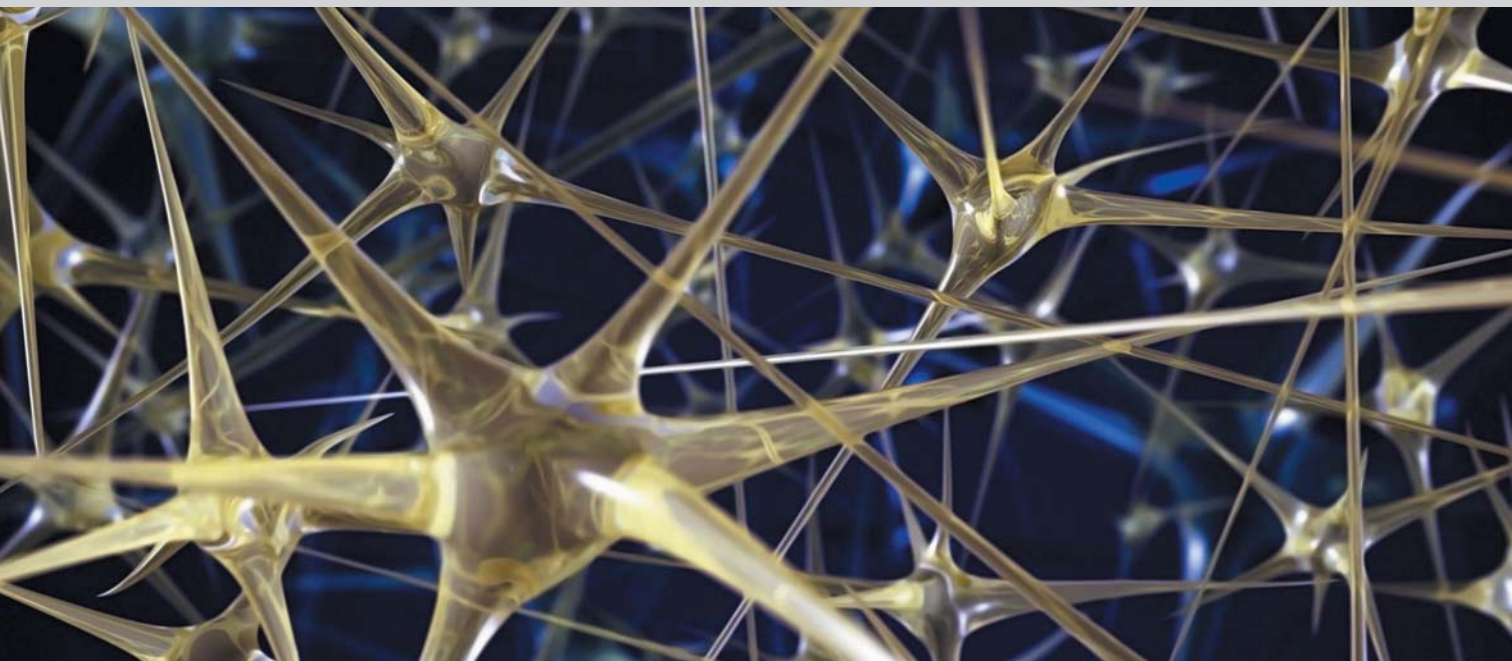


ДЖИММИ У. КИ (JIMMY W. KEY)

ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМИ ПРОЦЕССАМИ ЧАСТЬ 2

Нейронно-сетевые технологии могут решить наиболее насущные проблемы управления технологическими и производственными процессами. В данной части статьи уделено внимание подготовке набора данных для обучения, выполнению процесса обучения нейронной сети, созданию модели и выполнению необходимой валидации. Кроме того, в статье рассмотрено внедрение модели нейронной сети на платформе управления и человеко-машинного интерфейса. Смотрите также первую часть публикации, в которой были рассмотрены архитектура нейронной сети, вопросы управления пространством, модельный ряд, типы данных и выбор набора данных.



