

Б.И. Пригляднов

ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ ПРИ ОБРАБОТКЕ ЭКОНОМИЧЕСКОЙ ИНФОРМАЦИИ

В настоящей статье автор надеется привлечь внимание к проблеме, поднимаемой не только со стороны потребителей нейросетевых технологий, но и их российских производителей, имеющих хороший опыт применения этих перспективных технологий при решении вопросов, во многом аналогичных задачам, стоящим в финансовой и банковской сферах.

Общие сведения о нейронных сетях

В конце XX в. прикладная математика обогатилась новыми методами, в том числе методами искусственного интеллекта (ИИ). В последние десятилетия стремительно развивается прикладная область математики, изучающая искусственные нейронные сети (НС).¹ В настоящее время НС находят применение в распознавании образов, адаптивном управлении, автоматизации, прогнозировании, создании экспертных систем, организации ассоциативной памяти и т.д.² В экономике и финансах применение НС позволяет создавать экспертные системы, например, предсказывать показатели биржевого рынка. В обработке экономической информации в развитых странах НС применяют довольно часто и успешно. К сожалению, в России использование НС в обработке экономической информации оставляет желать лучшего.

Приведем некоторые сведения о НС для понимания сути вопроса. Основу каждой НС составляют относительно простые, в большинстве случаев однотипные элементы (ячейки), имитирующие работу нейронов мозга. Далее под нейроном будет подразумеваться искусственный нейрон, то есть элемент НС.

Каждый нейрон характеризуется своим текущим состоянием по аналогии с нервными клетками головного мозга, которые могут быть возбуждены или заторможены. Он обладает группой синапсов – однонаправленных входных связей, соединенных с выходами других нейронов, а также имеет аксон – выходную связь данного нейрона, с которой сигнал (возбуждения или торможения) поступает на синапсы следующих нейронов. Общий вид нейрона приведен на рис. 1. Каждый синапс

характеризуется величиной синаптической связи или ее весом w_i , который определяет значимость этой связи.



Рис 1. Модель нейрона

Текущее состояние нейрона определяется как взвешенная сумма его входов.

$$s = \sum_{i=1}^n x_i \cdot w_i \quad (1)$$

Выход нейрона есть функция его состояния:

$$y = f(s) \quad (2)$$

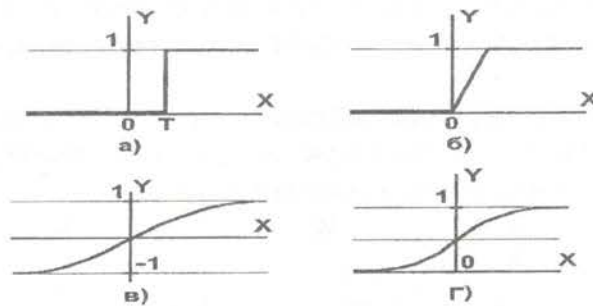


Рис.2 а) функция единичного скачка; б) линейный порог (гистерезис); в) сигмоид – гиперболический тангенс; г) сигмоид – формула (3)

Нелинейная функция f называется активационной и может иметь различный вид (рис. 2). Одной из наиболее распространенных является нелинейная функция с насыщением, так называемая сигмоида (то есть функция S-образного вида)[2]:

$$f(x) = \frac{1}{1 + e^{-\alpha x}} \quad (3)$$

При уменьшении α сигмоид становится более пологим, в пределе при $\alpha=0$ вырождаясь в горизонтальную линию на уровне 0.5, при увеличении α сигмоида приближается по внешнему виду к функции единичного скачка с порогом T в точке $x=0$. Из выражения для сигмоида очевидно, что выходное значение нейрона лежит в диапазоне $[0,1]$. Одно из ценных свойств сигмоидной функции – простое выражение для ее производной, применение которого будет рассмотрено в дальнейшем.

$$f'(x) = \alpha \cdot f(x) \cdot (1 - f(x)) \quad (4)$$

Следует отметить, что сигмоидная функция дифференцируема на всей оси абсцисс, что используется в некоторых алгоритмах обучения. Кроме того она обладает свойством усиливать слабые сигналы лучше, чем большие, и предотвращает насыщение от больших сигналов, так как они соответствуют областям аргументов, где сигмоид имеет пологий наклон.

