Учебный пример номер один

Наилучший способ объяснить, как работать с NeuroShell 2, - это построить приложение на основе нейронной сети. Поэтому ниже следует учебный пример использования NeuroShell 2, в котором создается сеть для предсказания ежедневной стоимости электричества в доме. Файл задачи под названием ELECTRIC входит в комплект примеров NeuroShell 2, поставляемых на дистрибутивной дискете, поэтому по мере чтения этого примера Вы можете выполнять все описываемые действия на компьютере.

Задача

Син ДюПри работает управляющим по продажам в фирме, разрабатывающей "умные" дома, имеющие встроенные компьютерные системы для контроля за безопасностью и за потреблением энергии, выключения и включения света и т.д. Как управляющий по продажам, он сам жил в одном из таких домов с 1992 года. Так как конкуренция на рынке недвижимости в его регионе обострилась, он хочет создать нейронную сеть, которая поможет ему продавать дома его фирмы.

Выходы

Стоимость электроэнергии в его регионе относительно высока, и он надеется, что его "умные" дома смогут существенно сэкономить деньги владельцев. Он решает создать сеть, которая будет предсказывать ежедневную стоимость электроэнергии в домах, которые он продает.

Входы

После того, как он решил, что он хочет предсказывать, Син должен решить, какие переменные следует принять во внимание для осуществления такого предсказания. Он думает, что на стоимость электроэнергии, потраченной в доме за день, влияют несколько факторов, в том числе средняя температура за день, количество жильцов в доме, их привычка оставлять включенными свет и бытовые приборы и т.д. Он хочет, чтобы при первой попытке построить приложение на основе нейронной сети задача оставалась простой, поэтому он решает использовать переменную, которую он считает наиболее важной, среднюю температуру за день, для предсказания ежедневной стоимости электроэнергии в своем собственном доме.

Заметим, что до того, как решиться использовать нейронную сеть, Син рассматривал возможность использования регрессионного анализа, однако решил, что один момент делает построение линейной модели сложным: если температура лежит в диапазоне от 32 до 45 градусов Фаренгейта (от 0 до 7 градусов Цельсия), то он топит свою печь дровами, поэтому ежедневная стоимость электроэнергии зависит от температуры нелинейно.

Запуск программы-примера

После того, как Син решил, что именно он хочет предсказать и какие переменные он собирается использовать для этого предсказания, он готов к использованию NeuroShell 2.

Вы можете вслед за ним выполнять его пример, щелкнув дважды мышью по значку группы программ NeuroShell 2, отображаемому в Диспетчере Программ Windows. Щелкните дважды по значку "мозга" NeuroShell 2. В меню Файл NeuroShell 2 выберите пункт Открыть задачу. Появятся несколько окон списков с именами подкаталогов и файлов. Щелкните дважды мышью по подкаталогу примеров NeuroShell 2 (EXAMPLES). Появится список файлов с расширением .DSC. Чтобы открыть Учебный пример номер один, щелкните мышью один раз по файлу "ELECTRIC.DSC". Щелкните по кнопке "OK", чтобы загрузить файл.

Появится Главное меню NeuroShell 2 со значками модулей для начинающего, для профессионала и модуля средств автономного использования. Вызовите Систему нейронных сетей для начинающего, дважды щелкнув по значку "трехколесный велосипед".

Обычный порядок действий для создания приложения в NeuroShell 2, как в Системе для

начинающего, так и в Системе для профессионала, состоит в последовательном переходе от значка к значку слева направо, сверху вниз.

Ввод данных (Таблица)

Поскольку Син собирается вводить данные непосредственно в NeuroShell 2, пропустите первый значок, Импорт файлов. Щелкните дважды по значку Ввод данных, появится форма типа электронной таблицы для ввода данных.

Обратите внимание на появившееся на экране предупреждение о том, что Таблицу следует использовать только для небольших задач. Задача Сина считается небольшой, поэтому для этого примера Таблица прекрасно подходит. В конце этого учебного примера мы расскажем Вам, как работать с файлами большего размера, используя Вашу обычную программу электронных таблиц.

