

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
Кафедра «Прикладная математика и информатика»

02.03.03 МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ И
АДМИНИСТРИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ

БАКАЛАВРСКАЯ РАБОТА

на тему «Разработка алгоритма диагностики сварки с использованием
нейронных сетей прямого распространения»

Студент	_____	А.В. Новоженин	_____
Руководитель	_____	В.С. Климов	_____
Консультант по аннотации	_____	Н.В. Яценко	_____

Допустить к защите
Заведующий кафедрой к.тех.н., доцент, А.В. Очеповский _____

« _____ » _____ 2017 г.

Тольятти 2017

АННОТАЦИЯ

Тема: «Разработка алгоритма диагностики сварки с использованием нейронных сетей прямого распространения».

В данной выпускной квалификационной работе исследуются пути совершенствования методов диагностики сварки. В исследовании рассматривается задача прогнозирования качества сварки с использованием нейронных сетей. При этом предложено наблюдать за изменением коэффициента мощности сварки и прогнозировать силу разрушения сварного соединения. В работе представлены результаты апробации алгоритма на практике.

Структура ВКР представлена введением, тремя главами, заключением, списком литературы.

Во введении описывается актуальность проводимого исследования, формулируется цель и ставятся задачи, которые необходимо решить.

В первой главе приводится обзор методов диагностики контактной сварки с помощью алгоритмов машинного обучения.

Во второй главе описывается математический аппарат нейронных сетей. Затем приводится описание разработанного алгоритма диагностики сварки на основе нейронных сетей.

В третьей главе будет проведена реализация и тестирование разработанного программного обеспечения, а также представлены результаты его работы.

Было доказано, что задача восстановления зависимости между измеряемыми и прогнозируемыми параметрами может быть решена с использованием нейронных сетей.

Данная бакалаврская работа состоит из пояснительной записки на 54 стр., включая 16 рисунков, списка 24 источников, в том числе 23 источника на иностранном языке и 2 приложения.

ABSTRACT

The title of the graduation work is “Development of an algorithm of diagnostics of welding with use of neural networks”.

In this graduation part work ways of improvement of methods of diagnostics of welding are investigated. In a research the problem of forecasting of quality of welding with use of neural network is considered. At the same time, it is offered to watch behind change of electrical power factor of welding and to predict force of corrupting of welded joint. In work results of approbation of an algorithm in practice are presented.

The structure of the graduation part is represented by an introduction, three parts, a conclusion and a list of literature. The introduction describes the relevance of the research conducted, the goal and the tasks.

In the first part review of methods of diagnostics of contact welding by means of algorithms of machine learning is provided.

The second part describes the mathematical apparatus of neural networks. Then the description of the developed algorithm of diagnostics of welding on the basis of neural networks is provided.

In the third part, the software was implemented and tested. Also, the results of the software were provided.

It was proved that the task of restoring the relationship between measured and predicted parameters can be solved using neural networks.

The graduation work consists of an explanatory note on 54 pages, including 16 figures, the list of 24 references including 23 foreign sources and 2 appendixes.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
ГЛАВА 1 АНАЛИЗ СОСТОЯНИЯ ВОПРОСА	7
1.1 Обзор предметной области	7
1.2. Обзор систем интеллектуальной диагностики сварки	9
ГЛАВА 2 РАЗРАБОТКА АЛГОРИТМА ДИАГНОСТИКИ СВАРКИ С ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ.....	22
2.1 Нейронная сеть прямого распространения как метод построения регрессионной модели	22
2.2 Источник диагностических признаков контактной сварки.....	27
2.3 Диагностика сварки с использованием нейронных сетей прямого распространения.....	31
ГЛАВА 3 ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ ПРЕДЛОЖЕННЫХ РЕШЕНИЙ	33
3.1 Программная реализация алгоритма.....	33
3.2 Апробация предложенных подходов и исследование точности получаемых регрессионных моделей.....	36
ЗАКЛЮЧЕНИЕ	40
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	42
ПРИЛОЖЕНИЕ А.....	45
ПРИЛОЖЕНИЕ Б.....	54

ВВЕДЕНИЕ

Контактная сварка позволяет производить неразъемные соединения деталей листовой формы.

Контактная сварка устроена следующим образом. Свариваемые детали друг относительно друга внахлест с учетом требуемого положения. Затем место планируемой сварки размещается между медными электродами (диаметр рабочей поверхности электрода составляет порядка 6 мм). Затем производится сжатие электродов с усилием порядка 300кгс с продавливанием свариваемых деталей для обеспечения плотного контакта. После этого через электроды подается несколько импульсов тока, обеспечивающих локальное расплавление деталей в зоне контакта (рисунок 1.1). После затвердевания расплавленного металла усилие электроды разжимаются. Длительность сварки – не более 0,2 секунды.



Рисунок 1.1 – Контактная точечная сварка на предприятии ПАО «АвтоВАЗ»
(на производстве модели Lada Vesta)

Основной проблемой контактной сварки является низкая стабильность получаемых сварных соединений. Это означает, что в одной партии изделий могут попадаться детали, как с качественной сваркой, так и с некачественной.

Для того, чтобы контролировать качество сварных соединений применяются различные методы контроля, которые делятся на разрушающие (механические испытания, металлография и химический анализ) и не разрушающие (визуально-оптический метод, радиационная дефектоскопия, ультразвуковая дефектоскопия, электромагнитная дефектоскопия).

Однако применение таких методов в условиях серийного производства осложнено по следующим причинам, главной из которых является невозможность их использования для сквозной диагностики абсолютно всех сварных соединений. Это, в сочетании с нестабильностью процесса сварки не позволяет гарантировать качество конечной продукции.

С развитием искусственного интеллекта, алгоритмы интеллектуального анализа данных стали активно применяться при синтезе систем управления и диагностики технологическими процессами.

В области диагностики контактной сварки алгоритмы интеллектуального анализа данных применяются для того, чтобы найти взаимосвязь между косвенными параметрами, характеризующими протекание процесса сварки, и качеством получаемых соединений.

Системы диагностики сварки, основанные на таком анализе, позволяют оценивать качества сварного соединения ментально, сразу же после окончания сварки.

Исследованиями на данную тему занимаются такие ученые, как P. Laurinen, H. Junno, L. Tuovinen, J. Rönning, A. Aravinthan, K. Sivayoganathan, D. Al-Dabass, V. Balendran, Б.Е. Патон, Н.В. Подола, В.С. Гавриш, М. El-Banna, D. Filev, R.V. Chinnam, Y.J. Park, H. Cho и др.

В большинстве известных исследований в качестве математического аппарата для анализа измеряемых в процессе сварки параметров

применяются нейронные сети, которые дают приемлемую точность прогнозирования различных характеристик соединений.

Однако основной проблемой данных исследований является сложность их внедрения на реальные производства. Это связано с необходимостью установки в сварочную машину дополнительных датчиков (тока, напряжения, силы сжатия электродов). Так датчики устанавливаются со стороны рабочего контура сварочной машины, то это приводит к захламлению полезного рабочего пространства вокруг сварочных электродов (что является критичным при использовании на производстве подвижных сварочных роботов). Кроме того, измерение электрических параметров осложнено влиянием электромагнитных помех, что приводит к необходимости усложнения оборудования для очистки измеряемых осциллограмм тока и напряжения от помех.

Для преодоления описанных выше недостатков в данной работе предлагается оценку сварки проводить по более легким для измерения параметрам – задержка импульса тока и длительность импульса тока. Причем их измерение предлагается проводить не со стороны электродов, а наоборот – со стороны подключения сварочной машины к сети питания.

В исследовании проверяется гипотеза о том, что нейронная сеть сможет самостоятельно восстановить зависимость прочности сварного соединения от задержки α и длительности λ протекания импульсов тока во время сварки.

Целью данной работы является разработка метода диагностики сварки, с использованием нейронных сетей, основанного на оценке кривой коэффициента мощности сварки.

В ходе выполнения исследований на языке C++ была разработана программа, позволяющая по данным экспериментальных сварок с помощью алгоритмов машинного обучения производить настройку нейронных сетей заданной конфигурации. Обученные нейронные сети можно использовать

для оценки прочности соединения по измеряемым в процессе сварки параметрам.

Исследования, проведенные в рамках данной работы, будут использованы при выполнении гранта РФФИ на тему “Управление металлургическими процессами с использованием искусственного интеллекта”.

Элементы исследования были внедрены на предприятии ПАО «АвтоВАЗ», что подтверждено актом о внедрении.

ГЛАВА 1 АНАЛИЗ СОСТОЯНИЯ ВОПРОСА

1.1 Обзор предметной области

Контактная сварка является основным технологическим процессом получения неразъемных соединений листовых деталей (корпуса, кузова и т.п.)

Основной проблемой контактной сварки является низкая стабильность получаемых сварных соединений. Это означает, что в одной партии изделий могут попадаться детали, как с качественной сваркой, так и с некачественной.

Для того, чтобы контролировать качество сварных соединений применяются различные методы контроля, которые делятся на разрушающие и не разрушающие.

К первой группе относятся следующие методы контроля:

- механические испытания, которые применяются для подбора режимов сварки на образцах-свидетелях, а также для оценки прочностных характеристик сваренных деталей.
- металлография и химический анализ, которые применяются для определения размеров литого ядра сварной точки, глубину проплавления, размер вмятины от электрода и наличие дефектов по макроструктуре.
- коррозионные испытания, которые применяются для определения коррозионной стойкости соединения.

К методам неразрушающего контроля относятся:

- Визуально-оптический метод, при котором проводится внешний осмотр сварного соединения на наличие видимых дефектов. Данный метод позволяет получить до 50% информации о качестве соединения и о ходе технологического процесса [10, 11].
- Радиационная дефектоскопия, при которой пучок рентгеновских или гамма-лучей направляется на контролируемое сварное соединение. Проходя сквозь него, лучи частично поглощаются и действуют на находящийся за соединением индикатор-фотопленку, люминесцентный экран

или ионизационный прибор. Дефекты соединения вследствие их меньшей поглощающей способности пропускают больше лучей, чем качественные сварные соединения. Различие интенсивности лучей фиксирует соответствующий индикатор [9, 10, 12].

- Ультразвуковая дефектоскопия, основанная на способности ультразвуковых колебаний распространяться в металле в виде направленных волн и отражаться от участков разной плотности, например, дефектов, при условии, что их размеры больше длины ультразвуковой волны. Ультразвуковой дефектоскопией можно определять следующие дефекты соединений, выполненных контактной сваркой: непровар, поры, раковины, трещины [13-18].

- Электромагнитная дефектоскопия, объединяющая большое число способов, которые можно разделить на две группы: намагничивание контролируемого соединения и выявления магнитных потоков рассеяния и наведения вихревых токов с оценкой электропроводности металла. Первая группа применяется при контроле соединений выполненных стыковой сваркой. Для точечной сварки установлена зависимость между диаметром литого ядра и электропроводностью зоны сварки [10, 12].

- Контроль проникающими веществами включает две группы методов: капиллярные и методы контроля течеисканием. Капиллярные методы контроля основаны на капиллярном проникновении жидкостей (пенетрантов) в дефекты и их контрастном изображении. Методы контроля течеисканием применяются для обнаружения сквозных дефектов [10].

Однако применение таких методов в условиях серийного производства осложнено по следующим причинам:

- Высокая трудоемкость, связанная с необходимостью задействования дополнительного персонала, непосредственно проводящий контроль описанными выше методами.

- Снижение производительности производства. Контроль качества – это дополнительная технологическая операция, увеличивающая технологическую цепочку получения готового изделия.
- Повышение стоимость производства. Для использования данных методов контроля необходимо использования дорогостоящего исследовательского оборудования.
- Невозможность их использования для сквозной диагностики абсолютно всех сварных соединений. Скорость протекания процесса контактной сварки (точечной) составляет порядка 0,2 секунды. Такая высокая производительность сварки не позволяет проводить контроль всех сварных соединений.

Однако развитие алгоритмов искусственного интеллекта начали появляться принципиально новые системы контроля качества сварки, которые лишены описанных выше недостатков.

1.2. Обзор систем интеллектуальной диагностики сварки

В настоящее время алгоритмы интеллектуального анализа данных стали активно применяться при синтезе систем управления и диагностики технологическими процессами.

В области диагностики контактной сварки алгоритма интеллектуального анализа данных применяются для того, чтобы найти взаимосвязь между косвенными параметрами, характеризующими протекание процесса сварки, и качеством получаемых соединений.

Системы диагностики сварки, основанные на таком анализе, позволяют оценивать качества сварного соединения ментально, сразу же после окончания сварки. Другими словами, производительность диагностики сопоставима с производительностью процесса сварки.

Рассмотрим существующие исследования по применению алгоритмов искусственного интеллекта для диагностики сварки.

В зарубежном исследовании (под авторством P, Laurinen) предложено проводить контроль качества путем анализа данных измеренных в процессе сварки с помощью байесовских сетей доверия.

Схема работы предложенного подхода представлена на рисунке 1.2.

В рассматриваемом исследовании диагностика качества сварки заключается в прогнозировании принадлежности диаметра сварного ядра контролируемого соединения к одному из нескольких классов.

При этом в процессе сварки предлагается измерять как электрические параметры (осциллограмму тока во вторичном контуре с помощью пояса Роговского и осциллограмму напряжения между электродами), так и механические параметры осциллограмма изменения силы сжатия между электродами.

- Недостатками такого подхода, ограничивающими использования данного метода на крупных производствах, являются:
 - Необходимость модификации привода сжатия для установки в него датчика давления.
 - Влияние электромагнитных помех на осциллограммы тока и напряжения. Это делает необходимым усложнения аппаратуры для измерения для очистки сигналов от помех.
 - Зависимость точности снятия осциллограммы силы сжатия электродов от износа подвижных частей сварочной машины.
 - Невозможность использования данного подхода на сварочных роботах из-за электромеханического привода сжатия электродов.

