**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**“РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА И**

**ГОСУДАРСВТЕННОЙ СЛУЖБЫ ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ”**

**Липецкий филиал**

Утверждена

Директор ЛФ РАНХиГС

\_\_\_\_\_\_­­­\_\_\_\_\_\_ Е.А. Гончарова

(подпись)

«28» августа 2023 г.

Утверждена

ученым советом ЛФ РАНХи ГС

Протокол от «28» августа 2023 г. №1

**ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА**

**повышения квалификации**

**Машинное обучение в цифровой трансформации государственного управления**

Липецк, 2023 г.

Разработчик

канд. пед. наук, доцент

доцент каф. ГиЕД \_\_\_\_\_\_\_\_ В.М Яворский

(ученая степень и (или) ученое звание, (подпись) (И.О. Фамилия)

должность, структурное подразделение)

Руководитель программы

канд. пед. наук, доцент

доцент каф. ГиЕД \_\_\_\_\_\_\_\_ В.М Яворский

(ученая степень и (или) ученое звание, (подпись) (И.О. Фамилия)

должность, структурное подразделение)

Руководитель структурного

подразделения

канд. пед. наук, доц.

и.о. зав. каф. ГиЕД \_\_\_\_\_\_\_\_\_ Е.В. Чудинова

(ученая степень и (или) ученое звание, (подпись) (И.О. Фамилия)

должность, структурное подразделение)

Дополнительная профессиональная программа рассмотрена и одобрена на заседании ученого совета Липецкого филиала РАНХиГС «28» августа 2023 г. протокол № 1.

**СОДЕРЖАНИЕ**

[1. Общая характеристика программы 4](#_Toc115627401)

[1.1. Цель реализации 4](#_Toc115627402)

[1.2. Нормативная правовая база 4](#_Toc115627403)

[1.3. Планируемые результаты обучения 5](#_Toc115627404)

[1.4. Категории слушателей 7](#_Toc115627405)

[1.5. Формы обучения и сроки освоения 7](#_Toc115627406)

[1.6. Период обучения и режим занятий 7](#_Toc115627407)

[1.7. Документ о квалификации 7](#_Toc115627408)

[2. Содержание программы 7](#_Toc115627409)

[2.1. Календарный учебный график 7](#_Toc115627410)

[2.2. Учебный план 8](#_Toc115627411)

[2.3. Рабочие программы дисциплин (модулей) 9](#_Toc115627412)

[3. Организационно-педагогическое сопровождение 10](#_Toc115627413)

[3.1. Кадровое обеспечение 10](#_Toc115627414)

[3.2. Материально-техническое и программное обеспечение реализации программы 12](#_Toc115627415)

[3.3. Учебно-методическое и информационное обеспечение программы 12](#_Toc115627416)

[4. Оценка качества освоения программы 13](#_Toc115627417)

[Приложение 1 Рецензии (внешняя и внутренняя) 44](#_Toc115627418)

**1. Общая характеристика программы**

**1.1. Цель реализации**

Целью освоения дополнительной профессиональной программы повышения квалификации «Машинное обучение в цифровой трансформации государственного управления» является совершенствование имеющихся компетенций, необходимых для профессиональной деятельности в корпоративных и внешних информационных системах и использования цифровых сервисов.

Задачи программы:

1. Формирование, развитие цифровых компетенций специалиста, направленных на повышение престижа профессии.

2. Обучение исполнителей рабочего процесса инструментам по использованию корпоративных и внешних информационных систем, цифровых сервисов.

3. Освоение инструментов развития внутренних информационно-коммуникационных технологий и с учетом основных требований информационной безопасности на уровне компетенций персонала.

**1.2. Нормативная правовая база**

1. Федеральный закон от 29 декабря 2012 №273-ФЗ «Об образовании в Российской Федерации».
2. Приказ Минобрнауки России от 01 июля 2013 №499 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам».
3. «Общероссийский классификатор видов экономической деятельности» (утв. Приказом Росстандарта от 31.01.2014 !14-ст) (ред. от 10 июня 2021).
4. Приказ Минздравсоцразвития РФ от 26.08.2010 №761н (ред. от 31 мая 2011) «Об утверждении Единого квалификационного справочника должностей руководителей, специалистов и служащих, раздел "Квалификационные характеристики должностей работников образования»
5. Постановление Минтруда России от 21 августа 1998 №37 (ред. от 27 марта 2018) «Квалификационный справочник должностей руководителей, специалистов и других служащих».
6. Методические рекомендации по использованию электронного обучения, дистанционных образовательных технологий при реализации дополнительных профессиональных образовательных программ Министерства образования и науки Российской Федерации от 10 апреля 2014 №06-381.
7. Методические рекомендации по разработке основных профессиональных образовательных программ дополнительных профессиональных программ с учетом соответствующих профессиональных стандартов (утв. Минобрнауки России 22 января 2015 №ДЛ-1/05вн).
8. Письмо Минобрнауки России от 21 апреля 2015 №ВК-1013/06 «Методически рекомендации по реализации дополнительных профессиональных программ с использованием дистанционных образовательных технологий, электронного обучения и в сетевой форме».
9. Приказ Министерства образования и науки Российской Федерации от 23 августа 2017 №816 «Об утверждении Порядка применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ».
10. Профессиональный стандарт «Специалист по большим данным» (Утвержден приказом Министерства труда и социальной защиты Российской Федерации от «06» июля 2020 г. № 405н).
11. Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 38.03.03 Управление персоналом (утв. Министерством науки и высшего образования Российской Федерации от 14 декабря 2015 г. № 1461).
12. Приказ РАНХиГС от 22 сентября 2017 №01-6230 «Об утверждении положения о применении в Академии электронного обучения. Дистанционных образовательных технологий при реализации образовательных программ».
13. Приказ РАНХиГС от 19 апреля 2019 №02-461 «Об утверждении локальных нормативных активов РАНХиГС по дополнительному профессионального образованию».
14. Приказ РАНХиГС от 13 августа 2021 г. №02-835 «Порядок разработки и утверждения в РАНХиГС дополнительных профессиональных программ профессиональной переподготовки, программ повышения квалификации»

**1.3. Планируемые результаты обучения**

Виды деятельности и перечень профессиональных компетенций в рамках. Имеющейся квалификации, качественное изменение которых осуществляется в результате обучения и (или) получение новой компетенции, необходимой для профессиональной деятельности, а также знания, умения и практический опыт, приобретаемый в результате освоения программы (таблица 1).

Таблица 1

**Планируемые результаты обучения**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Виды деятельности** | **Общепрофессиональные / профессиональные компетенции**  **ОПК, ПК или трудовые функции (ПСК и СК) (формируются и (или) совершенствуются)** | **Знания** | **Умения** | **Практический опыт** |
| Создание и применение технологий больших данных | ПСК-1[[1]](#footnote-1) Анализ больших данных с использованием существующей в организации методологической и технологической инфраструктуры | З1 – Современный опыт использования анализа больших данных  З2 – Теоретические и прикладные основы анализа данных  З3 – Основы бизнес-интеллекта, типы систем бизнес-интеллекта  З4 – Типы анализа больших данных, виды аналитики  З5 – Современные методы и инструментальные средства анализа больших данных  З6 – Стандарты проведения анализа данных  З7 – Методы оценки временных и стоимостных характеристик технологий больших данных  З8 – Методы интерпретации и визуализации больших данных | У1 – Выявление требований заказчика к результатам анализа, определение возможностей применения анализа больших данных в предметной области и конкретных задачах заказчика  У2 – Выбор методов и инструментальных средств анализа больших данных для проведения аналитических работ  У3 – Разработка, поверка, оценка используемых моделей больших данных  У4 – Адаптация и развертывание моделей больших данных в предметной среде  У5 – Выбор средств представления результатов аналитики больших данных | В1 – Разрабатывать и оценивать модели больших данных  В2 – Проводить сравнительный анализ методов и инструментальных средств анализа больших данных  В3 – Проводить анализ больших данных в соответствии с утвержденными требованиями к результатам аналитического исследования  В4 – Планировать аналитические работы с использованием технологий больших данных |
| **ПК – профессиональные компетенции (формируются и (или) совершенствуются)** | | | | |
| **Код и наименование компетенции** | | **Знания** | **Умения** | **Практический опыт** |
| ПК-27. владением методами и программными средствами обработки деловой информации, навыками работы со специализированными кадровыми компьютерными  программами, способностью взаимодействовать со службами информационных технологий и эффективно использовать корпоративные информационные системы при решении задач управления персоналом [[2]](#footnote-2) | | З1 - сущность методов и программных средств обработки деловой информации  З2 - сущность и содержание основных методов взаимодействия со службами информационных технологий и эффективно использовать корпоративные информационные системы | У1 – анализировать, деловую информацию, навыками работы со специализированными кадровыми компьютерными  программами  У2- эффективно использовать корпоративные информационные системы | В1 – владение способами работы с деловой информацией  В2 – владеть основами использования корпоративных информационных систем |
| ПК-28. знанием корпоративных коммуникационных каналов и средств передачи информации, владением навыками информационного обеспечения процессов внутренних  коммуникаций [[3]](#footnote-3) | | З1 – основы корпоративных коммуникационных каналов и средств передачи информации  З2 – основные элементы информационного обеспечения процессов внутренних  коммуникаций | У1 – работы с корпоративными коммуникационными каналами и средствами передачи информации  У2 – работы с информационным обеспечением процессов внутренних | В1 – владеть методами активной работы с корпоративными коммуникационными каналами и средствами передачи информации  В2 – владеть основами организации информационного обеспечения процессов внутренних коммуникаций |

**1.4. Категории слушателей**

К освоению программы допускаются лица, имеющие среднее профессиональное образование и/или высшее образование.

