

МАШИННАЯ КЛАССИФИКАЦИЯ СИГНАЛОВ ЧИСЛОВОГО КОДА В ЭЛЕКТРОТЕХНИЧЕСКИХ СИСТЕМАХ ЛОКОМОТИВНОЙ СИГНАЛИЗАЦИИ

Системы автоматической локомотивной сигнализации, применяемые на железных дорогах России, используют электрическую рельсовую цепь в качестве среды для передачи сигналов на локомотив. Сигналы в такой электротехнической системе в значительной степени подвержены влиянию помех, создаваемых тяговым током и различными внешними факторами, что приводит к сбоям в работе существующих систем и нарушению движения поездов. Для повышения устойчивости работы локомотивных систем, принимающих сигналы из электрических рельсовых цепей, предложено применить алгоритм машинной классификации на основе нейронной сети.

В статье представлены основные стадии данного алгоритма и рассмотрена архитектура нейронной сети для классификации принятого электрического сигнала. Также продемонстрирована работа предложенного алгоритма при приеме из рельсовой цепи электрических сигналов с помехами, вызывающими сбои в существующих системах автоматической локомотивной сигнализации.

Ключевые слова: автоматическая локомотивная сигнализация, числовой код, машинное обучение, нейронная сеть, железнодорожная автоматика, конечный автомат.

Введение. На железнодорожном транспорте в системах интервального регулирования движения поездов широко применяется принцип передачи информации о параметрах движения поезда (требуемый скоростной режим, уклон впередилежащего блок-участка, расстояние до следующей точки остановки и др. [1, 2]) на борт локомотива по рельсовым нитям. Данный принцип позволяет своевременно информировать машиниста о текущей поездной ситуации, а также формировать рекомендации по ведению поезда на впередилежащем участке.

Так, например, информация о требуемом скоростном режиме ведения поезда передается на борт локомотива в виде специальных электрических сигналов, амплитудно модулированных числовым кодом. В результате дешифрации таких сигналов на борту локомотива при помощи системы индикации в кабине машинист проводит соответствующие мероприятия по регулированию скорости движения поезда согласно установленным требованиям [3].

В качестве примера на рис. 1 показана осциллограмма электрического сигнала числового кода (далее — кодового сигнала), зафиксированного на борту локомотива до дешифрации. Кодовые сигналы подобного вида являются циклическими.

Существенной проблемой при расшифровке кодовых сигналов является возникновение регулярных сбоев дешифрирующей аппаратуры [4–6]. Данные сбои, как правило, вызваны помехами различной природы [7]. Помехи, воздействуя на кодовый сигнал в процессе его распространения

по рельсовым нитям, влекут за собой нарушение нормативных параметров кодового сигнала (длительностей его импульсных и интервальных частей [8]).

Как следствие, машинист, не владея информацией о параметрах движения поезда, вынужден в целях обеспечения безопасности движения снижать скорость и неэффективно использовать тяговые ресурсы, что напрямую влияет как на себестоимость перевозок, так и на выполнение графика движения.

Известны различные варианты решения данной проблемы, которые представлены, например, в работах [5, 6, 9, 10]. Однако в настоящее время перспективным представляется подход, который подразумевает дополнение дешифрирующей аппаратуры локомотива интеллектуальными функциями [11], [12]. Под интеллектуальными функциями подразумеваются такие алгоритмы принятия решения устройством, которые имитируют принятие решения человеком [13]. Тогда имеющие искажения кодовые сигналы при условии, что их осциллограммы могут быть однозначно расшифрованы человеком, не будут приводить к сбоям дешифрирующей аппаратуры. Это, в свою очередь, позволит значительно повысить безопасность и эффективность перевозочного процесса.

Дешифрация кодовых сигналов, передаваемых на борт локомотива из рельсовых нитей, может быть реализована на основе интеллектуальных алгоритмов распознавания изображений. Такие алгоритмы имеют достаточно широкое практическое

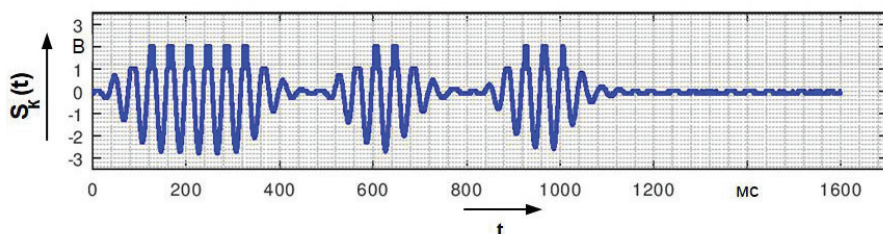


Рис. 1. Оциллограмма электрического сигнала числового кода

применение в различного рода задачах: автоматическое распознавание рукописных цифр почтового индекса на конвертах писем, человеческих лиц на фотографиях, знаков дорожного движения при движении беспилотных автомобилей и др. [14–19] Разработка подобных систем распознавания является одной из ключевых задач машинного обучения [14], [20]. Методологически, задачи распознавания изображений в практике машинного обучения наилучшим образом разрешимы при помощи нейронных сетей [16, 21, 22].

Таким образом, задачу классификации кодовых сигналов на борту локомотива можно формализовать в виде задачи классификации изображений, а решить — при помощи нейронной сети.

Выбор архитектуры нейронной сети. Для рассматриваемой задачи классификации кодовых сигналов на борту локомотива определим архитектуру нейронной сети. Выберем трехслойную нейросеть, в которой последовательно располагаются три уровня обработки входных данных — входной, скрытый и выходной слои (рис. 2).

Число нейронов во входном слое ($L = 1$) соответствует числу отсчетов в одном цикле кодового сигнала. В качестве примера выберем данное значение равным 1600, что соответствует длительности одного цикла кодового сигнала, равной 1,6 с, и частоте дискретизации, равной 1000 Гц. Число нейронов в выходном слое ($L = 3$) принято равным 3, поскольку в системах интервального регулирования движения поездов преимущественно применяются три типа кодовых сигналов («З», «Ж» и «КЖ») [1], дешифрацию которых необходимо реализовать на борту локомотива. Количество нейронов скрытого слоя ($L = 2$) выбрано равным 10, что обусловлено рекомендациями, сформированными по результатам решения задач аналогичной сложности, например, в [15, 18, 19].