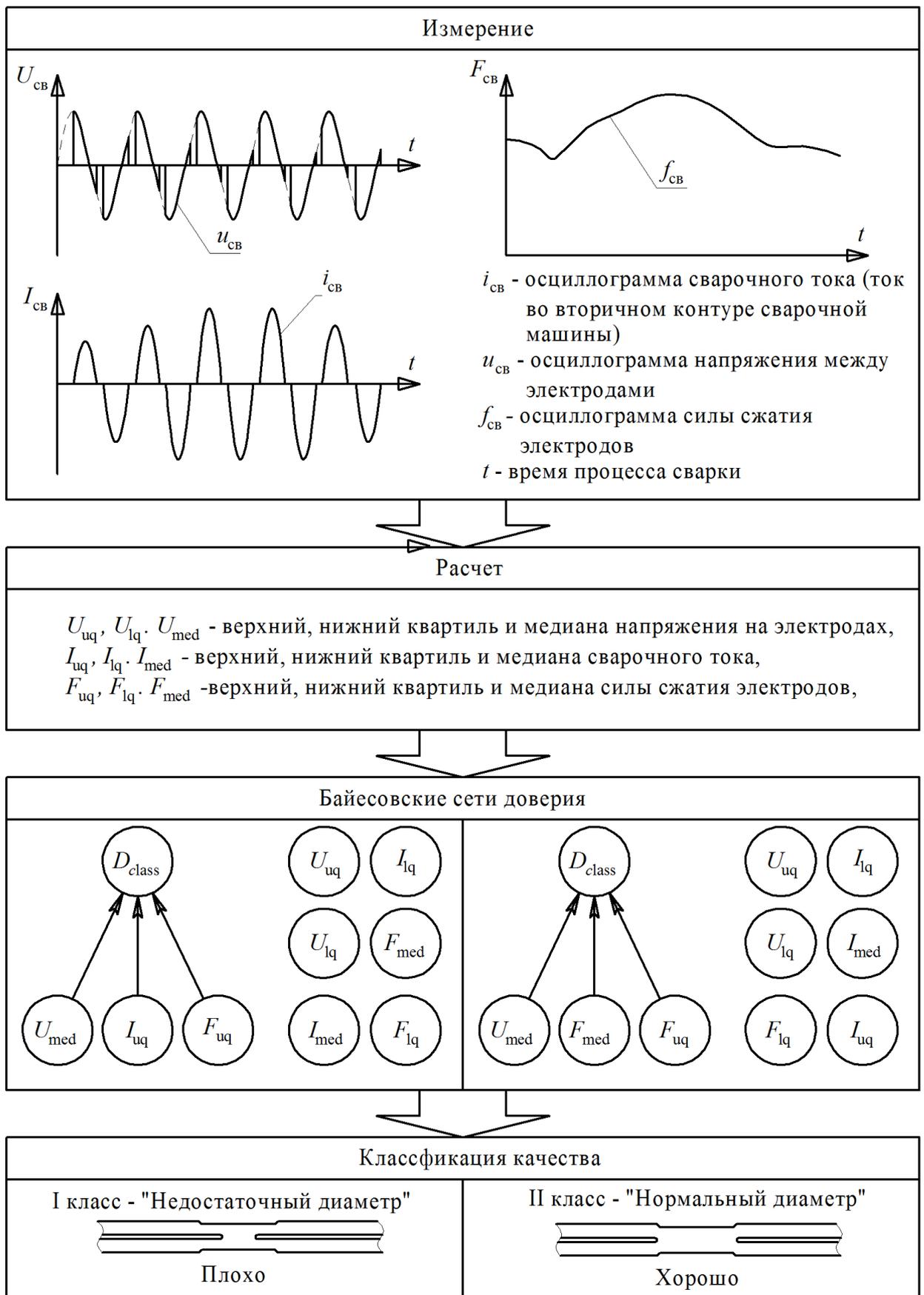


Рисунок 1.2 – Диагностика с помощью Байесовских сетей доверия

В зарубежном исследовании под авторством А. Aravinthan докладывается об успешном контроле качества сварки с помощью двухслойной нейронной сети прямого распространения.

Схема работы предложенного подхода представлена на рисунке 1.3.

В рассматриваемом исследовании диагностика качества сварки заключается в прогнозировании для контролируемого соединения таких прочностных характеристик, как предельная сила разрушения соединения.

В данном способе контроля в процессе сварки предложено измерять осциллограмму тока во вторичном контуре с помощью пояса Роговского и осциллограмму напряжения между электродами (рисунок 1.3).

Недостатками такого подхода, ограничивающими использования данного метода на крупных производствах, являются:

- Влияние электромагнитных помех на измерение электрических параметров в процессе сварки делает необходимым построение сложной аппаратуры для измерений (которая способна очищать сигналы от помех).
- Измерительные датчики приводят к загромождению рабочей области сварочной машины, что является существенным недостатком при использовании на сварочных работах.
- Для обучения двухслойной нейронной сети прямого распространения требуется накапливание обучающей выборки большого размера.

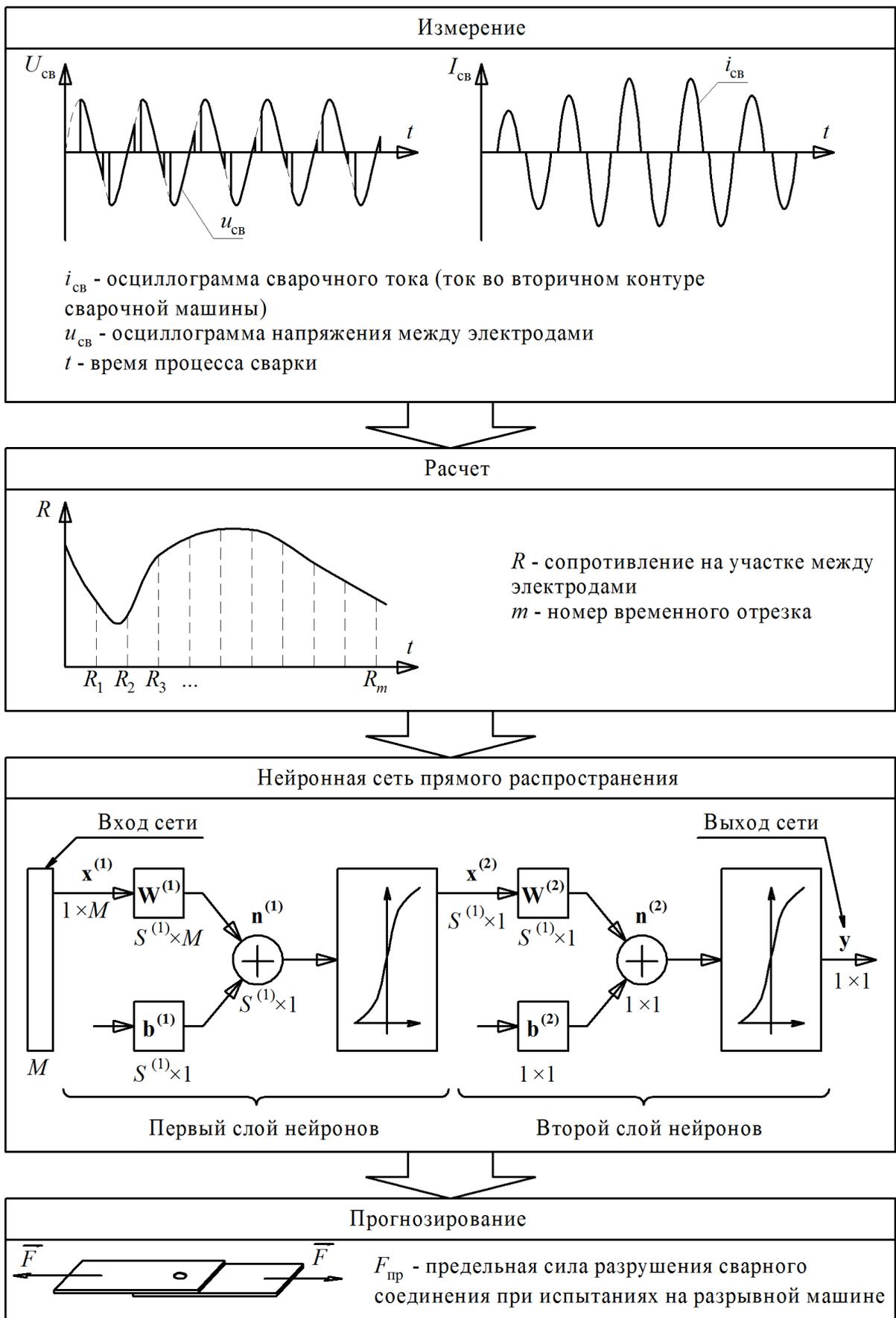


Рисунок 1.3 – Диагностика с помощью нейронной сети прямого распространения

В другом зарубежном исследовании под авторством Б.Е. Патон предложено проводить контроль качества путем анализа данных измеренных в процессе сварки с помощью нейронной сети прямого распространения.

Схема работы предложенного подхода представлена на рисунке 1.4.

В рассматриваемом исследовании диагностика качества сварки заключается в прогнозировании для контролируемого соединения таких прочностных характеристик, как предельная сила разрушения соединения. Однако по сравнению с предыдущим исследованием - не производится расчет сопротивления на участке между электродами.

В данном способе контроля в процессе сварки предложено измерять осциллограмму тока во вторичном контуре с помощью пояса Роговского и осциллограмму напряжения между электродами (рисунок 1.4).

Недостатками такого подхода, ограничивающими использования данного метода на крупных производствах, являются:

- Влияние электромагнитных помех на измерение электрических параметров в процессе сварки делает необходимым построение сложной аппаратуры для измерений (которая способна очищать сигналы от помех).
- Измерительные датчики приводят к загромождению рабочей области сварочной машины, что является существенным недостатком при использовании на сварочных работах.
- Для обучения двухслойной нейронной сети прямого распространения требуется накапливание обучающей выборки большого размера.
- Диаметр литого ядра является субъективным параметром качества сварки, который лишь косвенно описывает прочностные характеристики соединения.

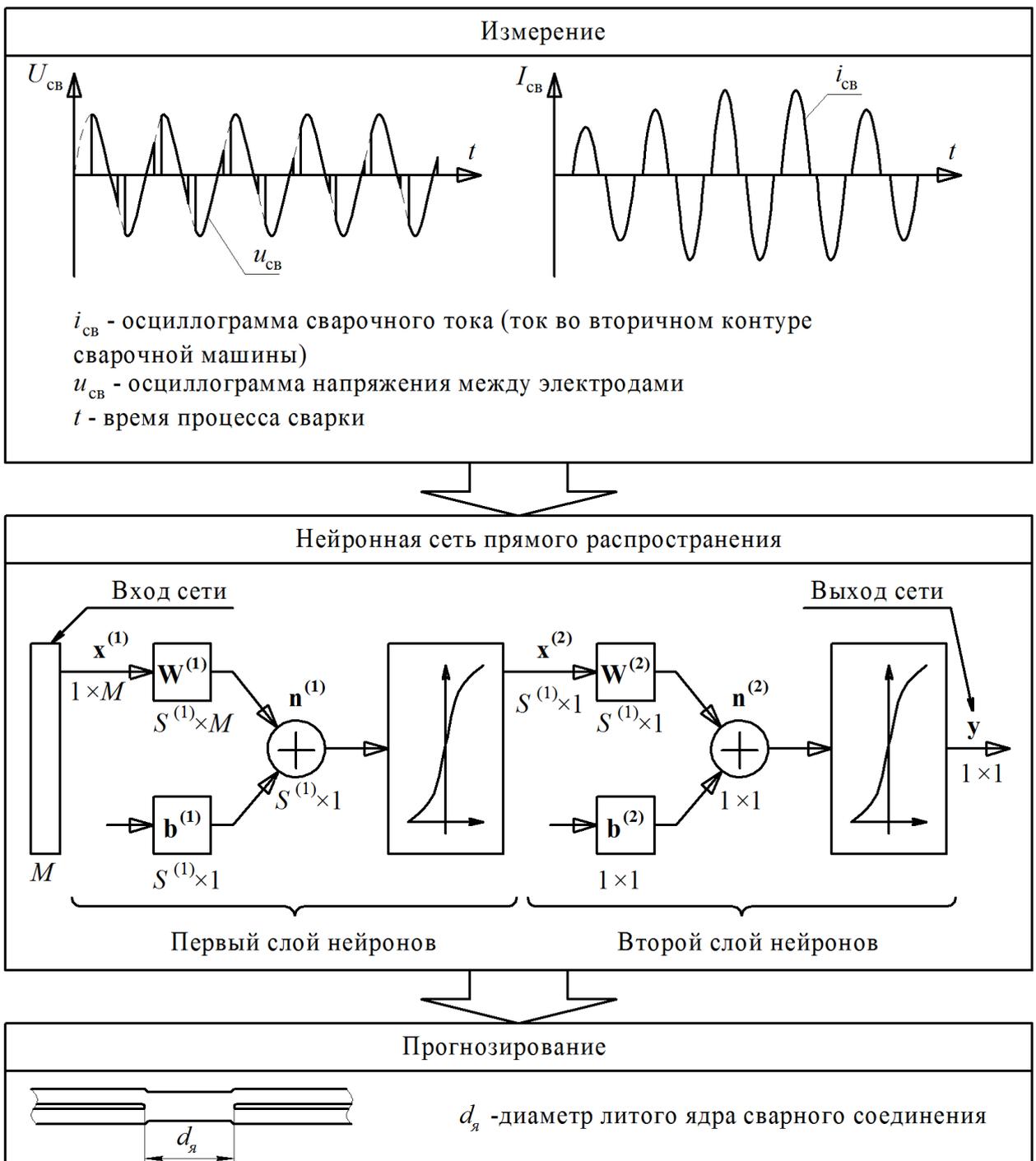


Рисунок 1.4 – Диагностика с помощью двухслойной нейронной сети прямого распространения

В другом зарубежном исследовании под авторством М. El-Banna докладывается об успешном контроле качества путем анализа данных, измеренных в процессе сварки, с помощью нейронной сети LVQ.

