Кукса П.П., Шмаков В.В., Панюшкин М.А.

Московский Государственный Технический Университет им. Н.Э. Баумана

E-mail: kouxa@online.ru
WWW: http://www.geocities.com/pkouxa

Применение нейронных сетей для кластеризации данных

За последнее десятилетие в области развития нейронных сетей всё отчётливее наблюдается стремительная интеграция теории нейронных сетей с другими, связанными с ней областями, в особенности со статистикой и обработкой сигналов. Основной задачей статистики является задача обработки информации, т.е. распределение собранных данных по различным категориям (категоризация или кластеризация данных). С этой задачей успешно справляются и обычные вычислительные машины, но нейронные сети решают её гораздо быстрее.

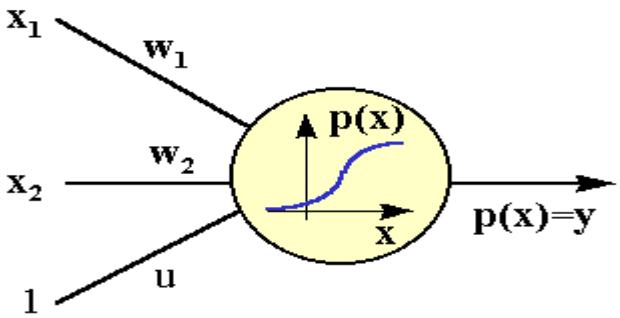

Рис.1. Математический нейрон.

Нейросеть строится на математических моделях нейронов. Такой нейрон — это несложный автомат, преобразующий входные сигналы в выходной сигнал (рис. 1). Сигналы $x_1, x_2, ..., x_n$, поступая на синапсы (участки мембраны на «хвосте» биологического нейрона, где размещаются области контакта данного нейрона с другими), преобразуются линейным образом, т.е. к телу нейрона поступают сигналы $w_1x_1, w_2x_2, ..., w_nx_n$ (здесь w_i — веса соответствующих синапсов). Для удобства к нейрону добавляют еще один вход (и еще один вес u), считая, что на этот вход всегда подается сигнал единицы. Нейрон суммирует эти сигналы, затем применяет к сумме некоторую фиксированную функцию p и выдает на выходе сигнал $y = p(w_1x_1 + w_2x_2 + ... + w_nx_n + u)$.

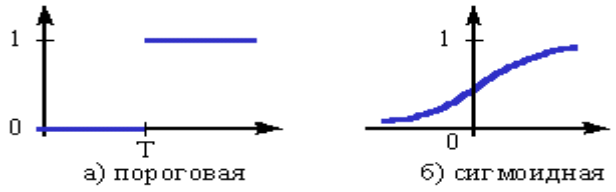

Рис.2. Передаточные функции нейронов.

Нейронная сеть — это набор нейронов, определенным образом связанных между собой. В качестве основного примера рассмотрим сеть, которая достаточно проста по структуре и в то же время широко используется для решения прикладных задач — трехслойный персептрон с n входами и одним выходом.

Как следует из названия, эта сеть состоит из трех слоев, изображенных на рис. 3. Собственно нейроны располагаются во втором (скрытом) и в третьем (выходном) слое. Первый слой только передает входные сигналы ко всем H нейронам второго слоя (здесь H = 4). Каждый нейрон второго слоя имеет n входов, которым приписаны веса $w_{i1}, w_{i2}, ..., w_{in}$ (для

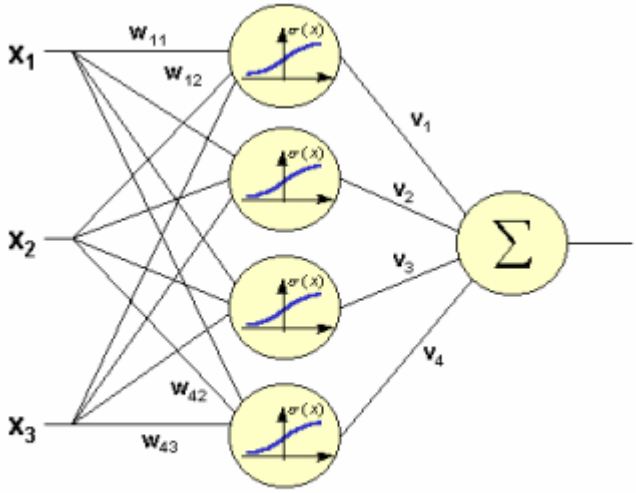

Рис.3. Трёхслойный персептрон

нейрона с номером *i*). Получив входные сигналы, нейрон суммирует их с соответствующими весами, затем применяет к этой сумме передаточную функцию (рис. 2) и пересылает результат на один из входов нейрона третьего слоя. В свою очередь, нейрон выходного слоя суммирует полученные от второго слоя сигналы с некоторыми весами v_i. Для определенности будем предполагать, что передаточные функции в скрытом слое являются сигмоидными, а в выходном слое используется функция $p(x) = x$, т. е. взвешенная сумма выходов второго слоя и будет ответом сети.