В первой части статьи (*CER* №3(63)'2016) мы остановились на завершении параметрического тестирования и сохранении полученных данных в исторической базе данных. Наша цель и следующий шаг направлены на то, чтобы получить записи набора данных для обучения и подтверждение достоверности (валидации) действия модели нейронной сети.

Интерпретация является первым этапом подготовки к обучению на основе набора данных. После того как запись набора данных выполнена, она должна быть соответствующим образом подготовлена для обучения модели нейронной сети. Запись набора данных должна быть случайным образом разобрана на три отдельных под-

множества (подкласса) записей: обучение, тестирование и валидацию (рис. 1).

Наборы данных для обучения и тестирования используются в процессе обучения модели. Для сходимости модели к целевой функции решения алгоритм обучения нейронной сети использует подмножество данных обучения. Подмножество

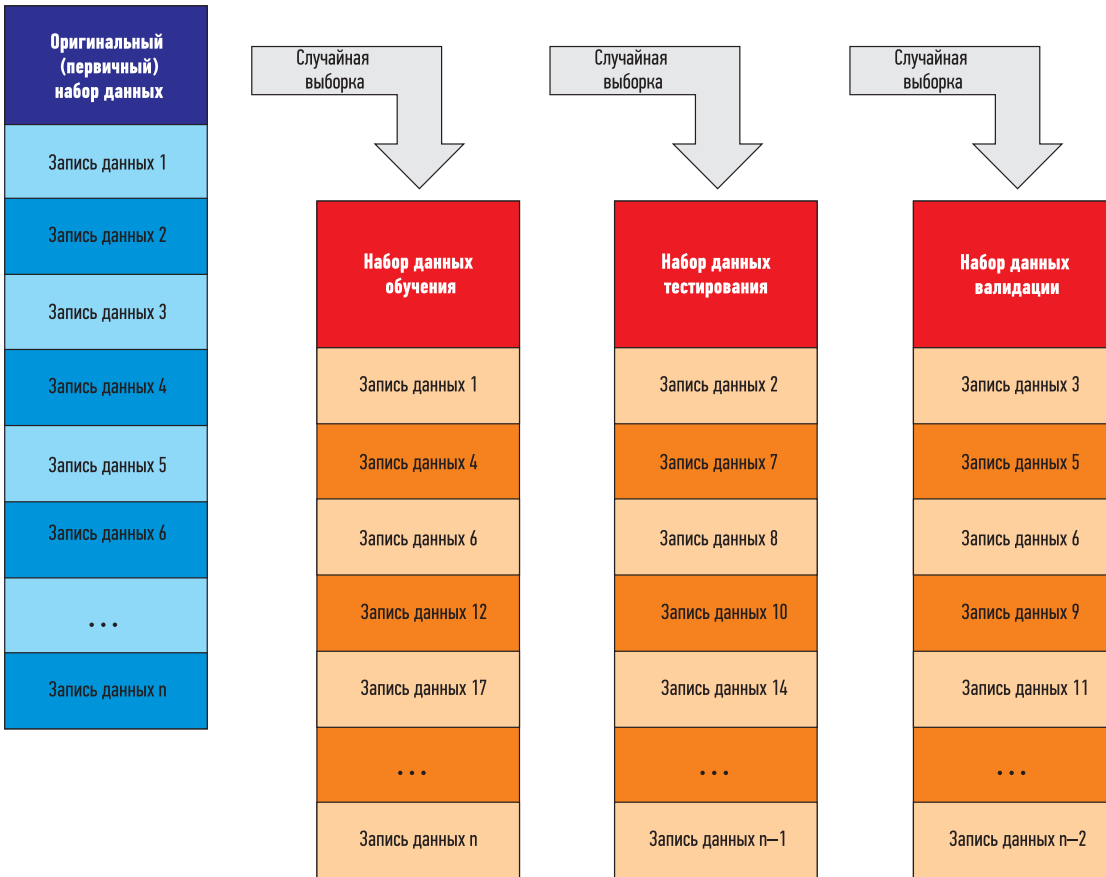


РИС. 1. Записи набора данных должны быть случайным образом разобраны на три отдельных подмножества — обучение, тестирование и валидация

