

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Хоружий Людмила Ивановна
Должность: Директор института экономики и управления АПК
Дата подписания: 21.03.2024 11:48:48
Уникальный программный ключ:
1e90b132d9b04dce67585160b015dddf2cb1e6a9

УТВЕРЖДАЮ:
Директор института экономики
и управления АПК
Людмила Хоружий
« 28 » августа 2023 г.

**Лист актуализации рабочей программы дисциплины
Б1.В.17 «Нейросетевые технологии анализа данных»**

для подготовки бакалавров
Направление: 09.03.02 «Информационные системы и технологии»
Направленность: «Компьютерные науки и интеллектуальный анализ данных
(Computer Science & Data Mining)»
Форма обучения: очная
Год начала подготовки: 2023
Курс 3
Семестр 5

- 1) В рабочую программу не вносятся изменения. Программа актуализирована для 2023 г. подготовки.
- 2) Программа будет распространена при организации учебного процесса на направленность (профиль): «Компьютерные науки и интеллектуальный анализ данных».

Разработчик (и):

Демичев В.В., канд. экон. наук, доцент
(ФИО, ученая степень, ученое звание)


(подпись)


«28» августа 2023 г.

Быков Д.В., ассистент
(ФИО, ученая степень, ученое звание)


(подпись)

«28» августа 2023 г.

Рабочая программа пересмотрена и одобрена на заседании кафедры статистики и кибернетики протокол № 11 от «28» августа 2023 г.

И.о. заведующего кафедрой статистики и кибернетики  А.В. Уколова

И.о. заведующего выпускающей
кафедрой статистики и кибернетики
А.В. Уколова, к.э.н., доцент



«28» августа 2023 г.



МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ –
МСХА имени К.А. ТИМИРЯЗЕВА»
(ФГБОУ ВО РГАУ - МСХА имени К.А. Тимирязева)

Институт экономики и управления АПК
Кафедра статистики и кибернетики

УТВЕРЖДАЮ:
Директор института экономики и
управления АПК
Институт экономики
и управления АПК
Л.И. Хоружий
«*Лаврушина*» 2022 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.В.17 «Нейросетевые технологии анализа данных»

для подготовки бакалавров

ФГОС ВО

Направление: 09.03.02 «Информационные системы и технологии»

Направленность: «Компьютерные науки и интеллектуальный анализ данных
(Computer Science & Data Mining)»

Курс 3

Семестр 5

Форма обучения: очная

Год начала подготовки: 2022

Москва, 2022

Разработчики:

Быков Д.В., ассистент
(ФИО, ученая степень, ученое звание)


(подпись)

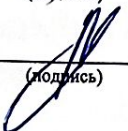
«23» августа 2022 г.

Демичев В.В., канд. экон. наук, доцент
(ФИО, ученая степень, ученое звание)


(подпись)

«23» августа 2022 г.

Харитонов А.Е., канд. экон. наук, доцент
(ФИО, ученая степень, ученое звание)


(подпись)

«23» августа 2022 г.

Рецензент:

Коломеева Е.С., канд. экон. наук, доцент
(ФИО, ученая степень, ученое звание)


(подпись)

«23» августа 2022 г.

Программа составлена в соответствии с требованиями ФГОС ВО, профессионального стандарта и учебного плана по направлению подготовки 09.03.02 «Информационные системы и технологии».

Программа обсуждена на заседании кафедры статистики и кибернетики. Протокол № 11 от «26» августа 2022 г.

И. о. зав. кафедрой Уколова А.В., канд. экон. наук, доцент
(ФИО, ученая степень, ученое звание)


(подпись)

«26» августа 2022 г.

Согласовано:

Председатель учебно-методической
комиссии института экономики и управления АПК

Корольков А.Ф., канд. экон. наук, доцент
(ФИО, ученая степень, ученое звание)


(подпись)

«26» августа 2022 г.

И. о. зав. выпускающей кафедрой
статистики и кибернетики
Уколова А.В., канд. экон. наук, доцент
(ФИО, ученая степень, ученое звание)


(подпись)

«26» августа 2022 г.

/Заведующий отделом комплектования ЦНБ



Ермилова Я.В.
(подпись)

СОДЕРЖАНИЕ

АННОТАЦИЯ.....	4
1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....	4
2. МЕСТО ДИСЦИПЛИНЫ В УЧЕБНОМ ПРОЦЕССЕ	5
3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ.....	5
4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ	11
4.1 РАСПРЕДЕЛЕНИЕ ТРУДОЁМКОСТИ ДИСЦИПЛИНЫ ПО ВИДАМ РАБОТ ПО СЕМЕСТРАМ	11
4.2 СОДЕРЖАНИЕ ДИСЦИПЛИНЫ.....	11
4.3 ЛЕКЦИИ/ПРАКТИЧЕСКИЕ ЗАНЯТИЯ.....	12
5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ	15
6. ТЕКУЩИЙ КОНТРОЛЬ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ	16
6.1. ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ И НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ	16
6.2. ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ КОНТРОЛЯ УСПЕВАЕМОСТИ, ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ	25
7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ.....	27
7.1 ОСНОВНАЯ ЛИТЕРАТУРА	27
7.2 ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА.....	27
8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....	27
9. ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ.....	28
10. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ.....	28
11. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ОБУЧАЮЩИМСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ .	29
Виды и формы отработки пропущенных занятий	29
12. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПРЕПОДАВАТЕЛЯМ ПО ОРГАНИЗАЦИИ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ.....	30

АННОТАЦИЯ

рабочей программы учебной дисциплины Б1.В.17 «Нейросетевые технологии анализа данных» для подготовки бакалавров по направлению 09.03.02 «Информационные системы и технологии» направленности «Компьютерные науки и интеллектуальный анализ данных (Computer Science & Data Mining)»

Цель освоения дисциплины. Основная цель дисциплины «Нейросетевые технологии анализа данных» – овладение студентами основными методами теории интеллектуальных систем, приобретение навыков по использованию интеллектуальных систем, изучение основных методов представления знаний и моделирования рассуждений.

Место дисциплины в учебном плане: дисциплина включена в обязательную часть учебного плана по направлению подготовки 09.03.02 «Информационные системы и технологии».

Требования к результатам освоения дисциплины: в результате освоения дисциплины формируются следующие компетенции (индикаторы): УК-1 (УК-1.1, УК-1.2, УК-1.3), ПКос-9 (ПКос-9.1, ПКос-9.2, ПКос-9.3).

Краткое содержание дисциплины: Основы искусственных нейронных сетей (ИНС). Инструменты создания ИНС. Принципы построения моделей ИНС. Архитектура сетей. Однослойные сети прямого распространения. Многослойные сети прямого распространения. Рекуррентные сети. Процессы обучения. Обучение, основанное на коррекции ошибок. Обучение на основе памяти. Обучение Хебба. Конкурентное обучение. Обучение Больцмана. Обучение с учителем, без учителя. Задачи обучения. Однослойный персептрон. Задача адаптивной фильтрации. Методы безусловной оптимизации. Линейный фильтр, построенный по методу наименьших квадратов. Алгоритм минимизации среднеквадратической ошибки. Графики процесса обучения. Изменения темпа обучения по модели отжига. Теорема о сходимости персептрона. Многослойный персептрон. Алгоритм обратного распространения ошибки. Аппроксимация функций. Перекрестная проверка. Методы упрощения структуры сети.

Распознавание образов. Представление изображения в виде массива чисел. Классификация с помощью нейронных сетей. Прогнозирование с помощью нейронных сетей. Кластеризация с помощью нейронных сетей. Карты самоорганизации. Алгоритм SOM. Контекстные карты.