**1.5. Формы обучения и сроки освоения**

Форма обучения: очная. Срок обучения составляет 24 часа. Контактная работа со слушателем 24 часа, из них:

очно – 24 часа.

Аттестационная работа 1 час.

**1.6. Период обучения и режим занятий**

Занятия проводятся в течение 3 дней.

**1.7. Документ о квалификации**

При успешном освоении программы выдается Удостоверение о повышении квалификации федерального государственного бюджетного образовательного учреждения высшего образования «Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации».

**2. Содержание программы**

**2.1. Календарный учебный график**

В календарном учебном графике использованы следующие обозначения:

УЗ – учебные занятия;

ТКУ – текущий контроль успеваемости;

ИА – итоговая аттестация.

Таблица 2

**Календарный учебный график**

|  |  |  |
| --- | --- | --- |
| **Период обучение – 3 дня** | | |
| 1 день | 2 день | 3 день |
| УЗ, ТКУ | УЗ, ТКУ | УЗ, ТКУ, ИА |

**2.2. Учебный план**

Таблица 3

**Учебный план**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№п/п** | **Наименование (модуля/раздела/дисциплины/темы), практики (стажировки)** | **Общая трудоемкость, час.** | **Контактная работа, час.** | | | | | **Самостоятельная работа, час** | **Контактная работа (с применением дистанционных образовательных технологий, электронного обучения), час.** | | | | | **Самостоятельная работа, час** | **Текущий контроль успеваемости** | **Промежуточная аттестация (форма/час)** | **Итоговая аттестация (вид /час.)** | **Код компетенции** |
| **Всего** | **В том числе** | | | | **Всего** | **В том числе** | | | |
| **Лекции** | **Лабораторные занятия (практикум)** | **Практические (семинарские) занятия /в интерактивной форме** | **Контактная самостоятельная работа, час** | **Лекции** | **Лабораторные занятия (практикум)** | **Практические (семинарские) занятия** | **Контактная самостоятельная работа, час** |
| 1. | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 16 | 17 | 18 | 19 | 20 |
| 1. | Машинное обучение в цифровой трансформации государственных служащих | 23 | 23 | 3 | 20 | - | - | - | - | - | - | - | - | - | ПЗ[[4]](#footnote-4) | - | - | ПСК-1  ПК-27  ПК-28 |
|  | Итого: | 23 | 23 | 3 | 20 | - | - | - | - | - | - | - | - | - | - | - | - | - |
|  | Итоговая аттестация | 1 | - | - | - | - | - | - | - | - | - | - | - | - | - | - | Зачет (Т/1) | - |
|  | Всего: | 24 |  | 3 | 20 | - | - | - | - | - | - | - | - | - |  | - | 1 | - |

**2.3. Рабочие программы дисциплин (модулей)**

**Рабочая программа дисциплины «Машинное обучение в цифровой трансформации государственного управления»»**

Таблица 4

**Содержание программы по модулям/разделам/ и (или) темам**

|  |  |
| --- | --- |
| Номер и названия разделов (темы) | Содержание разделов (темы) |
| 1. Цифровые технологии в государственном управлении | Сквозные технологии и цифровая экономика. Введение в машинное обучение (machine learning, ML). Цифровизация как средство достижения целей современной организации.  Информационная безопасность государственных данных. Техническая защита информации. Цифровой Этикет.  Этика работа с данными. Нормы этики и морали в сфере разработки ML. |
| 2. Технологии ML в госуправлении | Введение в теорию ML. Основные термины и понятия. Принципы ML.Цифровые технологии и инструменты в профессиональной деятельности государственного служащего. Разбор кейсов по применению технологий ML в государственном управлении и иных приоритетных отраслях экономики.  Обработка массивов данных с помощью ML Нейросети для анализа таблиц, построения диаграмм и упрощения работы с электронными таблицами.  Системы, сервисы и платформы ML. Выбор и комбинирование.  Внедрение ML-моделей в процессы организации и их сопровождение. Оценка качества ML-систем. |
| 3. Инструменты в государственном управлении | Обзор рынка ML и перспективных технологий, опыт зарубежных стран. Разбор часто встречающихся задач в сфере государственного управления в контексте их возможной автоматизации на базе ML.  Инструменты анализа данных и визуализации данных. Цифровые инструменты в сфере государственного управления: сервисы для управления целями и задачами в цифровой среде.  Цифровые инструменты в сфере государственного управления: продукты, предназначенные для создания, просмотра, редактирования файлов.  Цифровые инструменты в сфере государственного управления: сервисы создания видеоконференций. |

**3. Организационно-педагогическое сопровождение**

**3.1. Кадровое обеспечение**

Таблица 5

**Сведения о профессорско-преподавательском составе**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Ф.И.О. преподавателя/ведущего специалиста** | **Специальность, присвоенная квалификация по диплому** | **Дополнительн/ая/ые квалификаци/я/и** | **Место работы, должность, основное/дополнительное место работы** | **Ученая степень, ученое (почетное) звание** | **Стаж работы в области профессиональной деятельности/по дополнительной квалификации** | **Стаж научно-педагогической работы** | | **Наименование преподаваемой дисциплины/темы (модуля), практики/стажировки (при наличии) по данной программе** |
| **Всего** | **В том числе по преподаваемой дисциплине (модулю)** |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Яворский Василий Михайлович | Общетехнические дисциплины и труд, Учитель общетехнических дисциплин | Использование СДО в образовательном процессе с применением электронного обучения и дистанционных образовательных технологий (ЭО и ДОТ)»  «Организация образовательного процесса в электронной информационно-образовательной среде вуза» | ЛФ РАНХиГС, доцент кафедры ГиЕД | Кандидат педагогических наук, доцент | 19 | 19 | 19 | Модуль 1 |

**3.2. Материально-техническое и программное обеспечение реализации программы**

Реализация дополнительной профессиональной программы повышения квалификации «Машинное обучение в цифровой трансформации государственного управления» осуществляется в аудитории, оснащённой модульной мебелью, видеопроекционным оборудованием для презентаций, средствами звуковоспроизведения, экраном, компьютером(ами), имеющим(ими) выход в сеть Интернет.

Для организации и проведения контактной и самостоятельной работы обучающихся используется:

а) лицензионное программное обеспечение:

|  |  |
| --- | --- |
| Наименование программного обеспечения | Назначение и тип лицензии программного обеспечения |
| Microsoft Windows | Операционная система. Лицензионная версия |
| Microsoft Office (Word, Excel, Power Point) | Офисный пакет приложений. Лицензионная версия |
| Adobe Acrobat Reader | Программное обеспечение для чтения, печати и рецензирования файлов PDF. Свободная лицензия |
| 7-Zip | Архиватор с высокой степенью сжатия. Свободная лицензия |
| Microsoft Edge  Google Chrome  Mozilla FireFox  Opera  Yandex | Браузеры. Свободная лицензия |
| Ramus Educational | Средство для создания диаграмм в формате IDEF0 и DFD. Свободная лицензия |
| AST | Программное обеспечение для тестирования. Лицензионная версия |
| Программный продукт «Альт-Инвест Прим»; | предназначен для использования в образовательном процессе при разработке инвестиционных проектов и обосновании их экономической эффективности |
| Программный продукт «Альт-Финансы»; | предназначен для использования в образовательном процессе по направлениям подготовки при осуществлении комплексного финансового анализа деятельности хозяйствующего субъекта |

б) информационные справочные системы:

|  |  |
| --- | --- |
| Наименование информационной справочной системы | Ссылка на ресурс |
| Справочная правовая система  «КонсультантПлюс» (официальный сайт); | <http://www.consultant.ru> |
| Справочная правовая система «Гарант» (официальный сайт)» | <http://www.garant.ru> |

**3.3. Учебно-методическое и информационное обеспечение программы**

**3.3.1. Учебно-методическое и информационное обеспечение «Машинное обучение в цифровой трансформации государственного управления»**

1. Абросимова М. А. Информационные технологии в государственном и муниципальном управлении. – 2013. (в наличии 20 шт - ЛФ РАНХиГС).
2. Бессмертный, И. А. Системы искусственного интеллекта: учебное пособие для вузов / И. А. Бессмертный. — 2-е изд., испр. и доп. — Москва: Издательство Юрайт, 2022. — 157 с. — (Высшее образование). — ISBN 978-5-534-07467-3. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/490657 (дата обращения: 02.10.2022).
3. Платонов, А. В. Машинное обучение: учебное пособие для вузов / А. В. Платонов. — Москва: Издательство Юрайт, 2022. — 85 с. — (Высшее образование). — ISBN 978-5-534-15561-7. — Текст электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/508804 (дата обращения: 02.10.2022).
4. 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 (дата обращения: 02.10.2022).