Схема работы предложенного подхода представлена на рисунке 1.5.

В рассматриваемом исследовании диагностика качества сварки заключается в отнесении анализируемого сварного соединения к одному из трех классов (два из этих класса соответствуют соединениям не удовлетворяющих критериям качества и один класс соответствует соединениям удовлетворяющим критериям качества).

В данном способе контроля в процессе сварки предложено измерять осциллограмму тока во вторичном контуре с помощью пояса Роговского и осциллограмму напряжения между электродами (рисунок 1.5).

Недостатками такого подхода, ограничивающими использования данного метода на крупных производствах, являются:

- Влияние электромагнитных помех на измерение электрических параметров в процессе сварки делает необходимым построение сложной аппаратуры для измерений (которая способна очищать сигналы от помех).
- Измерительные датчики приводят к загромождению рабочей области сварочной машины, что является существенным недостатком при использовании на сварочных работах.
- Нейронная сеть LVQ (в силу своей архитектуры) не позволяет давать количественную оценку качества сварки.

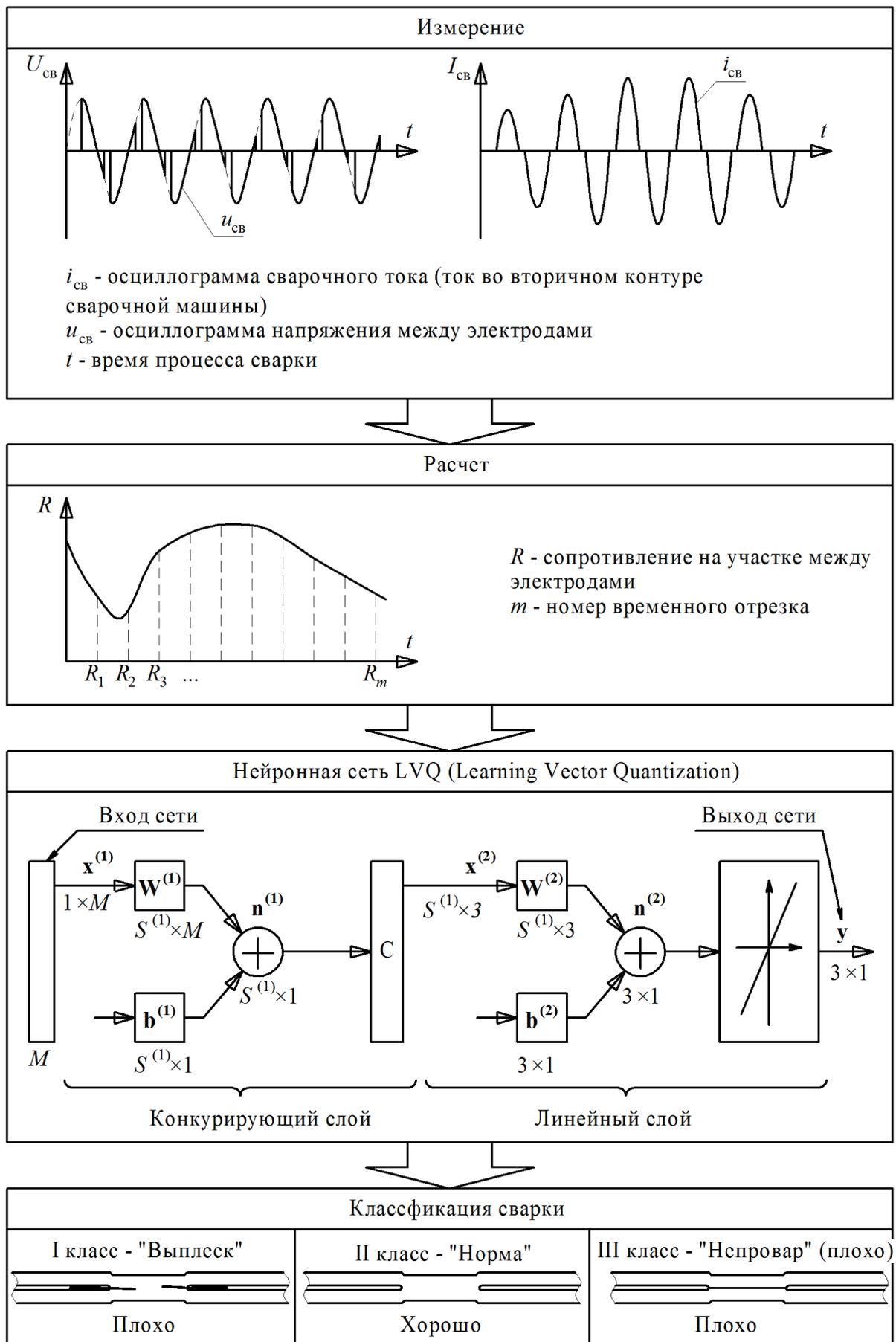


Рисунок 1.5 – Диагностика с помощью нейронной сети LVQ

В зарубежном исследовании, автором которого является Y.J. Park, докладывается об успешном контроле качества путем анализа данных, измеренных в процессе сварки, с помощью нейронной сети LVQ.

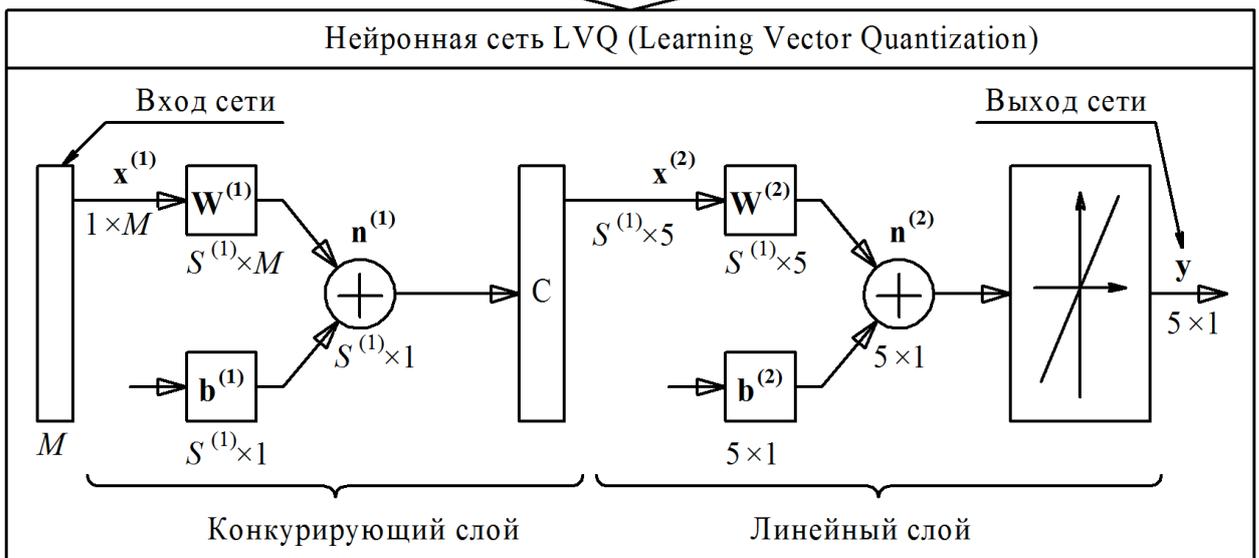
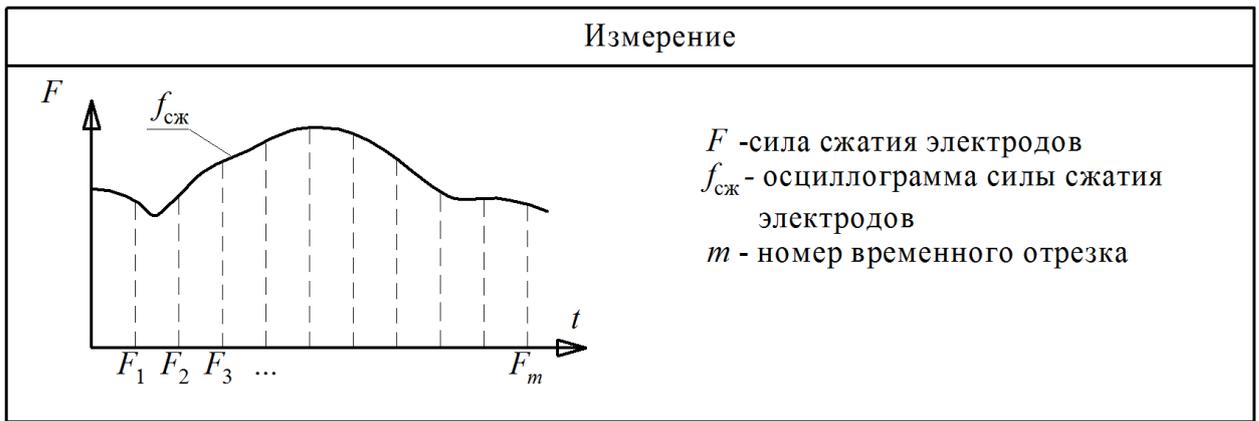
Схема работы предложенного подхода представлена на рисунке 1.6.

В рассматриваемом исследовании диагностика качества сварки заключается в отнесении анализируемого сварного соединения к одному из пяти классов: “Insufficient welding”, “Poor welding”, “Good welding”, “Rich welding”, “Excessive welding”. Каждому классу соответствует свой набор физико-механических характеристик соединений. Наилучшим считается отнесение сварки к классу “Good welding”.

В данном способе контроля в процессе сварки измеряется осциллограмма изменения силы сжатия электродов (рисунок 1.6).

Недостатками такого подхода, ограничивающими использования данного метода на крупных производствах, являются:

- Необходимость модернизации сварочного аппарата для установки датчика давления.
- Невозможность использования данного подхода на сварочных работах (только на стационарных машинах) из-за массивного привода сжатия электродов.
- Нейронная сеть LVQ (в силу своей архитектуры) не позволяет давать количественную оценку качества сварки



Классификация сварки

I класс - "Insufficient welding": отсутствует образование литого ядра соединения. Что можно обнаружить при визуальном осмотре	II класс - "Poor welding": литое ядро соединения недостаточных размеров, чтобы обеспечить требуемую прочность	III класс - "Good welding" литое ядро соединения обеспечивает нужные прочностные характеристики
IV класс - "Rich welding": Нужные прочностные характеристики обеспечены. Но внешний некоторых соединений испорчен выплесками жидкого металла	V класс - "Excessive welding": На всех сварных соединениях присутствуют выплески. Требуется проверка качества сварки	I class "Insufficient welding" II class "Poor welding" III class "Good welding" IV class "Rich welding" V class "Excessive welding"

Рисунок 1.6 – Диагностика с помощью нейронной сети LVQ

В большинстве известных исследований в качестве математического аппарата для анализа измеряемых в процессе сварки параметров применяются нейронные сети, которые дают приемлемую точность прогнозирования различных характеристик соединений.

Однако основной проблемой данных исследований является сложность их внедрения на реальные производства. Это связано с необходимостью установки в сварочную машину дополнительных датчиков (тока, напряжения, силы сжатия электродов). Так датчики устанавливаются со стороны рабочего контура сварочной машины, то это приводит к захламлению полезного рабочего пространства вокруг сварочных электродов (что является критичным при использовании на производстве подвижных сварочных роботов). Кроме того, измерение электрических параметров осложнено влиянием электромагнитных помех, что приводит к необходимости усложнения оборудования для очистки измеряемых осциллограмм тока и напряжения от помех.

Для преодоления описанных выше недостатков в данной работе предлагается оценку сварки проводить по более легким для измерения параметрам – задержка импульса тока и длительность импульса тока. Причем их измерение предлагается проводить не со стороны электродов, а наоборот – со стороны подключения сварочной машины к сети питания.

В исследовании проверяется гипотеза о том, что нейронная сеть сможет самостоятельно восстановить зависимость прочности сварного соединения от задержки α и длительности λ протекания импульсов тока во время сварки.

Целью работы является снижение трудоёмкости контроля качества сварных соединений за счет разработки новых подходов к диагностике сварки с помощью нейронных сетей прямого распространения.

Поставленная цель достигается путем последовательного решения ряда задач:

1. Анализ состояния вопроса об использовании интеллектуального анализа данных в системах диагностики сварки.
2. Разработка алгоритма диагностики сварки с использование нейронной сети прямого распространения.
3. Проектирование, разработка, апробация программного обеспечения для реализации предложенных решений.

ГЛАВА 2 РАЗРАБОТКА АЛГОРИТМА ДИАГНОСТИКИ СВАРКИ С ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ

2.1 Нейронная сеть прямого распространения как метод построения регрессионной модели

Формально задача построения регрессионной модели можно описать следующим образом. Существует множество входных переменных X – описаний объектов, а также множество Y выходных переменных (значения которых зависят от входных переменных).

В задаче регрессии существует семейство функций, которое является отображением вида:

$$f : W \times X \rightarrow Y, \quad (2.1)$$

где $w \in W$ – пространство настраиваемых параметров, $x \in X$ – пространство входных переменных, $y \in Y$ – пространство выходных переменных, f – семейство функций вида $f(w, x)$.