В общем случае НС работают по принципу параллельной обработки сигналов, который достигается путем объединения большого числа нейронов в так называемые слои, соединенные определенным образом.

В качестве примера простейшей НС рассмотрим перцептрон из трех нейронов (рис. 3), то есть такую сеть, нейроны которой имеют активационную функцию в виде единичного скачка.

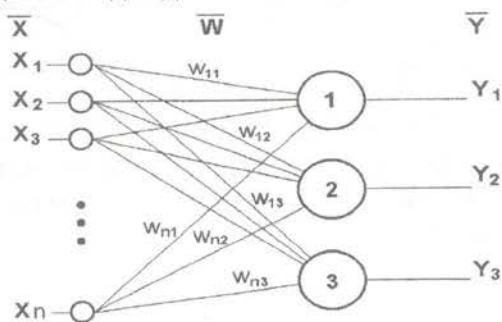


Рис 3. Трехнейронный перцептрон

На n входов поступают некие сигналы, проходящие по синапсам на три нейрона, образующие единственный слой этой НС и выдающие три выходных сигнала:

$$y_j = f \left[\sum_{i=1}^n x_i \cdot w_{ij} \right], \quad j=1...3 \quad (5)$$

Очевидно, что все весовые коэффициенты синапсов одного слоя нейронов можно свести в матрицу W , в которой каждый элемент w_{ij} задает величину i -ой синаптической связи j -ого нейрона. Таким образом, процесс, происходящий в НС, может быть записан в матричной форме:

$$Y=F(XW), \quad (6)$$

где X и Y – соответственно входной и выходной сигнальные векторы, $F(V)$ – активационная функция, применяемая поэлементно к компонентам вектора V .

Теоретически число слоев и число нейронов в каждом слое может быть произвольным, однако фактически оно ограничено ресурсами компьютера или специализированной микросхемы, на которых обычно реализуется НС. Чем сложнее НС, тем масштабнее задачи, подвластные ей.

Выбор структуры НС осуществляется в соответствии с особенностями и сложностью задачи. Для решения некоторых отдельных типов задач уже существуют оптимальные на сегодняшний день конфигурации.² Если же задача не может быть сведена ни к одному из известных типов, разработчику приходится решать сложную проблему синтеза новой конфигурации. При этом он руководствуется несколькими основополагающими принципами: возможности сети возрастают с увеличением числа ячеек сети, плотности связей между ними и числом выделенных слоев, введением обратных связей и др. Вопрос о необходимых и достаточных свойствах сети для решения того или иного рода задач представляет собой целое направление нейрокомпьютерной науки. Так как проблема синтеза НС сильно зависит от решаемой задачи, дать общие подробные рекомендации затруднительно. В большинстве случаев оптимальный вариант получается на основе интуитивного подбора.

Очевидно, что процесс функционирования НС, то есть сущность действий, которые она способна выполнять, зависит от величин синаптических связей, поэтому, задавшись определенной структурой НС, отвечающей какой-либо задаче, разработчик сети должен найти оптимальные значения всех переменных весовых коэффициентов (некоторые синаптические связи могут быть постоянными).

Этот этап называется обучением НС, и от того, насколько качественно он будет выполнен, зависит способность сети решать поставлен-

ные перед ней проблемы во время эксплуатации. На этапе обучения кроме параметра качества подбора весов важную роль играет время обучения. Как правило, эти два параметра связаны обратной зависимостью и их приходится выбирать на основе компромисса.

Обучение НС может вестись с учителем или без него. В первом случае сети предъявляются значения как входных, так и желательных выходных сигналов, и она по некоторому внутреннему алгоритму подстраивает веса своих синаптических связей. Во втором случае выходы НС формируются самостоятельно, а веса изменяются по алгоритму, учитывающему только входные и производные от них сигналы.