**4. Оценка качества освоения программы**

Таблица 6

**Планируемые результаты обучения дисциплины (модуля)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Виды деятельности** | **Общепрофессиональные / профессиональные компетенции**  **ОПК, ПК или трудовые функции (ПСК и СК) (формируются и (или) совершенствуются)** | **Знания** | **Умения** | **Практический опыт** |
| Создание и применение технологий больших данных | ПСК-1[[5]](#footnote-5) Анализ больших данных с использованием существующей в организации методологической и технологической инфраструктуры | З1 – Современный опыт использования анализа больших данных  З2 – Теоретические и прикладные основы анализа данных  З3 – Основы бизнес-интеллекта, типы систем бизнес-интеллекта  З4 – Типы анализа больших данных, виды аналитики  З5 – Современные методы и инструментальные средства анализа больших данных  З6 – Стандарты проведения анализа данных  З7 – Методы оценки временных и стоимостных характеристик технологий больших данных  З8 – Методы интерпретации и визуализации больших данных | У1 – Выявление требований заказчика к результатам анализа, определение возможностей применения анализа больших данных в предметной области и конкретных задачах заказчика  У2 – Выбор методов и инструментальных средств анализа больших данных для проведения аналитических работ  У3 – Разработка, поверка, оценка используемых моделей больших данных  У4 – Адаптация и развертывание моделей больших данных в предметной среде  У5 – Выбор средств представления результатов аналитики больших данных | В1 – Разрабатывать и оценивать модели больших данных  В2 – Проводить сравнительный анализ методов и инструментальных средств анализа больших данных  В3 – Проводить анализ больших данных в соответствии с утвержденными требованиями к результатам аналитического исследования  В4 – Планировать аналитические работы с использованием технологий больших данных |
| **ПК – профессиональные компетенции (формируются и (или) совершенствуются)** | | | | |
| **Код и наименование компетенции** | | **Знания** | **Умения** | **Практический опыт** |
| ПК-27. владением методами и программными средствами обработки деловой информации, навыками работы со специализированными кадровыми компьютерными  программами, способностью взаимодействовать со службами информационных технологий и эффективно использовать корпоративные информационные системы при решении задач управления персоналом [[6]](#footnote-6) | | З1 - сущность методов и программных средств обработки деловой информации  З2 - сущность и содержание основных методов взаимодействия со службами информационных технологий и эффективно использовать корпоративные информационные системы | У1 – анализировать, деловую информацию, навыками работы со специализированными кадровыми компьютерными  программами  У2- эффективно использовать корпоративные информационные системы | В1 – владение способами работы с деловой информацией  В2 – владеть основами использования корпоративных информационных систем |
| ПК-28. знанием корпоративных коммуникационных каналов и средств передачи информации, владением навыками информационного обеспечения процессов внутренних  коммуникаций [[7]](#footnote-7) | | З1 – основы корпоративных коммуникационных каналов и средств передачи информации  З2 – основные элементы информационного обеспечения процессов внутренних  коммуникаций | У1 – работы с корпоративными коммуникационными каналами и средствами передачи информации  У2 – работы с информационным обеспечением процессов внутренних | В1 – владеть методами активной работы с корпоративными коммуникационными каналами и средствами передачи информации  В2 – владеть основами организации информационного обеспечения процессов внутренних коммуникаций |

При реализации дополнительной профессиональной программы повышения квалификации «Машинное обучение в цифровой трансформации государственного управления» предусматриваются следующие формы аттестации.

Текущий контроль – обеспечивается посредством практических заданий, проводимых в ходе освоения дополнительной профессиональной программы повышения квалификации.

Итоговый контроль – обеспечивается комплексным тестированием, позволяющим оценить индикаторы достижения компетенций.

**Учебно-методическое обеспечение для самостоятельной работы слушателей по дисциплине**

**Кейс**

Тема: **Монтаж художественных образов с помощью нейронных сетей ipynb.**

Цель: научиться использовать нейронные сети для монтажа художественных образов.

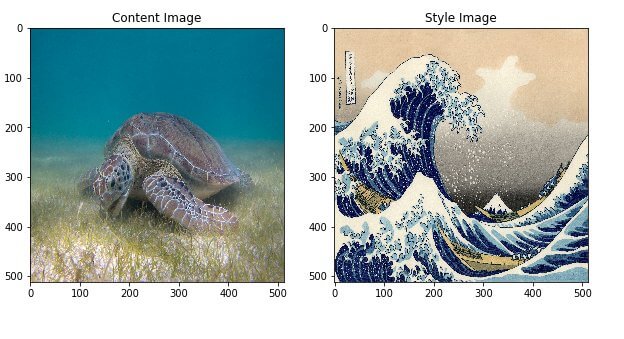
Материальное и дидактическое оснащение: Методические рекомендации по выполнению практической работы.

**ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

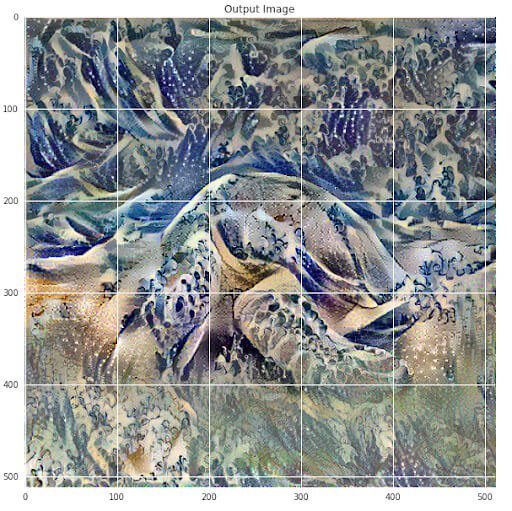
В [этом примере](https://colab.research.google.com/github/tensorflow/models/blob/master/research/nst_blogpost/4_Neural_Style_Transfer_with_Eager_Execution.ipynb) рассказывается о том, как использовать deep learning для стилизации изображения по заданному образцу. Это возможно благодаря нейронному переносу стиля (англ. neural style transfer). Эта техника описана в статье [Leon A. Gatys, A Neural Algorithm of Artistic Style](https://arxiv.org/abs/1508.06576).

Нейронная передача стиля — это процесс оптимизации, который работает с 3 изображениями: картинкой содержания, картинкой стиля (например, произведением художника) и входной картинкой. Если «смешать» их, то получится входная картинка, подогнанная по композиции под картинку содержания в образе копируемого стиля.

Для примера возьмём фотографию черепахи и гравюру Кацусики Хокусая «Большая волна в Канагаве»:



И что бы вышло, если бы художник решил стилизовать фотографию черепахи под свою гравюру? У него получилось бы что-то подобное:



Принцип передачи стиля заключается в определении двух функций расстояния. Одна из них описывает, насколько друг от друга отличаются содержания двух изображений (Lcontent). Вторая функция описывает разницу между двумя стилями изображений (Lstyle). Получив три изображения (желаемый стиль, желаемый контент и входное изображение), сеть пытается преобразовать входное изображение так, чтобы минимизировать его расстояние Lcontent с изображением контента и расстояние Lstyle с изображением стиля.

**О чём пример**

Пример освещает следующие аспекты:

* моментальное исполнение (англ. Eager Execution) — использование библиотеки TensorFlow, которая позволяет выполнять операции незамедлительно, без построения графов. [Тут](https://www.tensorflow.org/guide/eager) можно узнать больше о моментальном исполнении, а увидеть в действии можно [тут](https://www.tensorflow.org/tutorials);
* работа с [functional API](https://keras.io/getting-started/functional-api-guide/" \t "_blank) для определения модели — вы будете использовать подмножество моделей, чтобы получить доступ к важным промежуточным функциям активации с помощью functional API;
* использование карт признаков подготовленной модели;
* создание собственных циклов обучения — вы научитесь минимизировать заданные потери входных параметров.

Выполняя перенос стиля, вы проделаете следующие шаги:

1. Визуализация данных.
2. Базовая предварительная обработка/подготовка данных.
3. Настройка функций потери.
4. Создание модели.
5. Оптимизация функции потери.

Примечание Этот пост рассчитан на тех, кто уже знаком с базовыми концепциями машинного обучения. Чтобы извлечь максимум из этой статьи, рекомендуется сначала ознакомиться со следующими материалами:

* [https://tproger.ru/translations/6-step-for-building-machine-learning-projects/](https://tproger.ru/translations/6-step-for-building-machine-learning-projects/https:/tproger.ru/translations/math-for-ai-linear-algebra/https:/tproger.ru/video/machine-learning-2014/?autoplay=1https://tproger.ru/digest/learning-neuroweb-all-for-begin/https://tproger.ru/experts/required-ml-skills/)
* [https://tproger.ru/translations/math-for-ai-linear-algebra/](https://tproger.ru/translations/6-step-for-building-machine-learning-projects/https:/tproger.ru/translations/math-for-ai-linear-algebra/https:/tproger.ru/video/machine-learning-2014/?autoplay=1https://tproger.ru/digest/learning-neuroweb-all-for-begin/https://tproger.ru/experts/required-ml-skills/)
* [https://tproger.ru/video/machine-learning-2014/?autoplay=1](https://tproger.ru/translations/6-step-for-building-machine-learning-projects/https:/tproger.ru/translations/math-for-ai-linear-algebra/https:/tproger.ru/video/machine-learning-2014/?autoplay=1https://tproger.ru/digest/learning-neuroweb-all-for-begin/https://tproger.ru/experts/required-ml-skills/)
* [https://tproger.ru/digest/learning-neuroweb-all-for-begin/](https://tproger.ru/translations/6-step-for-building-machine-learning-projects/https:/tproger.ru/translations/math-for-ai-linear-algebra/https:/tproger.ru/video/machine-learning-2014/?autoplay=1https://tproger.ru/digest/learning-neuroweb-all-for-begin/https://tproger.ru/experts/required-ml-skills/)
* [https://tproger.ru/experts/required-ml-skills/](https://tproger.ru/translations/6-step-for-building-machine-learning-projects/https:/tproger.ru/translations/math-for-ai-linear-algebra/https:/tproger.ru/video/machine-learning-2014/?autoplay=1https://tproger.ru/digest/learning-neuroweb-all-for-begin/https://tproger.ru/experts/required-ml-skills/)

**Код**

Вы сможете найти полные исходники [здесь](https://github.com/tensorflow/models/tree/master/research/nst_blogpost). Если вы хотите детально разобрать примеры из этой статьи, то можно перейти на [Colab](https://colab.research.google.com/github/tensorflow/models/blob/master/research/nst_blogpost/4_Neural_Style_Transfer_with_Eager_Execution.ipynb" \t "_blank).