Так как регрессионный анализ предполагает поиск зависимости математического ожидания случайной величины от свободных переменных:

$$E(y | x) = f(x) \quad (2.2)$$

, то в её состав входит аддитивная случайная величина ε . Таким образом:

$$y = f(w, x) + \varepsilon. \quad (2.3)$$

Регрессионная модель является настроенной, когда зафиксированы её параметры, то есть модель задаёт отображение для фиксированных значений W :

$$f : X \rightarrow Y \quad (2.4)$$

Исходными данными для построения регрессионной модели является обучающая выборка D^m вида:

$$D^m = (x_1, y_1), \dots, (x_m, y_m), \quad (2.5)$$

где m – размер обучающей выборки, x_i – вектор значений входных переменных для i -го объекта из обучающей выборки, y_i – значение выходного параметра для i -го объекта из обучающей выборки. При этом

вектор x_i описывается набором значений числовых атрибутов P , а y_i – описывается единственным числовым значением:

$$x_i = (P_1, P_2, \dots, P_n), \quad (2.6)$$

где n – количество атрибутов, описывающих объект из обучающей выборки.

Одним из возможных способов построения регрессионных моделей является использование нейронных сетей прямого распространения. Такая нейронная сеть состоит из однотипных вычислительных элементов – нейронов.

Математическая модель нейрона была предложена американскими учеными У. Маккалоком и У. Питтсом. Модель нейрона представлена на рисунке 2.1.

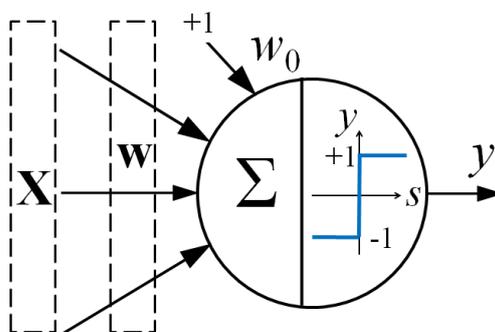


Рисунок 2.1 – Модель нейрона со ступенчатой функцией активации

Такой нейрон работает следующим образом:

1. На его входа подается набор (вектор) входных сигналов: $\mathbf{X} = (x_1, x_2, \dots, x_n)$
2. Каждый входной сигнал умножается на соответствующий весовой коэффициент из вектора $\mathbf{w} = (w_1, w_2, \dots, w_n)$
3. Затем взвешенные сигналы суммируются со смещением w_0 нейрона:

$$s = w_0 + \sum_{i=1}^n x_i w_i, \quad (2.7)$$

где n – количество входных сигналов нейрона.

4. Затем к полученной сумме сигналов применяется функция активации, чтобы определить выходной сигнал нейрона:

$$y = f(s) \quad (2.8)$$

Поздней в место применения ступенчатой функции активации (как на рисунке 2.1) исследователями было предложено использование гладких функций активации, таких как сигмоида, представленная на рисунке 2.2:

$$f(x) := \frac{2}{1 + e^{-2x}} - 1$$

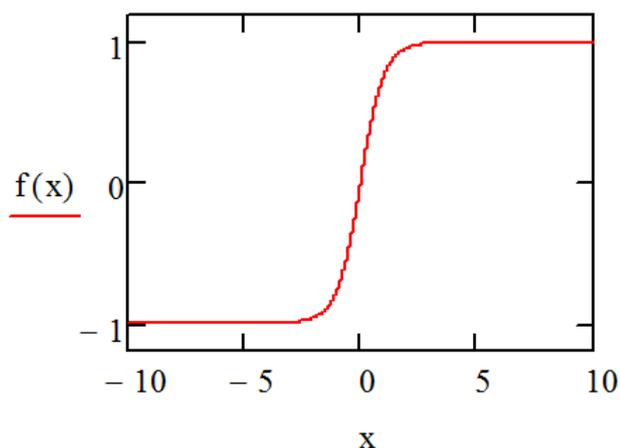


Рисунок 2.2 – Сигмоидная функция

При использовании с увеличением количества слоев нейронов и количества нейронов увеличивается вычислительные возможности нейронной сети. Пример архитектуры с использованием нескольких слоев представлен на рисунке 2.5.

Построение регрессионной модели с помощью нейронной сети заключается в поиске таких значений набора всех весовых коэффициентов \mathbf{W} , при которых нейронная сеть на своем выходе выдавала максимально точный сигнал зависимой переменной y_i на всей обучающей выборке $D^m = (x_1, y_1), \dots, (x_m, y_m)$, $i = 1, 2, \dots, m$.

Способов настройки весовых коэффициентов существует несколько, среди которых наиболее известные метод Левенберга-Маркара, метод Байесовской регуляризации и метод масштабируемых сопряженных градиентов.

Метод Левенберга-Маркара работает следующим образом. Сначала все весовые коэффициенты задаются случайным образом. Затем на каждой

итерации производится корректировка весовых коэффициентов на величину Δw . Изменения в регрессионной модели (функции $f(w, x_n)$) на каждой итерации будут выглядеть так:

$$f(w + \Delta w, x) \approx f(w, x) + J \Delta w, \quad (2.9)$$

где J – якобиан функции $f(w, x_n)$ в точке w , представленная матрицей $N \times R$:

$$J = \begin{bmatrix} \frac{\partial f(w, x_1)}{\partial w_1} & \dots & \frac{\partial f(w, x_1)}{\partial w_R} \\ \dots & \dots & \dots \\ \frac{\partial f(w, x_N)}{\partial w_1} & \dots & \frac{\partial f(w, x_N)}{\partial w_R} \end{bmatrix} \quad (2.10)$$

В данном случае вектор параметров $w = [w_1, \dots, w_R]^T$.

От метода требуется на каждой итерации определение таких значений Δw , которые обеспечивают наименьшую E сумму квадратов ошибок на обучающей выборке.

В данном методе это достигается путем решения системы линейных уравнений:

$$\Delta w = (J^T J + \lambda I)^{-1} J^T (y - f(w)), \quad (2.11)$$

где $\lambda \geq 0$ – параметр регуляризации, который изменяется на каждой итерации метода, а I – единичная матрица

Итерации по уточнению весовых коэффициентов продолжается до тех пор пока Δw выше заданного значения.

В методе Байесовской регуляризации задача нахождения весовых коэффициентов формулируется как минимизация среднеквадратичной ошибки по параметрам нейронной сети. Для обеспечения решения вводится функция вида $\Omega(w)$. Она отвечает за предпочтительность выбора решения $h(x)$. Таким образом задача рассматривается как минимизация ошибки:

$$F = \sum_{i=1}^L (y_i - h(x_i))^2 + \lambda \Omega(h), \quad (2.11)$$

где λ – множитель отвечающий за масштабирование значимости $\Omega(h)$ относительно F

При этом выбор решения h ведется в рамках модели H , задающей ограничение на его вид. Вероятность выбора решения h для обучающей выборки D рассчитывается по формуле Байеса:

$$P(h | D, H) = \frac{P(D | h, H)P(h | H)}{P(D | H)} \quad (2.12)$$

Алгоритм работы метода масштабируемых сопряженных градиентов следующий:

1. $K=0$. Инициализация матрицы весовых коэффициентов W . Расчет градиента $G = \text{grad } E(W)$ и вектора направления $p_n = -\frac{G}{\|G\|}$. E – ошибка работы нейронной сети на обучающей выборке.

2. Поиск значения α , обеспечивающего минимизацию ошибки $E(W_0 + \alpha p)$. Корректировка значений W : $W(K + 1) = W(K) + \alpha p(K)$

3. Если достигнута требуемое точность работы нейронной сети, то остановка алгоритма и иначе рассчитать $G(k + 1) = \text{grad } E(W(k + 1))$

4. Вычислить β и вектор направления P_{k+1} :

$$\beta = \frac{G(K + 1)^T G(K + 1)}{G(K)^T G(K)}; P_{k+1} = -\frac{G(k + 1) + \beta p(k)}{\| -G(k + 1) + \beta p(k) \|}$$

5. Присвоить $p(k)$ значением $p(k + 1)$. Присвоить $G(k)$ значением $G(k + 1)$. Вернуться на первый пункт.

Стоит отметить, что заранее нельзя сказать, какой метод настройки нейронной сети даст наилучшие результаты.

2.2 Источник диагностических признаков контактной сварки

Основная часть контактной сварочной машины – трансформатор, который запитывается от двух фаз трехфазного источника питания. Таким образом, напряжение на первичной обмотке трансформатора имеет синусоидальную форму с действующим значением равным 380В.

Трансформатор преобразует данное напряжение в значения порядка нескольких вольт со стороны вторичного контура (как раз, где расположены электроды). При этом сварочный ток во вторичном контуре (сварочный ток) благодаря трансформатору может достигать значений в десятки кА.

Для управления действующим значением сварочного тока в разрыв первичного контура установлен управляемый контактор. Он может находиться в двух состояниях: выключенном, тогда напряжение питания не подается на сварочный трансформатор и включенном, тогда трансформатор включен в цепь.

Во включенное состояние контактор переводится с помощью системы управления сваркой. Если требуется обеспечение малых значений сварочного тока, то контактор переводится во включенное состояние с большей задержкой α . И наоборот, если требуется большие значения сварочного тока, то задержка α устанавливается небольшой. Задержка всегда α отсчитывается от момента перехода сетевого напряжения через ноль.

Примечательным с точки зрения диагностики является следующая особенность работы контактора – он остается включенным до тех пор, пока через него течет ток. Когда мгновенное значение тока равно нулю – контактор выключается. Длительность λ включенного состояния контактора является неуправляемым параметром, который зависит от сопротивления R_{Σ} на участке между электродами (рисунок 2.3). Это сопротивление R_{Σ} в процессе сварки все время меняется, и эти изменения обусловлены металлургическими в зоне сварки – нагревом и плавлением металла (рисунок 2.4).

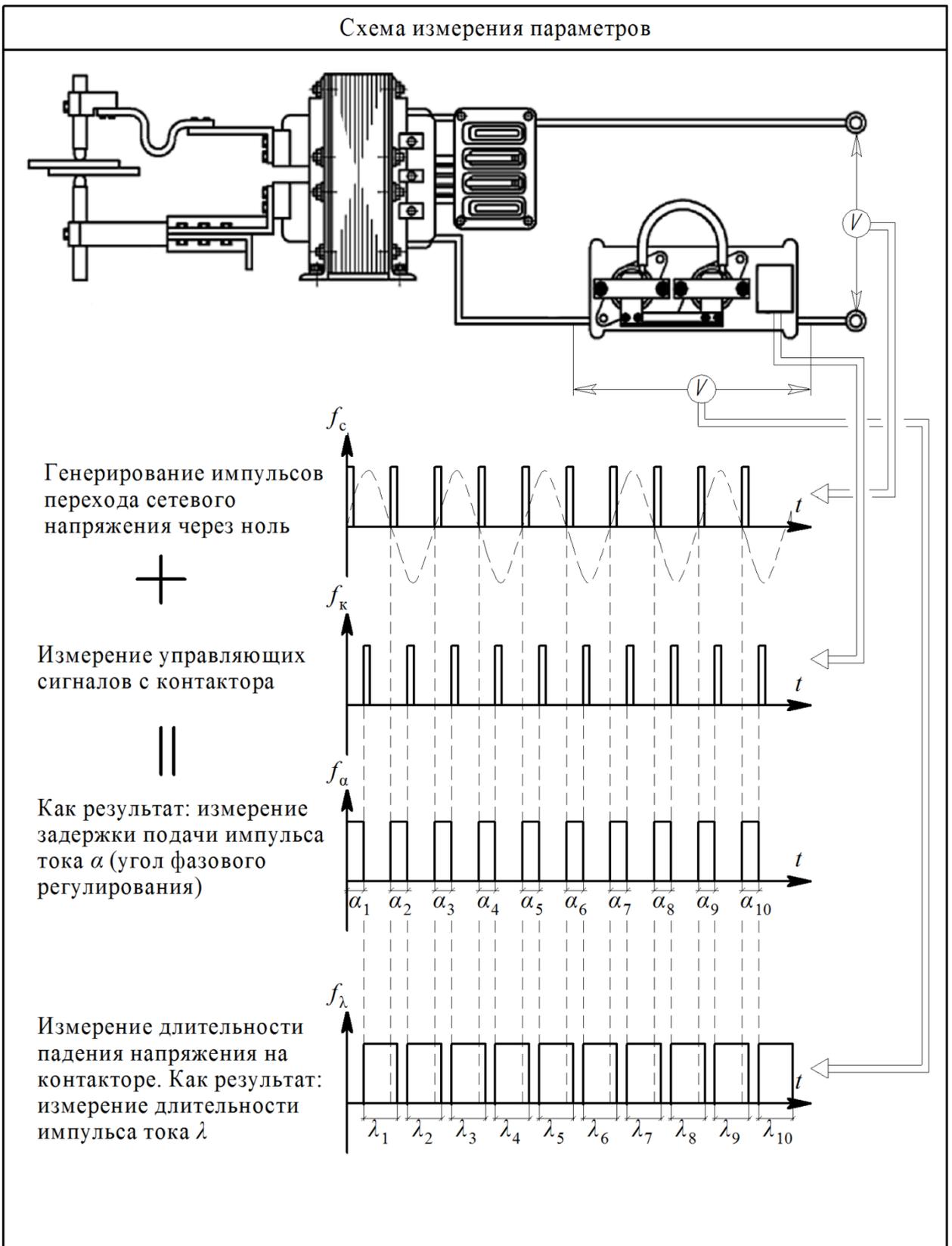


Рисунок 2.3 – Формирование диагностических сигналов

Таким образом, интересным для диагностики сварки является анализ параметров α и λ , которые для каждого экземпляра сварки представлены в виде векторов $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_m)$ и $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$. Здесь m – количество периодов сварки (периодов сетевого напряжения, в течение которых проводилась сварка).

Для того чтобы сократить размерность анализируемых данных (вместо двух векторов значений использовать один) с помощью α и λ для каждого периода сварки удобно рассчитывать коэффициент мощности $\cos(\varphi)$ сварки с использованием равенства:

$$\sin(\lambda + \alpha - \varphi) / \sin(\alpha - \varphi) = e^{-\lambda \cdot \text{ctg} \varphi} \quad (2.13)$$

Таким образом, каждая сварка будет описываться набором значений $\cos(\varphi)$. Но для диагностики сварки мало интересны абсолютные значения $\cos(\varphi)$, большей значимостью обладает относительное динамическое изменения $\cos(\varphi)$ в процессе сварки.