Для удобства программной реализации расчета нейронной сети входной и скрытый слои дополнены единичными базовыми нейронами (показаны окружностями с единицей внутри).

Весовые коэффициенты нейронной сети, ассоциированные с каждой линией перехода между нейронами, обозначены как матрицы $W^{[1]}$ и $W^{[2]}$. Согласно рис. 2 и [14], матрица $W^{[1]}$ содержит 10 строк и 1601 столбец, а матрица $W^{[2]}$ — 3 строки и 11 столбцов.

Система машинной классификации кодовых сигналов. Для дешифрации кодовых сигналов на борту локомотива предлагается построение системы классификации, структурная схема которой показана на рис. 3. АЦП предназначен для непрерывного преобразования кодового сигнала, принимаемого на локомотиве из рельсовой линии (рис. 1), в цифровые отсчеты с заданной частотой дискретизации (1000 Гц). Распознавание кодового сигнала ведется по совокупности отсчетов, число которых

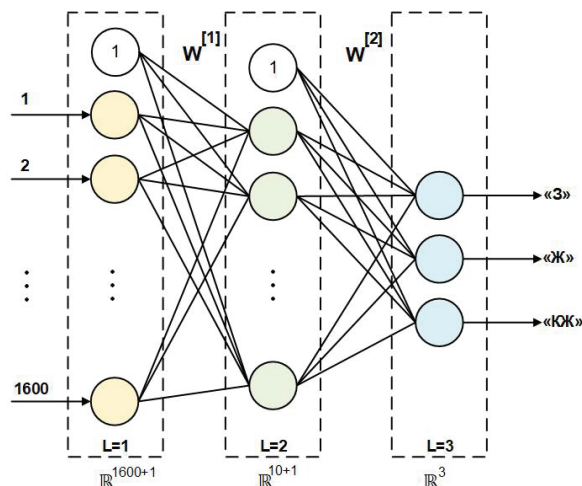


Рис. 2. Архитектура нейронной сети для классификации кодовых сигналов

соответствует длительности одного кодового цикла. Для накопления требуемого числа отсчетов перед их дальнейшей обработкой служит блок буферизации данных (ББД). Сигнал числового кода, который подлежит распознаванию, выделяется из принятого кодового сигнала путем амплитудной демодуляции в блоке амплитудной демодуляции (БАДМ). Блок нормализации (БН) выполняет предварительную обработку сигнала числового кода, выделенного в БАДМ путем приведения значений его отсчетов к интервалу $[0, 1]$ (рис. 4).

При распознавании нормализованного кодового сигнала нейронной сетью, реализуемой блоком распознавания (БР), требуется, чтобы отсчеты сигнала, соответствующие импульсам и интервалам, располагались систематическим образом в пределах кодового цикла, то есть кодовый сигнал, подлежащий распознаванию, должен быть синхронизирован относительно начала кодового цикла. Эта задача решается блоком синхронизации (БС). Устройство управления (УУ) предназначено для организации взаимодействия между компонентами системы между собой и формирования результата распознавания.

Рассмотрим более детально некоторые процессы преобразования кодового сигнала в представленной системе.

Амплитудная демодуляция кодового сигнала. Информационной составляющей передаваемого на локомотив кодового сигнала является цифровой код. Таким образом, классификация кодовых сигналов на борту локомотива в конечном итоге сводится к классификации сигналов числового кода. Выделение сигнала числового кода из оцифрованного кодового (рис. 1) возможно за счет амплитудной демодуляции последнего.

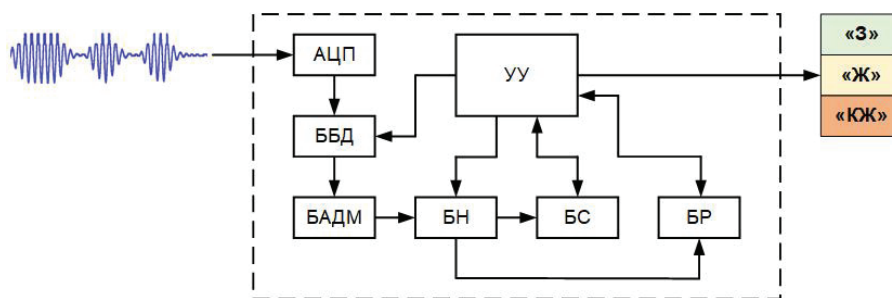


Рис. 3. Структурная схема системы классификации кодовых сигналов (АЦП — аналого-цифровой преобразователь; ББД — блок буферизации данных; БАДМ — блок амплитудной демодуляции; БН — блок нормализации; БС — блок синхронизации; БР — блок распознавания; УУ — устройство управления)

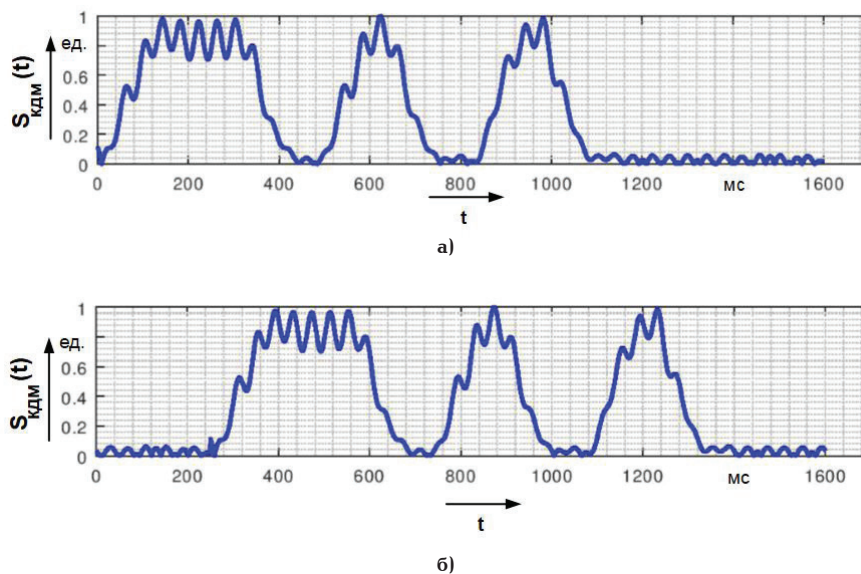


Рис. 4. Кодовый сигнал после:
а) — амплитудной демодуляции и нормализации;
б) — смещения относительно стартового отсчета