**Реализация**

Начать стоит с включения [моментального исполнения](https://www.tensorflow.org/guide/eager). Это позволит вам работать с техникой переноса стиля наиболее эффективным и понятным образом.

tf.enable\_eager\_execution()

print("Eager execution: {}".format(tf.executing\_eagerly()))

# Изображения контента и стиля, которые будут использованы:

plt.figure(figsize=(10,10))

content = load\_img(content\_path).astype('uint8')

style = load\_img(style\_path)

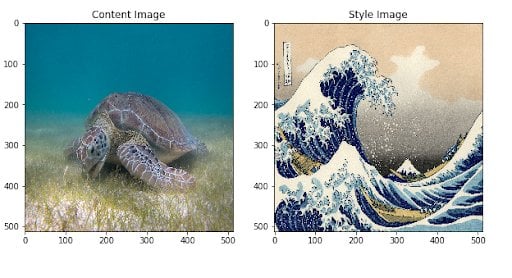
plt.subplot(1, 2, 1)

imshow(content, 'Content Image')

plt.subplot(1, 2, 2)

imshow(style, 'Style Image')

plt.show()



**Определите представления содержания и стиля**

Чтобы получить представление контента и стиля картинки, в первую очередь нужно посмотреть на промежуточные слои модели. Промежуточные слои представляют собой карты признаков, которые по мере углубления становятся более упорядоченными. В этом случае стоит использовать сетевую архитектуру VGG19 — предварительно подготовленную сеть классификации изображений. Промежуточные слои играют важную роль в определении представлений. Для входного изображения нужно сопоставить соответствующие представления на этих промежуточных слоях.

**Почему именно промежуточные слои?**

Вы можете задаться вопросом: почему эти промежуточные выводы дают возможность определить стиль и контент изображения? Чтобы сеть могла классифицировать изображение (чему она уже была обучена), она должна понимать это изображение. Это включает в себя построение из группы пикселей сложных представлений объектов на изображении. Отчасти это объясняет, почему свёрточные нейронные сети могут хорошо обобщать: они способны заметить постоянство и определить особенности, характерные для какого-либо класса (чтобы отличить, например, кота от собаки), не обращая внимания на фоновый шум. Таким образом, где-то между подачей изображения на вход и выводом результата классификации этого изображения, стоит модель, которая находит признаки во входных данных. Соответственно, обращаясь к этой самой промежуточной точке (т. е. слоям), можно без труда получить представление стиля и содержания изображения.

Вот как выглядит работа с промежуточными слоями сети:

# Слой контента, в который помещается карта объектов

content\_layers = ['block5\_conv2']

# Со слоем стиля немного по-другому

style\_layers = ['block1\_conv1',

'block2\_conv1',

'block3\_conv1',

'block4\_conv1',

'block5\_conv1'

]

num\_content\_layers = len(content\_layers)

num\_style\_layers = len(style\_layers)

**Модель**

Сначала нужно загрузить [VGG19](https://keras.io/applications/#vgg19) и подать тензор на вход модели. Это даст возможность получать карты признаков, а впоследствии — представления стиля и контента.

Плюсом VGG19 является её относительная простота (по сравнению с ResNet, Inception и им подобным). Поэтому карты признаков будут лучше подходить для переноса стиля.

Чтобы получить доступ к промежуточным слоям, соответствующим картам признаков стиля и контента, нужно получить характерные выходные данные, используя [Keras functional API](https://keras.io/getting-started/functional-api-guide/" \t "_blank) для определения модели с требуемыми выходными функциями активации.

Благодаря functional API определение модели сводится к банальному определению входных и выходных данных:

model = Model(inputs, outputs).

def get\_model():

""" Создание модели с доступом к промежуточным слоям

Эта функция будет подгружать модель VGG19 и давать доступ к промежуточным слоям.

В дальнейшем эти слои будут использоваться для создания собственной модели для изображения.

Возвращает данные с промежуточных слоёв VGG19 модели.

"""

# Тут подгружается модель (weights=’imagenet’)

vgg = tf.keras.applications.vgg19.VGG19(include\_top=False, weights='imagenet')

vgg.trainable = False

# Получение соответствующих слоёв стиля и контента

style\_outputs = [vgg.get\_layer(name).output for name in style\_layers]

content\_outputs = [vgg.get\_layer(name).output for name in content\_layers]

model\_outputs = style\_outputs + content\_outputs

# Построение модели

return models.Model(vgg.input, model\_outputs)

В приведённом выше коде подгружается подготовленная сеть классификации изображений. После этого нужно взять необходимые слои, про которые говорилось ранее. Затем нужно определить модель. Это можно сделать, настроив входы для изображения и выходы для слоёв стиля и контента. Таким образом вы сможете создать модель, которая на входе принимает изображение, а на выходе выдаёт промежуточные слои для стиля и контента.

**Определение и создание функций потерь (расстояний Lcontent и Lstyle)**

**Функция потерь для контента**

Определить функцию потерь для содержимого на самом деле довольно просто. Нужно передать сети два изображения: изображение желаемого стиля и базовое. После этого вы получите промежуточные слои вашей модели. И единственное, что остаётся, это рассчитать Евклидово расстояние между двумя промежуточными представлениями этих изображений.



[Евклидова, L1 и Чебышёва — 3 основные метрики, которые пригодятся в Data Science](https://tproger.ru/translations/3-basic-distances-in-data-science/)

[tproger.ru](https://tproger.ru/translations/3-basic-distances-in-data-science/)

Если быть точным, то функция потерь описывает расстояние содержимого (Lcontent) между входным изображением x и изображением контентаp.Пусть тогда Cₙₙ будет предварительно обученной глубокой свёртываемой нейронной сетью. Опять же в этом случае будет использоваться VGG19.

Допустим, X— это любое изображение, тогда Cₙₙ(x) — это сеть, на вход которой подаётся X. Пусть тогда Fˡᵢⱼ(x) ∈ Cₙₙ(x)иPˡᵢⱼ(x) ∈ Cₙₙ(x) описывает соответствующие промежуточные представления объектов сети, принимающей X и P. Тогда Lcontent можно будет рассчитать по следующей формуле:



Таким образом обратное распространение обеспечивается так, чтобы минимизировать потерю контента. Нужно менять первоначальное изображение до тех пор, пока оно не сгенерирует аналогичный выход.

Реализовать это довольно просто. Как и в прошлом случае, на вход нужно подать карту признаков со слоя L сети со входом X, входное изображение и P — изображение контента. На выходе получится расстояние Lcontent.

def get\_content\_loss(base\_content, target):

return tf.reduce\_mean(tf.square(base\_content - target))

**Функция потерь для стиля**

Расчёт функции потерь для стиля немного сложнее, но базируется на том же принципе. В этот раз на вход сети нужно подавать входное изображение и картинку стиля. Но теперь, вместо того чтобы сравнивать «сырые» данные с выходов базового и стиля изображения, нужно сравнить матрицы Грама этих двух выходов.

С математической точки зрения этот процесс заключается в описании функции потерь для стиля главного изображения (X) и изображения стиля (A) и расстояния между представлениями (матрица Грама) стиля этих двух картинок.

Представление стиля картинки можно описать как корреляцию между различными ответами фильтра матрицы Gˡ, где Gˡᵢⱼ — это внутреннее произведение между векторизированной картой признаков i и j в слое L.

Чтобы создать стиль для входного изображения, нужно выполнить градиентный спуск от изображения содержимого. Это нужно для того, чтобы трансформировать входное изображение в нечто похожее на изображение стиля. Это можно сделать, минимизировав среднее квадратичное расстояние между объектом корреляции карты стиля и входным изображением. Суммарное влияние каждого слоя на функцию потерь можно описать следующей формулой:



где Gˡᵢⱼ и Aˡᵢⱼ — это соответствующие представления на слое L входного изображения X и изображения стиля A. Nl описывает количество карт объектов, каждая из которых имеет размер Ml = высота \* ширина. Исходя из этого, функция потерь всех слоёв будет такой:



где взвешивается влияние потери каждого слоя от какого-либо фактора wl. В этом случае все слои «взвешиваются» одинаково:



А вот, собственно, и реализация:

def gram\_matrix(input\_tensor):

# Сначала идёт канал изображения

channels = int(input\_tensor.shape[-1])

a = tf.reshape(input\_tensor, [-1, channels])

n = tf.shape(a)[0]

gram = tf.matmul(a, a, transpose\_a=True)

return gram / tf.cast(n, tf.float32)

def get\_style\_loss(base\_style, gram\_target):

"""Принимает два изображения измерений h, w, c"""

# высота, ширина и количество фильтров в каждом слое

height, width, channels = base\_style.get\_shape().as\_list()

gram\_style = gram\_matrix(base\_style)

return tf.reduce\_mean(tf.square(gram\_style - gram\_target))

**Градиентный спуск**

Если вы не знакомы с градиентным спуском или обратным распространением, то [вот ресурс](https://developers.google.com/machine-learning/crash-course/reducing-loss/gradient-descent), чтобы это исправить.

Чтобы минимизировать потери при переносе стиля, понадобится оптимизатор [Adam](https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adam" \t "_blank). Для минимизации нужно многократно обновлять выходное изображение: не стоит как-либо изменять веса в сети. Вместо этого можно тренировать вход изображения. Чтобы это сделать, нужно понять, каким образом рассчитываются потери и градиенты. Используя Adam, можно понять функциональность autograd/gradient tape в собственных циклах обучения.

