

РЕАЛИЗАЦИЯ МОДЕЛИ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ В ОБЛАСТИ СЕРВИСНОГО ОБСЛУЖИВАНИЯ БАНКОМАТОВ

Рассматриваются основные подходы к построению модели системы поддержки принятия решений в специфичной области сервисного обслуживания банкоматов путем использования нейронных сетей. Приводится пример построения нейронной сети для реализации модели заданной системы.

1. Введение

Современный человек живет в мире техники, это касается всех сторон его существования - работы, дома и отдыха. Соответственно производство и продажа техники с каждым годом увеличиваются и в плане количества, и в плане расширения ассортимента. Но любая техника имеет свойство отказывать, что является нормальным. Надежной считается не та техника, которая не ломается, такой просто не бывает, а та, которую можно починить. Поэтому техническое обслуживание, которое в течение длительного времени не рассматривалось в качестве элемента экономической эффективности, сегодня является важнейшей областью, позволяющей эффективно использовать ресурсы предприятия. Хорошо организованная сервисная служба, которая поддерживает высокую степень надежности техники, может оказаться тем ключевым элементом, наличие или отсутствие которого означает успех или неудачу.

2. Постановка задачи

Эффективность сервисной службы во многом зависит от степени автоматизации ее работы.

Цель исследования – раскрыть содержание проблемы управления ресурсами в сфере сервисного обслуживания банкоматов и разработать механизм управления ресурсами предприятия в современных условиях хозяйствования. Такой механизм должен решать задачи координации поставки запасных частей и комплектующих, регулирования отпускных цен на запасные части для дилеров, финансовых расчетов с авторизованными Сервисными Центрами, контроля качества и сроков выполнения ремонтов.

3. Преимущества нейронных сетей для решения рассматриваемой задачи

На практике в сфере сервисного обслуживания банкоматов часто встречаются задачи, связанные с наблюдением случайных величин. Для подобных задач не удастся построить детерминированные модели, поэтому применяется принципиально иной, вероятностный подход. Параметры вероятностных моделей - это распределения случайных величин, их средние значения, дисперсии и т.д. Как правило, эти параметры изначально неизвестны, а для их оценки используются статистические методы, применяемые к выборкам наблюдаемых значений (историческим данным) [5].

К сожалению, классические методики оказываются малоэффективными во многих практических задачах. Это связано с тем, что невозможно достаточно полно описать реальность с помощью небольшого числа параметров модели, либо расчет модели требует слишком много времени и вычислительных ресурсов. В частности, рассмотрим одну из основных задач, решаемую в области сервисного обслуживания банкоматов - оптимальное распределение ресурсов и инвестиций, вкладываемых в развитие такого бизнеса.

В такой задаче ни одна из функций не известна точно - известны лишь приблизительные или ожидаемые значения используемых ресурсов (как материальных, так и человеческих). А однозначно определить размер ожидаемой прибыли в результате проведения ремонтных работ тем более тяжело. Для того чтобы избавиться от неопределенности, мы вынуждены зафиксировать функции, теряя при этом в точности описание задачи.

Детерминированный алгоритм для поиска оптимального решения (симплекс-метод) применим только в том случае, если все данные функции линейны. В реальных задачах оценки используемых ресурсов это условие не выполняется. Хотя данные функции можно аппроксимировать линейными, решение в этом случае будет далеким от оптимального.

Если одна из функций нелинейная, то симплекс-метод неприменим, и остается два традиционных пути решения этой задачи.

Первый путь – использовать метод градиентного спуска для поиска максимума прибыли. В данном случае область определения функции прибыли имеет сложную форму, а сама функция – несколько локальных максимумов, поэтому градиентный метод может привести к неоптимальному решению. Вторым путем – провести полный перебор вариантов распределения ресурсов. Если каждая из 10 функций задана в 100 точках, то придется проверить около 10^{20} вариантов, что потребует не менее нескольких месяцев работы современного компьютера.

Отметим также, что статистические методы хорошо развиты только для одномерных случайных величин. Если же мы хотим учитывать несколько взаимосвязанных факторов, то придется обратиться к построению многомерной статистической модели. Однако такие модели либо предполагают гауссовское распределение наблюдений (что не выполняется на практике), либо не обоснованы теоретически [5].