Поэтому набор значения $\cos(\varphi)$ для каждой сварки предложено нормировать к диапазону $[0; 1]$ (или, если в %, то к диапазону $[0\%; 100\%]$).

Таким образом, для каждой сварки предложено получать вектор \mathbf{P} , содержащий в себе нормированные в масштабах одной сварки значения $\cos(\varphi)$ (рисунок 2.4).

Планируется использовать векторы \mathbf{P} для построения регрессионной модели (в виде нейронной сети) предсказывающей по этим значениям прочность соединений.

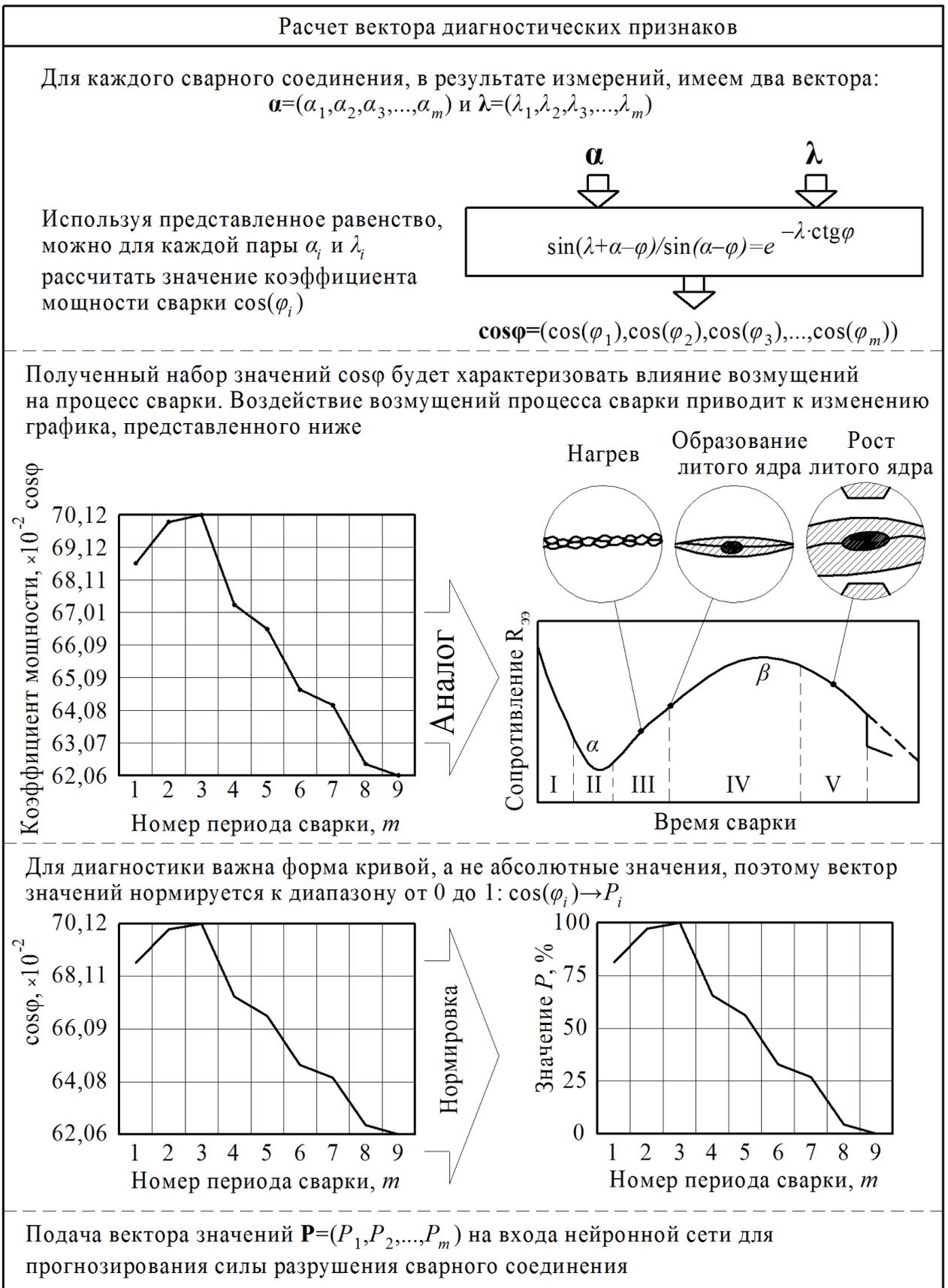


Рисунок 2.4 – Формирование вектора \mathbf{P} входных сигналов нейронной сети

2.3 Диагностика сварки с использованием нейронных сетей прямого распространения

Для того чтобы обучить нейронную сеть проводится экспериментальная сварка заготовок в режиме стабилизации тока на различных значениях сварочного тока в диапазоне $\pm 50\%$ от рекомендуемого значения тока сварки. При этом для каждого периода сварки измеряются характеристики импульса тока - фазовая задержка подачи импульса и длительность его протекания, по которым рассчитывают коэффициент мощности сварки $\cos\varphi$.

Таким образом, для каждой экспериментальной сварки получают вектор значений $\cos\varphi$ (количество компонентов вектора соответствует количеству периодов сетевого напряжения, при котором подавались импульсы тока). Вектор $\cos\varphi$ от каждой экспериментальной сварки нормируется путем линейного преобразования к диапазону от 0% до 100% (где 0% соответствует наименьшему значению компоненты в векторе $\cos\varphi$, а 100% - соответствует наибольшему значению). Затем полученные в результате экспериментальной сварки соединения исследуются методами контроля, необходимыми для определения соответствия сварки выбранным критериям качества. После исследования всех экспериментальных сварок получают массив данных, состоящий для каждой экспериментальной сварки из: вектора нормированных значений $\cos\varphi$ вида (P_1, P_2, \dots, P_n) , где количество компонентов n равно количеству периодов сварки и значения прочностной характеристики соединения.

Затем полученный массив данных для обучения нейронной сети. Обученная нейронная сеть – регрессионная модель прогнозирования прочности соединения в зависимости от значений компонентов P_1, P_2, \dots, P_n .

Архитектура используемой нейронной сети представлена на рисунке 2.5.

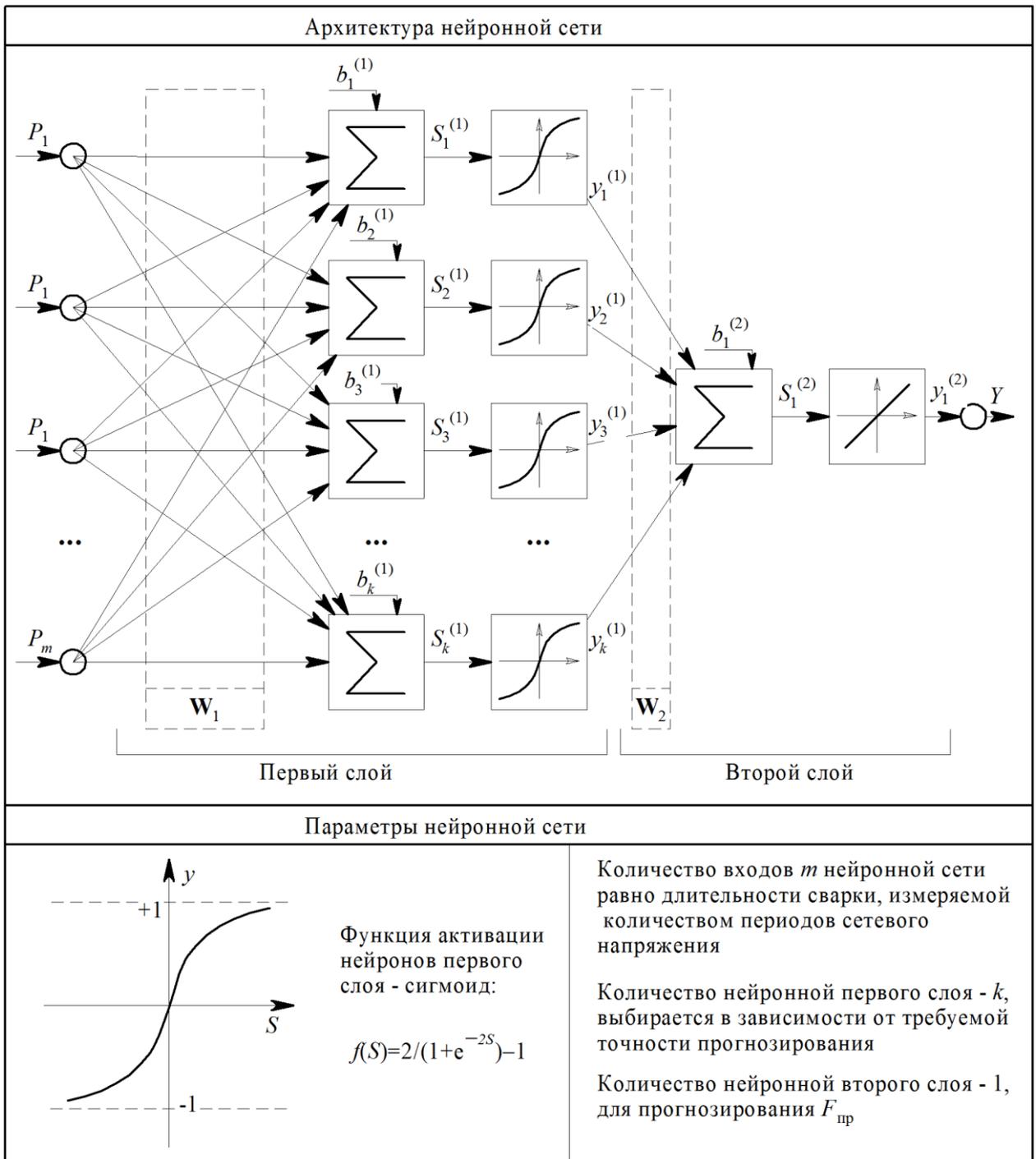


Рисунок 2.5 – Архитектура нейронной сети для прогнозирования прочности соединения.

ГЛАВА 3 ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ ПРЕДЛОЖЕННЫХ РЕШЕНИЙ

3.1 Программная реализация алгоритма

В ходе выполнения исследований по выпускной квалификационной работе в интегрированной среде разработки Qt Creator на языке программирования C++ была спроектирована, разработана и протестирована программа для построения регрессионных моделей в виде нейронных сетей прямого распространения. Программа обладает следующими характеристиками:

- программа реализована в виде консольного приложения;
- поддерживаемы архитектуры нейронных сетей – однослойные и многослойные нейронные сети прямого распространения;
- поддерживаемые функции активации – сладкие сигмовидные функции;
- возможность задания количества слоев нейронов – до 10 слоев;
- возможность задания количество нейронов в каждом слое – до 255 нейронов;
- функция автоматического импорта обучающей выборки из txt-файла (с определением размерности вектора входных сигналов);
- запись процесса поите рационального процесса обучения нейронной сети в текстовый файл;
- поддерживаемый метод обучения сети – метод обратного распространения ошибки;
- сохранения параметров нейронной сети в текстовый файл (смещение нейронов, весовые коэффициенты каждого слоя).

Программный код представлен в приложении А.

Чтобы запустить процесс обучения нейронной сети необходимо запустить командную строку, перейти в папку, содержащую исполняемый файл программы и ввести команду training. При этом в качестве параметра необходимо указать до текстового файла с параметрами обучения и путь до текстового файла с результатами обучения нейронной сети. Например: «training recordset.txt >temp.txt» (рисунок 3.1).

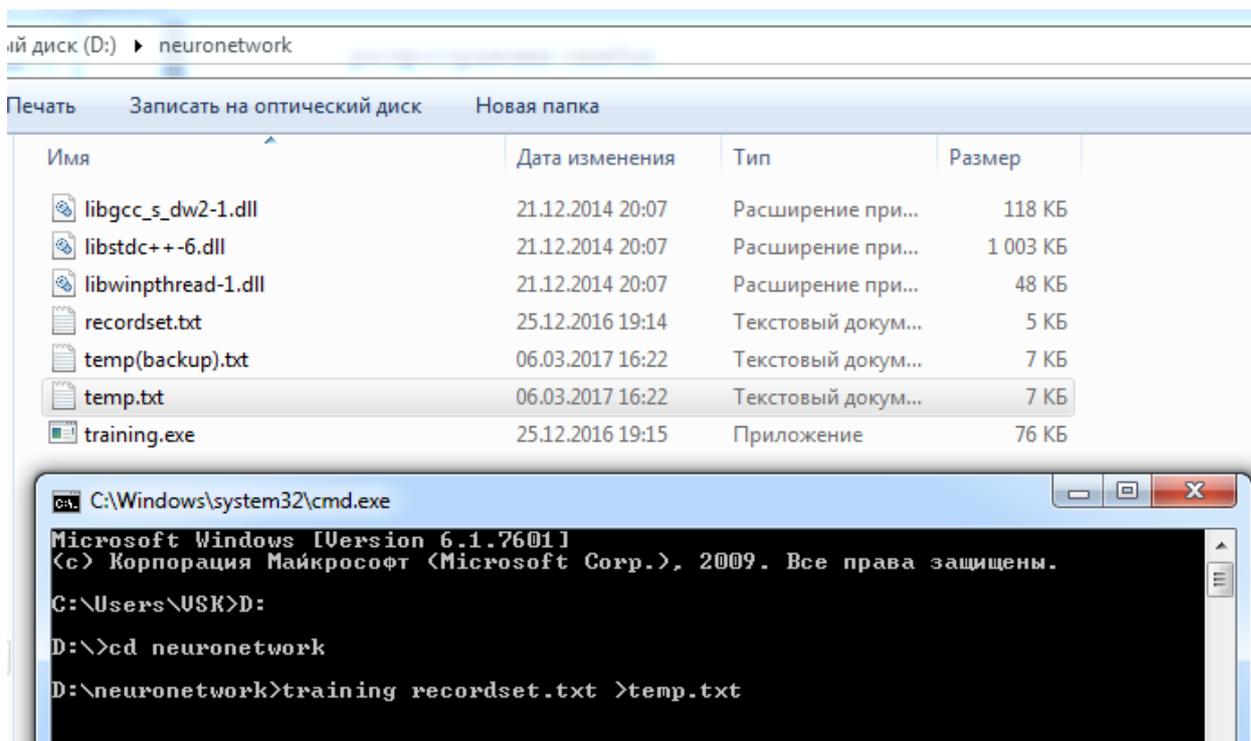


Рисунок 3.1 – Запуск обучения нейронной сети

Формат текстового файла, содержащего параметры обучения нейронной сети (рисунок 3.2):

1. Скорость обучения.
2. Пробел либо табуляция.
3. Относительная ошибка обучения при достижении которой процесс обучения останавливается.
4. Пробелы, табуляции либо переводы строки.
5. Количество входов нейронной сети.
6. Пробелы, табуляции либо переводы строки.