Общая трудоемкость дисциплины составляет 3 зачетные единицы (108 часов).

Промежуточный контроль: зачет.

1. Цель освоения дисциплины

Целью освоения дисциплины «Нейросетевые технологии анализа данных» является овладение студентами основными методами построения моделей искусственных нейронных сетей (ИНС), приобретение навыков по использованию ИНС при решении задач анализа данных.

2. Место дисциплины в учебном процессе

Дисциплина «Нейросетевые технологии анализа данных» относится к части Блока 1 «Дисциплины (модули)» учебного плана, формируемой участниками образовательных отношений. Дисциплина «Нейросетевые технологии анализа данных» реализуется в соответствии с требованиями ФГОС, ОПОП ВО и Учебного плана по направлению 09.03.02 «Информационные системы и технологии».

Предшествующими курсами, на которых непосредственно базируется дисциплина «Нейросетевые технологии анализа данных» являются «Основы науки о данных (Data Science)», «Анализ экономических данных с использованием современных информационных технологий».

Дисциплина «Нейросетевые технологии анализа данных» является основополагающей для изучения следующих дисциплин: «Методы искусственного интеллекта», «Интеллектуальный анализ данных и статистика», «Большие данные», а также подготовки выпускной квалификационной работы.

Особенностью дисциплины является изучение инструментов создания искусственных нейронных сетей, разработка и программирование моделей искусственных нейронных сетей для решения задач в области анализа данных.

Рабочая программа дисциплины «Нейросетевые технологии анализа данных» для инвалидов и лиц с ограниченными возможностями здоровья разрабатывается индивидуально с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся.

3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

Образовательные результаты освоения дисциплины обучающимся, представлены в таблице 1.

Таблица 1

Требования к результатам освоения учебной дисциплины

№ п/п	Код компетенции	Содержание компетенции (или её части)	Индикаторы компетенций	В результате изучения учебной дисциплины обучающиеся должны:		
				знать	уметь	владеть
1.	УК-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач.	УК-1.1 Знать: методики поиска, сбора и обработки информации; актуальные российские и зарубежные источники информации в сфере профессиональной деятельности; метод системного анализа	методики поиска, сбора и обработки информации; актуальные российские и зарубежные источники информации в сфере профессиональной деятельности; метод системного анализа		
			УК-1.2 Уметь: применять методики поиска, сбора и обработки информации; осуществлять критический анализ и синтез информации, полученной из разных источников; применять системный подход для решения поставленных задач		применять методики поиска, сбора и обработки информации; осуществлять критический анализ и синтез информации, полученной из разных источников; применять системный подход для решения поставленных задач	
			УК-1.3 Владеть: методами поиска, сбора и обработки, критического анализа и синтеза информации; методикой системного			методами поиска, сбора и обработки, критического анализа и синтеза информации;

			подхода для решения поставленных задач			методикой системного подхода для решения поставленных задач
2.	ПКос-9	Способен осуществлять разработку, отладку и рефакторинг кода программного обеспечения, интеграцию программных модулей и компонент, в том числе взаимодействующих с внешней средой, средствами выбранных языков программирования	ПКос-9.1 Знать: методы и приемы формализации и алгоритмизации поставленных задач; нотации и программные продукты для графического отображения алгоритмов; алгоритмы решения типовых задач, области и способы их применения; методологии разработки программного обеспечения; синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования; особенности выбранной среды программирования; методы и приемы отладки программного кода, повышения читаемости программного кода; типы и форматы сообщений об ошибках, предупреждений	методы и приемы формализации и алгоритмизации поставленных задач; нотации и программные продукты для графического отображения алгоритмов; алгоритмы решения типовых задач, области и способы их применения; методологии разработки программного обеспечения; синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования; особенности выбранной среды программирования;		

				методы и приемы отладки программного кода, повышения читаемости программного кода; типы и форматы сообщений об ошибках, предупреждений		
			ПКос-9.2 Уметь: использовать методы и приемы формализации и алгоритмизации поставленных задач; использовать программные продукты для графического отображения алгоритмов; применять стандартные алгоритмы в соответствующих областях; применять выбранные языки программирования для написания программного кода; использовать выбранную среду программирования; применять инструментарий для создания и актуализации исходных текстов программ; выявлять ошибки в программном коде, интерпретировать сообщения об ошибках, предупреждения, записи технологических журналов; применять методы		использовать методы и приемы формализации и алгоритмизации поставленных задач; использовать программные продукты для графического отображения алгоритмов; применять стандартные алгоритмы в соответствующих областях; применять выбранные языки программирования для написания программного кода; использовать выбранную среду программирования; применять	

			и приемы отладки программного кода		инструментарий для создания и актуализации исходных текстов программ; выявлять ошибки в программном коде, интерпретировать сообщения об ошибках, предупреждения, записи технологических журналов; применять методы и приемы отладки программного кода	
			ПКос-9.3 Владеть навыками; составления формализованных описаний решений поставленных задач в соответствии с требованиями технического задания; разработки алгоритмов решения поставленных задач в соответствии с требованиями технического задания или других принятых в организации нормативных документов; создания программного кода в соответствии с техническим заданием (готовыми			навыками составления формализованных описаний решений поставленных задач в соответствии с требованиями технического задания; разработки алгоритмов решения поставленных задач в соответствии с требованиями технического задания или других принятых в организации

			<p>спецификациями); оптимизации программного кода с использованием специализированных программных средств; анализа и проверки исходного программного кода; отладки программного кода на уровне программных модулей и межмодульных взаимодействий и взаимодействий с окружением</p>			<p>нормативных документов; создания программного кода в соответствии с техническим заданием (готовыми спецификациями); оптимизации программного кода с использованием специализированны х программных средств; анализа и проверки исходного программного кода; отладки программного кода на уровне программных модулей и межмодульных взаимодействий и взаимодействий с окружением</p>
--	--	--	---	--	--	--

4. Структура и содержание дисциплины

4.1 Распределение трудоёмкости дисциплины по видам работ по семестрам

Общая трудоёмкость дисциплины составляет 3 зачетные единицы (108 часов), их распределение по видам работ семестрам представлено в таблице 2.

ОЧНАЯ ФОРМА ОБУЧЕНИЯ

Таблица 2а

Распределение трудоёмкости дисциплины по видам работ по семестрам

Вид учебной работы	Трудоёмкость (семестр № 5)/*
	час.
Общая трудоёмкость дисциплины по учебному плану	108/4
1. Контактная работа	53,25/4
Аудиторная работа	53,25/4
<i>лекции (Л)</i>	16
<i>практические занятия (ПЗ)</i>	34/4
<i>курсовой проект (КП) (консультация, защита)</i>	3
<i>контактная работа на промежуточном контроле (КРА)</i>	0,25
2. Самостоятельная работа (СРС)	54,75
<i>самостоятельное изучение разделов, самоподготовка (проработка и повторение лекционного материала и материала учебников и учебных пособий, подготовка к практическим занятиям)</i>	27,75
<i>курсовой проект (КП) (подготовка)</i>	18
<i>подготовка к зачету</i>	9
Вид промежуточного контроля:	Зачет