В качестве алгоритма, реализующего амплитудную демодуляцию кодового сигнала, удобно использовать синхронное детектирование [23]. Данный принцип демодуляции заключается в умножении исходного сигнала на опорный, в результате чего образуется сумма низкочастотной и высокочастотной составляющих (1). Далее проводится фильтрация результирующего сигнала согласно формулам (2), (3), в результате которой удаляется высокочастотная составляющая, а из исходного сигнала выделяется низкочастотный сигнал числового кода.

$$S_{cg}(t) = S_k(t) \cdot \cos(2\pi F_N t) = \frac{1}{2} S_m(t) + \frac{1}{2} S_m(t) \cdot \cos(4\pi F_N t), \quad (1)$$

$$[b, a] = \text{butter}(5, 2 \cdot F_N / F_A), \quad (2)$$

$$S_{kgm}(t) = \text{filtfilt}(b, a, S_{cg}(t)), \quad (3)$$

где $S_k(t)$ — кодовый сигнал, принятый на локомотиве из рельсовой линии и подлежащий амплитудной демодуляции; F_N — частота несущего сигнала в $S_k(t)$; F_A — частота дискретизации; $S_{cg}(t)$ — результирующий сигнал, образующийся в результате умножения $S_k(t)$ на опорный сигнал $\cos(2\pi F_N t)$; $\text{butter}(5, 2$

$F_N / F_A)$ — функция, реализующая расчет параметров b и a цифрового фильтра Баттерворта нижних частот 5-го порядка (синтаксис языка программирования Octave); $\text{filtfilt}(b, a, S_{cg}(t))$ — функция, имитирующая воздействие цифрового фильтра с параметрами b и a на сигнал $S_{cg}(t)$ (синтаксис языка программирования Octave); $S_{kgm}(t)$ — амплитудно-демодулированный кодовый сигнал.

Отметим, что двукратное уменьшение амплитуды сигнала $S_{kgm}(t)$ по отношению к $S_k(t)$ будет учтено на этапе нормализации.

Нормализация значений отсчетов кодового сигнала. Следующим этапом после амплитудной демодуляции кодового сигнала является нормализация значений его отсчетов. Целью данной процедуры является повышение скорости настройки блока распознавания за счет уменьшения времени расчета его параметров. Это, в свою очередь, позволит повысить эффективность использования вычислительных ресурсов системы классификации.

Нормализация значений отсчетов кодового сигнала может быть выполнена по формуле (4):

$$x_i^{\text{норм}} = \frac{x_i - \min(S_{kgm}(t))}{\max(S_{kgm}(t)) - \min(S_{kgm}(t))}, \quad (4)$$

где i — число отсчетов в $S_{кдм}(t)$ (равно 1600); $x_i^{норм}$ — нормализованное значение отсчета в $S_{кдм}(t)$, $\min(S_{кдм}(t))$, $\max(S_{кдм}(t))$ — минимальное и максимальное значения отсчетов в $S_{кдм}(t)$ соответственно.

Сигнал, полученный в результате амплитудной демодуляции и нормализации принятого на локомотиве сигнала, показан на рис. 4а.

Поиск стартового отсчета кодового сигнала. Для того чтобы обеспечить корректную работу нейронной сети при классификации очередного кодового сигнала, необходимо осуществлять поиск отсчета, соответствующего началу цикла кода — стартового отсчета. Примем, что такой отсчет находится в середине длинного интервала цикла кодового сигнала. Тогда один цикл сигнала числового кода, который подлежит распознаванию, будет иметь вид, показанный на рис. 4б.

Алгоритм поиска стартового отсчета внутри цикла кодового сигнала предлагается реализовать на основе детерминированного временного конечного автомата [24].

Детерминированный временной конечный автомат представляет собой формальное описание переходов системы между состояниями под воздействием разрешенных сигналов. Термин «временной» указывает, что очередной переход системы между состояниями сопровождается соответствующим изменением переменных времени — таймеров.

Математическое описание указанного конечного автомата представляется кортежем $\langle \Sigma_c, S_c, S_0, C_c, E_c, F_c \rangle$, в котором:

- Σ_c — конечный входной алфавит символов автомата;
- S_c — конечное множество внутренних состояний автомата;
- S_0 — начальное состояние автомата, являющееся элементом множества S_c ;
- C_c — конечное множество таймеров автомата;
- E_c — конечное множество переходов автомата между состояниями S_c ;
- F_c — конечное множество допустимых (конечных) состояний автомата.

Определим формальное описание конечного автомата, используемого для поиска стартового отсчета кодового сигнала.

Конечный входной алфавит автомата Σ_c содержит два символа — $\{0,1\}$. Символ «0» соответствует значению анализируемого сигнала ниже порогового, а символ «1» — выше. Отсчеты сигнала, значения которых ниже порогового, интерпретируются как принадлежащие интервальной части сигнала, остальные отсчеты считаются принадлежащими импульсной части. В данном исследовании пороговое значение при обработке нормализованного кодового сигнала выбрано равным 0,3. Результат такой

обработки для кодового сигнала, изображенного на рис. 4, представлен на рис. 5. На рис. 5 пороговое значение показано штриховой линией.

Конечное множество внутренних состояний автомата S_c содержит четыре элемента — $\{S_0, S_1, S_2, S_3\}$. Состояние S_0 является начальным, то есть таким, с которого автомат начинает свою работу при очередном запуске или в которое он возвращается в случае нарушения условий успешного поиска стартового отсчета. В состоянии S_1 автоматом производится проверка корректной длительности искомой интервальной части кодового сигнала, в S_2 — следующей за ней импульсной. В рассматриваемом примере длительности указанных частей кодового сигнала выбраны равными 500 и 100 отсчетам соответственно. Таким образом, стартовый отсчет будет обнаружен в случае, если на протяжении 500 отсчетов анализируемого сигнала их значения не превышают пороговое, после чего на протяжении 100 отсчетов — превышают. Состояние S_3 является конечным и достигается системой в случае успешного поиска стартового отсчета.