7. Список количества нейронов на каждом из слоёв. Последний слой не указывается, его размер неявно равен единице.

8. Пробелы, табуляции либо переводы строки.

9. Описание обучающей выборки. Записи разделены новой строкой, числа в записи - табуляцией либо пробелом. Нецелые числа используют точку для отделения целой и дробной частей. Последнее число в строке - ожидаемое выходное значение нейронной сети.

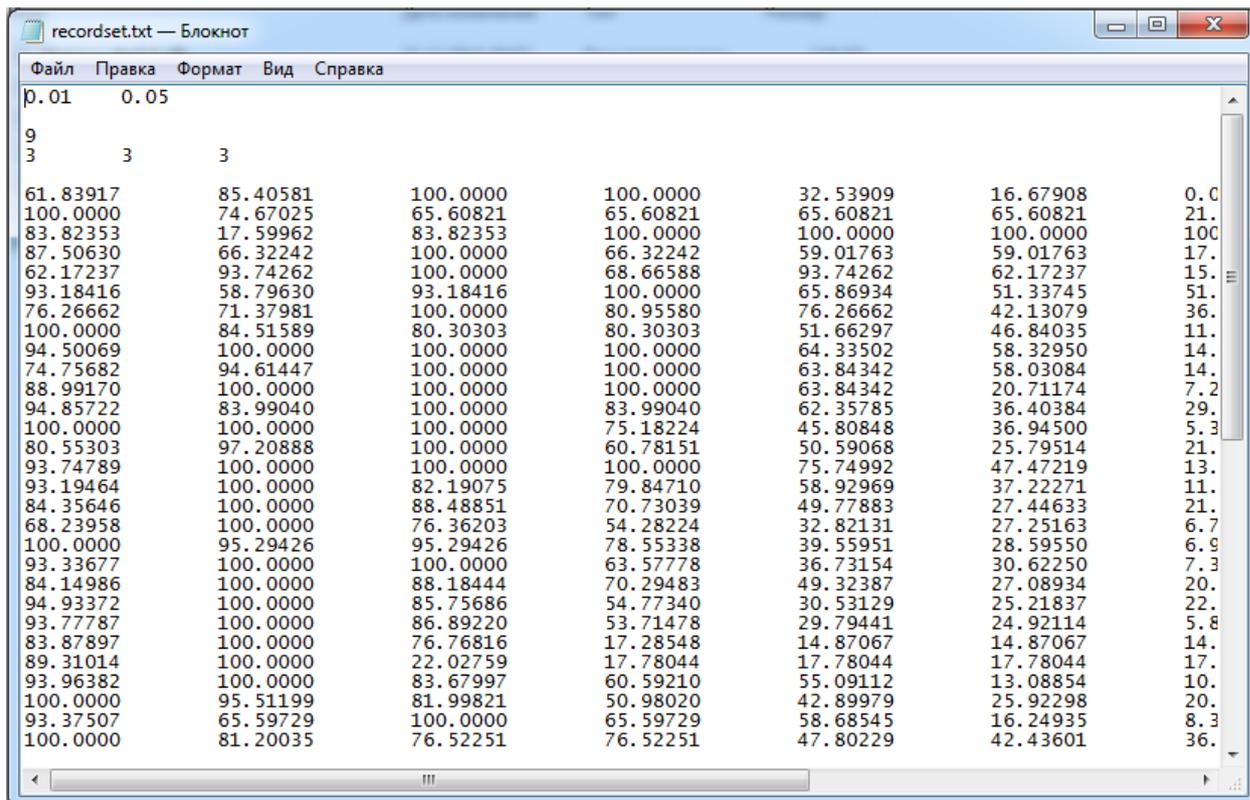
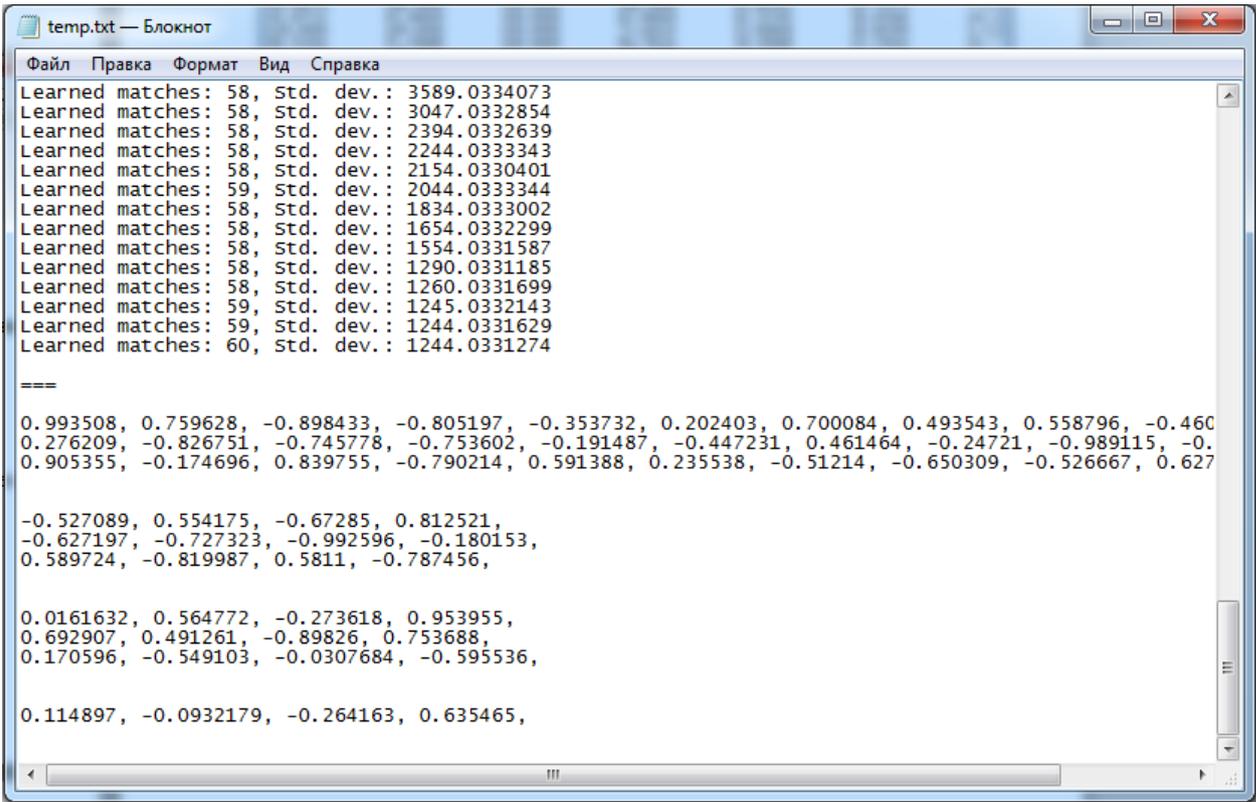


Рисунок 3.2 – Содержание файла с параметрами обучения сети

Формат текстового файла, содержащего результаты обучения нейронной сети (рисунок 3.3):

1. Ошибка MSE для каждой итерации обучения нейронной сети.
2. После разделителя «===» приведены результаты обкчения нейронной сети в следующем формате. На первой строчке смещение b и значение весовых коэффициентов для входов первого нейрона, на второй строчке смещение b и значение весовых коэффициентов для входов второго

нейрона и т.д. Пустая строка указывает на то, что далее будут идти параметры нейронов второго слоя и т.д.



```
temp.txt — Блокнот
Файл  Правка  Формат  Вид  Справка
Learned matches: 58, Std. dev.: 3589.0334073
Learned matches: 58, Std. dev.: 3047.0332854
Learned matches: 58, Std. dev.: 2394.0332639
Learned matches: 58, Std. dev.: 2244.0333343
Learned matches: 58, Std. dev.: 2154.0330401
Learned matches: 59, Std. dev.: 2044.0333344
Learned matches: 58, Std. dev.: 1834.0333002
Learned matches: 58, Std. dev.: 1654.0332299
Learned matches: 58, Std. dev.: 1554.0331587
Learned matches: 58, Std. dev.: 1290.0331185
Learned matches: 58, Std. dev.: 1260.0331699
Learned matches: 59, Std. dev.: 1245.0332143
Learned matches: 59, Std. dev.: 1244.0331629
Learned matches: 60, Std. dev.: 1244.0331274

===

0.993508, 0.759628, -0.898433, -0.805197, -0.353732, 0.202403, 0.700084, 0.493543, 0.558796, -0.460
0.276209, -0.826751, -0.745778, -0.753602, -0.191487, -0.447231, 0.461464, -0.24721, -0.989115, -0.
0.905355, -0.174696, 0.839755, -0.790214, 0.591388, 0.235538, -0.51214, -0.650309, -0.526667, 0.627

-0.527089, 0.554175, -0.67285, 0.812521,
-0.627197, -0.727323, -0.992596, -0.180153,
0.589724, -0.819987, 0.5811, -0.787456,

0.0161632, 0.564772, -0.273618, 0.953955,
0.692907, 0.491261, -0.89826, 0.753688,
0.170596, -0.549103, -0.0307684, -0.595536,

0.114897, -0.0932179, -0.264163, 0.635465,
```

Рисунок 3.3 – Содержание файла с результатами обучения нейронной сети

3.2 Апробация предложенных подходов и исследование точности получаемых регрессионных моделей

Экспериментальная сварка проводилась на пакете заготовок из стали 08кп толщиной 0,8+0,8 мм. Ток сварки выбирался из диапазона 5,5 кА до 11 кА с шагом 0,5 кА. Длительность сварки – 0,18 с. Сварка в режиме стабилизации тока. Сварка проводилась под действием различных возмущений. Всего экспериментальных сварок – 118 штук.

Формирование обучающей выборки проводилось согласно описанию в предыдущей главе (рисунок 2.4).

Архитектура нейронной сети, с помощью которой анализировалась обучающая выборка, представлена на рисунке 2.5.

Однако необходимо выбрать количество нейронов k в первом слое.

Для поиска оптимального количества k нейронов в первом слое было несколько сотен запусков обучения нейронов сети на одной и той же обучающей выборке, при этом оценивалась ошибка MSE работы обученной нейронной сети

$$MSE = \frac{1}{n} \sum_{i=1}^n (p_i - a_i)^2, \quad (3.1)$$

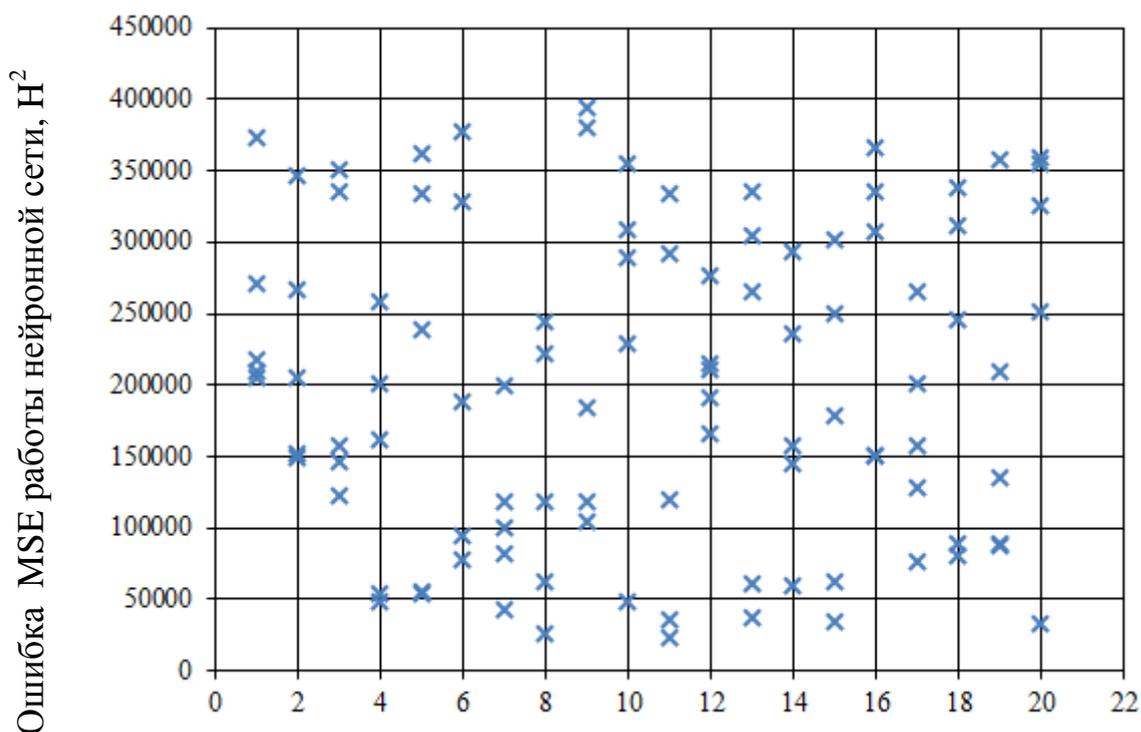
где a_i – фактические значения выходного параметра для i -ого объекта тестовой выборки, p_i – выходное значение, выдаваемое регрессионной моделью для входных параметров i -ого объекта тестовой выборки, n – количество объектов тестовой выборки.