* в том числе практическая подготовка

4.2 Содержание дисциплины

Таблица 3

Тематический план учебной дисциплины

Наименование разделов и тем дисциплин (укрупнённо)	Всего	Аудиторная работа			Внеаудиторная работа СР
		Л	ПЗ/*	ПКР	
Раздел 1 Введение в искусственные нейронные сети	46,75/4	14	16/4	-	16,75
Раздел 2 Анализ данных с помощью нейронных сетей	40	2	18	-	20
Курсовой проект (КП) (консультация, защита)	21	-	-	3	18
Контактная работа на промежуточном контроле	0,25	-	-	0,25	-
Итого по дисциплине	108	16	34	3,25	54,75

* в том числе практическая подготовка

Раздел 1 Введение в искусственные нейронные сети

Тема 1 Инструменты и способы создания искусственных нейронных сетей

Основы искусственных нейронных сетей (ИНС). Инструменты создания ИНС. Принципы построения моделей ИНС. Архитектура сетей. Однослойные сети прямого распространения. Многослойные сети прямого распространения. Рекуррентные сети. Процессы обучения. Обучение, основанное на коррекции ошибок. Обучение на основе памяти. Обучение Хебба. Конкурентное обучение. Обучение Больцмана. Обучение с учителем, без учителя. Задачи обучения. Однослойный персептрон. Задача адаптивной фильтрации. Методы безусловной оптимизации. Линейный фильтр, построенный по методу наименьших квадратов. Алгоритм минимизации среднеквадратической ошибки. Графики процесса обучения. Изменения темпа обучения по модели отжига. Теорема о сходимости персептрона. Многослойный персептрон. Алгоритм обратного распространения ошибки. Аппроксимация функций. Перекрестная проверка. Методы упрощения структуры сети.

Раздел 2 Анализ данных с помощью нейронных сетей

Тема 1 Применение моделей искусственных нейронных сетей для решения задач анализа данных

Распознавание образов. Представление изображения в виде массива чисел. Классификация с помощью нейронных сетей. Прогнозирование с помощью нейронных сетей. Кластеризация с помощью нейронных сетей. Карты самоорганизации. Алгоритм SOM. Контекстные карты.

4.3 Лекции/практические занятия

Таблица 4

Содержание лекций/практических занятий и контрольные мероприятия

№ п/п	Название раздела, темы	№ и название лекций/ лабораторных/ практических/ семинарских занятий	Формируемые компетенции	Вид контрольного мероприятия	Кол-во часов/ из них практическая подготовка
1.	Раздел 1 Введение в искусственные нейронные сети		УК-1.1, УК-1.2, УК-1.3, ПКос-9.1, ПКос-9.2, ПКос-9.3		30/4
	Тема 1 Инструменты и способы создания искусственных нейронных сетей	Лекция № 1. Основы искусственных нейронных сетей (ИНС). Инструменты создания ИНС. Принципы построения моделей ИНС.	УК-1.1, ПКос-9.1		2
		Практическое занятие № 1. Классы в Python. Основы	УК-1.1, УК-1.2,	устный опрос, защита	4/2

№ п/п	Название раздела, темы	№ и название лекций/ лабораторных/ практических/ семинарских занятий	Формируемые компетенции	Вид контрольного мероприятия	Кол-во часов/ из них практическая подготовка
		матричной алгебры. Реализация класса алгебраических операций с матрицами средствами языка Python.	УК-1.3, ПКос-9.1, ПКос-9.2, ПКос-9.3	практической работы	
		Лекция № 2. Архитектура сетей. Однослойные сети прямого распространения. Многослойные сети прямого распространения. Рекуррентные сети.	УК-1.1, ПКос-9.1		4
		Практическое занятие № 2. Построение простой модели ИНС средствами языка Python. Подготовка наборов исходных данных. Инициализация сети. Весовые коэффициенты. Подготовка тренировочных данных. Подготовка к обучению сети.	УК-1.1, УК-1.2, УК-1.3, ПКос-9.1, ПКос-9.2, ПКос-9.3	устный опрос, защита практической работы	4/2
		Лекция № 3. Процессы обучения. Обучение, основанное на коррекции ошибок. Обучение на основе памяти. Обучение Хебба. Конкурентное обучение. Обучение Больцмана. Обучение с учителем, без учителя. Задачи обучения.	УК-1.1, ПКос-9.1		4
		Практическое занятие № 3. Обучение ИНС средствами языка Python. Тестирование нейронной сети. Оценка качества модели ИНС.	УК-1.1, УК-1.2, УК-1.3, ПКос-9.1, ПКос-9.2, ПКос-9.3	устный опрос, защита практической работы	4
		Лекция № 4. Однослойный персептрон. Задача адаптивной фильтрации. Методы безусловной оптимизации. Линейный фильтр, построенный по методу наименьших квадратов. Алгоритм минимизации среднеквадратической	УК-1.1, ПКос-9.1		2

№ п/п	Название раздела, темы	№ и название лекций/ лабораторных/ практических/ семинарских занятий	Формируемые компетенции	Вид контрольного мероприятия	Кол-во часов/ из них практическая подготовка
		ошибки. Графики процесса обучения. Изменения темпа обучения по модели отжига. Теорема о сходимости персептрона.			
		Практическое занятие № 4. Изменение конфигурации ИНС средствами языка Python. Поиск оптимальных параметров сети.	УК-1.1, УК-1.2, УК-1.3, ПКос-9.1, ПКос-9.2, ПКос-9.3	устный опрос, защита практической работы	4
		Лекция № 5. Многослойный персептрон. Алгоритм обратного распространения ошибки. Аппроксимация функций. Перекрестная проверка. Методы упрощения структуры сети.	УК-1.1, ПКос-9.1		2
2.	Раздел 2 Анализ данных с помощью нейронных сетей		УК-1.1, УК-1.2, УК-1.3, ПКос-9.1, ПКос-9.2, ПКос-9.3		20
	Тема 1 Применение моделей искусственных нейронных сетей для решения задач анализа данных	Практическое занятие № 5. Построение и применение модели ИНС для распознавания образов средствами языка Python. Преобразование массивов значений пикселей в изображения.	УК-1.1, УК-1.2, УК-1.3, ПКос-9.1, ПКос-9.2, ПКос-9.3	устный опрос, защита практической работы	4
		Лекция № 6. Задачи прогнозирования, классификации и кластеризации, решаемые с помощью нейронных сетей.	УК-1.1, ПКос-9.1		2
		Практическое занятие № 6. Применение модели ИНС для распознавания подготовленных образов средствами языка Python. Подготовка собственных изображений. Преобразование изображений в массивы значений пикселей.	УК-1.1, УК-1.2, УК-1.3, ПКос-9.1, ПКос-9.2, ПКос-9.3	устный опрос, защита практической работы	4

№ п/п	Название раздела, темы	№ и название лекций/ лабораторных/ практических/ семинарских занятий	Формируемые компетенции	Вид контрольного мероприятия	Кол-во часов/ из них практическая подготовка
		Практическое занятие № 7. Прогнозирование с помощью нейронных сетей. Сохранение обученной ИНС. Загрузка и применение сохраненной обученной модели ИНС.	УК-1.1, УК-1.2, УК-1.3, ПКос-9.1, ПКос-9.2, ПКос-9.3	устный опрос, защита практической работы	6
		Практическое занятие № 8. Многомерная классификация с помощью нейронных сетей.	УК-1.1, УК-1.2, УК-1.3, ПКос-9.1, ПКос-9.2, ПКос-9.3	устный опрос, защита практической работы	4