Существует великое множество различных алгоритмов обучения, которые однако делятся на два больших класса: детерминистские и стохастические.³ В первом из них подстройка весов представляет собой жесткую последовательность действий, во втором – она производится на основе действий, подчиняющихся некоторому случайному процессу. Сети также можно классифицировать по числу слоев. На рис. 4 представлен двухслойный перцептрон, полученный из перцептрона с рис. 3 путем добавления второго слоя, состоящего из двух нейронов. Здесь уместно отметить важную роль нелинейности активационной функции, так как если бы она не обладала данным свойством или не входила в алгоритм работы каждого нейрона, результат функционирования любой p -слойной НС с весовыми матрицами $\mathbf{W}^{(i)}$, $i=1,2,\dots,p$ для каждого слоя i сводился бы к перемножению входного вектора сигналов \mathbf{X} на матрицу

$$\mathbf{W}^{(\Sigma)} = \mathbf{W}^{(1)} \cdot \mathbf{W}^{(2)} \cdot \dots \cdot \mathbf{W}^{(p)}, \quad (7)$$

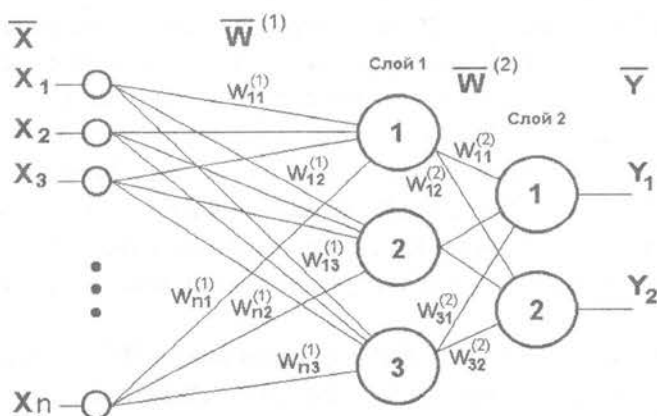


Рис. 4. Двухслойный перцептрон

то есть фактически такая p -слойная НС эквивалентна однослойной НС с весовой матрицей единственного слоя $\mathbf{W}^{(\Sigma)}$:

$$\mathbf{Y} = \mathbf{XW}^{(\Sigma)} \quad (8)$$

Продолжая рассматривать нелинейность, можно отметить, что она иногда вводится и в синаптические связи. Большинство известных на сегодняшний день НС используют для нахождения взвешенной суммы входов нейрона формулу (1), однако в некоторых приложениях НС полезно ввести другую запись, например:

$$s = \sum_{i=1}^n x_i^2 \cdot w_i \quad (9)$$

или даже

$$s = \sum_{i=1}^n x_i \cdot x_{((i+1) \bmod n)} \cdot w_i \quad (10)$$

Важный вопрос в том, чтобы разработчик НС четко понимал, для чего он это делает, какими ценными свойствами он тем самым дополнительно наделяет нейрон и каких лишает. Введение такого рода нелинейности увеличивает вычислительную мощь сети, то есть позволяет из меньшего числа нейронов с «нелинейными» синапсами сконструировать НС, выполняющую работу обычной НС с большим числом стандартных нейронов и более сложной конфигурации.

Теперь рассмотрим один нюанс, преднамеренно опущенный ранее. Из рисунка функции единичного скачка видно, что пороговое значение T в общем случае может принимать произвольное значение. Более того, оно должно принимать некое произвольное, неизвестное заранее значение, которое подбирается на стадии обучения вместе с весовыми коэффициентами. То же самое относится и к центральной точке сигмоидной зависимости, которая может сдвигаться вправо или влево по оси X , а также и ко всем другим активационным функциям. Это, однако, не отражено в формуле (1), которая должна была бы выглядеть так:

$$s = \sum_{i=1}^n x_i \cdot w_i - T \quad (11)$$

Такое смещение обычно вводится путем добавления к слою нейронов еще одного входа, возбуждающего дополнительный синапс каждого из нейронов, значение которого всегда равняется 1. Присвоим этому входу номер 0. Тогда $s = \sum_{i=0}^n x_i \cdot w_i$

$$\quad (12)$$

где $w_0 = -T$, $x_0 = 1$.

