Содержание

Цели и задачи освоения дисциплины

Цели освоения дисциплины: развитие навыков участия в научно-исследовательской деятельности, знакомство с современными методами машинного обучения и их практическими применениями, развитие навыков презентации результатов исследования и оформления презентационных материалов.

Задачи:

- освоение основных методов машинного обучения и искусственного интеллекта;

- приобретение навыков применения методов машинного обучения и искусственного интеллекта для решения финансовых задач.

Место дисциплины в структуре ОПОП

Учебная дисциплина «Научно-исследовательский семинар» (1-й курс магистратуры, 2-й семестр) относится к обязательной части блока дисциплин (модулей).

Для изучения данной учебной дисциплины необходимы знания, умения и навыки, формируемые дисциплинами «Методы оптимизации для машинного обучения», «Избранные вопросы теории вероятностей и математической статистики», «Питон для анализа данных», «Основы нейронных сетей».

Знания и навыки, полученные в ходе изучения данной дисциплины, могут использоваться для решения профессиональных задач в научно-исследовательской, научно-производственной и проектной деятельности.

Образовательные технологии

При проведении лекций и практических занятий используются следующие образовательные технологии:

— мультимедийные лекции

— электронные формы контроля

— самотестирование студентов

Учебный процесс базируется на концепции компетентностного обучения, ориентированного на формирование конкретного перечня профессиональных компетенций, актуализацию получаемых теоретических знаний. Развертывание компетентностной модели обучения предполагает широкое применение инновационных способов организации учебного процесса, в т.ч. применение метода проектного обучения, технологий управляемого самостоятельного обучения в том числе балльно-рейтинговой системы, а также внедрение системы онлайн-поддержки внеаудиторной работы студентов.

Дисциплина может быть реализована частично или полностью с использованием ЭИОС Университета (ЭО и ДОТ). Аудиторные занятия и другие формы контактной работы обучающихся с преподавателем могут проводиться с использованием платформ Microsoft Teams и MOODLE, в том числе, в режиме онлайн-лекций и онлайн- семинаров.

Тематика научных докладов по разделам

Раздел 1. Классические методы машинного обучения и практические прикладные задачи

Тема 1.1. Регрессии

Применение регрессий: линейной, логистической, LASSO, ElasticNet. Вопросы регуляризации и переобучения.

Темы докладов:

1. Нахождение линейной регресии при анализе данных
2. Нахождение логистической регрессии при анализе данных
3. Применение регрессии LASSO в прикладных задачах
4. Применение регрессии ElasticNet в прикладных задачах
5. Классические методы машинного обучения
6. Регуляризация данных
7. Переобучение данных

Тема 1.2. Деревья

Деревья, беггинг, бустинг.

Раздел 2. Глубокое обучение и его применение

Примерная тематика докладов:

1. Методы обучения нейронных сетей
2. Архитектура нейронных сетей
3. Практическое применение нейронных сетей при распознавании текста
4. Практическое применение нейронных сетей при анализе текста
5. Практическое применение нейронных сетей при распознавании изображений

Тема 2.1. Нейронные сети

Анализ статей посвященных нейронным сетям и методам их обучения. Практическое применение нейронных сетей в финансовых задачах и других областях. Обучение нейросетей.

Тема 2.2. Обработка текста

Обработка текста.

Раздел 3. Представление результатов проектов

Тематика проектов по разделам

Интегральный проект по тематике искусственного интеллекта и машинного обучения

Примерная тематика проектов:

1. Анализ информации методами машинного обучения по прикладным отраслям
2. Применение метода кластеризации для обработки данных, полученных из Интернета
3. Применение метода кластеризации для обработки текстовой информации
4. Применение метода кластеризации для обработки графической информации
5. Использование нейросетей для распознавания изображений
6. Использование нейросетей для принятия решений в конкретной прикладной сфере
7. Разработка чат-бота с определенными параметрами
8. Повышение качества изображений с помощью нейронных сетей

Тема 3.1. Описание проекта на английском языке

Представление описания проекта. Ответы на вопросы и замечания.

Тема 3.2. Презентация проекта на английском языке

Презентация проекта. Ответы на вопросы и замечания.

Тема 3.3. Заключительная дискуссия

Презентация результатов проекта. Обсуждение результатов и перспектив их использования.

Требования к презентации

Все слайды презентации должны иметь сквозную нумерацию (кроме титульного слайда), презентацию необходимо представить в формате pdf, структура презентации:

- титульный слайд (с указанием темы и автора);

- перечень анализируемых статей (по каждой статье необходимо указать автора(ов) статей, их название, выходные данные и ссылки на статьи).

Требования к оформлению обзора и доклада

Содержание каждого обзора определяется самой темой, ее спецификой. Обзор должен отражать основные идеи каждой статьи, достоинства и недостатки, критический анализ с мнением автора обзора и выводами. Автору доклада необходимо уметь объяснять излагаемый материал.