Таким образом, приходим к необходимости применения усовершенствованного математического аппарата.

Применение нейросетевой технологии уместно в случаях, когда формализация процесса решения трудна или вообще невозможна. Они являются очень мощным инструментом моделирования, поскольку нелинейны по своей природе. Как уже было сказано, линейное моделирование долгое время являлось основным в большинстве областей, поскольку для него существует большое число методов оптимизации. Однако для задачи анализа прибыли и распределения ресурсов предположение о линейности исходных данных в подавляющем большинстве случаев оказывается неверным. Кроме того, для нейронных сетей не существует проблемы «проклятия размерности», не позволяющей моделировать линейные зависимости от большого числа переменных.

Нейронная сеть применяется в первую очередь тогда, когда неизвестен точный вид связи между входом и выходом. Достаточно лишь точно знать, что связь между входными и выходными данными существует. При этом сама зависимость будет выведена в процессе обучения нейронной сети [2].

Подводя итог сказанному выше, можно сделать вывод, что по сравнению с традиционными технологиями нейронные сети обладают следующими преимуществами:

- Универсальность. Нейронные сети не зависят от свойств входных данных, для них не существует требования к определенному типу распределения исходных данных либо требования к линейности целевых функций.

- Не существует проблемы «проклятия размерности». Они способны моделировать зависимости в случае большого числа переменных.

- В отличие от статистических исследований не требуют большого объема данных.

- Ускоряют процесс нахождения зависимости за счет одновременной обработки данных всеми нейронами.

Несмотря на то, что нейронные сети способны решить практически любые задачи, во многих случаях их применение не является целесообразным. Для большинства задач более эффективным оказывается применение других математических моделей. Нейронные сети особенно хорошо зарекомендовали себя при решении задач классификации, прогнозирования, кодирования и декодирования информации [3].

4. Элементы построения нейронной сети

В работе [1] было введено понятие юнита (Unit) как минимального обучаемого элемента, способного самостоятельно обрабатывать информацию в гранулированном виде, и кластера (Cluster), как совокупности таких элементов.

Модель юнита, согласно [1], может быть представлена набором множеств

$$M_c = \{X, W_c, H_x, S_c, H_y, BF, y\}, \quad (1)$$

где $X = \{x_0, \dots, x_n\}$ – множество входных параметров – микроситуаций; $W_c = \{w_0, \dots, w_v\}$ – множество регулируемых весов (min); $H_x = \{h_{x1}, \dots, h_{xq}\}$ – множество скрытых входных параметров, соответствующих информационным гранулам на входах; $H_y = \{h_{y1}, \dots, h_{ye}\}$ – множество скрытых выходных параметров, соответствующих информационным гранулам на выходах; $S_c = (s_{c1}, \dots, s_{cr})$ – множество связей скрытых входных и выходных информационных гранул, определяющих цепочку преобразований гранул при активизации этой связи; $BF = (bf_1, \dots, bf_k)$ – множество базисных функций; y – выходной параметр юнита.

Модель кластера, согласно [1], также может быть представлена с помощью (1), где $X=(X_1 \vee X_2 \vee \dots \vee X_M)$ представляет собой совокупность входов юнитов, входящих в кластер; W_N, H_x, H_y – множества весов и скрытых параметров кластера; S_N – множество связей кластера, объединяющее связи юнитов; BF – набор базисных функций активаторных элементов; скалярный выход y заменен на векторный Y – объединенное множество выходных параметров кластера.

Получаем, что юнит является универсальным преобразователем информации (адаптивным аппроксиматором), имеющим n -входов и один выход. Он соответствует в модельном плане биологическому нейрону с его сетью синапсов, через которые организуются связи с другими нейронами. В свою очередь кластер с частичным объединением юнитов можно считать однослойной сетью клеток с входами, параллельно подведенными к каждой клетке (от каждого входа к каждой клетке).