Итак, подавая на входы персептрона любые числа $x_1, x_2, ..., x_n$, мы получим на выходе значение некоторой функции $F(x_1, x_2, ..., x_n)$, которое является ответом (реакцией) сети. Очевидно, что ответ сети зависит как от входного сигнала, так и от значений ее внутренних параметров — весов нейронов.

Точный вид этой функции: $F(x_1, x_2, ..., x_n) = \sum_{i=1}^{H} v_i \cdot a(w_{i1}x_1 + w_{i2}x_2 + ... + w_{in}x_n + u_i$ (*)

Как видно из этого равенства, такой персептрон реализует только функции определенного вида, а именно суммы значений сигмоидных функций, где в качестве аргументов подставляются линейные комбинации входных сигналов. Например, функцию $F(x_1, x_2) = x_1x_2$ не удается с ходу представить в таком виде.

В принципе, нейронные сети могут вычислить любую функцию, имеющую решение, иными словами, делать все, что могут делать традиционные компьютеры.

На практике для того, чтобы применение нейронной сети было оправдано, необходимо, чтобы задача обладала следующими признаками:

- отсутствует алгоритм или не известны принципы решения задач, но накоплено достаточное число примеров;
- проблема характеризуется большими объемами входной информации;
- данные неполны или избыточны, зашумлены, частично противоречивы.

Таким образом, нейронные сети хорошо подходят для распознавания образов и решения задач классификации, оптимизации и прогнозирования, а также для решения большого числа дифференциальных уравнений.

Целью кластеризации является образование схожих между собой групп данных. Единицы данных могут быть представлены в различных видах, например, посредством представления характеристик объекта компонентами вектора. В этом случае весь объект характеризуется этим вектором. На основании характеристик производится принятие решения о том, к какой группе или классу необходимо отнести данный объект. Это значит, что классификатор относит объект к одному из классов в соответствии с определенным разбиением N – мерного пространства, которое называется пространством входов. Размерность этого пространства равна количеству компонент вектора, представляющего объект. Поэтому сжатие данных (уменьшение степени их избыточности) и выделение независимых признаков объекта значительно облегчает работу с данными.

Для проведения кластеризации необходимым является определение уровня сложности системы (уровень линейной разделимости, уровень нелинейной разделимости или уровень вероятностной разделимости).

Получение линейно – разделимой задачи приводит к значительному упрощению построения классификатора. При решении реальных задач количество примеров зачастую ограничено, что приводит к невозможности достижения линейной разделимости образцов. Поэтому разумным вариантом является использование нейронных сетей с прямой связью, так как они являются универсальным средством аппроксимации (приближения) функций.

При применении нейронных сетей в практических задачах возникает ряд проблем, например, заранее неизвестна необходимая сложность сети для точной реализации отображения данных. В частности, простейшие однослойные персептроны (обычно под персептроном понимают нейронную сеть, в которой нейроны имеют активационную функцию в виде единичного скачка) способны решать лишь линейно – разделимые задачи. При использовании многослойных нейронных сетей это ограничение снимается, так как, например, в сети со скрытыми слоями входной вектор преобразуется слоем в некоторое новое пространство, уже имеющее нужную размерность. После этого внешние слои производят разбиение нового

пространства на подпространства (классы) посредством гиперплоскостей. Таким образом, сеть распознает не только характеристики исходных данных, но и вторичные характеристики, сформированные скрытыми слоями.

При построении классификатора необходимо определить параметры, на основании которых происходит разбиение на классы. При этом характерны две проблемы: во-первых, если количество исходных данных мало, может возникнуть неоднозначность, поскольку один и тот же набор данных может соответствовать примерам, находящимся в разных классах. В этом случае невозможно провести правильное обучение нейронной сети, система не будет корректно работать. Во-вторых, исходные данные обязательно должны быть непротиворечивы. Для решения этой проблемы необходимо увеличивать размерность пространства признаков принадлежности тому или иному образцу. В этом случае количество исходных примеров может стать недостаточным, что приведет к тому, что сеть просто запомнит примеры из обучающей выборки и не сможет корректно функционировать.

Следующим шагом является определение способа представления данных для нейронных сетей. Так как в большинстве случаев, нейронные сети работают с данными, расположенными в диапазоне [0-1], то для её корректной работы необходимо преобразование исходных данных, имеющих произвольную размерность, именно к такому диапазону. При нормировании обычно осуществляется комплексный анализ параметров и нелинейная нормировка в зависимости от влияния параметров друг на друга.