Обзор необходимо представить в едином стиле, с размером шрифта 14 пт.

Пример обзора 1:

Как заменить регулярные выражения нейронной сетью

Python *Программирование *Машинное обучение *Natural Language Processing *

Наиболее часто используемый инструмент для поиска подстроки определенного вида в тексте – это регулярные выражения. Но можно ли вместо регулярного выражения использовать нейронную сеть, которая бы выполняла ту же самую задачу?

Задача: найти в тексте описание стоимости недвижимости, то есть численное обозначение и стоимость, записанную прописью. Например, *2 050 000 (два миллиона пятьдесят тысяч) руб., 00 коп.* Задача усложняется тем, что «рубли» и «копейки» могут быть в любом месте (перед скобками или после) и могут быть сокращены.

Чтобы решить данную задачу, будем использовать NLP (Natural Language Processing), морфологический анализатор и нейронную сеть. Подключаем соответствующие библиотеки:

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
import pymorphy2
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
```

Прежде всего необходимо выполнить обработку текста.

1. Токенизируем текст с помощью nltk.

```
f = open('input/'+file, 'r', encoding='ansi')
strings = f.read().replace('\n', ' ')
words = word_tokenize(strings, language='russian')
```

2. Уберем стоп-слова из текста и знаки препинания, которые нам не нужны (например, : или “).

```
words = [word for word in words if not word in new_stopwords]
```

3. После этого пройдемся по тексту и выберем фрагменты, в которых встречается слово «рубли» в полном варианте или в сокращенном. В итоге получаем фрагменты предложений, которые содержат одинаковое количество слов/знаков/чисел. Именно в этих фрагментах мы и будем искать стоимость.
Следующим шагом нужно представить «слова» в виде чисел, так как нейронная сеть работает только с числами. Для этого пройдемся по каждому полученному фрагменту и определим части речи для каждого «слова». В этом нам поможет морфологический анализатор pymorphy2.

```
morph = pymorphy2.MorphAnalyzer()
def set_morphema(x):
    morphema = str(morph.parse(x)[0].tag).split(',')[0]
    if morphema.find(' ') != -1:
        morphema = morphema.split(' ')[0]
    return morphema
```

При анализе выделим 6 групп значений:

- PNCT – знаки пунктуации: ( ) . ,
- NUMB – числа
- NUMR – числительные
- NOUN – существительные: «тысяча», «миллион», «миллиард»
- NOUN – существительные: «рубль», «копейка» и их сокращенные формы
- Все остальные части речи, которые не встречаются в нужных нам фрагментах

Каждое «слово» в зависимости от того, в какую группу оно попало, представим в виде вектора значений из 6 чисел, содержащего 0 и 1 – 0, если число не относится к данной группе, 1 – если относится. Получается, что каждое «слово» закодировано пятью нулями и одной единицей. Каждый фрагмент, в котором мы будем искать стоимость, содержит 23 «слова», соответственно получаем 138 чисел для одного фрагмента. Именно эти значения будем подавать на вход нейронной сети.
Чтобы обучить нейронную сеть, составим выборку. Входные данные уже имеются, остается составить выходные. Выходные данные для одного фрагмента будут представлены в виде 23 чисел – 0 и 1. Единицами обозначим тот фрагмент, который в итоге нужно выбрать из текста и который содержит стоимость.
Как преобразовывались данные:
Фрагмент, содержащий стоимость:

```
['Цена', 'Договора', 'порядок', 'расчетов', '.', '2.1', '.', 'Стоимость', 'Объекта', 'составляет', '810', '000', '(', 'Восемьсот', 'десять', 'тысяч', ')', 'рублей', '.', 'Цена', 'является', 'окончательной', 'изменению']
```

Фрагмент, преобразованный в числа:

```
[0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0,
0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1]
```

Выходные данные для фрагмента:

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0]
```

Создаем модель нейронной сети. Она будет состоять из 4 плотно связанных слоев Dense, с учетом входного и выходного. Используем наиболее распространенную функцию активации «relu» для каждого слоя, кроме выходного. Также добавим слой Dropout, чтобы предотвратить переобучение модели.
Важным критерием работы нейронной сети оказалась функция потерь. Наиболее стабильный и достоверный результат получился при функции потерь MSE (средняя квадратическая ошибка).

```
model = keras.Sequential([
    layers.Dense(138, activation='relu'),
    layers.Dense(138, activation='relu'),
    layers.Dropout(0.1),
    layers.Dense(23, activation='relu'),
    layers.Dense(23, activation='sigmoid')])
optimizer = tf.optimizers.Adam()
model.compile(loss=tf.keras.losses.MSE, optimizer=optimizer, metrics=['accuracy'])

history = model.fit(train_data, vyhod_data_train, epochs=1000)
```