Прежде всего, Син решает включить имена переменных как названия столбцов с данными. Он набирает 1 в поле правки "Номер строки с именами переменных". Он набирает 2 в поле правки "Первая строка, содержащая используемые данные".

Когда Син открывает новую задачу, в Таблице в строке 1 отображаются фиктивные имена столбцов Имя 1, Имя 2 и Имя 3, а в строках 2 и 3 под именами столбцов отображаются нулевые значения.

В строке 1 он вводит имена "Ср. температура" и "Стоимость/день". Начиная со строки 2, он набирает данные за прошлый год. Он вводит ежедневные отсчеты температуры за 1995 год, рассортированные в порядке от самой низкой до самой высокой температуры, а также стоимость за каждый день. Данные каждого дня вводятся на отдельной строке. Его окончательная электронная таблица содержит 1 строку имен и 365 строк данных. Он вызывает пункт Сохранить файл меню Файл для сохранения данных под именем ELECTRIC.PAT. (Этот файл за Вас уже создан.)

Син также хочет создать другой набор данных, который он будет использовать для проверки натренированной сети. (Этот файл, под названием ELECTRIC.PRO, также уже создан.) Все еще находясь в модуле Таблица, он выбирает из меню Файл пункт Новый. В отличие от первого раза, когда Син открыл Таблицу, появляется электронная таблица без каких-либо данных в ячейках.

Когда Вы открываете модуль Таблица первоначально, и появляется электронная таблица, в ней отображаются фиктивные имена столбцов и данные, которые означают, что в Таблице уже был установлен формат таким образом, что ячейки строки 1 имеют текстовый тип, а ячейки строк 2, 3 и т.д. содержат числа с десятичными знаками. Если Вы открываете второй лист Таблицы из меню Файл Таблицы, эта фиктивная информация отображаться не будет, и Вам придется использовать меню Формат Таблицы, чтобы установить для строки 1 текстовый тип, если Вы хотите ввести и отображать имена столбцов.

Син щелкает по строке 1, чтобы целиком пометить ее. Затем он выбирает из меню Формат пункт Тип данных, чтобы выбрать тип Текст, который позволит ему вводить имена столбцов. Он выбирает для текста Выравнивание слева, чтобы это соответствовало установкам в другой его программе электронных таблиц.

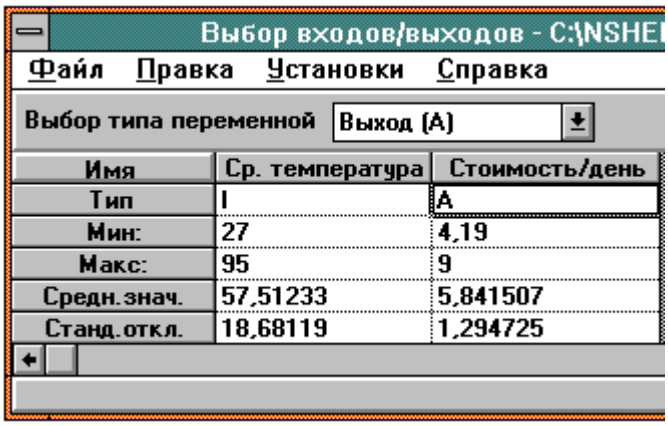
Син вводит в Таблице имена столбцов "Ср. температура" и "Стоимость/день". Син выбирает 46 дней из данных за 1996 год так, чтобы они перекрывали весь диапазон температур. Он вводит температуру и стоимость за день для каждого из 46 дней. Затем он вызывает из меню Файл пункт Сохранить файл как, чтобы сохранить файл под именем ELECTRIC.PRO.

Выбор входов и выходов

Теперь Сину нужно проинформировать NeuroShell 2 о том, какие из столбцов являются входами, а какие - выходами. Щелкните дважды по значку модуля Выбор входов/выходов. Модуль покажет на экране имена всех столбцов. Вам необходимо ввести Ваш выбор для каждого столбца в строке Тип переменной. Выберите с помощью мыши в поле списка один из вариантов: Вход (I - Input), Выход (A - Actual Output) или Не используется (пробел). Сделав выбор, щелкните по ячейке в строке Тип переменной под именем столбца, чтобы отметить Ваш выбор.