При обучении критерии остановки настройки нейронной сети являлись:

- не возможность улучшения результатов (снижения суммы квадратов ошибок) на следующей итерации;
- или достижение количества итерации значения равному 500.

Обучение проводилось методом обратного распространения ошибки.

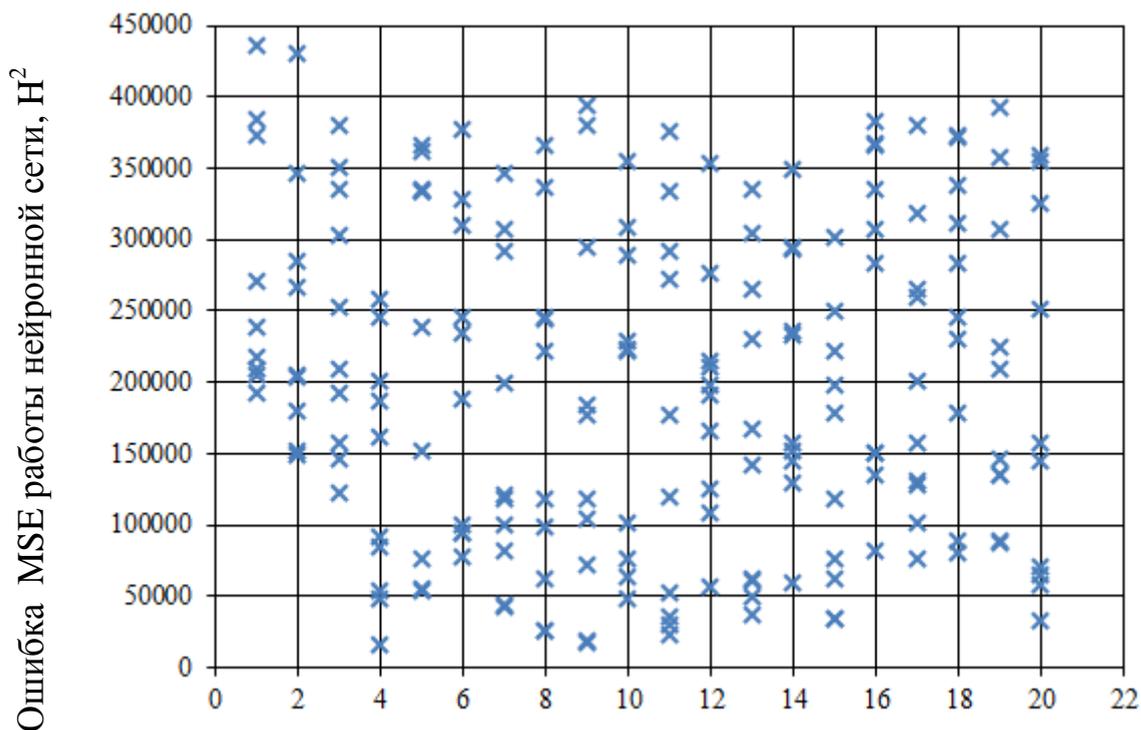
Результаты вычислительных экспериментов для первых 100 запусков приравлены на рисунке 3.4.



Количество нейронов k в первом слое, шт.

Рисунок 3.4 – Результаты 100 экспериментов обучения нейронной сети

Как видно из рисунка 3.4, никакой закономерности между точностью работы нейронной сети и количеством k нейронов в первом слое не наблюдается. Попробуем провести еще 100 экспериментов обучения нейронной сети (рисунок 3.5).



Количество нейронов k в первом слое, шт.

Рисунок 3.5 – Результаты 200 экспериментов обучения нейронной сети

Из рисунка 3.5 видно, что для построения регрессионной модели на основе имеющихся данных по сварке количество нейронов в первом слое следует выбирать не менее 4. Так как в противном случае не удастся добиться снижения ошибки MSE ниже значения 120000 Н².

Стохастичность результатов обучения нейронной сети связана со случайным выбором первоначальных значений весовых коэффициентов (которые на каждой следующей итерации уточняются). Выбор фиксированных значений весовых коэффициентов перед началом каждого

обучения не является выходом, так как в этом случае возможен эффект постоянного попадания в локальный минимум. Это приведет к тому, что не удастся достигнуть более точной работы нейронной сети, которая могла бы быть при других начальных значениях весовых коэффициентов.

Итак, когда мы убедились, на собственном опыте, в стохастичности результатов обучения нейронной сети было принято решение провести по 1000 запусков обучения нейронной сети для каждого значения k . По результатам данного длительного эксперимента были отобраны те нейронные сети, которые обеспечивали наименьшую ошибку MSE:

- 18 нейронов, MSE=4.64e-22;
- 16 нейронов MSE=2.37e-24;
- 12 нейронов MSE=1.93e-22;
- 11 нейронов MSE=3.31e-4;
- 10 нейронов MSE=6.6e-24 (приложение Б);
- 6 нейронов MSE=255.7;
- 5 нейронов MSE=4253;
- 4 нейрона MSE=11322.

Параметры данных нейронных сетей сохранены для дальнейшего использования при диагностике сварки.

ЗАКЛЮЧЕНИЕ

Основные выводы по проведенным в ВКР исследованиям:

1. На основе изучения научных статей по теме исследования установлено, что для интеллектуальной диагностики сварки применяются нейронные сети. Их использование позволяет установить зависимость между электрическими процессами сварки и получаемыми прочностными характеристиками.

2. Показано, что диагностику сварки можно представить, как задачу регрессии по восстановлению зависимости между параметрами. При этом входными параметрами регрессионной модели являются измеряемые характеристики процесса сварки, а выходным – прогнозируемые прочностные характеристики.

3. Доказано, что задача восстановления зависимости между измеряемыми и прогнозируемыми параметрами может быть решена с использованием нейронных сетей. Параметры обученной нейронной сети прямого распространения направленной на прогнозирование прочности сварки представлены в приложении Б. При этом входными сигналами для нейронной сети являются нормированные значения коэффициента мощности сварки $\cos\varphi$, который рассчитывается для каждого i -го периода сварки. Таким образом, каждая сварка описывается вектором $P = (P_1, P_2, \dots, P_i)$. Выходным сигналом является прочность сварного соединения.

4. С использованием технологического оборудования лаборатории “Контактная сварка” института машиностроения ТГУ было проведено около 100 экспериментальных сварок пакета заготовок из стали 08кп. Использовались в роли обучающей выборки для обучения нейронной сети.

При вычислительных экспериментах по обучению нейронных сетей было подтверждено, что результаты обучения носят стохастический характер.

Однако количество нейронов в первом слое следует выбирать не менее 4. Так как в противном случае не удастся добиться снижения ошибки MSE прогнозирования силы разрушения соединения ниже значения 120000 Н^2 .

5. В ходе проведения множественных экспериментов по обучению нейронных сетей удалось добиться следующей точности работы при различном количестве нейронов в первом слое:

6. 18 нейронов, $MSE=4.64e-22$; 16 нейронов $MSE=2.37e-24$; 12 нейронов $MSE=1.93e-22$; 11 нейронов $MSE=3.31e-4$; 10 нейронов $MSE=6.6e-24$ (параметры сети приведены в приложении Б); 6 нейронов $MSE=255.7$; 5 нейронов $MSE=4253$; 4 нейрона $MSE=11322$.

7. С использованием свободно-распространяемой среды разработки Qt Creator на языке программирования C++ разработано программное обеспечение, на практике реализующее предложенные подходы по построению регрессионных моделей для диагностики качества сварки с использованием нейронных сетей.

8. Исследования, проведенные в рамках данной работы, будут использованы при выполнении гранта РФФИ на тему “Управление металлургическими процессами с использованием искусственного интеллекта”.

9. Получен акт о внедрении исследований на предприятии ПАО «АвтоВАЗ»

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Iatan, F.I. Issues in the Use of Neural Networks in Information Retrieval / F.I. Iatan. – Springer International Publishing, 2017 – 199 p.
2. Silva, I.N. Artificial Neural Networks: A Practicle course / Ivan Nunes da Silva, Danilo Hernane Spatti, Rogerio Andrade Flauzino, Luisa Helena Bartocci Liboni, Silas Franco dos Reis Alves. – Springer International Publishing, 2017 – 307 p.
3. Kong, X. Principal Component Analysis Networks and Algorithms / Xiangyu Kong, Changhua Hu, Zhansheng Duan. – Springer Singapore, 2017 – 323 p.
4. Laurinen, P. Studying the Quality of Resistance Spot Welding Joints Using Bayesian Networks / P. Laurinen, H. Junno, L. Tuovinen, J. Röning // Proceedings of Artificial Intelligence and Applications. – 2004. – С. 705–711.
5. Aravinthan, A. A neural network system for spot weld strength prediction / A. Aravinthan, K. Sivayoganathan, D. Al-Dabass, V. Balendran // UKSIM2001: Conference Proceedings of the UK Simulation Society. - 2001. - P. 156-160
6. Патон, Б.Е. Оценка качества контактной точечной сварки с помощью нейронных сетей / Б.Е. Патон, Н.В. Подола, В.С. Гавриш // Автоматическая сварка. – 1998. – № 12. – С. 3–10.
7. El-Banna, M. Online qualitative nugget classification by using a linear vector quantization neural network for resistance spot welding / M. El-Banna, D. Filev, R.B. Chinnam // The International Journal of Advanced Manufacturing Technology. - 2008. - №36. - P. 237-248.
8. Park, Y.J. Quality evaluation by classification of electrode force patterns in the resistance spot welding process using neural networks / Y.J. Park, H. Cho // Proceedings of the Institution of Mechanical Engineers B, Journal of Engineering Manufacture. – 2004. – С. 151–162.
9. Börgers, Ch. An Introduction to Modeling Neuronal Dynamics / Christoph Börgers. – Springer International Publishing, 2017 – 457 p.

10. Kowaliw, T. Growing Adaptive Machines: Combining Development and Learning in Artificial Neural Networks / Taras Kowaliw, Nicolas Bredeche, René Doursat. – Springer Berlin Heidelberg, 2014 – 261 p.
11. Siddique, N. Intelligent Control: A Hybrid Approach Based on Fuzzy Logic, Neural Networks and Genetic Algorithms / Nazmul Siddique. – Springer International Publishing, 2014 – 282 p.
12. Ławryńczuk, M. Computationally Efficient Model Predictive Control Algorithms: A Neural Network Approach / Maciej Ławryńczuk. – Springer International Publishing, 2014 – 316 p.
13. Krawczak, M. Multilayer Neural Networks: A Generalized Net Perspective / Maciej Krawczak. – Springer International Publishing, 2013 – 182 p.
14. Liu, J. Radial Basis Function (RBF) Neural Network Control for Mechanical Systems: Design, Analysis and Matlab Simulation. – Springer Berlin Heidelberg, 2014 – 365 p.
15. Hirose, A. Complex-Valued Neural Networks / Akira Hirose. – Springer Berlin Heidelberg, 2012 – 175 p.
16. Terzic, E. A Neural Network Approach to Fluid Quantity Measurement in Dynamic Environments / Edin Terzic, Jenny Terzic, Romesh Nagarajah, Muhammad Alamgir. – Springer London, 2012 – 138 p.
17. Anastassiou, G. A. Intelligent Systems: Approximation by Artificial Neural Networks / George A. Anastassiou. – Springer Berlin Heidelberg, 2011 – 107 p.
18. Luo, C. J. Mathematical Modeling and Applications in Nonlinear Dynamics / Albert C.J. Luo, Hüseyin Merdan. – Springer International Publishing, 2016 – 205 p.
19. Melin, P. Design of Intelligent Systems Based on Fuzzy Logic, Neural Networks and Nature-Inspired Optimization / Patricia Melin, Oscar Castillo, Janusz Kacprzyk. – Springer International Publishing, 2015. – 637 p.
20. Advances in Neural Networks: Computational and Theoretical Issues

21. Tripathi, B. K. High Dimensional Neurocomputing / Bipin Kumar Tripathi. – Springer India, 2015. – 165 p.

22. Du, K. L. Neural Networks and Statistical Learning / Ke-Lin Du, M. N. S. Swamy. – Springer London, 2014. – 824 p.

23. Mrugalski, M. Advanced Neural Network-Based Computational Schemes for Robust Fault Diagnosis / Marcin Mrugalski. – Springer International Publishing, 2014. – 181 p.

24. Shanmuganathan, S. Artificial Neural Network Modelling / Subana Shanmuganathan, Sandhya Samarasinghe. – Springer International Publishing, 2016. – 472 p.