Очевидно, что различие формул (1) и (12) состоит лишь в способе нумерации входов.

Теперь более подробно рассмотрим вопрос обучения НС, например, используя перцептрон с рис. 3.

Рассмотрим алгоритм обучения с учителем. Задача обучения сводится к следующему.

1. Проинициализировать элементы весовой матрицы (обычно небольшими случайными значениями).

2. Подать на входы один из входных векторов, которые сеть должна научиться различать, и вычислить ее выход.

3. Если выход правильный, перейти на шаг 4.

Иначе, вычислить разницу между идеальным и полученным значениями выхода:

$$\delta = Y_i - Y.$$

Модифицировать веса в соответствии с формулой:

$$w_{ij}(t+1) = w_{ij}(t) + v \cdot \delta \cdot x_i.$$

где t и $t+1$ – номера соответственно текущей и следующей итераций; v – коэффициент скорости обучения, $0 < v \leq 1$; i – номер входа; j – номер нейрона в слое.

Очевидно, что если $Y_i > Y$, весовые коэффициенты будут увеличены, тем самым они уменьшат ошибку. В противном случае они будут уменьшены, и Y тоже уменьшится, приближаясь к Y_i .

4. Вернуться на шаг 2, в случае неприемлемой ошибки.

На втором шаге на разных итерациях поочередно в случайном порядке предъявляются все возможные входные вектора. К сожалению, нельзя заранее определить число итераций, которые потребуются выполнить, а в некоторых случаях и гарантировать полный успех.

Применение нейронных сетей

Одно из важных применений НС – решение задачи классификации, которая представляет собой отношение образца к одному из нескольких попарно не пересекающихся множеств. Примеры таких задач: определение кредитоспособности клиента банка; управление портфелем ценных бумаг (продать, купить или «придержать» акции в зависимости от ситуации на рынке); определение жизнеспособных и склонных к банкротству фирм.

При решении задач классификации необходимо отнести имеющиеся *статические образцы* (характеристики ситуации на рынке, данные медосмотра, информация о клиенте) к *определенным классам*. Таким путем образец представляется вектором. Компоненты этого вектора представляют собой различные характеристики образца, которые влияют на принятие решения о том, к какому классу можно отнести данный образец. Например, для медицинских задач в качестве компонентов этого вектора могут быть данные из медицинской карты больного. На основании некоторой информации о примере необходимо оп-

ределить, к какому классу его можно отнести. Классификатор таким способом относит объект к одному из классов в соответствии с определенным раскладыванием N -мерного пространства, которое называется *пространством входов*, а размерность этого пространства является количеством компонент вектора.⁴

Прежде всего нужно определить уровень сложности системы. В реальных задачах часто возникает ситуация, когда количество образцов ограничено, что осложняет определение сложности задачи. Возможно выделить три основных уровня сложности. Первый (самый простой), когда классы можно разделить прямыми линиями (или гиперплоскостями, если пространство входов имеет размерность больше двух), — так называемая *линейная делимость*. Во втором случае классы невозможно разделить линиями (плоскостями), но их возможно отделить с помощью более сложного деления — это *нелинейная делимость*. В третьем случае классы пересекаются, можно говорить только о *вероятностной делимости*.

В идеальном варианте после предварительной обработки необходимо получить линейно делимую задачу, так как после этого значительно упрощается построение классификатора. К сожалению, при решении реальных задач имеется ограниченное количество образцов, на основании которых и производится построение классификатора. При этом не представляется возможным провести такую предобработку данных, при которой будет достигнута линейная делимость образцов.

Сети с прямой связью являются универсальным средством аппроксимации функций, что позволяет их использовать в решении задач классификации. Как правило, нейронные сети оказываются наиболее эффективным способом классификации, так как генерируют фактически большое число регрессионных моделей используемых в решении задач классификации статистическими методами.