Обучаем модель и выполняем предсказания для тестовых данных. На выходе получаем последовательность из 23 чисел для каждого фрагмента. Числа, которые больше 0,9 – нужные нам значения, заменим их на 1. Находим начало и конец последовательности единиц. Далее по индексам начала и конца последовательности единиц выбираем стоимость из фрагмента текста.
Что получается при обработке тестовых данных?
Фрагмент, в котором ищем стоимость

```
['Цена', 'Объекта,', 'являющаяся', 'предметом', 'настоящего', 'Договора', ',', 'составляет', '2',
'100', '000', '(', 'два', 'миллиона', 'сто', 'тысяч', ')', 'рублей', ',', 'именно', 'цена', 'земельного',
'участка']
```

Фрагмент, преобразованный в числа

```
[0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1]
```

Значения, полученные после работы нейронной сети

```
[5.8360482e-14 5.7737207e-19 7.7303122e-18 6.7739243e-18 5.8828078e-14
 1.1499115e-18 2.8125218e-13 9.2836785e-08 1.0000000e+00 1.0000000e+00
```

```
1.0000000e+00 1.0000000e+00 1.0000000e+00 1.0000000e+00 1.0000000e+00
1.0000000e+00 1.0000000e+00 1.0000000e+00 5.1083343e-10 2.7263733e-10
3.0139889e-14 2.2163703e-13 1.4157570e-14]
```

Выбранный фрагмент стоимости

```
['2', '100', '000', '(', 'два', 'миллиона', 'сто', 'тысяч', ')', 'рублей']
```

Процент корректно выбранных фрагментов стоимости составляет 94% по результатам работы данной нейронной сети.

Пример обзора 2:

Еще раз о Code Review
Не так давно сидел я делал ревью кода одного из коллег. Это было не первое мое ревью, но в этот раз я задался вопросом как все таки формализовать подход и на что конкретно стоит обращать внимание и как аргументировать и формулировать предложения и замечания. Сформулировал я для себя вот такие пункты:
Соблюдение принципов SOLID
На всякий случай напомню, SOLID - это аббревиатура для 5 основных принципов ООП:
Single Responsibility Principle - принцип единой ответственности. Этот принцип говорит, что для класса должна быть определена единственная ответственность. Или еще его иногда формулируют как “для внесения изменений в класс должна быть только одна причина”
Open Closed Principle - принцип открытости-закрытости. Этот принцип говорит, что программные сущности должны быть открыты для расширения, но закрыты для модификации.
Liskov Substitution Principle - принцип подстановки Барбары Лисков. Роберт Мартин формулирует этот принцип так: “Функции, которые используют базовый тип, должны иметь возможность использовать подтипы базового типа, не зная об этом.”
Interface Segregation Principle - принцип разделения интерфейсов. Этот принцип говорит, что “много интерфейсов специального назначения лучше, чем один интерфейс общего назначения”.
Dependency Inversion Principle - принциц инверсии зависимостей. Этот принцип говорит, что абстракции не должны зависеть от реализаций, наоборот, реализации должны зависеть от абстракций.
Отсутствие дурных запахов
Обычно выделяют около 20 запахов и в идеале бы, конечно, знать и отслеживать их все. Их все я здесь перечислять не буду, но вот наиболее часто встречающиеся и сильнее других бросающиеся в глаза:

- ✓ Его величество дублирование кода
- ✓ Магические числа
- ✓ Операторы типа switch
- ✓ Длинный метод
- ✓ Длинный список параметров
- ✓ Неинформативные имена переменных

Дурные запахи (не все) и принципы SOLID взаимосвязаны и иногда наличие запаха может сигнализировать о нарушении одного из принципов, к примеру дублирование кода может говорить о нарушении принципа единой ответственности. Однако, на мой взгляд все равно стоит оставлять комменты и о запахе и о нарушении.
Правильное использование языка

Сюда я отнес использование условных языковых идиом, хороший практик и рекомендаций из документации. Приведу пару примеров. Я пишу на питоне, так что примеры тоже будут на питоне.

Использовать литеральную форму задания словаря, вместо циклов, когда это возможно:

```
chars = ['a', 'b', 'c']
# Bad
d = {}
for i in range(len(chars)):
  d[i] = chars[i]
# Good
d = {i: char for i, char in enumerate(chars)}
```

Пример выше нужен лишь для иллюстрации, он искусственный и создать такой словарь можно конечно же по другому

Использовать подходящие методы встроенный типов:

```
colors_codes = {
  'red': '#FF0000',
  'green': '#008000'
}
white_name = 'white'
white_code = '#FFFFFF'

# Bad
if white_name not in colors_codes:
  colors_codes[white_name] = white_code
print(colors_codes[white_name])

# Good
print(colors_codes.setdefault(white_name, white_code))
```

Использовать менеджеры контекста:

```
#Bad
...
fin = open(path, 'rt')
text = fin.read()
print(text)
...
# Good
...
with open(path) as fin:
  text = fin.read()
  print(text)
```

и т.д.