данных тестирования используется для предотвращения ее так называемой «перетренированности» — излишне длительного обучения, что, в конечном итоге, может повлиять на надежность модели. После того как модель нейронной сети обучена, она должна пройти через жесткий процесс валидации, что даст уверенность пользователю в том, что она будет хорошо и надежно работать в условиях эксплуатации. Подмножество данных валидации доказывает, что модель может путем аппроксимации приблизиться к целевому значению с использованием входящих данных уже независимо от процесса обучения.

Подмножество, используемое для обучения, составляет от 60 до 80% объема всех имеющихся записей. Остальные записи, как правило, распределены между подмножествами данных тестирования и проверки. Каждая из записей должна быть выбрана случайным образом из набора данных и помещена в одно из трех подмножеств. (Экспериментальная проверка

показала более высокие результаты в том случае, когда имел место равный процент записей в каждом подмножестве, чем когда он был выше и ниже целевого базового уровня.) После того как запись была выбрана, она удаляется из исходного массива данных, и, тем самым, не может быть выбрана повторно.

ОБУЧЕНИЕ, ВАЛИДАЦИЯ, ОТЧЕТНОСТЬ

Теперь, когда подмножества созданы, мы должны выполнить следующие четыре шага, обязательные для каждой тренировочной итерации:

1. Инициализация.
2. Нормализация.
3. Обучение.
4. Проверка на перетренированность (создание ранней точки остановки обучения).

После обучения необходимо выполнить три последующих действия:

1. валидация.
2. Отчет об обучении.
3. Анализ обучения.

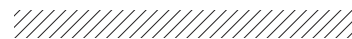
Инициализация

Перед обучением может иметь место ситуация, когда нейроны сети должны быть инициализированы случайным посевом их значений (называемых еще весами) и их случайным смещением (т. е. разбросом, отклонением). Первоначальный случайный посев считается в этом случае наилучшим вариантом, потому что целевая функция нам неизвестна.

Нормализация

Все переменные, входы и целевые опорные уровни должны быть соответствующим образом нормализованы еще до использования их в модели для целей обучения. Нормализация значений для каждого из входов должна быть проведена в пределах передаточной функции скрытого слоя. Выходные данные модели потом преобразуются обратно в инженерные единицы измерения с помощью снятия такой нормализации на выходе.

Если перцептрон (разъяснение данного термина дано в ч.1 статьи)



использует сигмоидальную функцию¹, то нормализация осуществляется в промежутке между «0» и «1». При использовании в качестве передаточной сигмоидной функции в виде гиперболического тангенса (tansig) нормализация осуществляется в промежутке между «-1» и «+1». Таким образом, диапазон инженерных установок каждой переменной процесса масштабируется в заданных пределах нормирования.

Напомним (см. ч. 1), персептрон, чтобы вычислить свой отклик, использует весовые коэффициенты и значения их смещения. Конечные значения генерируются посредством итеративного процесса обратного распространения. В процессе обучения значения показателей персептрона и их смещения корректируются так, чтобы свести к минимуму суммы квадратов ошибок (Sum Squared Error, SSE) между целевой функцией и моделью.

Алгоритмы обучения, которые используют оптимизационные подходы, позволяют найти наилучший путь к глобальным минимумам SSE. Тем не менее, время от времени в ходе обучения находятся локальные минимумы, и процесс «спотыкается». Это происходит из-за случайного высева значений и смещения основной функции, а также пути, пройденного алгоритмом оптимизации. Если встречается локальный минимум, модель покажет плохую корреляцию с целевой функцией, и учебная сессия должна быть повторена.

Ранняя точка остановки

Выбор более ранней точки остановки — это стратегия подготовки цикла выхода из обучения. Она предназначена для обеспечения оптимального сокращения SSE от обучающих данных, при этом модель остается достаточно надежной и устойчивой, а ее перетренированность предотвращается.