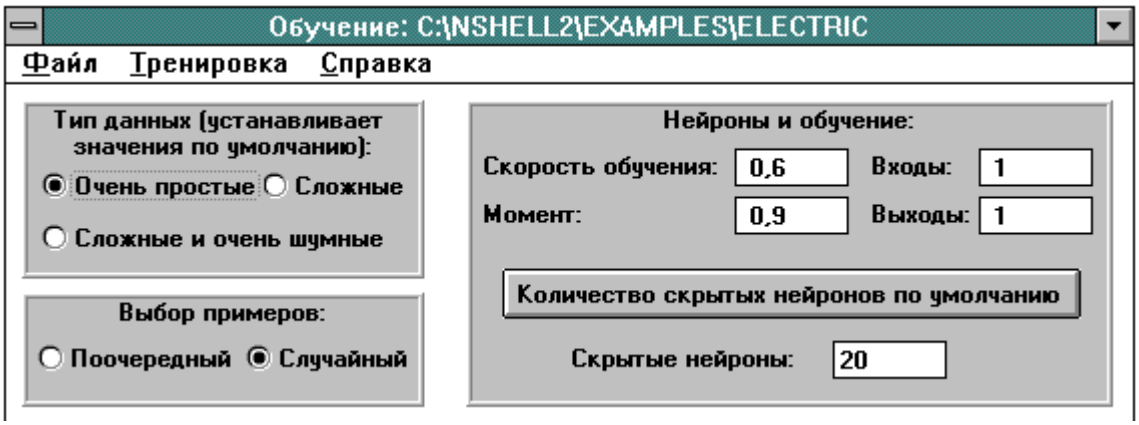
Син решает обозначить столбец "Ср. температура" как вход, I, а столбец "Стоимость/день" как столбец, который сеть пытается предсказать (выход, A).

Следующим шагом является ввод минимального и максимального значений для каждой переменной в строках Минимум и Максимум. Поскольку нейронные сети требуют перевода переменных путем масштабирования в диапазоны от 0 до 1 или от -1 до 1, сети необходимо знать истинный диапазон значений переменной. В этом модуле Вы можете ввести минимальное и максимальное значения для каждой переменной, которая должна использоваться сетью, или Вы можете вычислить диапазон автоматически из Ваших данных, вызывая пункт Расчет мин/макс меню Установки. Выбор этого пункта меню также приводит к вычислению среднего и стандартного отклонения для каждой переменной.



В общем случае, используйте диапазон, границы которого вплотную примыкают к Вашим данным. (Вы можете захотеть указать значения минимума и максимума, которые будут чуть меньше и чуть больше соответствующих значений в Вашем файле данных, чтобы предусмотреть более широкий диапазон для будущих предсказаний, или Вы можете предпочесть выбрать диапазон более узкий, чтобы исключить выбросы, которые могут повлиять на точность работы сети. За подробностями обращайтесь к разделу **Выбор входов и выходов**.) Если Вы не установите значения минимума и максимума вплотную к данным, сеть может потерять способность отслеживать мелкие различия в данных.

Син переходит к тренировке сети, щелкая дважды по значку Обучение, чтобы вызвать модуль тренировки. При появлении этот модуль уже должен знать о количестве входов и выходов из .MMX-файла (созданного в модуле Выбор входов/выходов). Тем не менее, есть еще несколько параметров, которые Сину необходимо задать, прежде чем он сможет начать тренировку. Во-первых, он должен указать сложность задачи, щелкнув по одному из переключателей в левой верхней части экрана. Хотя его задача не принадлежит к "игрушечным" задачам того типа, какие любят использовать для целей проверки многие изобретатели нейронных сетей и журналисты, специализирующиеся на их описании, вроде задачи Исключающее ИЛИ, Син решает, что его данные очень простые. Он щелкает мышью по кнопке Очень простые. Обратите внимание, что при этом значения скорости обучения и момента автоматически устанавливаются равными 0,6 и 0,9.