5. Локальные и распределенные сети

Локальная сеть (LN – Local Network) содержит связанные юниты и кластеры. Она рассматривается как самостоятельный модуль обработки образов. Информационная модель локальной сети может быть представлена набором множеств

$$M_{LN} = \{P_X(X, C_X), W_{LN}, S_{LN}, P_Y(Y, C_Y)\}, \quad (2)$$

где X и C_X – вектор образа P_X и связи между его координатами, т.е. множество входных параметров и связи между ними, определяющие входной (преобразуемый сетью) образ P_X ; W_{LN} – множество регулируемых весов связей между юнитами и кластерами, входящими в локальную сеть; S_{LN} – множество связей юнитов и кластеров, входящих в локальную сеть; Y и C_Y – вектор образа P_Y и связи между его координатами, т.е. множество выходных параметров и связей между ними, определяющих выходной (преобразованный сетью) образ P_Y .

Структурно локальная сеть состоит из связанных юнитов и кластеров, а также обучающего, формирующего множество весов связей W_{LN} элементов сети. Наличие разноуровневых компонентов (юнитов и кластеров) в локальной сети позволяет строить достаточно сложные преобразующие структуры с полносвязной и слоистой организацией. Локальные сети с полносвязной организацией используются для ассоциативного запоминания образов. Локальные сети со слоистой организацией предназначены для классификации образов или аппроксимации заданных преобразований образов [2].

Распределенные сети (DN – Distributed Network) имеют кустовую организацию. Они включают несколько локальных сетей (кустов), обрабатывающих образы, между которыми имеют место слабые (одинарные) связи. Так могут реализоваться процессы, определяющие сложное поведение системы. Распределенные сети могут преобразовывать образы замкнуто: образы, проходящие через кусты, возвращаются преобразованные опять на первый куст. Динамический процесс преобразования образов в распределенной сети определяется цепочкой отображений

$$P_{X/t_1} \rightarrow F_{k1}(P_X, P_{Z1}) \rightarrow \dots \rightarrow F(P_X, P_{Zq}) \rightarrow P_{Y/t_2} \rightarrow (OY) \rightarrow P_{S/t_2} \rightarrow P_{X/t_2}, \quad (3)$$

где t_1 и t_2 – моменты времени, соответствующие 1-му и 2-му циклам преобразований; F_{ki} – отображенные образы в i -м кусте; P_S – образ состояния объекта управления (OY); P_X, P_Y, P_Z – входной, выходной и внешний образы распределенной сети.

При этом имеют место слабые связи между кустами, определяемые малыми весами $w_{i,i+1}$.

Такая организация сетей позволяет учесть влияние на процесс управления дополнительных факторов, изменяющих это управление через вводимые в кусты внешние образы P_Z .

6. Структура нейронной сети

Рассмотрим реализацию нейронной сети для системы поддержки принятия решений в области сервисного обслуживания банкоматов. Нейронная сеть должна реализовать нечетко-нейронное отображение входного n -мерного вектора X в выходной m -мерный вектор Y .

Нейронная сеть включает:

- фазификатор (процедура FUZ) - в качестве гранулятора [статья];
- активатор, состоящий из блока ассоциативного вывода и реализующий процедуру взвешенной агрегации (процедура INCL), объединения и памяти весов;
- дефазификатор (процедура DFUZ) - в качестве дегранулятора;
- блок вычисления текущих весов, реализующий процедуру коррекции весов (процедура UPD), и память образов.

Перечисленные процедуры имеют следующее содержание:

$$FUZ = \text{fuz}(x_i, \mu^q(x_i)) \quad (4)$$

– процедура, вычисляющая степень принадлежности переменной x_i к q -й нечеткой грануле этой переменной;

$$INCL = \text{incl}(\ln, w_i, \mu_{x_i}^q; i=1, \dots, I; q=1, \dots, K) \quad (5)$$