Таблица 5

Перечень вопросов для самостоятельного изучения дисциплины

№ п/п	Название раздела, темы	Перечень рассматриваемых вопросов для самостоятельного изучения
Раздел 1 Введение в искусственные нейронные сети		
1.	Тема 1 Инструменты и способы создания искусственных нейронных сетей	1. Преимущества языка программирования Python при создании моделей искусственных нейронных сетей. (УК-1.1, УК-1.2, УК-1.3, ПКос-9.1, ПКос-9.2, ПКос-9.3) 2. Искусственные нейронные сети. Особенности и отличия от стандартных алгоритмов. (УК-1.1, УК-1.2, УК-1.3, ПКос-9.1, ПКос-9.2, ПКос-9.3) 3. Состав искусственной нейронной сети. (УК-1.1, УК-1.2, УК-1.3, ПКос-9.1, ПКос-9.2, ПКос-9.3) 4. Виды искусственных нейронных сетей. (УК-1.1, УК-1.2, УК-1.3, ПКос-9.1, ПКос-9.2, ПКос-9.3)
Раздел 2 Анализ данных с помощью нейронных сетей		
2.	Тема 1 Применение моделей искусственных нейронных сетей для решения задач анализа данных	1. Задачи анализа данных, позволяющие применять искусственные нейронные сети (УК-1.1, УК-1.2, УК-1.3, ПКос-9.1, ПКос-9.2, ПКос-9.3) 2. Функция активации. Виды функций активации. (УК-1.1, УК-1.2, УК-1.3, ПКос-9.1, ПКос-9.2, ПКос-9.3)

5. Образовательные технологии

Таблица 6

Применение активных и интерактивных образовательных технологий

№ п/п	Тема и форма занятия		Наименование используемых активных и интерактивных образовательных технологий (форм обучения)
1.	Практическое занятие № 1. Классы в Python. Основы матричной алгебры. Реализация класса алгебраических операций с матрицами средствами языка Python.	ПЗ	Компьютерная симуляция
2.	Практическое занятие № 2. Построение простой модели ИНС средствами языка Python.	ПЗ	Компьютерная симуляция
3.	Практическое занятие № 3. Обучение ИНС средствами языка Python.	ПЗ	Компьютерная симуляция
4.	Практическое занятие № 4. Изменение конфигурации ИНС средствами языка Python.	ПЗ	Компьютерная симуляция
5.	Практическое занятие № 5. Построение и применение модели ИНС для распознавания готовых образов средствами языка Python.	ПЗ	Компьютерная симуляция
6.	Практическое занятие № 6. Применение модели ИНС для распознавания подготовленных образов средствами языка Python.	ПЗ	Компьютерная симуляция
7.	Практическое занятие № 7. Прогнозирование с помощью нейронных сетей.	ПЗ	Компьютерная симуляция
8.	Практическое занятие № 8. Многомерная классификация с помощью нейронных сетей.	ПЗ	Компьютерная симуляция

6. Текущий контроль успеваемости и промежуточная аттестация по итогам освоения дисциплины

6.1. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений и навыков и (или) опыта деятельности

1) Примерная тематика курсовых проектов

1. Разработка модели нейронной сети для распознавания образов.
2. Разработка нейронной сети для прогнозирования временных рядов.
3. Применение многослойного персептрона для классификации объектов.
4. Применение рекуррентной нейронной сети для прогнозирования временных рядов.
5. Применение сверточной нейронной сети для распознавания образов.
6. Сравнительный анализ библиотек машинного обучения.
7. Распознавание сортов сельскохозяйственных культур с помощью нейронных сетей.
8. Обработка спутниковых снимков посредством нейронных сетей.

9. Кластерный анализ сельскохозяйственных организаций с помощью самоорганизующихся карт Кохонена.
10. Многослойные персептрон для обработки изображений.
11. Сверточные нейронные сети для обработки изображений.
12. Рекуррентные нейронные сети для обработки изображений.
13. Сеть радиально-базисных функций для аппроксимации функции.
14. Анализ данных с помощью многослойного персептрона.
15. Анализ данных с помощью самоорганизующихся карт Кохонена.
16. Анализ данных с помощью сверточных нейронных сетей.
17. Анализ данных с помощью рекуррентных нейронных сетей.
18. Анализ данных с помощью рекурсивных нейронных сетей.
19. Анализ данных с помощью сети радиально-базисных функций.
20. Анализ данных с помощью нейронных сетей адаптивного резонанса.
21. Разработка и применение модели многослойного персептрона.
22. Разработка и применение модели самоорганизующихся карт Кохонена.
23. Разработка и применение модели сверточной нейронной сети.
24. Разработка и применение модели рекуррентной нейронной сети.
25. Разработка и применение модели рекурсивной нейронной сети.
26. Разработка и применение модели сети радиально-базисных функций.
27. Разработка и применение модели нейронной сети адаптивного резонанса.
28. Сравнительный анализ эффективности применения многослойного персептрона и сверточной нейронной сети при распознавании образов.
29. Сравнительный анализ эффективности применения многослойного персептрона и рекуррентной нейронной сети при распознавании образов.
30. Сравнительный анализ эффективности применения многослойного персептрона и рекуррентной нейронной сети при прогнозировании временных рядов.
31. Сравнительный анализ эффективности применения многослойного персептрона и сети радиально-базисных функций при аппроксимации функции.

2) Вопросы для защиты практических работ

32. Понятие класса в объектно-ориентированном программировании.
33. Определение класса, создание экземпляра класса средствами Python.
34. Способы реализации основных алгебраических операций с матрицами средствами Python.
35. ИНС. Основные элементы ИНС.
36. Логические вычисления с помощью нейронов.
37. Персептрон.
38. Многослойный персептрон и обратная связь.
39. Типы и структура ИНС.
40. Топологии ИНС.

41. Входные и выходные сигналы сети.
42. Весовые коэффициенты.
43. Функция активации.
44. Метод обратного распространения ошибки.
45. Обучение нейронной сети.
46. Метод градиентного спуска.
47. Применение матричной алгебры для построения и обучения нейронной сети.
48. Тренировочный и тестовый наборы данных.
49. Функция ошибки.
50. Темп обучения, число эпох обучения.
51. Диапазоны значений сигналов в сети для сигмоидальной функции активации.
52. Выбор значений для основных параметров сети.
53. Распознавание образов.
54. Представление изображения в виде массива чисел. Отличие размеченного от неразмеченного набора данных.
55. Прогнозирование с помощью нейронных сетей.

3) Перечень вопросов, выносимых на промежуточную аттестацию (зачет)

1. Инструменты создания ИНС.
2. Архитектура сети.
3. Однослойные сети прямого распространения.
4. Многослойные сети прямого распространения.
5. Рекуррентные сети.
6. Обучение сети.
7. Обучение, основанное на коррекции ошибок.
8. Обучение на основе памяти.
9. Обучение Хебба.
10. Конкуренционное обучение.
11. Обучение Больцмана.
12. Обучение с учителем, без учителя.
13. Задачи обучения.
14. Однослойный персептрон.
15. Задача адаптивной фильтрации.
16. Методы безусловной оптимизации.
17. Линейный фильтр, построенный по методу наименьших квадратов.
18. Алгоритм минимизации среднеквадратической ошибки.
19. Графики процесса обучения.
20. Изменения темпа обучения по модели отжига.
21. Теорема о сходимости персептрона.
22. Многослойный персептрон.
23. Алгоритм обратного распространения ошибки.