Для того чтобы отнести объект к одному из двух существующих классов достаточно сети с одним нейроном в выходном слое, который может принимать одно из двух значений, ноль или один, в зависимости от того, к какому классу принадлежит образец. При наличии большего количества классов возникает проблема, связанная с представлением данных для выхода сети. Наиболее распространенным и простым является представление результата в виде вектора, компоненты которого соответствуют различным номерам классов. При этом *k*-я компонента вектора соответствует *k*-му классу. Все остальные компоненты приравниваются нулю. Например, второму классу будет соответствовать единица на втором выходе сети и ноль на остальных. В реальных условиях обычно считается, что номер класса определяется номером выхода сети, на котором появилось максимальное значение. Например, если в сети с тремя выходами мы имеем вектор выходных значений 0.1, 0.9, 0.4, тогда считается, что классифицируемый объект относится ко второму классу. При таком способе кодирования важную роль играет понятие *уверенности* сети в том, что пример относится к данному классу. Наиболее простым методом определения уверенности является вычисление разности между максимальным значением выхода, и значением другого выхода, которое является ближайшим к максимальному. Другими словами, в нашем случае уверенность равна 0.9 - 0.4 = 0.5. Соответственно, чем выше уверенность, тем больше вероятность того, что сеть дала правильный результат. Данный метод кодирования является самым простым, но далеко не самым оптимальным способом представления данных.

Гораздо более экономичен способ, при котором выходной вектор представляет собой номер класса, записанный в двоичной форме. Тогда, например, для классификации восьми классов потребуется вектор всего из 3-х элементов. Однако, в данном случае при получении неверного значения хотя бы на одном из выходов, весь номер класса становится ошибочным (впрочем, как и в случае прямого кодирования). Поэтому для повышения надежности необходимо применять кодирование выхода по коду Хэмминга, который позволяет существенно повысить надежность классификации.

Другим подходом является разбиение задачи с *k* классами на *k*(k-1)/2* подзадач с двумя классами каждая. То есть исходный вектор разбивается на группы по два компонента в каждой, таким образом, чтобы в них вошли всевозможные комбинации компонент выходного вектора. Число таких групп равняется числу неупорядоченных выборок по два из исходных компонент., т. е. $A=k(k-1)/2$

Тогда, например, для задачи с четырьмя классами мы имеем 6 выходов (подзадач) распределенных следующим образом:

№ подзадачи (выхода)	Компоненты выхода
1	1-2
2	1-3
3	1-4
4	2-3
5	2-4
6	3-4

При этом единица на выходе говорит о наличии одной из компонент. В этом случае мы можем перейти к номеру класса следующим образом: определяем единичные входы и определяем, какие подзадачи были активированы. При этом искомым классом будет тот, который вошел в наибольшее количество подзадач, например, для нашего примера:

№ класса	Активные выходы
1	1,2,3
2	1,4,5
3	2,4,6
4	3,5,6

Данное кодирование в большинстве случаев дает лучший результат, нежели классические способы кодирования.

Следующей задачей, которую необходимо решать для классифицирования, является правильный выбор объема сети. Если размер сети будет слишком велик, то данная сеть будет просто запоминать примеры из обучающей выборки и не производить аппроксимацию, что, естественно, приведет к некорректной работе классификатора. Слишком малая сеть не даст необходимой точности.

Правильный выбор объема сети практически невозможно произвести сразу. Поэтому обычно используют метод последовательного приближения в двух вариациях – конструктивной и деструктивной. В первом случае выбирается сеть минимального размера, а затем она постепенно увеличивается для достижения необходимой точности. При каждом шаге ее обучают заново. При деструктивном же подходе сначала строится сеть большего, нежели необходимо, объема, и постепенно производится удаление из нее узлов и связей, мало влияющих на решение. При этом необходимо, чтобы число примеров в обучаемом множестве всегда было больше числа настраиваемых входов. Иначе вместо проведения аппроксимации сеть просто запомнит данные, и результат будет не определен для примеров, не входящих в начальную выборку.

Алгоритм 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 (возможно разбиение на 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
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 функция единичного скачка)
3. Выбрать алгоритм обучения сети
4. Оценить качество работы сети на основе подтверждающего множества или другому критерию, оптимизировать архитектуру (уменьшение весов, прореживание пространства признаков)
5. Остановится на варианте сети, который обеспечивает наилучшую способность к обобщению 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 вернутся на этап 2, изменив способ представления образцов или изменив базу данных.

Для того чтобы построить качественный классификатор, необходимо иметь качественные данные (примеры обучения). Никакой из методов построения классификаторов, основанный на нейронных сетях или статистический, никогда не даст классификатор нужного качества, если имеющийся набор примеров не будет достаточно полным и представительным для той задачи, с которой придется работать системе.

Литература

1. Галушкин А. Современные направления развития нейрокомпьютерных технологий в России // Открытые системы. — 1997 г., №4.

2. Научная сессия МИФИ – 99. Всеросийская научно – техническая конференция нейроинформатика – 99. Дискуссия о нейрокомпьютерах. М.: МИФИ, 2000 г.

3. Борисов Ю., Кашкаров В., Сорокин С. Нейросетевые методы обработки информации и средства их программно-аппаратной поддержки // Открытые системы. — 1997 г., №4.

4. Горбань А. Нейроинформатика.

5. Терехов. А. Типовые задачи для информационного моделирования с использованием нейронных сетей. Снежинск. 2000 г.