Затем следует установить количество скрытых нейронов. Щелчок по кнопке вычисления по умолчанию вызывает вычисление количества по встроенной формуле NeuroShell 2, которая дает хорошее начальное приближение для будущих задач. Количество скрытых нейронов по умолчанию для 3-слойной сети вычисляется по следующей формуле: Количество скрытых нейронов = 1/2 (входы + выходы) + корень квадратный из количества примеров в .TRN-файле, если он существует, иначе в .PAT-файле.

Син решает использовать случайный порядок представления данных сети, так как данные отсортированы в порядке от низких температур к высоким, а он хочет, чтобы сеть могла хорошо предсказывать при всех температурах.

Для своего первого сеанса тренировки Син устанавливает Интервал **Калибровки** равным 0, хотя это мощный инструмент, который следует использовать для большинства задач.

Пора начать тренировку, выбирая пункт Начать тренировку из меню Тренировка. Статистика на тренировочном наборе обновляется каждую эпоху (один полный проход тренировки по всем данным).

Основной статистический показатель тренировки - это внутренний средний "показатель ошибки". Вы не можете сами вычислить или точно воспроизвести этот средний показатель на тренировочном наборе, да это и совершенно не нужно. Тем не менее, это среднее по всем примерам значение квадрата ошибки всех выходов, вычисленных в пределах внутреннего интервала NeuroShell 2. Значение этого числа само по себе бесполезно. Полезно в процессе тренировки видеть, улучшается ли качество сети, т.к. по мере улучшения сети показатель уменьшается.

Когда следует остановить тренировку? Син замечает, что после 1000 эпох (эпоха это полный проход через все 365 тренировочных примеров) минимальная средняя ошибка равна примерно 0,0017 и что количество эпох после достижения минимальной ошибки равно примерно 400. Сеть, похоже, прекратила дальнейшее уменьшение минимальной средней ошибки. Чтобы остановить тренировку, он выбирает пункт Прервать тренировку из меню Тренировка. (Сети для профессионала имеют механизмы, автоматически останавливающие тренировку за Вас.)

Применение к файлу

Син хочет посмотреть, хорошие ли результаты дает его сеть. Он дважды щелкает по значку Применение к файлу и выбирает пункт Начать применение из меню Работа. По умолчанию флажки "Вычислять R квадрат и т.д.", "Включать выходы в .OUT-файл" и "Включать в .OUT-файл разности между выходами и ответами сети" включены.

Модуль Применение по умолчанию обрабатывает .PAT-файл, который представляет собой первый набор данных, введенный Сином. Температура каждого дня (вход) обрабатывается натренированной сетью, которая вычисляет стоимость электричества за этот день. На экране отображаются статистические показатели, измеряющие точность работы натренированной сети.

Натренированная сеть дает значение R квадрат на .PAT-файле 0,9539. Он записывает значение R квадрат, чтобы сравнивать эту сеть с другими, которые он может создать позже.

R квадрат, коэффициент множественной детерминации, - это статистический индикатор, обычно используемый при анализе методом множественной регрессии. Он сравнивает точность модели с точностью тривиальной реперной модели, для которой предсказание представляет собой просто среднее по всем примерам. При безупречном совпадении предсказаний с желаемыми значениями R квадрат будет равен 1, при хорошем совпадении - близок к 1, а при очень плохом совпадении близок к 0. Если предсказания Вашей нейросетевой модели хуже, чем можно было бы предсказать, просто используя среднее значение выхода по всем Вашим примерам, значение R квадрат будет равно 0.