Конечное множество таймеров автомата C_c содержит только один элемент «count», являющийся переменной времени, которое выражается числом отсчетов обработанных автоматом. Значение переменной «count» изменяется в соответствии с условиями, проверяющимися при очередном переходе автомата между состояниями. Таким образом, $C_c = \{\text{count}\}$.

Конечное множество E_c задается множеством функций переходов автомата $\delta_1 - \delta_8$, которые представлены следующими соотношениями (5):

$$\begin{aligned} \delta_1(S_0, 0) &= S_1, \\ \delta_2(S_1, 0, \text{count} < 1600) &= S_1, \\ \delta_3(S_1, 0, \text{count} = 1600) &= S_0, \\ \delta_4(S_1, 1, \text{count} < 500) &= S_0, \\ \delta_5(S_1, 1, \text{count} = 500) &= S_2, \\ \delta_6(S_2, 1, \text{count} < 100) &= S_2, \\ \delta_7(S_2, 1, \text{count} = 100) &= S_3, \\ \delta_8(S_2, 0) &= S_0. \end{aligned} \quad (5)$$

Конечное множество допустимых (конечных) состояний автомата является подмножеством S_c и содержит один элемент — состояние S_3 . Таким образом, $F_c = \{S_3\}$. Граф переходов конечного автомата, соответствующего приведенному описанию, показан на рис. 6.

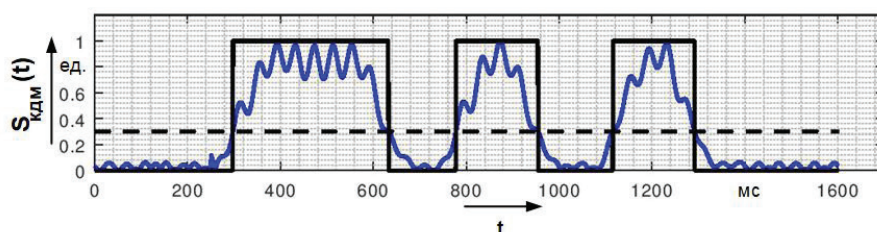


Рис. 5. Результат сравнения значений отсчетов кодового сигнала с пороговым

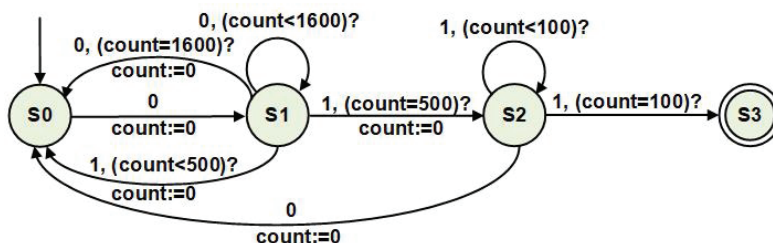


Рис. 6. Граф переходов конечного автомата для поиска стартового отсчета кодового сигнала

```
for v = 1:length(y)
    y_vec(y(v),v) = 1;
endfor
```

Лист. 1. Программный код предварительной обработки y_{train} (y — исходный вектор ответов y_{train} , y_{vec} — вектор ответов y_{train} после преобразования)

Распознавание кодового сигнала на основе нейронной сети. Ключевым этапом работы системы классификации кодовых сигналов является распознавание очередного цикла сигнала на основе нейронной сети, архитектура которой представлена на рис. 2. Для этого необходимо предварительно определить значения параметров $W^{[1]}$ и $W^{[2]}$. Очевидно, что значения данных параметров должны быть такими, при которых точность работы нейронной сети максимальна, то есть быть оптимальными. Процесс подбора оптимальных параметров классификатора в практике машинного обучения получил название «обучение» [15, 16, 20].

Для обучения нейронной сети необходима специальная выборка данных, которая, согласно своему назначению, получила название «обучающая». Обучающая выборка представляет собой двумерный массив данных X_{train} , дополненный вектором ответов y_{train} . Каждая строка массива X_{train} является одним обучающим примером, то есть содержит отсчеты, образующие один цикл сигнала числового кода, пример которого показан на рис. 6. Поскольку каждый обучающий пример содержит 1600 отсчетов кодового сигнала, то данное значение определяет параметр n — размерность пространства, в котором предполагается выполнять классификацию. Таким образом, n — это число признаков классификации, и равно 1600.

Вектор ответов y_{train} содержит целочисленные значения «1», «2» и «3» для каждого обучающего примера массива X_{train} . Значение «1» соответствует кодовому сигналу типа «3», «2» — «Ж», 3 — «КЖ».

Для обучения нейронной сети предварительно была сформирована обучающая выборка X_{train} размерности $m \times n$ и y_{train} размерности m , где $m = 96000$ — число обучающих примеров, $n = 1600$ — количество признаков классификации.

Важной особенностью процедуры обучения нейронной сети, используемой в задачах классификации данных с числом классов ≥ 3 , является предварительная обработка вектора y_{train} . Данная обработка заключается в преобразовании каждого целочисленного значения ответа для обучающего примера в новый вектор, размерность которого совпадает с числом выходов нейронной сети. Тот элемент вектора, который располагается на позиции, номер которой равен правильному ответу, обраба-

$$W = \text{rand}(L_{\text{out}}, L_{\text{in}} + 1) * 2 * 0.12 - 0.12;$$

Лист. 2. Программный код инициализации параметров нейронной сети (L_{out} — количество нейронов в слое L ; L_{in} — количество нейронов в слое $L-1$)

ется в единицу, а оставшиеся элементы вектора — в нули. В рассматриваемом случае классификации кодовых сигналов, результат предварительной обработки одного правильного ответа из y_{train} для каждого типа будет иметь вид (6):

Ответ — «1» Ответ — «2» Ответ — «3»

$$y = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad y = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad y = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad (6)$$

Программный код, реализующий данную обработку вектора ответов y_{train} для синтаксиса языка программирования Octave, приведен в лист. 1.

Таким образом, вектор ответов y_{train} преобразуется в массив данных размерности $k \times m$, где k (число классов данных) равно 3. Для упрощения применяемой математической нотации обозначение вектора ответов y_{train} после преобразования оставим без изменений.

