310 экз. Заказ № 53673.

Издательский отдел Объединенного института ядерных исследований
141980, г. Дубна, Московская обл., ул. Жолио-Кюри, 6.
E-mail: publish@pds.jinr.ru
www.jinr.ru/publish/