Программный код

```
/* INCLUDES *****/

#include <algorithm>
#include <array>
#include <cassert>
#include <cmath>
#include <chrono>
#include <cstdlib>
#include <fstream>
#include <iostream>
#include <map>
#include <numeric>
#include <random>
#include <vector>

/* DECLARATIONS *****/

struct Neuron;

struct Input
{
    double weight;
    Neuron* source;
};

struct Neuron
{
    double baseWeight;

    std::vector<Input> inputs;
    double output;

    void process()
    {
        double accumulator = baseWeight;
        for(const Input& input : inputs)
            accumulator += (input.source->output * input.weight);

        output = std::tanh(accumulator);
    }

    double derivative()
```

```

    {
        double accumulator = baseWeight;
        for(const Input& input : inputs)
            accumulator += (input.source->output * input.weight);

        const double a = 1.0 / std::cosh(accumulator);
        return a * a;
    }
};

using layer_t = std::vector<Neuron>;
using network_t = std::vector<layer_t>;

struct ReferenceRecord
{
    std::vector<double> inputs;
    double expectedOutput;
};

// =====

static std::vector<size_t> perLayerNeuronsCount;
static std::vector<ReferenceRecord> reference;

static double LearnSpeed;
static double ResultEpsilon;

/* PUBLIC FUNCTIONS *****/

void readReferenceData(std::istream& in)
{
    in >> LearnSpeed;
    in >> ResultEpsilon;
    while(!std::isdigit(in.peek()))
        in.get();

    size_t inputsCount;
    in >> inputsCount;
    perLayerNeuronsCount.push_back(inputsCount);

    while(!std::isdigit(in.peek()))
        in.get();

    while(in.peek() != '\n')

```

```

    {
        double count;
        in >> count;
        perLayerNeuronsCount.push_back(count);
    }
    perLayerNeuronsCount.push_back(1); // The last layer always has the only
neuron

while(!std::isdigit(in.peek()))
    in.get();

while(in.peek() != std::char_traits<char>::eof())
{
    ReferenceRecord record;
    record.inputs.resize(inputsCount);

    for(size_t i = 0; i < inputsCount; i++)
    {
        double value;
        in >> value;
        record.inputs[i] = value;
    }

    double value;
    in >> value;
    record.expectedOutput = value;

    reference.push_back(record);
}
}

void normalizeReferenceData()
{
    double minIn = reference[0].inputs[0];
    double maxIn = reference[0].inputs[0];

    double minOut = reference[0].expectedOutput;
    double maxOut = reference[0].expectedOutput;

    for(auto& record : reference)
    {
        for(double& input : record.inputs)
        {
            minIn = std::min(minIn, input);
            maxIn = std::max(maxIn, input);

```

```

    }

    minOut = std::min(minOut, record.expectedOutput);
    maxOut = std::max(maxOut, record.expectedOutput);
}

// Original:
//
// const double aIn = -minIn;
// const double kIn = 1 / (maxIn - minIn);
// input = (input + aIn) * kIn * 2 - 1; // To [-1; 1] range
//
// => input = (input - minIn) * (1 / (maxIn - minIn)) * 2 - 1
// => input = (input - minIn) / (maxIn - minIn) * 2 - 1
//
//          -----
//          aIn      kIn

const double aIn = -minIn;
const double kIn = 2.0 / (maxIn - minIn);

const double aOut = -minOut;
const double kOut = 2.0 / (maxOut - minOut);

for(auto& record : reference)
{
    for(double& input : record.inputs)
        input = (input + aIn) * kIn - 1.0;

    record.expectedOutput = (record.expectedOutput + aOut) * kOut -
1.0;
}
}

static void printNumberCell(double num, std::ostream& out)
{
    if(num >= 0.0)
    {
        out << ' ';
        out.width(9);
    }
    else
        out.width(10);

    out << num;
}

```

```

static void printNormalizedReferenceData()
{
    for(const auto& record : reference)
    {
        for(const double& input : record.inputs)
            printNumberCell(input, std::cout);

        std::cout << " | ";
        printNumberCell(record.expectedOutput, std::cout);
        std::cout << '\n';
    }
}

static network_t doLearning()
{
    const unsigned seed =
std::chrono::system_clock::now().time_since_epoch().count();
    std::default_random_engine generator(seed);
    std::uniform_real_distribution<double> distribution(-1.0, 1.0);
    const auto getRandom = [&distribution, &generator]() {return
distribution(generator);};

    normalizeReferenceData();

    network_t network(perLayerNeuronsCount.size());
    network[0].resize(perLayerNeuronsCount[0]);
    for(size_t layerId = 1; layerId < perLayerNeuronsCount.size(); layerId++)
    {
        network[layerId].resize(perLayerNeuronsCount[layerId]);

        // By the way, fill random coefficients
        for(Neuron& neuron : network[layerId])
        {
            neuron.baseWeight = getRandom();

            neuron.inputs.resize(perLayerNeuronsCount[layerId - 1]);
            for(size_t i = 0; i < perLayerNeuronsCount[layerId - 1]; i++)
            {
                neuron.inputs[i].weight = getRandom();
                neuron.inputs[i].source = &network[layerId - 1][i];
            }
        }
    }
}

```

```

// Do learning. Each pass uses the whole reference records sequentially
bool continueLearning = true;
const size_t originalPrecision = std::cout.precision();
while(continueLearning)
{
    double squaresSum = 0.0, prevSquaresSum =
std::numeric_limits<double>::quiet_NaN();
    size_t learnedMatches = 0;
    for(auto record : reference)
    {
        // Feed the first layer with input values then process into hidden
layers
        for(size_t i = 0; i < record.inputs.size(); i++)
        {
            assert(i < network[0].size());
            network[0][i].output = record.inputs[i];
        }

        // Propagate values forward, layer by layer. Since inputs are just
// pointers, a neuron will automatically take input values from
// the previous layer.
        for(size_t i = 1; i < network.size(); i++)
        {
            for(Neuron& neuron : network[i])
                neuron.process();
        }

        // Test if the network gave the appropriate result
        const double y = network.back()[0].output;
        const double e = record.expectedOutput - y;
        squaresSum += e * e;
        if(std::abs(e) > ResultEpsilon)
        {
            // Do backward error propagation
            std::map<const Neuron*, double> deltas;
            auto it = deltas.find(&network.back()[0]);
            if(it == deltas.end())
                it = deltas.insert({ &network.back()[0], e }).first;

            for(size_t i = network.size() - 1; i > 0; i--)
            {
                for(const Neuron& neuron : network[i])
                {
                    for(const Input& input : neuron.inputs)

```

```

it->second += input.weight *
deltas[&neuron];
    }
}

// Now perform forward coefficients correction
for(size_t i = 1; i < network.size(); i++)
{
    for(Neuron& neuron : network[i])
    {
        for(Input& input : neuron.inputs)
            input.weight += LearnSpeed *
deltas[&neuron] * neuron.derivative() * input.source->output;

        neuron.process();
    }
}
else
    learnedMatches++;
}

if(learnedMatches == reference.size())
    continueLearning = false;

std::cout.precision(0);
std::cout << "Learned matches: ";
std::cout.width(2);
std::cout << learnedMatches << " (";
std::cout.width(3);
std::cout << ((double)learnedMatches * 100.0 / reference.size());
std::cout.precision(originalPrecision);

const double standardDeviation = std::sqrt(squaresSum /
reference.size());
std::cout << "%)\tStd. dev.: " << standardDeviation << "\n";

prevSquaresSum = squaresSum;
}

return network;
}

static void printNeuronsInfo(const network_t& network)
{

```

```

std::cout << "\n===";
for(auto layerIt = std::next(network.begin()); layerIt != network.end();
layerIt++)
{
    std::cout << "\n\n";

    for(const Neuron& neuron : *layerIt)
    {
        std::cout << neuron.baseWeight << ", ";
        for(const Input& input : neuron.inputs)
            std::cout << input.weight << ", ";
        std::cout << '\n';
    }
}

int main(int argc, char *argv[])
{
    if(argc == 2)
    {
        std::ifstream in(argv[1]);
        if(in)
        {
            std::cout.setf(std::ios_base::left, std::ios_base::adjustfield);
            std::cout.setf(std::ios_base::fixed, std::ios_base::floatfield);
            std::cout.precision(4);

            readReferenceData(in);
            normalizeReferenceData();

            std::cout <<
                "Normalized reference data\n"
                "-----\n"
                "\n";
            printNormalizedReferenceData();

            std::cout <<
                "\n"
                "\n"
                "Learning progress\n"
                "-----\n"
                "\n";
            const network_t network = doLearning();

            std::cout <<

```

```

        "\n"
        "\n"
        "Final neuron coefficients\n"
        "-----\n"
        "\n";
printNeuronsInfo(network);

return EXIT_SUCCESS;
}
else
{
    std::cerr << "File not found\n";
    return EXIT_FAILURE;
}
}
else
{
    std::cerr <<
        "USAGE: training FILE\n"
        "\n"
        "  where FILE is a text file with the following layout:\n"
        "\n"
        "    LEARN_SPEED RESULT_EPSILON\n"
        "\n"
        "    INPUTS_COUNT\n"
        "    LAYER1_SIZE [LAYER2_SIZE [...]]\n"
        "\n"
        "    IN1 [IN2 [...]] EXPECTED_OUT\n"
        "    [IN1 [IN2 [...]] EXPECTED_OUT]\n"
        "    [...]\n"
        "\n"
        "    All numbers have a point as a decimal separator.\n";
    return EXIT_FAILURE;
}
}

```

ПРИЛОЖЕНИЕ Б

Параметры нейронной сети

Архитектура представлена на рисунке 2.5. $k=10$. Ошибка работы сети с данными параметрами $MSE=6.6e-24$. На основе вектора входов $X = (x_1, x_2, \dots, x_9)$ нейронная сеть формирует значение Y

Формирования входов x_i нейронной сети на основе значений P_i (из обучающей выборки), где $i=1\dots 9$, с путем следующих вычислений $x_i = (P_i - os_i) \cdot g_i + m$, где

$$os = (61.8391660461652 \ 17.5996204933584 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0),$$

$$G = (0.0524097560975608 \ 0.0242717328727691 \ 0.02 \ 0.02 \ 0.02 \ 0.02 \ 0.02 \ 0.0461627906976744 \ 0.157043832823649);$$

$$m = -1;$$

Смещения b нейронов первого слоя

$$b^{(1)} = (3.0683974048678002 \ -0.27423273112427027 \ 1.5115634033487764 \ -0.38504969485354673 \ -0.47669551895823126 \ -1.1156876063771957 \ -1.3597102456447809 \ -0.19559472186387705 \ 3.6574740285182377 \ 1.7640743723224406);$$

Векторы $W_k^{(1)}$ весовых коэффициентов k -ого нейрона ($k=1\dots 10$):

$$W_1^{(1)} = (-0.068487973865354809 \ -0.54349740533179425 \ -1.1236306325269185 \ -1.6769025835659823 \ -0.80524540559016078 \ -0.48665736284057665 \ -0.19179174597836773 \ -0.78955510466223378 \ -0.69056026667588788);$$

$$W_2^{(1)} = (1.8155412899174841 \ -0.68009695016411686 \ -0.25983000650386467 \ -4.7917322524034871 \ 2.0417580642936879 \ -0.76925458614065956 \ 0.50985190788874701 \ -4.3147826640756275 \ -0.27514472192221184);$$

$$W_3^{(1)} = (0.17308751577672116 \ -0.24753554287718371 \ 1.1818091629421976 \ 1.5369956551476138 \ 1.3562310364947554 \ 0.16243951488522521 \ -0.32547010388098563 \ -0.099790765156405586 \ -1.8655385829258433);$$

$$W_4^{(1)} = (-0.56861810188760598 \ -0.25567741380395081 \ 1.105570560359951 \ 2.6176065591901678 \ -0.071296653803467128 \ 0.69021976600501278 \ -0.40900460620005641 \ 0.92247707227865805 \ 0.24198267368838169);$$

$$W_5^{(1)} = (-2.0692003826072694 \ 3.4062307875723903 \ 1.8433719149300949 \ 2.0567191897368393 \ 2.385564340508783 \ 3.4336979884967116 \ 1.1330931970942011 \ -3.8922409174588459 \ 1.483003136160689);$$

$$W_6^{(1)} = (-0.39579962911579003 \quad 0.85218558372013997 \quad 2.8676572512156273 \\ 5.8574617315920472 \quad -0.78750102960844104 \quad 0.92612949010273604 \\ -2.0925714443060865 \quad 3.9375844736205732 \quad 2.0665562629685557);$$

$$W_7^{(1)} = (-4.5786798257057466 \quad 1.7450593411758353 \quad 1.350441456000711 \\ 0.080332955051055102 \quad 3.3358049188766299 \quad -0.20151490001487718 \\ 1.7768533327918716 \quad -2.5483009583225704 \quad -0.1265727684562212);$$

$$W_8^{(1)} = (-1.2333331191859327 \quad -4.1845238206284145 \quad -2.2034669004940102 \\ 0.84962085099049645 \quad 1.1180801170396448 \quad -4.2084535860658718 \\ -1.2885947338092631 \quad -4.1379851834096 \quad 0.59564852281100145);$$

$$W_9^{(1)} = (0.43062286264241623 \quad -1.3083903968116906 \quad 0.3763445761976823 \quad - \\ 0.57078927793048773 \quad -0.0087744046128371122 \quad 0.304767294567591 \\ 1.9434923790452379 \quad 0.66251937673096872 \quad -0.90073479742963236);$$

$$W_{10}^{(1)} = (-0.79978780715586106 \quad 0.72732245420814501 \quad -1.19668235986862 \\ 1.9605899217752609 \quad -2.9127889972111092 \quad 0.020752434274668299 \\ -1.0860124276905005 \quad -0.012519833441071094 \quad 0.65668730552502452);$$

Смещения b нейронов второго слоя

$$b^{(2)} = -0.84932148535999064;$$

Весовые коэффициенты второго слоя

$$W_1^{(2)} = (-1.5510974514542455 \quad 0.13045834137880297 \quad -0.25129865830526538 \\ -2.2127274865211977 \quad 0.3767248619650429 \quad 1.3397124029914027 \\ 0.4622597516915799 \quad 0.325444462273355 \quad 2.3414525841171394 \\ 0.80142437263375832);$$

Расчет значения F_{np} на основе выхода сети Y осуществляется путем

$$\text{расчета } F_{np} = \left(\frac{Y - m}{g} \right) + os$$

$$m = -1;$$

$$g = 0.000476303881876637;$$

$$os = 4238;$$