Согласно [14, 16, 25], наиболее эффективно обучение нейронной сети может быть реализовано с применением алгоритма обратного распространения. Данный алгоритм позволяет реализовать поиск оптимальных параметров нейронной сети на основе минимизации функции ошибки методом градиентного спуска. Охарактеризуем особенности построения алгоритма обратного распространения применительно к рассматриваемой задаче.

Подготовительной стадией для применения алгоритма обратного распространения является инициализация параметров нейронной сети, то есть $W^{[1]}$ и $W^{[2]}$. Установлено, что в качестве начальных параметров нейронной сети должны быть присвоены случайные значения, принадлежащие некоторому диапазону $[-\epsilon, \epsilon]$. Величина ϵ выбирается индивидуально для каждого массива параметров нейронной сети, ассоциированного с соответствующим переходом между слоями [14].

Однако в целях упрощения вычислительных операций на данном этапе величина ϵ принята единой для $W^{[1]}$ и $W^{[2]}$ и составляет 0,12 [15]. Программный код инициализации параметров $W^{[1]}$ и $W^{[2]}$ для синтаксиса языка программирования Octave приведен в лист. 2.

Функция ошибки, минимизация которой является ключевой задачей алгоритма обратного распространения, отражает степень неточности работы нейронной сети при определении класса, к которому принадлежат данные на ее входе. Для рассматриваемой архитектуры нейронной сети (рис. 2) функция ошибки $J(\mathbf{W}^{[1]}, \mathbf{W}^{[2]})$ имеет вид (7) [14]:

$$J(\mathbf{W}^{[1]}, \mathbf{W}^{[2]}) = \frac{1}{m} \left[-\mathbf{y}_{\text{train}} \cdot \ln h(\mathbf{X}_{\text{train}}, \mathbf{W}^{[1]}, \mathbf{W}^{[2]}) - (1 - \mathbf{y}_{\text{train}}) \cdot \ln(1 - h(\mathbf{X}_{\text{train}}, \mathbf{W}^{[1]}, \mathbf{W}^{[2]})) \right] + \frac{\lambda}{2m} \left[\sum_{i=1}^{10} \sum_{j=1}^{1600} (\mathbf{w}_{i,j}^{[1]})^2 + \sum_{i=1}^3 \sum_{j=1}^{10} (\mathbf{w}_{i,j}^{[2]})^2 \right], \quad (7)$$

где $h(\mathbf{X}_{\text{train}}, \mathbf{W}^{[1]}, \mathbf{W}^{[2]})$ — функция, описывающая разделяющую поверхность в пространстве признаков классификации; λ — параметр регуляризации (в данном случае выбран равным 1), предназначенный для контроля обобщающей способности классификатора [14].

Минимизации функции ошибки в задачах обучения нейронных сетей выполняется методом градиентного спуска [14–16, 26]. При этом необходимым условием для его применения является расчет градиента минимизируемой функции. Под градиентом функции ошибки (7) подразумевается некоторый вектор **grad**, элементы которого являются ее частными производными по каждому параметру из матриц $\mathbf{W}^{[1]}$ и $\mathbf{W}^{[2]}$. Таким образом, размерность вектора **grad** определяется как $10 \times 1601 + 3 \times 11$ и равна 16043.

Очевидно, что расчет градиента функции (7) в явном виде весьма затруднителен. Алгоритм обратного распространения позволяет существенно упростить решение данной задачи. Расчет градиента функции ошибки на основе алгоритма обратного распространения состоит из двух этапов: расчет функций активации нейронов и расчет частных производных, ассоциированных с каждой связью между нейронами сети.

Расчет функций активаций нейронов осуществляется в соответствии с концепцией прямого распространения данных через нейронную сеть, то есть от входного слоя к выходному. Функции активации нейронов, расположенных в слоях $L \geq 2$, могут быть рассчитаны по формуле (8):

$$\mathbf{a}^{[L]} = g(\mathbf{z}^{[L]}), \quad (8)$$

$$\mathbf{z}^{[L]} = \mathbf{W}^{[L-1]} \cdot \mathbf{a}^{[L-1]}, \quad (9)$$

где $\mathbf{z}^{[L]}$ — вектор, элементы которого представляют собой «взвешенные» значения выходов предшествующего слоя (9); $g(\cdot)$ — нелинейное преобразование элементов вектора $\mathbf{z}^{[L]}$.

Функции активации нейронов, располагающихся во входном слое ($L = 1$), принимаются равными значениям признаков классификации обучающих примеров, то есть (10):

$$\mathbf{a}^{[1]} = \mathbf{X}_{\text{train}}. \quad (10)$$

В качестве функции активации для нейронов скрытого и выходного слоев выбрано сигмоидальное преобразование (11) [14]:

$$g(z) = \frac{1}{1 + e^{-z}}. \quad (11)$$

Таким образом, итоговый алгоритм расчета функций активаций для нейронной сети выбранной архитектуры (рис. 2) представляет собой последовательный расчет формул (12):

$$\mathbf{a}^{[1]} = \mathbf{X}_{\text{train}},$$

$$\mathbf{z}^{[2]} = \mathbf{W}^{[1]} \cdot \mathbf{a}^{[1]},$$

$$\mathbf{a}^{[2]} = g(\mathbf{z}^{[2]}), \quad (12)$$

$$\mathbf{z}^{[3]} = \mathbf{W}^{[2]} \cdot \mathbf{a}^{[2]},$$

$$\mathbf{a}^{[3]} = g(\mathbf{z}^{[3]}),$$

при этом $h(\mathbf{X}_{\text{train}}) = \mathbf{a}^{[3]}$.

Расчет частных производных, ассоциированных с каждой линией связи между нейронами, осуществляется от выходного слоя сети к входному. Интерпретировать данный этап вычислений можно как определение степени причастности отдельного нейрона к итоговой ошибке сети при классификации данных [26].

Для выходного слоя нейронной сети ($L = 3$) искомые частные производные определяются по формуле (13), а для скрытого — (14):

$$\delta^{[3]} = \mathbf{a}^{[3]} - \mathbf{y}_{\text{train}}, \quad (13)$$

$$\delta^{[L]} = \mathbf{W}^{[L]T} \cdot \delta^{[L+1]} \cdot g'(\mathbf{z}^{[L]}), \quad (14)$$

где $g'(\mathbf{z}^{[L]})$ — первая производная для $g(\mathbf{z}^{[L]})$; T — символ транспонирования.