Регулирование

Регулирование — это метод забравки значений, которые становятся несоизмеримо большими во время обучения. Несмотря на то, что этот подход заслуживает внимания, использование раннего прекращения обучения, о котором говорилось выше, иногда бывает более предпочтительным в качестве средства

повышения надежности и устойчивости.

Валидация

Валидация (проверка на соответствие) является наиболее важной и заключительной частью процесса и является подтверждением того, что модель будет работать именно так, как это ожидалось. Подмножество проверки не зависит от цикла обучения и используется для генерации результатов проверки. После анализа находится в фазе (не имеет значительного запаздывания) и в тесном приближении к цели, ее можно считать успешной. Кривая проверки является последней частью отчета об обучении.

Отчет об обучении

Отчеты об обучении должны быть получены и проверены для всех моделей. Отчет должен включать в себя записи набора данных, архитектуру нейронной сети, финальный (окончательный) график обучения, а также данные по сокращению SSE на итерацию для обучения и тестовое подмножество для целей регулирования.

Анализ обучения

Модель и цель должны совпадать (находиться в фазе), иметь минимальную ошибку и характеризоваться высокой корреляцией. Кривые регрессии должны показать, насколько хорошим является совпадение между моделью и целью. Если модель находится вне фазы с целью или имеет большую погрешность, она должна быть переучена начиная с этапа введения случайного набора данных для обучения. Если имеет место значительное расхождение в результатах анализа модели, то это является основанием к увеличению количества нейронов в скрытом слое и/или пересмотру выбора переменной процесса, как это описано в первой части статьи (см. главу «Выбор массива данных»).

ИМПЛЕМЕНТАЦИЯ МОДЕЛИ НЕЙРОННОЙ СЕТИ

Соображения касательно внедрения нейронной сети в конкретном индустриальном применении включают в себя:

- конфигурацию нейронной сети (алгоритм);
- проверку качества входов;
- тестирование управления пространством;
- планирование альтернативной стратегии управления/измерения;
- соответствующий пользовательский интерфейс.

Алгоритм

Большинство платформ управления уже имеют набор команд для программирования алгоритма нейронной сети. Число нейронов в сети устанавливается в процессе обучения. Уравнение для одного нейрона в скрытом слое:

$$F_{sigmoid}(PxW_{1,n}^T + bias_{1,n}) = a_n, \quad (1)$$

где P — входной вектор.

Каждый из нейронов скрытого слоя создает выходной сигнал, который является элементом для «А» в уравнении выходного слоя:

$$F_{linear}(Ax_{w2}T + bias_2) = \text{Выход модели}. \quad (2)$$

Рассчитанный по уравнению (2) результат является моделью целевой переменной.

Проверка качества входов

Любая модель зависит от надежных и точных ее входов. Однако нейронные сети будут без проблем принимать и неточные, искаженные или смещенные входы, но до тех пор, пока они являются повторяющимися и модель обучалась с учетом этого. Все сказанное означает, что для установления известного базового уровня все входы должны быть соответствующим образом откалиброваны перед вводом обучающего набора данных. Во время работы все входы в модели нейронной сети для подтверждения их должного качества должны пройти валидацию, которая выполняется посредством системы управления.

Тестирование управления пространством

В рабочей модели испытание должно быть выполнено таким образом, чтобы доказать, что входы действительно соответствуют управлению пространством. Напомним (см. ч. 1, глава «Управление пространством»),

¹ Сигмоида — это гладкая монотонная нелинейная функция, имеющая форму буквы «S». Используется для сглаживания процессов. — Прим. перев.

что модель имеет лишь те знания, относящиеся к области управления пространством, где она проходила обучение. Эффективный метод тестирования — использование набора центроидов векторов точки (более известных как радиус-вектор, который описывает координаты точки в данной системе координат) с отклонениями. Центроиды и их отклонения генерируются исходя из распределения векторов точки.