**Расчёт потери и градиентов**

Нужно создать всего лишь одну маленькую функцию, которая будет подгружать изображения стиля и контента, а потом передавать их сети. В будущем это даст представления признаков стиля и контента в модели:

def get\_feature\_representations(model, content\_path, style\_path):

"""Функция, которая рассчитывает признаки стиля и контента

Эта функция будет просто предварительно подгружать и обрабатывать содержимое и стиль.

Затем эти представления пройдут через сеть, чтобы получить промежуточные слои.

Аргументы:

model: Используемая модель.

content\_path: Путь к изображению содержимого.

style\_path: Путь к изображению стиля.

Возвращает:

Признаки стиля и контента.

"""

# Подгрузка изображений

content\_image = load\_and\_process\_img(content\_path)

style\_image = load\_and\_process\_img(style\_path)

# Одновременная обработка признаков стиля и контента

stack\_images = np.concatenate([style\_image, content\_image], axis=0)

model\_outputs = model(stack\_images)

# Получение представлений признаков

style\_features = [style\_layer[0] for style\_layer in model\_outputs[:num\_style\_layers]]

content\_features = [content\_layer[1] for content\_layer in model\_outputs[num\_style\_layers:]]

return style\_features, content\_features

Для расчёта градиента тут используется [tf.GradientTape](https://www.tensorflow.org/guide/eager" \l "computing_gradients" \t "_blank). Этот способ даёт преимущество использования автоматического дифференцирования, доступного благодаря трассировке последующих вычислений градиента. В этом случае во время прямого прохода операции кэшируются. Это даст возможность рассчитать потери градиента на обратном проходе.

def compute\_loss(model, loss\_weights, init\_image, gram\_style\_features, content\_features):

"""Эта функция рассчитывает полную потерю.

Аргументы:

model: Модель с нужными промежуточными слоями.

loss\_weights: Вес каждого компонента для каждой функции потерь.

(вес для стиля, для контента и общий).

init\_image: Первичное изображение. Это то изображение, которое в процессе оптимизации будет обновляться.

gram\_style\_features: Предварительные вычисления матрицы Грама соответствующих слоёв.

content\_features: Предварительные вычисления нужных слоёв контента.

Возвращает:

Общие потери, потери для стиля, контента и вариационные потери

"""

style\_weight, content\_weight, total\_variation\_weight = loss\_weights

# Прогонка изображение через модель. Это даст представления контента и стиля.

# Из-за использования мгновенного выполнения, эта модель вызывается как и любая другая функция.

model\_outputs = model(init\_image)

style\_output\_features = model\_outputs[:num\_style\_layers]

content\_output\_features = model\_outputs[num\_style\_layers:]

style\_score = 0

content\_score = 0

# Суммирует потерю стиля со всех слоёв

# Тут одинаково взвешиваются потери каждого слоя.

weight\_per\_style\_layer = 1.0 / float(num\_style\_layers)

for target\_style, comb\_style in zip(gram\_style\_features, style\_output\_features):

style\_score += weight\_per\_style\_layer \* get\_style\_loss(comb\_style[0], target\_style)

# Суммирование потерь контента со всех слоёв

weight\_per\_content\_layer = 1.0 / float(num\_content\_layers)

for target\_content, comb\_content in zip(content\_features, content\_output\_features):

content\_score += weight\_per\_content\_layer\* get\_content\_loss(comb\_content[0], target\_content)

style\_score \*= style\_weight

content\_score \*= content\_weight

total\_variation\_score = total\_variation\_weight \* total\_variation\_loss(init\_image)

# Получение суммарной потери

loss = style\_score + content\_score + total\_variation\_score

return loss, style\_score, content\_score, total\_variation\_score

В итоге расчёт градиента сводится к этому:

def compute\_grads(cfg):

with tf.GradientTape() as tape:

all\_loss = compute\_loss(\*\*cfg)

# Расчёт градиента изображения

total\_loss = all\_loss[0]

return tape.gradient(total\_loss, cfg['init\_image']), all\_loss

**Запуск процесса переноса стиля**

Вот так выглядит фактический запуск сети:

def run\_style\_transfer(content\_path,

style\_path,

num\_iterations=1000,

content\_weight=1e3,

style\_weight = 1e-2):

display\_num = 100

# В этом случае не нужно обучать каждый слой модели. Поэтому параметр trainability нужно выставить в false.

model = get\_model()

for layer in model.layers:

layer.trainable = False

# Получение представлений признаков стиля и контента (из промежуточных слоёв)

style\_features, content\_features = get\_feature\_representations(model, content\_path, style\_path)

gram\_style\_features = [gram\_matrix(style\_feature) for style\_feature in style\_features]

# Загрузка изначального изображения

init\_image = load\_and\_process\_img(content\_path)

init\_image = tfe.Variable(init\_image, dtype=tf.float32)

# Создание оптимизатора

opt = tf.train.AdamOptimizer(learning\_rate=10.0)

# Отображение промежуточных изображений

iter\_count = 1

# Сохранение лучшего результата

best\_loss, best\_img = float('inf'), None

# Создание конфигурации

loss\_weights = (style\_weight, content\_weight)

cfg = {

'model': model,

'loss\_weights': loss\_weights,

'init\_image': init\_image,

'gram\_style\_features': gram\_style\_features,

'content\_features': content\_features

}

# Отображение

plt.figure(figsize=(15, 15))

num\_rows = (num\_iterations / display\_num) // 5

start\_time = time.time()

global\_start = time.time()

norm\_means = np.array(1)

min\_vals = -norm\_means

max\_vals = 255 - norm\_means

for i in range(num\_iterations):

grads, all\_loss = compute\_grads(cfg)

loss, style\_score, content\_score = all\_loss

# grads, \_ = tf.clip\_by\_global\_norm(grads, 5.0)

opt.apply\_gradients([(grads, init\_image)])

clipped = tf.clip\_by\_value(init\_image, min\_vals, max\_vals)

init\_image.assign(clipped)

end\_time = time.time()

if loss < best\_loss:

# Обновление лучшей потери и изображения

best\_loss = loss

best\_img = init\_image.numpy()

if i % display\_num == 0:

print('Iteration: {}'.format(i))

print('Total loss: {:.4e}, '

'style loss: {:.4e}, '

'content loss: {:.4e}, '

'time: {:.4f}s'.format(loss, style\_score, content\_score, time.time() - start\_time))

start\_time = time.time()

# Отображение промежуточных изображений

if iter\_count > num\_rows \* 5: continue

plt.subplot(num\_rows, 5, iter\_count)

# Используйте метод .numpy(), чтобы получить конкретный numpy-массив

plot\_img = init\_image.numpy()

plot\_img = deprocess\_img(plot\_img)

plt.imshow(plot\_img)

plt.title('Iteration {}'.format(i + 1))

iter\_count += 1

print('Total time: {:.4f}s'.format(time.time() - global\_start))

return best\_img, best\_loss

На этом всё!

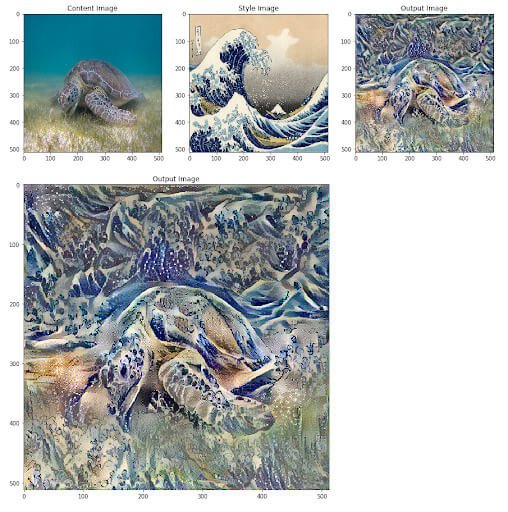
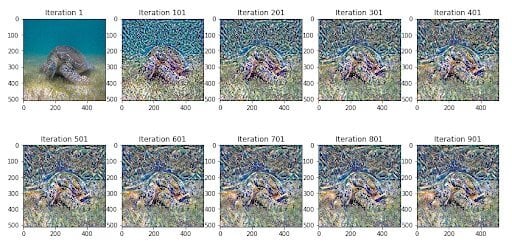
Чтобы запустить нейронный перенос стиля, нужно просто вызвать функцию, передав ей пути к входным изображениям:

best, best\_loss = run\_style\_transfer(content\_path,

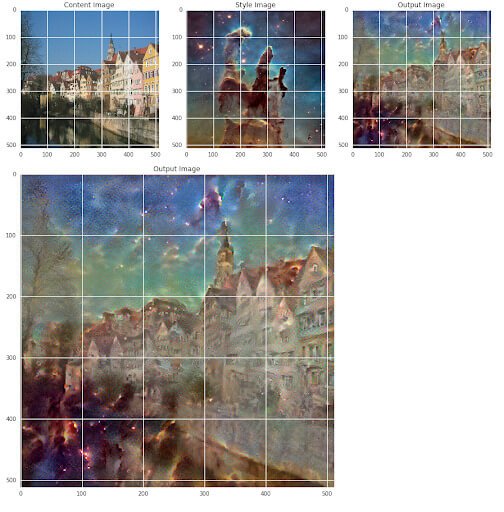
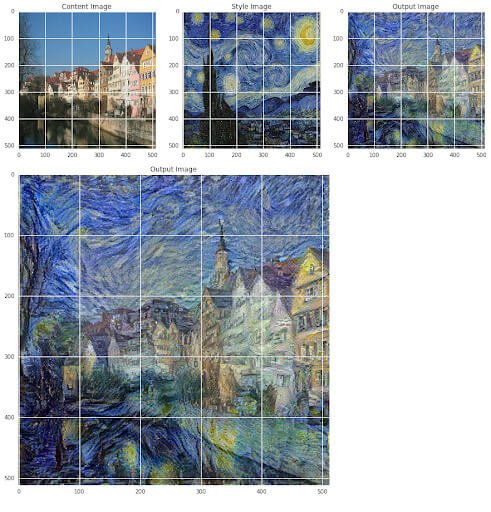
style\_path,

verbose=True,

show\_intermediates=True)



Вот ещё крутые примеры работы сети:



**Ключевые моменты**

В этом примере были разобраны следующие этапы:

* создание нескольких различных функций потерь и использование обратного распространения для входного изображения;
* для этого использовалась предварительно обученная модель и изученные карты признаков для описания содержимого на изображении;
* функциями потерь в основном являлись вычисления расстояний различных представлений;
* всё это выполнялось благодаря собственной модели и моментальным исполнениям;
* построение модели осуществлялось благодаря Functional API;
* моментальное исполнение позволило динамически работать с тензорами, используя естественный поток управления Python;
* управление тензорами велось напрямую, а это в свою очередь облегчило отладку и работу в целом.