- 24.Аппроксимация функций.
- 25.Перекрестная проверка.
- 26.Методы упрощения структуры сети.
- 27.Распознавание образов. Представление изображения в виде массива чисел.
- 28.Классификация с помощью нейронных сетей.
- 29.Прогнозирование с помощью нейронных сетей.
- 30.Кластеризация с помощью нейронных сетей.
- 31.Карты самоорганизации.
- 32.Алгоритм SOM.
- 33.Контекстные карты.

Пример работ

Практическое задание № 1 «Классы в Python. Основы матричной алгебры»

Цель: изучить основы создания и применения классов в языке программирования Python, изучить особенности реализации операций с матрицами средствами Python.

Работа с матрицами в Python

Для работы с матрицами применяется библиотека `numpy`.

Создать матрицу можно с помощью функции `numpy.array()`.

В результате будет создан объект типа `numpy.ndarray` (экземпляр класса `numpy.ndarray`).

Для создания вектора (одномерной матрицы) в качестве аргумента данной функции необходимо передать список элементов (одномерный массив), например, список `[1, 2, 3]`:

```
numpy.array([1, 2, 3])
```

Для создания двумерной матрицы в качестве аргумента функции `numpy.array()` необходимо передать список списков (двумерный массив), например, список списков `[[1, 2, 3], [4, 5, 6]]`:

```
numpy.array([[1, 2, 3], [4, 5, 6]])
```

Можно сказать, что передаваемый список списков `[[1, 2, 3], [4, 5, 6]]` – это список строк матрицы. То есть, в результате будет создана матрица с двумя строками: `[1, 2, 3]` – первая строка, `[4, 5, 6]` – вторая строка.

Примеры:

```

1 # Подключение библиотеки
2 import numpy as np
3
4 # Создание матриц
5 mx1 = np.array([1, 2, 3])
6 print('\nМатрица 1x3:\n', mx1)
7
8 mx1 = np.array([[1, 2, 3], [1, 2, 3]])
9 print('\nМатрица 2x3:\n', mx1)
10
11 mx1 = np.array([[1, 2], [3, 4], [5, 6]])
12 print('\nМатрица 3x2:\n', mx1)

```

Матрица 1x3:
[1 2 3]

Матрица 2x3:
[[1 2 3]
[1 2 3]]

Матрица 3x2:
[[1 2]
[3 4]
[5 6]]

Создать матрицу со случайными целыми числами можно с помощью функции `numpy.random.randint()`:

```
In [59]: x = np.random.randint(low = 0, high = 100, size = (10, 10))
```

```
In [60]: x
```

```
Out[60]:
array([[94, 77, 57, 18, 20, 82, 40, 39, 20, 96],
       [ 5, 63, 68, 98, 13,  9, 98, 47,  2, 42],
       [61, 85, 21,  2, 23, 42, 47, 74, 47, 53],
       [21, 67, 27,  2, 87, 42, 50, 75, 31, 65],
       [34, 55, 48, 13, 83, 29, 38, 11, 87, 93],
       [87, 70, 20, 22, 67, 18, 93, 34, 73, 68],
       [15, 94, 96, 83, 34, 81, 24, 93, 25, 28],
       [24,  1, 18, 96, 47, 84, 20, 62, 41, 86],
       [18, 71, 29, 45, 72, 63, 38, 78, 82, 61],
       [49, 20, 38, 18, 79, 60, 54, 56,  1, 13]])
```

Узнать размер матрицы можно с помощью атрибута `shape` класса `numpy.ndarray`:

```
матрица.shape
```

Атрибут `shape` представляет собой кортеж значений. Число строк матрицы можно узнать, обратившись к первому элементу кортежа `shape[0]`, а число столбцов – ко второму элементу `shape[1]`.

Для отбора определенных строк и столбцов из матрицы, можно использовать следующую запись:

```
матрица[индекс_строки, индекс_столбца]
```

Для отбора всех строк или всех столбцов используется символ «:», поэтому, чтобы отобразить все строки и лишь определенный столбец, например, с индексом 0, то можно воспользоваться следующей записью:

матрица[:, 0]

Выбрать определенную строку (например, с индексом 0) можно, указав один индекс – индекс строки:

матрица[0]

Для отбора всех строк и нескольких столбцов необходимо указывать индексы столбцов в виде списка. Например, следующая запись отбирает все строки и столбцы с индексами 1, 2, 3:

матрица[:, [1, 2, 3]]

Примеры:

```
In [36]: mx1
```

```
Out[36]: array([[1, 2],  
              [3, 4],  
              [5, 6]])
```

```
In [37]: mx1[0, 0]
```

```
Out[37]: 1
```

```
In [38]: mx1[0, 1]
```

```
Out[38]: 2
```

```
In [39]: mx1[2, 1]
```

```
Out[39]: 6
```

```
In [40]: mx1[:, 0]
```

```
Out[40]: array([1, 3, 5])
```

```
In [41]: mx1[1, :]
```

```
Out[41]: array([3, 4])
```

```
In [42]: mx1[1]
```

```
Out[42]: array([3, 4])
```

```
In [43]: mx1[1, [0, 1]]
```

```
Out[43]: array([3, 4])
```

Операции с матрицами при помощи библиотеки numpy

1. Сложение матриц осуществляется через оператор «+»:

```
16 # 1. Сложение матриц  
17  
18 A = np.array([[1, 2], [3, 4], [5, 6]])  
19 print('\nМатрица A:\n', A)  
20  
21 B = np.array([[4, 4], [2, 2], [1, 1]])  
22 print('\nМатрица B:\n', B)  
23  
24 C = A + B  
25 print('\nМатрица C = A + B:\n', C)
```

Матрица A:

```
[[1 2]
 [3 4]
 [5 6]]
```

Матрица B:

```
[[4 4]
 [2 2]
 [1 1]]
```

Матрица C = A + B:

```
[[5 6]
 [5 6]
 [6 7]]
```

2. Умножение матрицы на число осуществляется через оператор «*»:

```
29 # 2. Умножение матрицы на число
30
31 A = np.array([[1, 2], [3, 4], [5, 6]])
32 print('\nМатрица A:\n', A)
33
34 C = A * 10
35 print('\nМатрица C = A * 10:\n', C)
```

Матрица A:

```
[[1 2]
 [3 4]
 [5 6]]
```

Матрица C = A * 10:

```
[[10 20]
 [30 40]
 [50 60]]
```

3. Умножение двух матриц осуществляется через функцию dot():

```
40 # 3. Умножение матриц
41
42 A = np.array([[1, 2], [3, 4], [5, 6]])
43 print('\nМатрица A:\n', A)
44
45 B = np.array([[4, 4, 4], [2, 2, 2]])
46 print('\nМатрица B:\n', B)
47
48 C = np.dot(A, B)
49 print('\nМатрица C = A * B:\n', C)
```

Матрица A:

```
[[1 2]
 [3 4]
 [5 6]]
```

Матрица B:

```
[[4 4 4]
 [2 2 2]]
```

Матрица C = A * B:

```
[[ 8  8  8]
 [20 20 20]
 [32 32 32]]
```

4. Транспонирование матрицы осуществляется через функцию transpose():

```
53 # 4. Транспонирование матрицы
54
55 A = np.array([[1, 2], [3, 4], [5, 6]])
56 print('\nМатрица A:\n', A)
57
58 A_t = np.transpose(A)
59 print('\nМатрица A_t:\n', A_t)
```

Матрица A:

```
[[1 2]
 [3 4]
 [5 6]]
```

Матрица A_t:

```
[[1 3 5]
 [2 4 6]]
```

Требуется:

1. Создать класс операций с матрицами, реализующий 4 основные операции: сложение матриц, умножение матрицы на число, скалярное произведение матриц, транспонирование матрицы.