– взвешенная агрегация, вычисляющая каждую степень принадлежности переменной y_j к q -й нечеткой грануле этой переменной путем агрегирования степеней принадлежности входных переменных x_i , с дальнейшим умножением результата на вес w_i и объединением; I, K – количество агрегируемых степеней принадлежности и нечетких гранул соответственно; \ln – номер слоя сети, в котором выполняется агрегация;

$$DFUZ = \text{dfuz}(\mu_{y_j}^q, y_j; q=1, \dots, K; j=1, \dots, M) \quad (6)$$

– дефазификация, вычисляющая значение выходного параметра y_j , исходя из степеней принадлежности значения этого параметра к нечетким гранулам его шкалы;

$$UPD = \text{upd}(w_p, \mu_y^*, \mu_y, p) \quad (7)$$

– коррекция веса при обучении; w_p – текущий вес; μ_y^*, μ_y – степени принадлежности правильного и текущего значений выходного параметра y к нечеткой грануле соответственно (правильное значение параметра берется из примера для обучения); p – номер примера; в результате операции находятся веса при обработке $p+1$ примера.

Такой модуль является относительно простым для программной и аппаратной реализации и может обучаться в реальном времени.

7. Модель нейронной сети

Для построения нейронной сети необходимо разработать ее топологию, определить механизм обучения и процедуру тестирования. Кроме того, для обучения нужны входные данные – параметры, описывающие микроситуации [1] предметной области.

Для сферы сервисного обслуживания банкоматов в качестве входных данных используются разработанный автором набор параметров:

- параметры установки банкоматов;
- параметры эксплуатации банкоматов;
- параметры, описывающие состояние узлов и блоков;
- количество зарегистрированных заявок;
- количество просроченных заявок;
- количество невыполненных заявок;
- среднее время в ремонте;
- количество использованных запчастей;
- среднее время выполнения заявки.

На начальной стадии исследования наиболее привлекательным является выбор трех-слойного персептрона и алгоритма обратного распространения в качестве обучающего. Каждый элемент нейронной сети [4] строит взвешенную сумму своих входов с поправкой в

виде слагаемого и затем пропускает эту величину активации через передаточную функцию, и таким образом получается выходное значение этого элемента.

Элементы организованы в послонную топологию с прямой передачей сигнала. Такую сеть легко можно интерпретировать как модель вход-выход, в которой веса и пороговые значения (смещения) являются свободными параметрами модели. Такая сеть может моделировать функцию практически любой степени сложности, причем число слоев и число элементов в каждом слое определяют сложность функции. Определение числа промежуточных слоев и числа элементов в них является вопросом при конструировании, они будут отражены в дальнейших исследованиях. Количество входных и выходных элементов определяется условиями задачи.

Самый известный вариант алгоритма обучения нейронной сети - так называемый алгоритм обратного распространения (back propagation; [2]). В алгоритме обратного распространения вычисляется вектор градиента поверхности ошибок. Этот вектор указывает направление кратчайшего спуска по поверхности из данной точки, поэтому если мы «немного» продвинемся по нему, ошибка уменьшится. Последовательность таких шагов (замедляющаяся по мере приближения ко дну) в конце концов приведет к минимуму того или иного типа. Определенную трудность здесь представляет вопрос о том, какую нужно брать длину шагов [6].

При большой длине шага сходимость будет более быстрой, но имеется опасность перепрыгнуть через решение или уйти в неправильном направлении. Классическим примером такого явления при обучении нейронной сети является ситуация, когда алгоритм очень медленно продвигается по узкому оврагу с крутыми склонами, прыгая с одной его стороны на другую. Напротив, при маленьком шаге, вероятно, будет схвачено верное направление, однако при этом потребуются очень много итераций. На практике величина шага берется пропорциональной крутизне склона (так что алгоритм замедляет ход вблизи минимума) с константой, которая называется скоростью обучения. Правильный выбор скорости обучения зависит от конкретной задачи и обычно осуществляется опытным путем; эта константа может также зависеть от времени, уменьшаясь по мере продвижения алгоритма.