Следует иметь в виду, что в применении нейронных сетей в практических задачах возникает ряд проблем. Во-первых, заранее не известно, какой сложности (размера) может потребоваться сеть для достаточно точной реализации отображения. Эта сложность может оказаться чрезмерно высокой, что потребует усложнения архитектуры сетей. Так, Минский в своей работе «Перцептроны» доказал, что простейшие однослойные нейронные сети способны решать только линейно делимые задачи. Это ограничение преодолимо при использовании многослойных нейронных сетей. В общем виде можно отметить, что в сети с одним скрытым слоем вектор, соответствующий входному образцу, преобразуется скрытым слоем в некоторое новое пространство, которое может иметь другую размерность, а затем гиперплоскости,

соответствующие нейронам выходного слоя, разделяют его на классы. Таким образом, сеть распознает не только характеристики исходных данных, но и «характеристики характеристик», сформированные скрытым слоем. Приведем один из разработанных алгоритмов построения классификатора на основе нейронных сетей.

1. Работа с данными.

1. Составить базу данных из примеров, характерных для данной задачи
2. Разбить всю совокупность данных на два множества: обучающее и тестовое (допускаются 3 множества: обучающее, тестовое и подтверждающее).

2. Предварительная обработка.

1. Выбрать систему признаков, характерных для данной задачи, и преобразовать данные соответствующим образом для подачи на вход сети (нормировка, стандартизация и т.д.). В результате желательно получить линейно отделяемое пространство множества образцов.
2. Выбрать систему кодирования выходных значений (классическое кодирование, 2 на 2 кодирование и т.д.)

3. Конструирование, обучение и оценка качества сети.

1. Выбрать топологию сети: количество слоев, число нейронов в слоях и т.д.
2. Выбрать функцию активации нейронов (например, «сигмоида»).
3. Выбрать алгоритм обучения сети.
4. Оценить качество работы сети на основе подтверждающего множества или по другому критерию, оптимизировать архитектуру (уменьшение весов, прореживание пространства признаков).
5. Остановиться на варианте сети, который обеспечивает наилучшую способность к обучению, и оценить качество работы по тестовому множеству.

4. Использование и диагностика.

1. Выяснить степень влияния различных факторов на принимаемое решение (эвристический подход).
2. Убедится, что сеть дает требуемую точность классификации (число неправильно распознанных примеров мало).
3. При необходимости вернуться на этап 2, изменив способ представления образцов или изменив базу данных.

6. Практически использовать сеть для решения задачи.

Рассмотрим также применение НС на примере прогнозирования объемов продажи определенного товара. Алгоритм прогнозирования выглядит так:

Первый шаг. В компьютер заносится «история», то есть сведения о том, как раньше при различном стечении обстоятельств этот товар продавался. (В жизни мы бы назвали это «опытом работы»).

Второй шаг. Запускается на обучение программа, моделирующая нейронную сеть. Она почти как человек пытается спрогнозировать методом проб и ошибок, преобразует этот «жизненный опыт» и приходит к выводу, что, например, летом лыжи продаются плохо. Важно то, что, в отличие от человека, который одновременно может принимать во внимание 5-6 факторов, нейронная сеть может оперировать сотнями, дать прогноз лучше. Таким образом, после обучения вы получаете программу-эксперт.

Третий шаг. Спрашиваете систему, а она по мере своих возможностей отвечает. Можно узнать, как изменятся объемы продаж спустя два-три дня или два-три месяца; можно узнать, как изменится спрос, если снизить или повысить цену на 10%, и т. д.

Четвертый шаг. Время от времени нужно опять проходить обучение (повышение квалификации).

Реальные задачи не всегда сводятся к теории. Бывают случаи, когда нейронная сеть не дает качественного анализа. Причин тут в основном две. Первая: нейронная сеть используется не для тех задач. Она хорошо подходит для тех случаев, когда события происходят достаточно часто и есть необходимая для ее обучения предыстория. Вторая причина: к сожалению, очень часто на вход нейросети даются недостоверные данные или только часть необходимой информации.