При этом не использовать методы операций с матрицами из библиотеки `numpy`, реализующие указанные операции.

2. Проверить созданный класс на примерах из текущего задания.

3. Создать матрицы с помощью функции `numpy.random.randint()` размером минимум 10x10. Реализовать основные операции с матрицами, при этом:

a. Применить методы собственного класса.

b. Проверить правильность полученных результатов путем применения методов библиотеки `numpy`.

Листинг 1. Пример создания и применения класса

```
74 # Подключение библиотеки
75 import numpy as np
76
77 # Определение класса
78 class Mx:
79
80     # Метод увеличения каждого элемента вектора на число value
81     def plus(vec1, value):
82         n = vec1.shape[0]
83         vec_res = vec1.copy() # Копирование вектора vec1
84         for i in range(n):
85             vec_res[i] = vec1[i] + value
86         return vec_res
87
88
89 v1 = np.array([1, 2, 3])
90 v2 = Mx.plus(vec1 = v1, value = 10)
91 print('v1 =', v1)
92 print('v2 =', v2)
```

Практическое задание № 2

«Построение простой нейронной сети. Подготовка к обучению сети»

Цель: изучить основы построения простейших нейронных сетей средствами Python.

Требуется:

1. Создать матрицы с исходными данными, содержащими значения таблиц истинности для следующих двоичных логических функций алгебры логики с двумя параметрами (a , b или x_1 , x_2):

- Конъюнкция (AND).
- Дизъюнкция (OR).
- Сложение по модулю два (исключающее ИЛИ) (XOR).

Разбить каждую матрицу исходных данных на две матрицы: матрицу входных сигналов (X), матрицу правильных выходных сигналов (Y). Матрица X будет содержать значения параметров логических функций, а матрица Y значения самой функции.

2. Создать класс для построения и обучения трехслойной нейронной сети (с одним скрытым слоем), с числом нейронов 2-2-1 (2 нейрона во входном слое, 2 нейрона в скрытом слое, 1 нейрон в выходном слое).

Класс должен иметь следующие атрибуты:

- 1) матрица Y – значения результирующего признака (в виде вектора-столбца),
- 2) матрица X – значения факторов,
- 3) матрица W_{ih} – весовые коэффициенты между входным (input) и скрытым (hidden) слоем,
- 4) матрица W_{ho} – весовые коэффициенты между скрытым и выходным слоем.
- 5) матрица H – значения входящих сигналов скрытого слоя,
- 6) матрица $H_{sigmoid}$ – значения исходящих сигналов скрытого слоя,
- 7) матрица O – значения входящих сигналов выходного слоя,
- 8) матрица $O_{sigmoid}$ – значения исходящих сигналов выходного слоя,
- 9) матрица E_o – ошибки выходного слоя (ошибки сети).
- 10) матрица E_h – ошибки скрытого слоя.

3. Создать метод заполнения $fit()$, с помощью которого можно будет сохранить матрицы X и Y в соответствующие атрибуты экземпляра класса. Метод должен принимать в качестве параметров 2 матрицы: матрицу входных сигналов (X), матрицу правильных выходных сигналов (Y).

4. Заполнить матрицы весовых коэффициентов случайными значениями в диапазоне $[-1; 1]$.

5. Создать метод для обучения сети. В методе реализовать расчет всех основных матриц (H , $H_{sigmoid}$, O , $O_{sigmoid}$, E_o , E_h). Указанные матрицы будут являться атрибутами экземпляра класса.

6. Создать метод для проверки сети, который в качестве параметров принимает 1 матрицу: матрицу входных сигналов (X). В методе реализовать расчет всех основных матриц для получения выходного сигнала сети (H , $H_{sigmoid}$, O , $O_{sigmoid}$). Указанные матрицы не будут являться атрибутами экземпляра класса, так как при вызове данного метода обновление значений атрибутов H , $H_{sigmoid}$, O , $O_{sigmoid}$ не требуется.

Проверить сеть:

- на нескольких отдельных парах значений параметров логической функции (например, для конъюнкции с матрицей входных сигналов, содержащей значения 1, 1, выходной сигнал сети должен быть равен 1).

- на всех парах значений параметров логической функции (вся таблица X).

Методические указания

Для лучшего транспонирования матриц-строк и матриц-столбцов рекомендуется представлять все матрицы в виде объектов `matrix` библиотеки `numpy`. Создание объекта `matrix` происходит так же, как и создание объекта `ndarray`:

```
import numpy as np
mx0 = np.array([1, 2])
mx1 = np.matrix([1, 2])
```

Матрицу случайных значений с диапазоном $[0; 1)$ можно создать с помощью функции `numpy.random.random_sample(size)`:

```
In [20]: np.random.random_sample(size = (3, 3))
Out[20]:
array([[0.04503118, 0.45325379, 0.0501128 ],
       [0.44839094, 0.77028716, 0.97096171],
       [0.3600896 , 0.06931329, 0.25210061]])
```

Для изменения диапазона значений элементов необходимо изменить полученную матрицу. Например, если вычесть 0,5 от каждого элемента матрицы, то диапазон значений станет $[-0,5; 0,5)$:

```
In [21]: np.random.random_sample(size = (3, 3)) - 0.5
Out[21]:
array([[ 0.46356637,  0.175796 , -0.37092527],
       [-0.20583565, -0.17492526, -0.06569389],
       [-0.26281252,  0.27804715,  0.33559061]])
```

Умножив каждый элемент последней матрицы на 2, мы получим диапазон значений $[-1; 1)$:

```
In [22]: (np.random.random_sample(size = (3, 3)) - 0.5) * 2
Out[22]:
array([[ -0.43590022, -0.24096225,  0.88032551],
       [ 0.53856507, -0.18656475, -0.27879637],
       [-0.44772787,  0.44940373, -0.1870465 ]])
```

Для расчета сигмоиды используется функция `expit` библиотеки `scipy`. Пример подключения элемента библиотеки и использования функции показан ниже:

```
from scipy.special import expit
expit(0.1)
```

6.2. Описание показателей и критериев контроля успеваемости, описание шкал оценивания

Для оценки знаний, умений, навыков и формирования компетенции по дисциплине применяется **балльно-рейтинговая** система контроля и оценки успеваемости студентов.

В основу балльно-рейтинговой системы (БРС) положены принципы, в соответствии с которыми формирование рейтинга студента осуществляется в ходе текущей работы в семестре.

Работы должны быть выполнены по своему варианту, оформлены в соответствии с требованиями стандартов по оформлению текстовых документов в текстовом редакторе MS Word. Работы сдаются в электронном виде.

По результатам защиты могут быть получены следующие баллы:

9-10 баллов – расчеты проведены корректно, результаты правильно интерпретированы. Работа оформлена в соответствии с требованиями стандартов по оформлению текстовых документов. Студент развернуто и свободно ответил на все вопросы при защите работы.

7-8 баллов – работа выполнена, но имеются незначительные ошибки в интерпретации результатов и/или оформлении. Студент в целом ответил на все поставленные вопросы, ориентируется в работе.

4-6 баллов – работа в целом выполнена, но имеются значительные ошибки в интерпретации полученных результатов и представления данных, оформления работы. Некоторые вопросы по работе вызывают затруднения.

1-3 балла – имеются грубые ошибки в методике выполнения, интерпретации полученных результатов и представления данных, оформления работы. Студент не отвечает на вопросы при защите. Либо работа выполнена не полностью.