Теперь Син просматривает .OUT-файл, в котором показаны реальные значения стоимости за день, которые он вводил в файл, предсказания сети (ее выход) и разница между ними. Вы можете дополнительно захотеть "приписать" выходной файл к исходному файлу данных, чтобы видеть исходный файл данных вместе с предсказаниями сети. Модуль Приписывание выходного файла сделает это за Вас, создавая при этом новый .OUT-файл. Для вызова этого модуля щелкните по значку Приписывание выходного файла. Поскольку Вы, вероятно, хотите увидеть результаты работы сети рядом со значениями входной переменной, щелкните по значку Приписать файл сбоку. (Правильные имена файлов должны отображаться автоматически. Исходный файл Сина - ELECTRIC.PAT, а файл ELECTRIC.OUT содержит ответы сети.)


Просмотр данных

Теперь Син готов посмотреть на .OUT-файл. Он может сделать это с помощью нашей Таблицы, дважды щелкнув по значку Просмотр данных.

Таблица не является электронной таблицей коммерческого класса, и при загрузке больших файлов работает довольно медленно. Если у Вас очень быстрый компьютер, то это не причинит Вам неудобств, в противном случае используйте Вашу программу работы с электронными таблицами.

Использование Вашей обычной программы работы с электронными таблицами

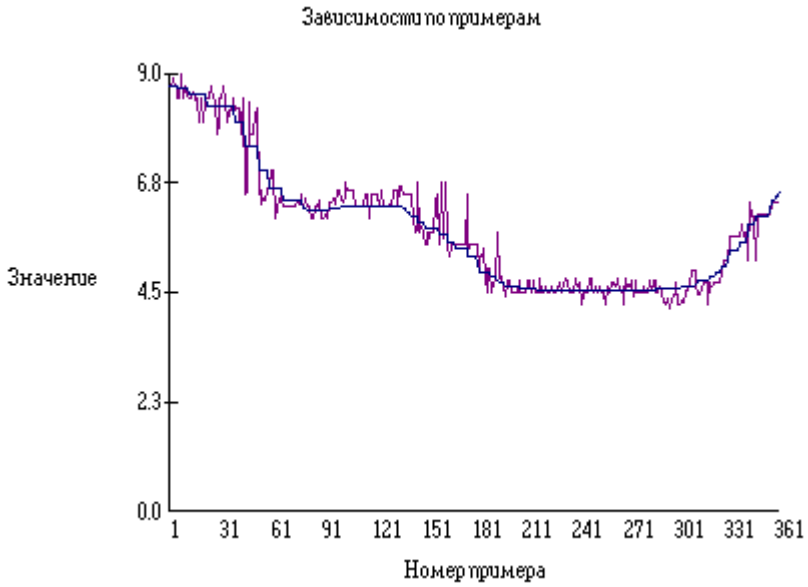
Вы можете изменить NeuroShell 2 так, чтобы он всегда вызывал вместо Таблицы Вашу электронную таблицу. Из главного меню NeuroShell 2 выберите Установки, Выбрать электронную таблицу. Появятся поля каталогов и списка файлов. Щелкните по .EXE-файлу, запускающему Вашу программу работы с электронными таблицами, и затем по кнопке OK.

Син также хочет применить сеть к .PRO-файлу. В модуле Применение к файлу он с помощью меню Файл выбирает другой файл данных, ELECTRIC.PRO. Затем он выбирает пункт Начать применение из меню Работа. Он замечает, что сеть дает на .PRO-файле значение R квадрата около 0,95.

Возможно, Син захочет оценить качество своей модели предсказания стоимости электричества каким-либо другим способом, не использующим R квадрат.

Он решает посмотреть на данные в графической форме. Он выходит из Системы для начинающего и возвращается в Главное меню NeuroShell 2, а затем выбирает Систему для

профессионала. В столбце Постобработка он выбирает значок Графики зависимостей. Он щелкает по значку График зависимостей по всем примерам. В поле списка Переменные/Столбцы он щелкает по названию Выход(1), а затем нажимает клавишу Ctrl и щелкает мышью по названию Ответ сети(1), чтобы выбрать оба столбца. Затем он щелкает по кнопке Построить. (График построен по .OUT-файлу, полученному при применении сети к .PAT-файлу. Прим. перев.)



Видно, что его модель на основе нейронной сети близко следует его реальным данным.