**ПРАКТИЧЕСКАЯ ЧАСТЬ РАБОТЫ**

В ходе изучения теоретических материалов теоретических сведений данной работы:

I. Ознакомьтесь с теоретическими сведениями.

II. Изучите инструкцию ниже.

III. Предоставьте ссылку на свою работу в https://colab.research.google.com/ онлайн, в процессе выполнения преподавателю на yvmranh48@gmail.com, через вкладку Share (поделиться). Для отслеживания Вашей работы преподавателем онлайн и возможности консультирования.

IV. Оформите работу, используя снимки экрана.

V. Ссылку на работу и отчет по работе выложите в ментальной карте.

VI. Сделайте алгоритм по Вашей работе в draw.io

VII. Выкладывайте отчет на странице своего сайта.

**Ход работы.**

**Во время работы не забывайте запускать на каждом этапе код.**

|  |  |
| --- | --- |
|  | Создайте новый ноутбук по ссылке:  <https://colab.research.google.com/>. |
| Назовите:  Монтаж художественных образов с помощью нейронных сетей (Ф.И.О.).ipynb.  Подпишите в скобках своим Ф.И.О. |
| Ваш новый ноутбук должен содержать комментарии и иметь законченный вид.  Текст для комментариев можем использовать из инструкции | |
|  | |
|  | Рассмотрим, как работают нейронные сети по монтажу художественных образов. Установим призму.  Добавьте новый код:  !pip install -q git+https://github.com/moritztng/prism.git |
|  | Появится надпись:  Building wheel for prism-style-transfer (setup.py) ... done |
|  | Осуществим импорт библиотек  Добавьте новый код:  from IPython.display import Image |
|  | Скачаем изображение:  Добавьте новый код:  images\_url = 'https://raw.githubusercontent.com/moritztng/prism/master/images'  !wget -q -nc $images\_url/content.jpg $images\_url/style.jpg |
|  | Перенос стиля в оболочке  Добавьте новый код:  !style-transfer content.jpg style.jpg --style\_weight 1000 --use\_amp |
|  | Выводим полученное изображение  Добавьте новый код:  Image('artwork.png') |
| 2021-05-30 15:51:08.846854: I tensorflow/stream\_executor/platform/default/dso\_loader.cc:53] Successfully opened dynamic library libcudart.so.11.0  Iteration: 50 Time: 0.192 Loss: 4.8e+05 Content Loss: 4.0e+05 Style Loss: 7.7e+01 Mean Abs Grad: 2.7e-02 Grad Zeros: 0 Grad Scale: 1.0e+03  Iteration: 100 Time: 0.22 Loss: 3.5e+05 Content Loss: 3.0e+05 Style Loss: 4.4e+01 Mean Abs Grad: 1.2e-02 Grad Zeros: 0 Grad Scale: 2.0e+03  Iteration: 150 Time: 0.242 Loss: 3.1e+05 Content Loss: 2.7e+05 Style Loss: 3.8e+01 Mean Abs Grad: 7.9e-03 Grad Zeros: 0 Grad Scale: 4.1e+03  Iteration: 200 Time: 0.247 Loss: 2.9e+05 Content Loss: 2.5e+05 Style Loss: 3.5e+01 Mean Abs Grad: 6.4e-03 Grad Zeros: 0 Grad Scale: 8.2e+03 | Осуществим тонкую настройку  Добавьте новый код:  # Use exactly the same parameters as previously, but add "--init\_img artwork.png --area 1024 --iter 200".  !style-transfer content.jpg style.jpg --style\_weight 1000 --use\_amp --init\_img artwork.png --area 1024 --iter 200 |
|  | Получим изображение  Добавьте новый код:  Image('artwork.png') |
|  | Передача стилей в Python.  Импортируем библиотеки  Добавьте новый код:  from PIL import Image  from style\_transfer.learn import StyleTransfer |
|  | Создание экземпляра объекта передачи стиля.  Добавьте новый код:  style\_transfer = StyleTransfer(lr=1,                                 content\_weight=1,                                 style\_weight=1e3,                                 content\_weights="{'relu\_4\_2':1}",                                 style\_weights="{'relu\_1\_1':1,'relu\_2\_1':1,'relu\_3\_1':1,'relu\_4\_1':1,'relu\_5\_1':1}",                                 avg\_pool=False,                                 feature\_norm=True,                                 weights='original',                                 preserve\_color='style',                                 device='auto',                                 use\_amp=False,                                 adam=False,                                 optim\_cpu=False,                                 logging=50) |
| Iteration: 50 Time: 0.089 Loss: 8.1e+05 Content Loss: 4.3e+05 Style Loss: 3.8e+02 Mean Abs Grad: 5.7e-01 Grad Zeros: 0 Grad Scale: 6.6e+04  Iteration: 100 Time: 0.0965 Loss: 5.6e+05 Content Loss: 3.9e+05 Style Loss: 1.7e+02 Mean Abs Grad: 8.4e-02 Grad Zeros: 0 Grad Scale: 6.6e+04  Iteration: 150 Time: 0.101 Loss: 4.9e+05 Content Loss: 3.6e+05 Style Loss: 1.3e+02 Mean Abs Grad: 4.2e-02 Grad Zeros: 0 Grad Scale: 6.6e+04  Iteration: 200 Time: 0.102 Loss: 4.6e+05 Content Loss: 3.5e+05 Style Loss: 1.1e+02 Mean Abs Grad: 2.8e-02 Grad Zeros: 0 Grad Scale: 6.6e+04  Iteration: 250 Time: 0.103 Loss: 4.4e+05 Content Loss: 3.4e+05 Style Loss: 9.8e+01 Mean Abs Grad: 2.3e-02 Grad Zeros: 0 Grad Scale: 6.6e+04  Iteration: 300 Time: 0.104 Loss: 4.3e+05 Content Loss: 3.4e+05 Style Loss: 9.1e+01 Mean Abs Grad: 2.9e-02 Grad Zeros: 0 Grad Scale: 6.6e+04  Iteration: 350 Time: 0.105 Loss: 4.2e+05 Content Loss: 3.3e+05 Style Loss: 8.7e+01 Mean Abs Grad: 1.6e-02 Grad Zeros: 0 Grad Scale: 6.6e+04  Iteration: 400 Time: 0.106 Loss: 4.2e+05 Content Loss: 3.3e+05 Style Loss: 8.4e+01 Mean Abs Grad: 2.1e-02 Grad Zeros: 0 Grad Scale: 6.6e+04  Iteration: 450 Time: 0.107 Loss: 4.1e+05 Content Loss: 3.3e+05 Style Loss: 8.2e+01 Mean Abs Grad: 1.0e-02 Grad Zeros: 0 Grad Scale: 6.6e+04  Iteration: 500 Time: 0.108 Loss: 4.1e+05 Content Loss: 3.3e+05 Style Loss: 8.0e+01 Mean Abs Grad: 9.6e-03 Grad Zeros: 0 Grad Scale: 6.6e+04 | Выполним перенос стиля.  Добавьте новый код:  artwork = style\_transfer(Image.open('content.jpg'), Image.open('style.jpg'),                           area=512, init\_random=False, init\_img=None, iter=500)  artwork.save('artwork.png') |
|  | Получим изображение.  Добавьте новый код:  artwork |
|  | Определим потери на участке.  Добавьте новый код:  %load\_ext tensorboard  %tensorboard --logdir runs |
| Мы натренировали модель нейронной сети для преобразования изображений.  Самостоятельно загрузим новое изображение в этот блокнот преобразуем изображение и выгрузим новое полученное изображение на Google Drive. | |
| **Составьте алгоритм. Работу выполняйте онлайн.**  **Ссылку на онлайн файл отправьте в чат ивстройте в ментальную карту. Можете встроить отчет в на страницу своего сайта.** | |