Альтернативные стратегии управления/измерения

Как уже говорилось в первой части статьи, реальное пространство управления может привести к созданию недопустимой точки вектора внешнего управления пространством. Если входной вектор точки находится вне допустимого пространства управления, то должен быть введен заранее подготовленный план действий, направленный на смягчение воздействия от необученных результатов моделирования. Таким образом, здесь необходимо иметь некий альтернативный контроль или стратегию измерения. На рис. 2 представлена модель базовой стратегии контроллера, которая в случае возникновения ситуации, находящейся вне ее пространства управления, подает аварийные сигналы оператору, и система автоматически переключается на альтернативный алгоритм управления. Такое переключение также может быть оставлено на усмотрение оператора, если характеристики процесса

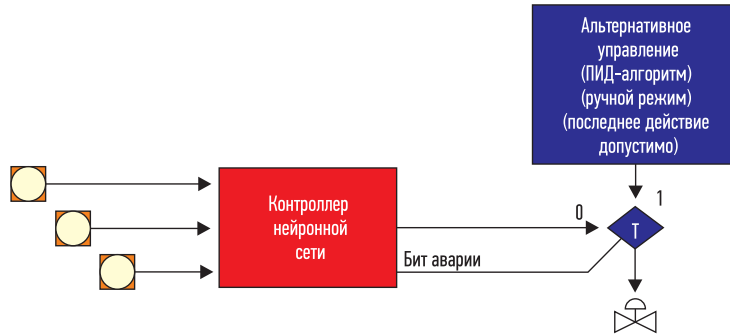


РИС. 2. Ручное или автоматическое переключение на альтернативную стратегию управления может потребоваться в том случае, если вводится управление пространством, не прошедшее обучение

являются, по его мнению, приемлемыми. Такая ситуация может иметь место потому, что имеются случайные флуктуации в части входного воздействия на модель, которые эффективно зажаты передаточной функцией скрытого слоя нейронной сети.

Интерфейс оператора

Рекомендуемый человеко-машинный интерфейс (Human-Machine Interface, HMI) должен иметь:

- заданную величину выходного сигнала модели;
- общую индикацию тревоги пространства управления;
- индивидуальный, с обслуживанием в порядке поступления, выход сигнала тревоги для каждого входа;
- переход на ручной режим управления;
- выбор альтернативной стратегии для выходного значения.

С помощью виртуального интерфейса измерительной системы

сигнал тревоги, представленный на лицевой панели HMI (рис. 3а), показывает, какой из входов вызывает беспокойственность. Передача управления доступна в случае, если оператор выбирает альтернативную стратегию. Ручной режим доступен для удержания последнего значения. MIMO-контроллер (Multiple Input/Multiple Output — «многоканальный вход/многоканальный выход») со множеством входов, приведенный на рис. 3б, является аналогичным вариантом, но он показывает тенденции управления и регулируемых переменных. Для ввода значений ограничений и других параметров сигнализации рекомендуется использовать отдельный экран настройки.