Предположим, он решил, что ответы недостаточно хороши. Что ему следует делать? Самая простое - это попробовать другие архитектуры сети из модуля Нейронные сети для профессионала (значок гоночного велосипеда).

С другими архитектурами он может получить несколько лучшие результаты, но на практике ему, вероятно, необходимо сделать что-то из того, что перечислено ниже в порядке убывания вероятности положительного эффекта:

1. **Используйте Калибровку**. NeuroShell 2 использует Калибровку для оптимизации сети путем применения текущей сети в процессе тренировки к независимому тестовому набору. (Вы можете создать тестовый набор данных автоматически, используя модуль **Выделение тестового набора** .) Калибровка позволяет найти оптимальную сеть для данных в тестовом наборе (что означает, что сеть способна хорошо обобщать и дает хорошие результаты на новых данных).

При использовании Калибровки это делается путем вычисления среднего квадратичного отклонения между реальными и предсказанными значениями для всех выходов по всем примерам. (Среднее квадратичное отклонение - это стандартный статистический показатель для определения близости подгонки.) При Калибровке вычисляется квадратичное отклонение для каждого выхода в данном примере, они суммируются, и затем вычисляется среднее значение этой величины по всем примерам в тестовом наборе.

Для сетей с обратным распространением ошибки сеть сохраняется всякий раз при достижении нового минимума средней ошибки (или среднего квадратичного отклонения). Чтобы использовать Калибровку, Вам необходимо установить интервал проверки Калибровки, т.е. как часто производится оценка на тестовом наборе. Мы предлагаем устанавливать его в диапазоне от 50 до 200. Вы также должны выбрать "Автоматическую

запись сети при наилучшем результате на тестовом наборе".

2. Включите в рассмотрение переменные, которые лучше предсказывают то, что Вы пытаетесь предсказать, и/или найдите лучшие способы представления переменных, которые у Вас уже есть. Син мог бы получить лучшие результаты, включив в рассмотрение переменную для количества людей в доме. Увеличение количества жильцов непосредственно влияет на потребление горячей воды для стирки, душа и т.д.

Если бы у него было большое количество входов, возможно, ему бы стоило подумать о преобразовании некоторых входов в отношения. Это дает больше информации при меньшем количестве переменных. Помните, что нейронные сети похожи на людей: чем проще Вы сделает входы, которые они должны выучить, тем легче сети выучить задание. Отношения служат этой цели.

3. Подойдите заново к вопросу о том, что именно Вы хотите предсказать. Для некоторых вещей это сделать проще, чем для других. Возможно, Вы получите более высокую точность, предсказывая процентное изменение стоимости электроэнергии, а не саму стоимость.

4. Соберите более качественный набор исторических данных или более представительный тестовый набор. Убедитесь в том, что Ваши переменные нормированы, если это необходимо. В примере с фондовым рынком это означает необходимость убедиться в том, что уровни, характерные для ситуации несколько лет назад, переведены в тот же диапазон, что и сегодняшние. В научной области нормировка может означать многие другие вещи. Вы можете обратиться за помощью к нам. Если Вы не будете нормировать данные, Вам придется предъявить сети гораздо большее количество примеров, с соответствующим увеличением времени обучения.

5. Попробуйте подобрать скорость обучения, момент и количество скрытых нейронов и посмотрите, не получатся ли сети более высокого качества. Попробуйте использовать TurboProp, который не требует установки скорости обучения и момента. Этот метод включен в модуль Проектирования в Системе для профессионала и работает для сетей с обратным распространением ошибки.

Замечание: Алгоритмы и методы, использованные в этом учебном примере, могли измениться со времени написания файла Справки. Обратитесь к разделу указателя файла Справки Изменения в программе, где отражены все последние изменения, которые могут включать другие способы предобработки данных или тренировки.

Данные, использованные в этой задаче, были созданы в методических целях и не были основаны на реальных счетах за электроэнергию. Имя Син ДюПри вымышленное и не относится ни к какому человеку, живому или умершему.