**Фонд оценочных средств текущего контроля успеваемости**

Таблица 9

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № п/п | Раздел, тема, вид занятия | Контролируемые компетенции, результаты обучения | Оценочное средство | Основные показатели оценки результата | Баллы (оценка) |
| **Контактная работа слушателей с преподавателем**  **(лекции, практические занятия)** | | | | | |
| 1 | Лекционное занятие.  Сквозные технологии и цифровая экономика.  Введение в машинное обучение (machine learning, ML).  Цифровизация как средство достижения целей современной организации Практическое занятие  Информационная безопасность государственных данных.  Техническая защита информации.  Цифровой Этикет | ПК-27. владением методами и программными средствами обработки деловой информации, навыками работы со специализированными кадровыми компьютерными  программами, способностью взаимодействовать со службами информационных технологий и эффективно использовать корпоративные информационные системы при решении задач управления персоналом | Кейс-задание | все задания кейса выполнены обучающимся без ошибок или сделано не менее 50% правильных заданий | 3-5 |
| 2 | Лекционное занятие.  Этика работы с данными.  Нормы этики и морали в сфере разработки ML | ПК-28. знанием корпоративных коммуникационных каналов и средств передачи информации, владением навыками информационного обеспечения процессов внутренних  коммуникаций | Кейс-задание | все задания кейса выполнены обучающимся без ошибок или сделано не менее 50% правильных заданий | 3-5 |
| 3 | Лекционное занятие.  Введение в теорию ML. Основные термины и понятия. Принципы ML.  Цифровые технологии и инструменты в профессиональной деятельности государственного служащего.  Разбор кейсов по применению технологий ML в государственном управлении и иных приоритетных отраслях экономики. | ПК-27. владением методами и программными средствами обработки деловой информации, навыками работы со специализированными кадровыми компьютерными  программами, способностью взаимодействовать со службами информационных технологий и эффективно использовать корпоративные информационные системы при решении задач управления персоналом | Кейс-задание | все задания кейса выполнены обучающимся без ошибок или сделано не менее 50% правильных заданий | 3-5 |
| 4 | Практическое занятие.  Обработка массивов данных с помощью ML.  Практическое занятие.  Нейросети для анализа таблиц.  Практическое занятие.  Нейросети для построения диаграмм и упрощения работы с электронными таблицами | ПК-28. знанием корпоративных коммуникационных каналов и средств передачи информации, владением навыками информационного обеспечения процессов внутренних  коммуникаций | Кейс-задание | все задания кейса выполнены обучающимся без ошибок или сделано не менее 50% правильных заданий | 3-5 |
| 5 | Лекционное занятие.  Системы, сервисы и платформы ML. Выбор и комбинирование  Практическое занятие.  Внедрение ML-моделей в процессы организации и их сопровождение.  Оценка качества ML-систем | ПК-27. владением методами и программными средствами обработки деловой информации, навыками работы со специализированными кадровыми компьютерными  программами, способностью взаимодействовать со службами информационных технологий и эффективно использовать корпоративные информационные системы при решении задач управления персоналом | Кейс-задание | все задания кейса выполнены обучающимся без ошибок или сделано не менее 50% правильных заданий | 3-5 |
| 6 | Практическое занятие.  Внедрение ML-моделей в процессы организации и их сопровождение.  Оценка качества ML-систем | ПК-28. знанием корпоративных коммуникационных каналов и средств передачи информации, владением навыками информационного обеспечения процессов внутренних  коммуникаций | Кейс-задание | все задания кейса выполнены обучающимся без ошибок или сделано не менее 50% правильных заданий | 3-5 |
| 7 | Практическое занятия  Инструменты анализа данных и визуализация данных.  Цифровые инструменты в сфере государственного управления: сервисы для управления целями и задачами в цифровой среде. | ПК-28. знанием корпоративных коммуникационных каналов и средств передачи информации, владением навыками информационного обеспечения процессов внутренних  коммуникаций | Кейс-задание | все задания кейса выполнены обучающимся без ошибок или сделано не менее 50% правильных заданий | 3-5 |
| 8 | Практическое занятие  Цифровые инструменты в сфере государственного управления: продукты, предназначенные для создания, просмотра, редактирования файлов.  Цифровые инструменты в сфере государственного управления: сервисы создания видеоконференций. | ПК-28. знанием корпоративных коммуникационных каналов и средств передачи информации, владением навыками информационного обеспечения процессов внутренних  коммуникаций | Кейс-задание | все задания кейса выполнены обучающимся без ошибок или сделано не менее 50% правильных заданий | 3-5 |

**Типовые практические задания**

*Средство оценивания - Кейс-задание.*

**Примерный вариант Кейс-задания**

Тема: **Работа с Google Drive и в среде обработки данных Google Colab Laboratory.**

Цель: уметь работать с Google Drive и в среде обработки данных Google Colab Laboratory.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Машинное обучение, нейронные сети, искусственный интеллект

Краткое руководство по запуску ваших скриптов Python в Google Colab Laboratory.

Если вы ищете интерактивный способ запуска скрипта Python, скажем, вы хотите запустить проект машинного обучения с парой друзей, не смотрите дальше -Google Colab лучшее решение для вас. Вы можете работать в Интернете и сохранять свой код на локальном диске Google, и это позволяет вам

• Запустите ваши скрипты с бесплатными графическими процессорами (и TPU!)

• Используйте предустановленные библиотеки Python и функции Jupyter Notebook

• Работайте где угодно, на облаках

• Делитесь кодами и сотрудничайте с коллегами

Google Colab = Jupyter Notebook + бесплатные графические процессоры

Ход работы.

1. Изучите инструкции по работе с Google Drive

https://support.google.com/drive/answer/9191990?hl=ru

https://youtu.be/ky94MN4dx3c

https://youtu.be/XYWDKAwzuv0

https://youtu.be/WvQIZBS7PSo

Создайте свой Google Drive, Google дакумент, Google таблицу.

2. Инструменты анализа больших данных в машинном обучении.

Бесплатная облачная платформа для нейросетей Google Colab | Нейросети на Python

https://youtu.be/rt4806DzfUY

https://youtu.be/GquJSJ4KU2E

Практическая часть работы.

1. Рекомендуем отчет подготавливать синхронно в процессе работы.

Создайте свой Google Drive.

Видео инструкции:

https://youtu.be/xo2lZW8s3GQ

2. В Google диске создайте папку, в которой должен располагаться Google документ с отчетом по проделанной работе, Google таблица с любыми экономическими данными

3. Предоставьте доступ у папке на Google диске преподавателю (gmail для доступа yvmranh48@gmail.com).

Видео инструкции:

https://youtu.be/JZJJZJbw5c8

Войдете в среду разработки Google Colab и начните работу

4. Пройдите по ссылке https://colab.research.google.com/notebooks/welcome.ipynb?hl=ru

5. Создайте свой первый ноутбук по ссылке https://colab.research.google.com/. Назовите

Мой первый ноутбук Ф.И.О. (предоставьте доступ для редактирования преподавателю gmail для доступа yvmranh48@gmail.com).

6. Выполните работу по рекомендациям к началу работы с Google Colab данными по ссылке https://colab.research.google.com/notebooks/welcome.ipynb?hl=ru в пункте 6

Видео инструкции:

https://youtu.be/7p6s7AsdtIo

https://youtu.be/NAohorPvUx0

7. Подготовьте отчет по проделанной работе и разместите в своем Google Drive.

**Примерный вариант Кейс-задания**

Тема: **Распознавание образов с помощью свёрточных нейронных сетей ipynb.**

Цель: научиться использовать свёрточные нейронные сети ipynb для распознавания образов.

Материальное и дидактическое оснащение: Методические рекомендации по выполнению практической работы.

В ходе изучения теоретических материалов теоретических сведений данной работы:

I. Ознакомьтесь с теоретическими сведениями после практического занятия для понимания теории просмотрите видео.

II. Изучите инструкцию ниже.

III. Предоставьте ссылку на свою работу в https://colab.research.google.com/ онлайн, в процессе выполнения преподавателю на yvmranh48@gmail.com, через вкладку Share (поделиться). Для отслеживания Вашей работы преподавателем онлайн и возможности консультирования