В течение периода обучения по дисциплине студент должен выполнить и защитить 8 практических заданий (индивидуальных или групповых проектов), каждое из которых оценивается максимум на 10 баллов. За посещение занятий добавляется 0,19 балла за каждый час ($53 \cdot 0,19$), участие в конференции с докладом с использованием методов разработки элементов искусственного интеллекта – 15 баллов. Таким образом, максимально возможная сумма баллов равна: $8 \cdot 10 + 53 \cdot 0,19 + 15 = 70 + 10 + 15 = 95$.

Зачет по дисциплине получают студенты, набравшие не менее 60% от максимального количества баллов, т.е. 57 балла и более.

Итоговая оценка по дисциплине выставляется преподавателем в соответствии со шкалой:

Текущий рейтинг	Оценка			
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»
в процентах	0-59	60-69	70-84	85-100
в баллах	0-56	57-65	66-79	80-95

Студенты, набравшие в течение семестра менее 57 баллов, пишут итоговую зачетную работу. К написанию итоговой зачетной работы допускаются студенты, **в случае выполнения всех практических работ.**

7. Учебно-методическое и информационное обеспечение дисциплины

7.1 Основная литература

1. Бессмертный, И. А. Интеллектуальные системы : учебник и практикум для вузов / И. А. Бессмертный, А. Б. Нугуманова, А. В. Платонов. — Москва : Издательство Юрайт, 2021. — 243 с. — (Высшее образование). — ISBN 978-5-534-01042-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/469867>

2. Ростовцев, В. С. Искусственные нейронные сети : учебник для вузов / В. С. Ростовцев. — 2-е изд., стер. — Санкт-Петербург : Лань, 2021. — 216 с. — ISBN 978-5-8114-7462-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/160142>

7.2 Дополнительная литература

1. Воронов, М. В. Системы искусственного интеллекта : учебник и практикум для вузов / М. В. Воронов, В. И. Пименов, И. А. Небаев. — Москва : Издательство Юрайт, 2022. — 256 с. — (Высшее образование). — ISBN 978-5-534-14916-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/485440>

2. Иванов, В. М. Интеллектуальные системы : учебное пособие для среднего профессионального образования / В. М. Иванов ; под научной редакцией А. Н. Сесекина. — Москва : Издательство Юрайт, 2022. — 93 с. — (Профессиональное образование). — ISBN 978-5-534-07819-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/494505>

3. Станкевич, Л. А. Интеллектуальные системы и технологии : учебник и практикум для вузов / Л. А. Станкевич. — Москва : Издательство Юрайт, 2022. — 397 с. — (Высшее образование). — ISBN 978-5-534-02126-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/489694>

4. Загорулько, Ю. А. Искусственный интеллект. Инженерия знаний: учебное пособие для вузов / Ю. А. Загорулько, Г.Б. Загорулько. - Москва: Издательство Юрайт, 2022. — 93 с. — (Высшее образование). — ISBN 978-5-534-07198-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/494205>

8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. Machine Learning Crash Course. – URL: <https://developers.google.com/machine-learning/crash-course>
2. Цифровые профессии: Искусственный интеллект. – URL: <https://steps.2035.university/collections/f6361b9a-ea2e-41b1-a18f-9a2f84a9fcd4>
3. Kaggle. – URL: <https://www.kaggle.com/>
4. Machine Learning Repository. – URL: <https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients>
5. TensorFlow library. <https://www.tensorflow.org/resources/libraries-extensions>
6. PyTorch. <https://pytorch.org/>
7. KERAS. <https://keras.io/>

9. Перечень программного обеспечения и информационных справочных систем

Таблица 9

Перечень программного обеспечения

№ п/п	Наименование раздела учебной дисциплины	Наименование программы	Тип программы	Автор	Год разработки
1	Разделы 1, 2	Python	расчетная, обучающая, контролирующая	Python Software Foundation	Текущая версия
2	Разделы 1, 2	Anaconda	расчетная, обучающая, контролирующая	Anaconda, Inc.	Текущая версия
3	Разделы 1, 2	Spyder	расчетная, обучающая, контролирующая	Spyder project contributors	Текущая версия
4	Разделы 1, 2	Microsoft Word	обучающая, контролирующая	Microsoft	Текущая версия
5	Разделы 1, 2	Microsoft Excel	обучающая	Microsoft	Текущая версия

10. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Для проведения практических занятий нужен компьютерный класс с доступом в «Интернет», оснащенный программным обеспечением в соответствии с разделом 9.

Таблица 10

Сведения об обеспеченности специализированными аудиториями, кабинетами, лабораториями

Наименование специальных помещений и помещений для самостоятельной работы (№ учебного корпуса, № аудитории)	Оснащенность специальных помещений и помещений для самостоятельной работы
1	2
<p><i>учебная аудитория для проведения занятий лекционного типа, учебная аудитория для групповых и индивидуальных консультаций, учебная аудитория для текущего контроля и промежуточной аттестации (2й учебный корпус, 102 ауд.)</i></p>	<ol style="list-style-type: none"> 1. Экран с электроприводом 1 шт. (Инв. №558771/2) 2. Проектор 1 шт. (без инв. №) – приобретался не за счет средств вуза 3. Вандалоустойчивый шкаф 1 шт. (Инв.№558850/7) 4. Системный блок iP-4 541 3200 Mhz/1024 Mb/ 80 Gb / DVD-R с монитором 1 шт. (Инв. №558777/9) 5. Стенд «Сергеев Сергей Степанович 1910-1999» 1 шт. (Инв.№591013/25) 6. Огнетушитель порошковый 1 шт. (Инв. №559527) 7. Подвесное крепление к огнетушителю 1 шт. (Инв. № 559528) 8. Жалюзи 2шт. (Инв. №1107-221225, Инв. №1107-221225) 9. Лавка 20 шт. 10. Стол аудиторный 20 шт. 11. Стол для преподавателя 1 шт. 12. Стул 2 шт. 13. Доска маркерная 1 шт. 14. Трибуна напольная 1 шт. (без инв. №)
<p><i>учебная аудитория для проведения занятий семинарского типа, учебная аудитория для групповых и индивидуальных консультаций, учебная аудитория для текущего контроля и промежуточной аттестации, помещение для самостоятельной работы (2й учебный корпус, 302 ауд.)</i></p>	<ol style="list-style-type: none"> 1. Системный блок Intel Core Intel Core i3-2100/4096Mb/500Gb/DVD-RW 10 шт. (Инв.№601997, Инв.№601998, Инв.№601999, Инв.№602000, Инв.№602001, Инв.№602002, Инв.№602003, Инв.№602004, Инв.№602005, Инв.№602006) 2. Монитор 10 шт. (без инв. №) - приобретались не за счет средств вуза 3. Шкаф 2 шт. (Инв.№594166, Инв.№594167) 4. Тумба 1 шт. (Инв.№594168) 5. Подвесное крепление к огнетушителю 1 шт. (Инв. № 559528) 6. Огнетушитель порошковый 1 шт. (Инв. №559527) 7. Жалюзи 1 шт. (Инв.№551557) 8. Доска магнитно-маркерная 1 шт. 9. Стол 5 шт. 10. Стол компьютерный 12 шт. 11. Стул офисный 21 шт. 12. Сейф 1 шт. (без Инв.№).
Студенческое общежитие	Комнаты для самоподготовки
ЦНБ имени Н.И. Железнова	Читальный зал

11. Методические рекомендации обучающимся по освоению дисциплины

Предполагается, что студент выполняет практическое задание в аудитории, дома оформляет и готовится по теоретическим вопросам к защите отчета на следующем занятии.