Интересно использование технологий НС в разработке системы FOREX-94, предназначенной для поддержки принятия решений при операциях на валютных рынках.⁵ С точки зрения нейросетевых технологий система FOREX-94 представляет собой большую нейронную сеть, распадающуюся на три основных блока. Первый обрабатывает входящую информацию о валютных курсах, поступающую в оперативном режиме по DDE--каналу системы REUTERS («Money 2000») или Tenfore. Этот блок играет ту же роль, что рецепторы в биологических нейронных сетях. Затем во втором блоке проводится предварительная обработка полученной информации с помощью методов технического анализа. Отметим, что результаты такого анализа представляют самостоятельный интерес, поскольку по полноте информации, получаемой о рынке с помощью этой процедуры, система FOREX-94 сопоставима с таким известным аналитическим инструментом, как «REUTERS/Technical Analysis», отличаясь от него лишь гораздо боль-

шим «дружелюбием». Пользователь уже на этапе технического анализа получает не только технический график, отражающий тот или иной метод, но и конкретный совет «купить», «продать» или «воздержаться от операции». Совокупность таких советов, полученных различными методами технического анализа и тремя нейронными сетями, исполняющими также роль анализаторов валютных курсов, поступает в третий блок -- на входные нейроны сети, проводящей окончательную классификацию ситуации на рынке. О результатах такой классификации пользователь информируется с помощью обобщенного индикатора «Радуга Рынка». Это цветовой сигнал, изменяющий цвет от чисто желтого (в моменты времени, соответствующие достижению наивысших котировок, после которых, по мнению системы, произойдет падение курса) до красного (в моменты наименьшего значения курса, предшествующего его росту).

Настройка нейронных сетей в настоящее время в большой степени искусство, чем наука. Для использования возможностей НС непрофессионалами (не математиками и не программистами) существует механизм генетических алгоритмов. Они проводят самый настоящий естественный отбор, в ходе которого выживает сильнейший, то есть лучшая сеть, наиболее подходящая для решения поставленной задачи. Пользователи освобождаются от необходимости изучать литературу по нейросетям, оставляя всю или почти всю рутинную работу генетическим алгоритмам.

Генетические алгоритмы -- это самообучающаяся модель, которая для решения поставленных перед ней задач использует механизмы естественного отбора, и здесь используются такие понятия, как популяция, особь, наследование и мутация. Сначала алгоритм выдвигает возможные решения. Затем в каждом поколении, подобно биологическим организмам, «сильнейшие» решения «скрещиваются», порождая новые решения, а «слабые» погибают. Генетические алгоритмы -- один из методов оптимизации. Их сфера не ограничивается только настройкой нейронных сетей, включая также решение самых различных задач, когда можно составить описание возможных вариантов решения в виде вектора параметров и известен (может быть вычислен) критерий, определяющий эффективность каждого варианта. Генетические алгоритмы вполне успешно применяются для составления расписаний и портфелей ценных бумаг, заполнения контейнеров при перевозке (пересылке), для выбора маршрутов движения и конфигурации оборудования и др.

Использование генетических алгоритмов совместно с нейронными сетями для подбора входных данных и выбора оптимальных конфигураций сетей и параметров обучения уже показало свои сильные стороны и получает все большее распространение. Такой тандем позволяет

упростить работу широкому кругу пользователей - от практически не знакомых с теорией искусственных нейронных сетей до профессионалов.

Технологии, базирующиеся на использовании НС, могут с успехом использоваться при решении ряда задач финансового и экономического анализа, а также при проведении многих стандартных банковских операций (оценка кредитных рисков, решение задач банковского маркетинга, контроль операций с кредитными карточками и т.д.).

ПРИМЕЧАНИЯ

1. Ф. Уоссермен. Нейрокомпьютерная техника. - М., 1992. - С. 132.
2. Итоги науки и техники: физические и математические модели нейронных сетей. Том 1. - М., 1990.
3. Artificial Neural Networks: Concepts and Theory, IEEE Computer Society Press. - 1992.
4. Фондовый рынок и нейросети. Степанов В.С. МИР ПК #12/98.
5. Банковские технологии. - 1996. - №1,3.