IV. Оформите работу, используя снимки экрана.

V. Ссылку на работу и отчет по работе выложите в ментальной карте.

VI. Сделайте алгоритм по Вашей работе в draw.io

VII. Выкладывайте отчет на странице своего сайта.

**Ход работы.**

**Во время работы не забывайте запускать на каждом этапе код.**

|  |  |
| --- | --- |
|  | Создайте новый ноутбук по ссылке:  <https://colab.research.google.com/>. |
| Назовите:  Распознавание образов с помощью свёрточных нейронных сетей (Ф.И.О.).ipynb.  Подпишите в скобках своим Ф.И.О. |
| Ваш новый ноутбук должен содержать комментарии и иметь законченный вид.  Текст для комментариев можем использовать из инструкции | |
|  | |
|  | Рассмотрим, как работают свёртки, создавая базовую свертку на 2D изображении в масштабе серого (из оттенков серого цвета). Сначала мы можем загрузить изображение, взяв пример под названием "восхождение" из пакета "Scipy". Это красивая, встроенная картинка с большим количеством углов и линий. Добавьте новый код:  import cv2  import numpy as np  from scipy import misc  i = misc.ascent() |
|  | Далее мы можем использовать библиотеку pyplot для того, чтобы нарисовать изображение, чтобы мы знали, как оно выглядит.  Добавьте новый код:  import matplotlib.pyplot as plt  plt.grid(False)  plt.gray()  plt.axis('off')  plt.imshow(i)  plt.show() |
|  | Получите изображение. |
|  | Изображение хранится в виде массива numpy, поэтому мы можем создать преобразованное новое изображение, просто скопировав этот массив. Давайте также получим размеры изображения, чтобы потом можно было пройти через значения массива/матрицы через итерирование в цикле.  Добавьте новый код:  i\_transformed = np.copy(i)  size\_x = i\_transformed.shape[0]  size\_y = i\_transformed.shape[1] |
|  | Теперь мы можем создать фильтр в виде массива/матрицы размером 3x3.  # Этот фильтр хорошо определяет границы в изображении.  # Он создаёт свертки, которые проходят только через острые края и прямые линии  # Поэкспериментируйте с различными значениями для практики.  Добавьте новый код:  #filter = [ [0, 1, 0], [1, -4, 1], [0, 1, 0]]  # Ещё пара фильтов :)  #filter = [ [-1, -2, -1], [0, 0, 0], [1, 2, 1]]  filter = [ [-1, 0, 1], [-2, 0, 2], [-1, 0, 1]]  # Если все цифры в фильтре не складываются до значения 0 или 1,  # вам скорее всего потребуется немало усилий, чтобы добиться этого.  # так, например, если ваши веса 1,1,1 1,2,1 1,1,1  # Они складываются до 10, так что вы бы установили вес .1, если хотите их нормализовать.  weight  = 1 |
|  | Теперь давайте создадим конволюцию. Выполним итерацию по изображению, оставив поле в 1 пиксел, и умножим каждого из соседей текущего пиксела на значение, заданное в фильтре.  Т.е. сосед текущего пиксела над ним и слева будет умножаться на верхний левый элемент в фильтре и т.д. и т.п. Затем умножим результат на вес, а затем убедимся, что результат находится в диапазоне 0-255  Наконец-то мы загрузим новое значение в преобразованное изображение.  Добавьте новый код:  for x in range(1,size\_x-1):    for y in range(1,size\_y-1):        convolution = 0.0        convolution = convolution + (i[x - 1, y-1] \* filter[0][0])        convolution = convolution + (i[x, y-1] \* filter[0][1])        convolution = convolution + (i[x + 1, y-1] \* filter[0][2])        convolution = convolution + (i[x-1, y] \* filter[1][0])        convolution = convolution + (i[x, y] \* filter[1][1])        convolution = convolution + (i[x+1, y] \* filter[1][2])        convolution = convolution + (i[x-1, y+1] \* filter[2][0])        convolution = convolution + (i[x, y+1] \* filter[2][1])        convolution = convolution + (i[x+1, y+1] \* filter[2][2])        convolution = convolution \* weight        if(convolution<0):          convolution=0        if(convolution>255):          convolution=255        i\_transformed[x, y] = convolution |
|  | Теперь мы можем построить изображение, чтобы увидеть эффект свёртки!  Добавьте новый код:  # Plot the image. Note the size of the axes -- they are 512 by 512  plt.gray()  plt.grid(False)  plt.imshow(i\_transformed)  #plt.axis('off')  plt.show() |
|  | Получите изображение. |
|  | Этот код покажет (2, 2) операцию подвыборки. Идея здесь в том, чтобы провести итерирование по изображению, и посмотреть на пиксель, а также те, которые находятся в непосредственной близости справа, внизу и справа-внизу по диагонали. Возьмите из них самый большой по значению и оставьте его, загрузив в новое изображение. Таким образом, новое изображение будет 1/4 размера старого - при этом размеры по каждой из осей X и Y будут уменьшены вдвое. Вы увидите, что, несмотря на это сжатие в 1/4, признаки изображения сохраняются!  Добавьте новый код:  new\_x = int(size\_x/2)  new\_y = int(size\_y/2)  newImage = np.zeros((new\_x, new\_y))  for x in range(0, size\_x, 2):    for y in range(0, size\_y, 2):      pixels = []      pixels.append(i\_transformed[x, y])      pixels.append(i\_transformed[x+1, y])      pixels.append(i\_transformed[x, y+1])      pixels.append(i\_transformed[x+1, y+1])      newImage[int(x/2),int(y/2)] = max(pixels)  # Plot the image. Note the size of the axes -- now 256 pixels instead of 512  plt.gray()  plt.grid(False)  plt.imshow(newImage)  #plt.axis('off')  plt.show() |
|  | Получите изображение. |
| **Составьте алгоритм. Работу выполняйте онлайн.**  **Ссылку на онлайн файл отправьте в чат и встройте в ментальную карту. Можете встроить отчет на страницу своего сайта.** | |

**Шкала оценивания текущего контроля**

|  |  |  |
| --- | --- | --- |
| Вид работ | Критерий оценивания | Оценка |
| Практическое задание | Демонстрирует знание основ машинного обучения | зачтено |

**Содержание итоговой аттестации**

Итоговая аттестация проводится в виде итогового тестирования. Трудоемкость итогового тестирования составляет 2 часа.

**Примерный список вопросов для итогового тестирования**

1. Выберете, какими могут быть числовые данные (несколько вариантов ответа):

1) числовые

+2) дискретные

3) символьные

+4) непрерывные

2. Команды Unix — Linux-удалить file

1) cd

+2) rm file

3) pwd

4) ls -al

3. Выберете, свойство, характеризующее объект (цвет глаз, температура воды):

+1) атрибут

2) числовые характеристики выборки

3) дихотомическое, порядковое

4) знания

4. Команды Unix — Linux-rm -r dir означает

1) удалить file

+2) удалить каталог dir

3) удалить форсированно file

4) удалить форсированно каталог dir

5. Что такое совокупность сведений, которые отображают целостность описаний, соответствующие некоторым уровням осведомленности об описываемом вопросе, предмете:

1) атрибут

+2) знания

3) правила, в соответствии с которыми объектам присваивается числа

4) все выше перечисленное

6. Показать информацию о пользователе

+1) finger user

2) uname -a

3) w

4) uname

7. Выберете, какими могут быть переменные данные (несколько вариантов ответа):

+1) числовые

+2) символьные

3) дискретные

4) непрерывные

8. Создать tar-архив с именем file.tar содержащий files

1) tar czf file.tar.gz files

2) tar xf file.tar нт 2

+3) tar cf file.tar files

4) gzip file 4

9. Выберете все типы существующих типов шкал измерений (несколько вариантов ответа):

1) процесс присвоения чисел, согласно определенному правилу

2) интеллектуальный анализ данных

+3) дихотомическое, порядковое

+4) интервальное, номинальное, относительное

10. Клавиатурные сочетания - разлогиниться, тоже самое, что и exit

1) Ctrl+Z

2) Ctrl+C

4) Ctrl+D

5) Ctrl+W

11. Статистика это:

+1) числовые характеристики выборки

2) правила, в соответствии с которыми объектам присваивается числа

3) числовые характеристики генер.совокупности

4) охватывает анализ данных, который полагается на агрегацию

12. Команда - повторить последнюю команду

+1) !!

2) Ctrl+U

3) Ctrl+C

4) Ctrl+W

13. Данные, состоящие из записей, каждый из которых состоит из набора атрибутов:

+1) табличные

2) числовые

3) атрибут

4) символьные

14. Специально разработанные Google специализированные интегральные схемы

1) Алгоритмы

+2) Tensor Processing Units

3) Библиотеки кодов

4) Свёрточные нейронные сети

15. Шкала это:

1) табличные данные

2) атрибут таблицы

+3) правила, в соответствии с которыми объектам присваивается числа

4) присвоение чисел, согласно определенному правилу

16. Операции реляционной алгебры - объединение

1) WHERE

2) JOI

+3) UNION

4) INTERSECT

17. Операции реляционной алгебры - пересечение

1) WHERE

2) JOI

3) UNION

+4) INTERSECT

18. Операции реляционной алгебры - разность

1) WHERE

2) JOI

+3) UNION

4) MINUS

19. Операции реляционной алгебры - декартово произведение

1) WHERE

+ 2) TIMES

3) UNION

4) INTERSECT

20. Операции реляционной алгебры - ограничение

+1) WHERE

2) JOI

3) UNION

4) INTERSECT

21. Операции реляционной алгебры - реляционное деление

1) WHERE

2) JOI

3) UNION

+4) DIVIDE BY

22. Бизнес- аналитика:

1) анализ данных

2) охватывает анализ данных, который полагается на агрегацию

3) числовые характеристики генер. совокупности

4) совокупность методов планирования эксперимента, сбора данных их предприятия и обобщения, а также анализа и получения вывода на основании этих данных

23. Множества всех атрибутов

+1) Заголовок таблицы

2) Кортеж

3) Атрибут

4) Домен

24. Процесс получения программы новых знаний – это:

+1) машинное обучение

2) анализ данных, который полагается на агрегацию

3) совокупность методов планирования эксперимента, сбора данных их предприятия и обобщения, а также анализа и получения вывода на основании этих данных

4) правила, в соответствии с которыми объектам присваивается числа

25. Процесс поддержки принятия решений, основанный на поиске в данных скрытых закономерностей

1) Рекомендательные системы

2) Машинное обучение

+3) Data Mining

4) Olap системы

Таблица 11

**Критерии оценивания слушателя при проведении итоговой аттестации**

|  |  |  |
| --- | --- | --- |
| Оценка | Шкала оценивания | Требования к знаниям |
| зачтено | Дано не менее 50% правильных ответов | Демонстрирует способность к анализу больших данных с использованием существующей в организации методологической и технологической инфраструктуры |
| незачтено | Дано менее 50% правильных ответов | Не демонстрирует способность к анализу больших данных с использованием существующей в организации методологической и технологической инфраструктуры |

**Приложение 1 Рецензии (внешняя и внутренняя)**

1. Компетенция сформулирована на основе профессионального стандарта «Специалист по большим данным». [↑](#footnote-ref-1)
2. Компетенция взята из ФГОС ВО 38.03.03.«Управление персоналом». [↑](#footnote-ref-2)
3. Компетенция взята из ФГОС ВО 38.03.03.«Управление персоналом». [↑](#footnote-ref-3)
4. ПЗ – практическое задание. [↑](#footnote-ref-4)
5. Компетенция сформулирована на основе профессионального стандарта «Специалист по большим данным». [↑](#footnote-ref-5)
6. Компетенция взята из ФГОС ВО 38.03.03.«Управление персоналом». [↑](#footnote-ref-6)
7. Компетенция взята из ФГОС ВО 38.03.03.«Управление персоналом». [↑](#footnote-ref-7)