В первом приближении конфигурация разработанной сети следующая:

Общие параметры	Параметры скрытого слоя	Параметры выходного слоя
Количество узлов во входном слое: 13	Коэффициент обучения: 0.25	Коэффициент обучения: 0.01
Число узлов в скрытом слое: 9	Коэффициент инерции: 0.6	Коэффициент инерции: 0
Число узлов в выходном слое: 3	Затухание:0	Затухание:0
Алгоритм обучения: Обратное распространение		
Функция активации: сигмоидальная		

8. Заключение

В результате проведенных исследований моделей и методов, используемых для повышения эффективности, надежности и функциональности создаваемых информационных технологий, был разработан набор входных параметров для исследуемой предметной области. Отметим, что при выборе параметров модели основное внимание уделялось не соответствию расчета выбранных коэффициентов общепринятым методикам анализа ресурсов, а влиянию исходного набора данных на качество обучения сети и как следствие эффективность ее прогнозирования. Проведена сравнительная характеристика методов анализа и оценки распределения ресурсов. Особое внимание было уделено преимуществам и недостаткам нейросетевых методов. Нейронные сети не являются панацеей, во многих случаях более эффективным будет использование традиционных методов статистики. Не-

смотря на это, во многих областях нейронные сети зарекомендовали себя как более рациональное решение и требуются дальнейшие исследования в данной области.

Список литературы: 1. *Станкевич Л.А.* Когнитивные нейробиологические системы управления. Проблемы нейрокибернетики. Материалы XII Международной конференции по нейрокибернетике (Ростов-на-Дону, октябрь 1999), Ростов-на-Дону, 1999. 2. *Haykin S.* (1994). *Neural Networks: A Comprehensive foundation.* Macmillan College Publishing Company Inc. 3. *Widrow B., Rumelhart D.E. and Lehr M.A.* (1994) *Neural networks: Application in industry, business and science, Communications of the ACM* 37 (3) : 93-105 4. *Черноруцкий И.Г.* Методы принятия решений. Л.: ЛПИ, 1989. 5. *Борисов Ю., Кашкаров В., Сорокин С.* Нейросетевые методы обработки информации и средства их программно-аппаратной поддержки / Открытые системы. 1997. №4.

Поступила в редколлегию 11.12.2008

Кузмин Александр Яковлевич, д-р техн. наук, проф. кафедры информатики, начальник инновационно-маркетингового отдела ХНУРЭ. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел.: 8 (057) 702-15-15, e-mail: kuzy@kture.kharkov.ua.

Головий (Гусарь) Наталья Владимировна, аспирантка кафедры информатики ХНУРЭ. Научные интересы: системный анализ данных. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. (057) 702 15 15, e-mail: rica1982@mail.ru.

Ясер Даюб, аспирант кафедры информатики ХНУРЭ. Научные интересы: системный анализ данных. Адрес: Украина, 61166, Харьков, пр.Ленина,14, тел. (057) 702 15 15, e-mail: kuzy@kture.kharkov.ua.

УДК 004.896, 004.932

И.А.ВЕРЕЩАК

МЕТОД НИЗКОУРОВНЕВОЙ ОБРАБОТКИ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ

Рассматриваются вопросы цифровой обработки графической информации. Описываются разработанные методы фильтрации и сегментации цветных изображений, приводятся результаты их тестирования.

Актуальность. Цифровая обработка графической информации используется во многих областях науки и техники для решения большого спектра задач.

Обработка графической информации способна дать много данных об объектах окружающего мира и единственная проблема, мешающая использовать видеоконтроль, заключается в сложности выбора необходимых методов и этапов преобразования получаемой графической информации.

Постановка задачи

Одна из задач обработки графической информации, требующих решения – это мониторинг пассажиропотока. Вовремя поступающая информация о плотности пассажирского потока способна повысить качество работы систем управления городским пассажирским транспортом.

В общем случае, решая задачу мониторинга пассажиропотока, необходимо:

- произвести предварительную обработку (фильтрацию) изображения,
- перевести графическую информацию из цветовой модели RGB в HSB,
- разложить входное изображение по цветам,
- провести сглаживание изображения,
- определить размеры выделенных областей (площадь),
- определить количество областей, размер которых больше заданного порога.

Методы

При решении данной задачи необходимо разработать или выбрать из существующих такие методы преобразования информации, которые бы позволили выделить на кадре пассажиров и определить их количество. Этот этап как правило называют сегментацией. Зачастую процессу сегментации предшествует процесс фильтрации и сглаживания изображения.