Тогда в рассматриваемом случае для единственного скрытого слоя ($L=2$) частные производные определяются согласно (15):

$$\delta^{[2]} = \mathbf{W}^{[2]T} \cdot \delta^{[3]} \cdot g'(\mathbf{z}^{[2]}). \quad (15)$$

Полученные значения $\delta^{[2]}$ и $\delta^{[3]}$ представляют собой накопленные по всем обучающим примерам суммы частных производных. Следовательно, для расчета результирующего градиента их, во-первых, необходимо усреднить по всем обучающим примерам (16), а во-вторых, — конкатенировать (17):

$$\mathbf{D}^{[i]} = \frac{\delta^{[i]}}{m},$$

$$\mathbf{D}^{[2]} = \frac{\delta^{[3]}}{m}, \quad (16)$$

$$\mathbf{grad} = \begin{bmatrix} \mathbf{D}^{[1]} \\ \mathbf{D}^{[2]} \end{bmatrix}, \quad (17)$$

где $\mathbf{D}^{[1]}$, $\mathbf{D}^{[2]}$ — усредненные по m $\delta^{[2]}$ и $\delta^{[3]}$ соответственно.

$\mathbf{D1} = [(1/m) \cdot \mathbf{d1}(:,1) \quad (1/m) \cdot \mathbf{d1}(:,2:\text{end})] + (\text{lambda}/m) \cdot \mathbf{W1}(:,2:\text{end});$

$\mathbf{D2} = [(1/m) \cdot \mathbf{d2}(:,1) \quad (1/m) \cdot \mathbf{d2}(:,2:\text{end})] + (\text{lambda}/m) \cdot \mathbf{W2}(:,2:\text{end});$

Лист. 3. Программный код регуляризации частных производных нейронной сети
(D1, D2 — D^[1], D^[2] соответственно,
d1, d2 — δ^[1], δ^[2] соответственно,
W1, W2 — W^[1], W^[2] соответственно,
lambda — параметр регуляризации λ)

Для предотвращения эффекта «переобучения» нейронной сети, при котором снижается ее обобщающая способность для примеров, не участвовавших в обучении [14], [16], необходимо произвести регуляризацию вычисленных по (16) частных производных. Для этого формулы (16) необходимо заменить формулами (18):

$$\begin{aligned}
 D_{ij}^{[1]} &= \frac{\delta^{[2]}}{m} \text{ при } j = 0, \\
 D_{ij}^{[1]} &= \frac{\delta^{[2]}}{m} + \lambda \frac{W_{ij}^{[1]}}{m} \text{ при } j > 0, \\
 D_{ij}^{[2]} &= \frac{\delta^{[3]}}{m} \text{ при } j = 0, \\
 D_{ij}^{[2]} &= \frac{\delta^{[3]}}{m} + \lambda \frac{W_{ij}^{[2]}}{m} \text{ при } j > 0.
 \end{aligned}
 \tag{18}$$

Программный код, реализующий регуляризацию частных производных нейронной сети для синтаксиса языка программирования Octave, приведен в лист. 3.

Работа алгоритма градиентного спуска заключается в итеративном расчете градиента функции ошибки. Итеративная работа алгоритма градиентного спуска осуществляется до тех пор, пока не будет выполнено условие останковки. Такими условиями могут быть следующие:

- число прошедших итераций достигло заданного значения;
- значение функции ошибки достигло некоторого порогового;
- значение функции ошибки на проверочной выборке данных удовлетворяет заданному условию.

В рассматриваемом случае для обучения нейронной сети выбрано первое условие, при котором число итераций составляет 50.

Программная реализация минимизации функции ошибки методом градиентного спуска с применением алгоритма обратного распространения в среде Octave может быть выполнена встроенными средствами. В данном исследовании была выбрана

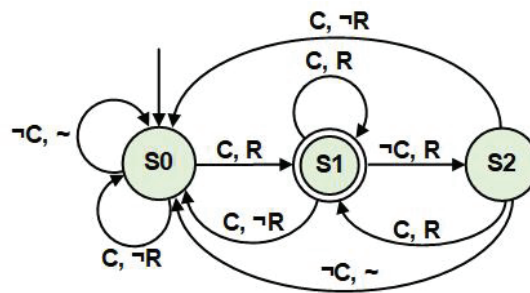


Рис. 7. Граф переходов конечного автомата, описывающего взаимодействие блоков синхронизации и распознавания

встроенная функция «fmincg», при помощи которой получены оптимальные значения параметров нейронной сети $W^{[1]}$ и $W^{[2]}$. После обучения распознавание нейронной сетью очередного цикла кодового сигнала заключается в однократном расчете функций активаций нейронов на основании оптимальных значений параметров $W^{[1]}$ и $W^{[2]}$. В результате данного расчета нейронная сеть формирует ответ, определяющий тип кодового сигнала.

Алгоритм взаимодействия блоков синхронизации и распознавания системы классификации кодовых сигналов. Алгоритм взаимодействия блоков синхронизации и распознавания в предлагаемой системе классификации кодовых сигналов (рис. 3) может быть реализован на основе детерминированного конечного автомата, граф которого показан на рис. 7.

Конечный входной алфавит символов автомата $\Sigma_{\text{вх}}$ содержит четыре символа — {C,R; C,-R; -C,R; -C-R}. Каждый символ данного алфавита представляет собой сочетание двух логических переменных C и R. При этом значение переменной C истинно, когда блоку синхронизации удалось успешно осуществить поиск стартового отсчёта кодового сигнала и ложно в противном случае. Значение переменной R истинно в случае успешной классификации блоком распознавания очередного кодового сигнала и ложно в противном случае.