Покрытие кода тестами

В этом пункте проверяю не количество тестов, а их качество: есть ли тесты для граничных случаев и есть ли непокрытые кейсы.

Конечно, есть и другие моменты, о которых можно задуматься: архитектура и вопросы производительности, к примеру, но их я еще не формализовал.

SOLID принципы конечно же не для всех случаев актуальны, да и с некоторыми запахами в определенных ситуациях можно не бороться. Пункты выше относятся к проектам, где они имеют смысл.

Критерии оценивания докладов, проектов:

Показатели для оценки докладов и проектов (требования):

•проявление знаний видов и особенностей проектов; проблематики проектной деятельности; предметного поля информатики и проблематики междисциплинарных проектов;

•проявление умений анализировать противоречия, выделять проблему, разрабатывать замысел проекта, оценивать его практическую значимость и перспективы;

•широта кругозора автора, понимание актуальных проблем и основ междисциплинарного подхода в проектной деятельности;

•полнота, глубина, всесторонность раскрытия идеи (наличие всех пунктов в содержании эссе);

•последовательность и логичность изложения идеи (все пункты должны быть содержательно связаны друг с другом и должны быть непротиворечивы);

•культура (стиль) письменного изложения материала

Критерии оценивания докладов:

25-30 баллов – в представленном докладе демонстрируется отличная способность анализировать профессиональную информацию для решения задач в области создания и применения технологий и систем искусственного интеллекта, выделять в ней главное, презентация правильно оформлена и структурирована, выводы и рекомендации обоснованы

17-24 балла – в представленном докладе демонстрируется хорошая способность анализировать профессиональную информацию для решения задач в области создания и применения технологий и систем искусственного интеллекта, выделять в ней главное, презентация правильно оформлена и структурирована, выводы и рекомендации недостаточно обоснованы

9-16 балла – в представленном докладе демонстрируется удовлетворительная способность анализировать профессиональную информацию для решения задач в области создания и применения технологий и систем искусственного интеллекта, выделять в ней главное, презентация правильно оформлена и структурирована, выводы и рекомендации не обоснованы

0-8 баллов – в представленном докладе – непонимание задач в области ИИ либо доклад не представлен

Критерии оценивания проектов:

31-40 баллов – в представленном проекте демонстрируется отличная способность исследовать современные проблемы и методы информатики, искусственного интеллекта и развития информационного общества, цифровой экономики

21-30 балла – в представленном проекте демонстрируется хорошая способность исследовать современные проблемы и методы информатики, искусственного интеллекта и развития информационного общества, цифровой экономики, анализ современных методов проведен неполный

11-20 балла – в представленном проекте демонстрируется удовлетворительная способность исследовать современные проблемы и методы информатики, искусственного интеллекта и развития информационного общества, цифровой экономики, анализ современных методов и само исследование неполны

0-10 баллов – в представленном проекте – непонимание задач в области ИИ либо проект не представлен

Список литературы

Основная литература.

Анализ данных : учебник для вузов / В. С. Мхитарян [и др.] ; под редакцией В. С. Мхитаряна. — Москва : Издательство Юрайт, 2020. — 490 с. — (Высшее образование). — ISBN 978-5-534-00616-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/450166 (дата обращения: 09.10.2021).

Дополнительная литература.

1. Крутиков В. Н. Анализ данных / В.Н. Крутиков; В.В. Мешечкин - Кемерово: Кемеровский государственный университет, 2014. - 138 с.

2. Экономико-математические методы и прикладные модели: Учеб.пособие / Под ред. В. В. Федосеева - М.: ЮНИТИ, 1999. - 392 с.

3 Подругина И.А., Ильичева И.В. Проектно-исследовательская деятельность: развитие одаренности: монография. Москва: МПГУ, 2017. 300 с. http://biblioclub.ru/index.php?page=book_red&id=469696&sr=1

4 Сибагатуллина А.М. Организация проектной и научно-исследовательской деятельности. Йошкар-Ола: ПГТУ, 2012. 93 с. http://biblioclub.ru/index.php?page=book_red&id=277052&sr=1

Перечень ресурсов сети Интернет, необходимых для освоения дисциплины

Университетская библиотека online:

http://biblioclub.ru/index.php?page=main_ub_red

MathWorld. http://mathworld.wolfram.com

https://habr.com/ru/post/

Электронно-библиотечная система (ЭБС) ЮРАЙТ www.biblio-online.ru