ИСПОЛЬЗОВАНИЕ НЕЙРОННЫХ СЕТЕЙ

Во многих приложениях, связанных с управлением технологически

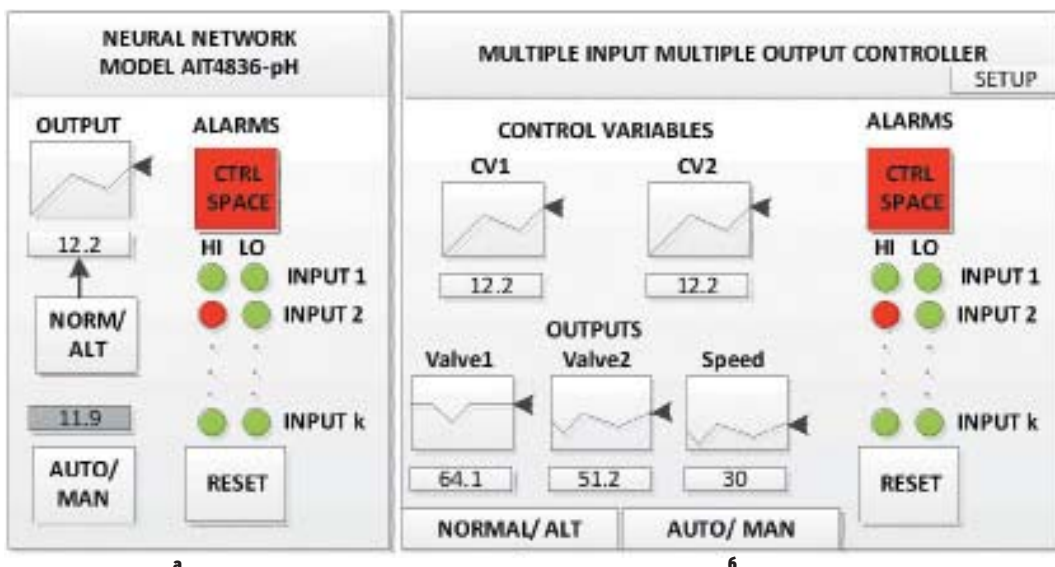
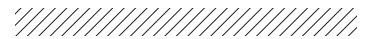


РИС. 3. Окно HMI для виртуальной измерительной системы (а) и многоканальный MIMO-контроллер (б)



ми производственными процессами, функционально-приближенная архитектура сети достаточно проста, и она хорошо справляется с моделированием производственных процессов. Нейронные сети должны быть обучены исходя из наборов данных, полученных с помощью самого лучшего из возможных вариантов параметрического тестирования. Чтобы смоделировать управление процессом должным образом, пользователь выбирает переменную процесса и формирует набор данных для обучения. Этот набор должен покрыть весь возможный диапазон обучения и соответствовать его функциональному назначению. Набор данных содержит данные, связанные со входами, и коррелирует их к целевому назначению. Входы, используемые в обучении, определяют пространство управления нейронной сетевой модели.

Все записи набора данных выполняются с временными метками, определяющими их целевой отсчет по времени захвата и значение параметра входа. Наборы данных должны быть привязаны к конфигурации оборудования, которая имела место в момент приобретения системы. Для повыше-

ния надежности модели, устранения помех и снижения стоимости критически важное значение имеет правильный выбор модели входов. При этом калибровка должна быть завершена для всех входов еще до получения данных. Это необходимо для того, чтобы установить их базовый опорный уровень.

После того как набор данных был получен, он должен быть разобран на три подмножества: обучение, испытание и валидацию. Обучение нейронной сети предполагает обратное распространение ошибки между выходом модели и целевым значением. Обучение представляет собой итеративный процесс обратного распространения, который требует стратегии, делающей упор на надежность и подтверждение достоверности. Значения (веса) и их смещения для каждого нейрона предоставляются в процессе обучения. Отчет о валидации должен быть рассмотрен еще до внедрения системы на платформе управления. Этот отчет должен показать совпадение целевой функции управления в фазе с моделью при минимальной погрешности.

Большинство программ обучения нейронной сети работа-

ют на настольных компьютерах, а основные платформы управления уже имеют весь необходимый набор команд для настройки модели нейронной сети. Там, где это возможно, кодирование нейронов на платформе управления должно быть выполнено в подпрограммах, позволяющих повторное использование кода на будущих моделях. В работающих моделях все входы должны быть проверены на соответствующие качественные показатели и допустимое пространство управления. Если входы находятся вне допустимого пространства управления, то должна предусматриваться альтернативная стратегия управления.

НМІ должен отображать выход модели, входы, сигналы тревоги, технологические уставки процесса, предоставлять возможность выбора режима (автоматический/ручной) и стратегии управления (обычная/альтернативная). Тенденции и логика с обслуживанием в порядке поступления являются весьма важным решением для оперативного информирования и диагностики. Для введения необходимых констант, сигнализации и ограничений рекомендуется использовать отдельный экран настройки. ●