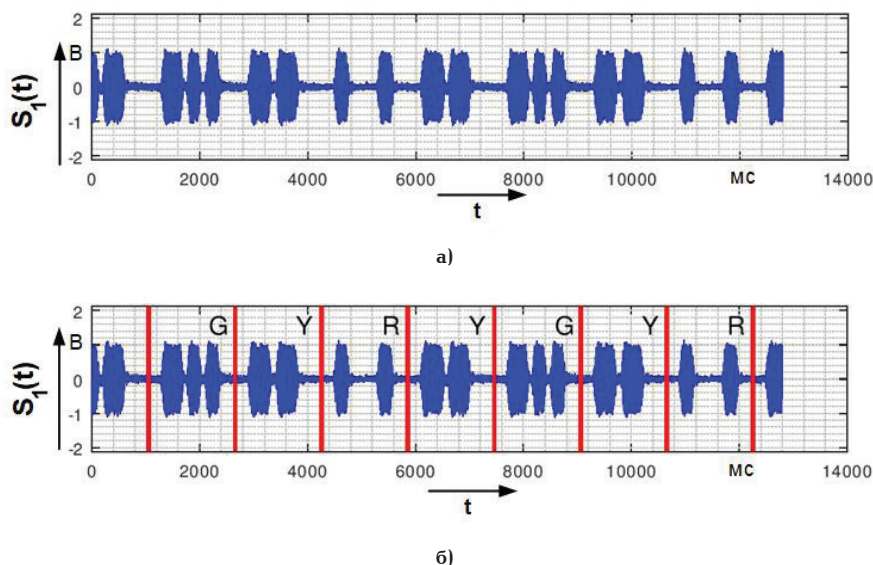


Рис. 8. Результат классификации последовательности кодовых сигналов $S_1(t)$: а) — исходная осциллограмма; б) — размеченная осциллограмма

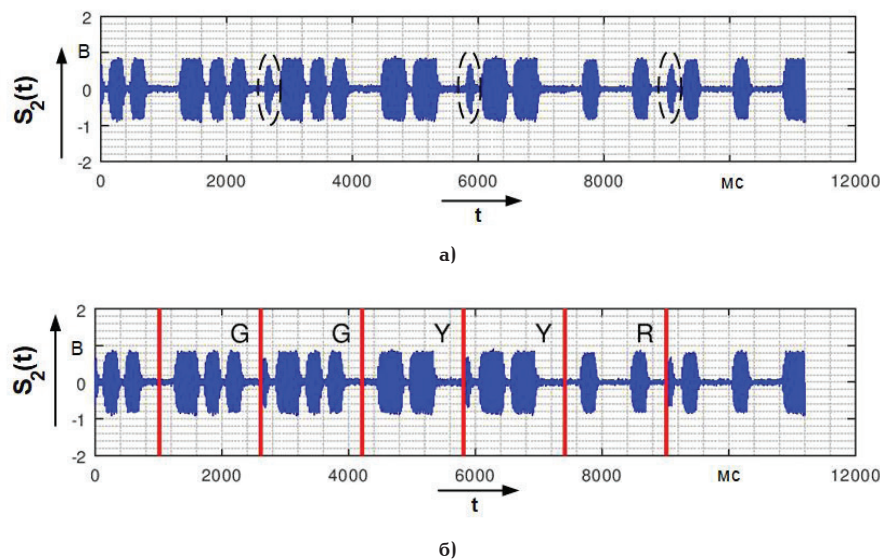


Рис. 9. Результат классификации последовательности кодовых сигналов $S_2(t)$: а) — исходная осциллограмма; б) — размеченная осциллограмма

Конечное множество внутренних состояний автомата S_{ck} содержит три элемента — $\{S_0, S_1, S_2\}$. Автомат начинает свою работу из начального состояния S_0 . Состояние S_1 характеризуется успешным поиском стартового отсчета, а также положительным результатом распознавания кодового сигнала. Состояние S_2 соответствует неудачному поиску стартового отсчета при положительном результате распознавания кодового сигнала. Если в состояниях S_0 и S_2 системе не удалось обнаружить стартовый отсчет, то распознавание не выполняется (на рис. 7 обозначено знаком «~»), и автомат переходит в начальное состояние S_0 .

Конечное множество переходов E_{ck} определяет множество функций переходов $\delta_1 - \delta_9$ (19):

$$\begin{aligned}
 \delta_1(S_0, -C-) &= S_0, \\
 \delta_2(S_0, C-R) &= S_0, \\
 \delta_3(S_0, CR) &= S_1, \\
 \delta_4(S_1, CR) &= S_1, \\
 \delta_5(S_1, C-R) &= S_0, \\
 \delta_6(S_1, -CR) &= S_2, \\
 \delta_7(S_2, C-R) &= S_0, \\
 \delta_8(S_2, CR) &= S_1, \\
 \delta_9(S_2, -C-) &= S_0.
 \end{aligned} \quad (19)$$

Конечное множество допустимых (конечных) состояний F_{ck} содержит один элемент — S_1 .

Демонстрация результатов работы системы классификации кодовых сигналов. С целью проверки и демонстрации результатов работы предлагаемой системы классификации передадим на ее вход осциллограмму электрического сигнала $S_1(t)$, содержащего несколько циклов различных кодовых сигналов (рис. 8а). В результате работы системы классификации на ее выходе получена размеченная осциллограмма электрического сигнала (рис. 8б).

Вертикальные линии соответствуют обнаруженным границам циклов кодового сигнала в составе последовательности, а буквы сверху — определенному нейронной сетью типу кодового сигнала («G» — код З, «Y» — код Ж, «R» — код КЖ).

Для демонстрации корректной классификации кодовых сигналов с помехами на вход системы классификации передадим электрический сигнал $S_2(t)$, осциллограмма которого приведена на рис. 9а. Для наглядности указанные помехи обведены штриховой линией. На рис. 9б приведен результат классификации переданного электрического сигнала. Из рисунка можно увидеть, что корректная классификация кодового сигнала может быть реализована даже при наличии помехи в длинном интервале. В электротехнических системах локомотивной сигнализации, эксплуатируемой в настоящее время, такие помехи приводят к сбою дешифрирующей аппаратуры локомотива.

Заключение. Предложенная система дешифрации кодовых сигналов реализует совокупность алгоритмов обработки сигналов принимаемых из электрической рельсовой цепи: демодуляция, поиск стартового отсчета, распознавание. При обработке сигнала с незначительным уровнем помех алгоритмы применяются последовательно. При значительной зашумленности принятого сигнала и невозможности поиска стартового отсчета его позиция рассчитывается по стартовому отсчету предыдущего кодового цикла и известной длине кодовых циклов. Корректность такого расчета проверяется при распознавании. Удалось продемонстрировать, что предложенная система дешифрации кодовых сигналов является более устойчивой к помехам по сравнению с применяемыми в настоящий момент системами дешифрации бортового оборудования локомотива.