Виды и формы отработки пропущенных занятий

Студент, пропустивший занятие, обязан предъявить преподавателю документы установленного образца, подтверждающие необходимость пропуска. Не допускается пропуск занятий без уважительной причины.

Студент, пропустивший занятия, осваивает материал самостоятельно (выполняет практическое задание по своему варианту в компьютерном классе кафедры в часы, свободные от занятий, изучает теоретические вопросы).

Студент, пропустивший лекцию, отвечает на вопросы по пропущенной теме.

12. Методические рекомендации преподавателям по организации обучения по дисциплине

На первом занятии преподаватель закрепляет за каждым студентом номер варианта для выполнения индивидуальных работ (как правило, номер варианта соответствует порядковому номеру студента в журнале преподавателя). По каждой индивидуальной работе должна быть поставлена оценка по факту ее защиты. Защиту рекомендуется проводить на следующем после получения задания занятии. Преподаватель обязан проверить соответствие выполненного задания исходным данным варианта студента. Таким образом, исключается вероятность плагиата.

Преподаватель должен стимулировать студентов к занятию научно-исследовательской работой, изучению научной литературы по теме искусственного интеллекта, в т.ч. отечественной и зарубежной периодики.

Программу разработали:

Быков Д.В., ассистент


(ФИО, ученая степень, ученое звание)

Демичев В.В., канд. экон. наук, доцент


(ФИО, ученая степень, ученое звание)

Харитонов А.Е., канд. экон. наук, доцент


(ФИО, ученая степень, ученое звание)



(подпись)



(подпись)



(подпись)

РЕЦЕНЗИЯ

на рабочую программу дисциплины «Нейросетевые технологии анализа данных»
ОПОП ВО по направлению 09.03.02 «Информационные системы и технологии»,
направленность «Компьютерные науки и интеллектуальный анализ данных (Computer
Science & Data Mining)»
(квалификация выпускника – бакалавр)

Колосовой Еленой Сергеевной, доцентом кафедры финансов (далее по тексту рецензент), проведена рецензия рабочей программы дисциплины «Нейросетевые технологии анализа данных» ОПОП ВО по направлению 09.03.02 – «Информационные системы и технологии», направленность «Компьютерные науки и интеллектуальный анализ данных (Computer Science & Data Mining)», разработанной в ФГБОУ ВО «Российский государственный аграрный университет – МСХА имени К.А. Тимирязева», на кафедре статистики и кибернетики (разработчики – Быков Денис Витальевич, ассистент кафедры статистики и кибернетики, Демичев Вадим Владимирович, кандидат экономических наук, доцент, доцент кафедры статистики и кибернетики Харитоновна Анна Евгеньевна, кандидат экономических наук, доцент, доцент кафедры статистики и кибернетики).

Рассмотрев представленные на рецензию материалы, рецензент пришел к следующим выводам:

1. Предъявленная рабочая программа дисциплины «Нейросетевые технологии анализа данных» (далее по тексту Программа) соответствует требованиям ФГОС ВО по направлению 09.03.02 – «Информационные системы и технологии». Программа содержит все основные разделы, соответствует требованиям к нормативно-методическим документам.

2. Представленная в Программе **актуальность** учебной дисциплины в рамках реализации ОПОП ВО не подлежит сомнению – дисциплина относится к части учебного цикла, формируемой участниками образовательных отношений – Б1.В.

3. Представленные в Программе **цели** дисциплины соответствуют требованиям ФГОС ВО направления 09.03.02 – «Информационные системы и технологии».

4. В соответствии с Программой за дисциплиной «Нейросетевые технологии анализа данных» закреплено: 1 универсальная **компетенция (3 индикатора)** 1 профессиональная **компетенция**, определяемая самостоятельно (**3 индикатора**). Дисциплина «Нейросетевые технологии анализа данных» и представленная Программа способна реализовать их в объявленных требованиях. Результаты обучения, представленные в Программе в категориях знать, уметь, владеть соответствуют специфике и содержанию дисциплины и демонстрируют возможность получения заявленных результатов.

5. Общая трудоёмкость дисциплины «Нейросетевые технологии анализа данных» составляет 3 зачётных единицы (108 часов).

6. Информация о взаимосвязи изучаемых дисциплин и вопросам исключения дублирования в содержании дисциплин соответствует действительности. Дисциплина «Нейросетевые технологии анализа данных» взаимосвязана с другими дисциплинами ОПОП ВО и Учебного плана по направлению 09.03.02 – «Информационные системы и технологии» и возможность дублирования в содержании отсутствует.

7. Представленная Программа предполагает использование современных образовательных технологий, используемые при реализации различных видов учебной работы. Формы образовательных технологий соответствуют специфике дисциплины.

8. Программа дисциплины «Нейросетевые технологии анализа данных» предполагает проведение занятий в интерактивной форме.

9. Виды, содержание и трудоёмкость самостоятельной работы студентов, представленные в Программе, соответствуют требованиям к подготовке выпускников, содержащимся во ФГОС ВО направления 09.03.02 – «Информационные системы и технологии».

10. Представленные и описанные в Программе формы *текущей* оценки знаний (выполнение и защита практических заданий), соответствуют специфике дисциплины и требованиям к выпускникам.

Форма промежуточного контроля знаний студентов, предусмотренная Программой, осуществляется в форме зачета, что соответствует статусу дисциплины, как дисциплины части учебного цикла, формируемой участниками образовательных отношений – Б1.В ФГОС ВО направления 09.03.02 – «Информационные системы и технологии».

11. Формы оценки знаний, представленные в Программе, соответствуют специфике дисциплины и требованиям к выпускникам.

12. Учебно-методическое обеспечение дисциплины представлено: основной литературой – 2 источника (базовый учебник), дополнительной литературой – 4 наименований, Интернет-ресурсы – 7 источников и соответствует требованиям ФГОС ВО направления 09.03.02 – «Информационные системы и технологии».

13. Материально-техническое обеспечение дисциплины соответствует специфике дисциплины «Нейросетевые технологии анализа данных» и обеспечивает использование современных образовательных, в том числе интерактивных методов обучения.

14. Методические рекомендации студентам и методические рекомендации преподавателям по организации обучения по дисциплине дают представление о специфике обучения по дисциплине «Нейросетевые технологии анализа данных».

ОБЩИЕ ВЫВОДЫ

На основании проведенной рецензии можно сделать заключение, что характер, структура и содержание рабочей программы дисциплины «Нейросетевые технологии анализа данных» ОПОП ВО по направлению 09.03.02 «Информационные системы и технологии», направленность «Компьютерные науки и интеллектуальный анализ данных (Computer Science & Data Mining)» (квалификация выпускника – бакалавр), разработанная Быковым Денисом Витальевичем, ассистентом кафедры статистики и кибернетики, Демичевым Вадимом Владимировичем, кандидатом экономических наук, доцентом, доцентом кафедры статистики и кибернетики, Харитоновой Анной Евгеньевной, кандидатом экономических наук, доцентом, доцентом кафедры статистики и кибернетики соответствует требованиям ФГОС ВО, современным требованиям экономики, рынка труда и позволит при её реализации успешно обеспечить формирование заявленных компетенций.

Рецензент: Коломеева Е.С., доцент кафедры финансов ФГБОУ ВО «Российский государственный аграрный университет – МСХА имени К.А. Тимирязева», кандидат экономических наук _____ «23» августа 2022г.

(подпись)