Библиографический список

1. Леонов А. А. Техническое обслуживание автоматической локомотивной сигнализации. 5-е изд., перераб. и доп. М.: Транспорт, 1982. 255 с.
2. Венцевич Л. Е. Локомотивные устройства обеспечения безопасности движения поездов и расшифровка информации.

- онных данных их работы. 2-е изд., стер. М., 2013. 327 с. ISBN 978-5-89035-703-8.
3. Правила технической эксплуатации железных дорог 2017 ПТЭ (ИСИ, ИДП). Новоуральск: Новоуральская типография, 2017. 574 с.
 4. Табунщиков А. К., Горенбейн Е. В., Стряпкин Л. И. Сбои АЛСН. Проблемы и пути их решения // Автоматика, связь, информатика. 2015. № 8. С. 21–22.
 5. Пат. 2618616 Российская Федерация, МПК В 61 L 25/06, Н 04 В 15/00. Устройство подавления импульсных помех на входе локомотивного приемника АЛС / Аргунов И. А., Вихрова Н. Ю., Горенбейн Е. В. [и др.]. № 2015154775; заявл. 21.12.15; опубл. 04.05.17, Бюл. № 13.
 6. Юсупов Р. Р. Цифровое устройство обработки сигналов автоматической локомотивной сигнализации повышенной помехозащищенности: дис. ... канд. техн. наук. Самара, 2003. 162 с.
 7. Брылеев А. М., Поупе О., Дмитриев В. С. [и др.]. Автоматическая локомотивная сигнализация и авторегулировка. М.: Транспорт, 1981. 320 с.
 8. Сороко В. И., Фотькина Ж. В. Аппаратура железнодорожной автоматики и телемеханики: справочник. В 4 кн. 4-е изд., перераб. и доп. М.: Планета, 2013. Кн. 2. 1046 с. ISBN 978-5-901307-22-9.
 9. Пат. 165420 Российская Федерация, МПК В 61 L 23/00. Приемное устройство автоматической локомотивной сигнализации / Леушин В. Б., Юсупов Р. Р., Блачев К. Э. № 2015155567/11; заявл. 23.12.15; опубл. 20.10.16, Бюл. № 29.
 10. Засов В. А., Железнов Д. В., Митрофанов А. Н., Белоголов А. С. Адаптивное подавление помех в приемных устройствах автоматической локомотивной сигнализации // Электротехника. 2017. № 3. С. 18–22.
 11. Розенберг Е. Н. Инновационная технология управления движением поездов // Автоматика, связь, информатика. 2017. № 10. С. 2–4.
 12. Розенберг Е. Н. Цифровая железная дорога — ближайшее будущее // Автоматика, связь, информатика. 2016. № 10. С. 4–7.
 13. Flach P. The art and science of algorithms that make sense of data. Cambridge: Cambridge University Press, 2012. 409 p. ISBN 978-1107422223.
 14. Goodfellow I., Bengio Y., Courville A. Deep learning. Cambridge: MIT Press, 2016. 775 p. ISBN 0262035618; 978-0262035613.
 15. Hastie T. The elements of statistical learning. 2nd ed. NY: Springer, 2009. 745 p. ISBN 0387848576; 978-0387848570.
 16. Bishop C. M. Pattern recognition and machine learning. NY: Springer, 2010. 738 p. ISBN 978-0-387-31073-2.
 17. Harrington P. Machine learning in action. NY: Manning Publications, 2012. 354 p. ISBN 1617290181; 9781617290183.
 18. Vaillant R., Monrocq C., Le Cunn Y. Original approach for the localisation of objects in images IEE Proceedings – Vision, Image and Signal Processing. 1994. Vol. 141, Issue 4. P. 245–250.
 19. Le Cunn Y., Jackel L. D., Bottou L. [et al.]. Learning algorithms for classification: a comparison on handwritten digit recognition. Neural networks: the statistical mechanics perspective. 1995. P. 261–276.
 20. Murphy K. P. Machine learning: a probabilistic perspective. Cambridge: MIT Press, 2012. 1067 p. ISBN 0262018020; 978-0262018029.
 21. Owen S., Anil R., Dunning T. [et al.]. Mahout in Action. NY: Manning Publications, 2011. 416 p. ISBN 978-1935182689.
 22. Shalev-Swartz S., Ben-David S. Understanding machine learning: from theory to algorithms. Cambridge: Cambridge University Press, 2014. 410 p. ISBN 978-1107057135.
 23. Сергиенко А. Б. Цифровая обработка сигналов. 3-е изд. СПб.: БХВ-Петербург, 2011. 768 с.
 24. Rajeev A., Dill D. L. A theory of timed automata // Theoretical computer science. 1994. Vol. 126. P. 183–235.
 25. Boyd S. P. Convex optimization. NY: Cambridge University Press, 2004. 716 p. ISBN 0521833787.
 26. Mitchel T. M. Machine learning. NY: McGraw-Hill Science, 1997. 432 p. ISBN 0070428077.

ПРИСУХИНА Илона Вадимовна, аспирантка кафедры «Автоматика и телемеханика».

SPIN-код: 8429-0259

AuthorID (РИНЦ): 907740

Адрес для переписки: ivprisukhina@mail.ru

БОРИСЕНКО Дмитрий Владимирович, кандидат технических наук, доцент (Россия), доцент кафедры «Автоматика и телемеханика».

SPIN-код: 9697-9830

AuthorID (РИНЦ): 533908

Адрес для переписки: borisenkodv@yandex.ru

Для цитирования

Присухина И. В., Борисенко Д. В. Машинная классификация сигналов числового кода в электротехнических системах локомотивной сигнализации // Омский научный вестник. 2019. № 4 (166). С. 39–47. DOI: 10.25206/1813-8225-2019-166-39-47.

Статья поступила в редакцию 14.06.2019 г.

© И. В. Присухина, Д. В. Борисенко