

Федеральное государственное бюджетное образовательное учреждение
высшего образования
Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики

УТВЕРЖДАЮ
декан факультета вычислительной
математики и кибернетики



/И.А. Соколов /
«27» сентября 2022г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине

Практикум по глубокому машинному обучению

Уровень высшего образования:

бакалавриат

Направление подготовки / специальность:

01.03.02 "Прикладная математика и информатика" (3++)

Направленность (профиль) ОПОП:

Искусственный интеллект и анализ данных

Форма обучения:

очная

Рассмотрен и утвержден

на заседании Ученого совета факультета ВМК

(протокол №7, от 27 сентября 2022 года)

Москва 2022

1. ФОРМЫ И ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

В процессе и по завершении изучения дисциплины оценивается формирование у студентов следующих компетенций:

Планируемые результаты обучения по дисциплине (модулю)		
Содержание и код компетенции.	Индикатор (показатель) достижения компетенции	Планируемые результаты обучения по дисциплине, сопряженные с индикаторами достижения компетенций
ПК-6. Способен создавать и поддерживать системы искусственного интеллекта на основе нейросетевых моделей и методов	ПК-6.1. Осуществляет оценку и выбор моделей искусственных нейронных сетей и инструментальных средств для решения поставленной задачи ПК-6.2. Разрабатывает системы искусственного интеллекта на основе моделей искусственных нейронных сетей и инструментальных средств	Знать основные современные методы глубинного обучения Уметь применять методы глубинного обучения для решения задач анализа данных Владеть навыками реализации алгоритмов анализа данных на языке Python с использованием библиотек глубинного обучения

1.1. Текущий контроль успеваемости

Текущий контроль успеваемости осуществляется путем оценки результатов выполнения заданий практических (семинарских) занятий, самостоятельной работы, предусмотренных учебным планом и посещения занятий/активность на занятиях.

В качестве оценочных средств текущего контроля успеваемости предусмотрены:

решение индивидуальных заданий

Примеры заданий

1. Необходимо будет реализовать полносвязную нейронную сеть, используя модульный подход. Для каждого слоя реализации прямого и обратного проходов алгоритма обратного распространения ошибки будут иметь следующий вид:

```
def layer_forward(x, w):
    """ Receive inputs x and weights w """
    # Do some computations ...
    z = # ... some intermediate value
    # Do some more computations ...
    out = # the output

    cache = (x, w, z, out) # Values we need to compute gradients

    return out, cache
def layer_backward(dout, cache):
    """
    Receive dout (derivative of loss with respect to outputs) and cache,
    and compute derivative with respect to inputs.
    """
    # Unpack cache values
    x, w, z, out = cache

    # Use values in cache to compute derivatives
```

```

dx = # Derivative of loss with respect to x
dw = # Derivative of loss with respect to w

return dx, dw

```

- Для полносвязного слоя реализуйте прямой проход. Протестируйте свою реализацию.
- Для полносвязного слоя реализуйте обратный проход. Протестируйте свою реализацию.
- Реализуйте полносвязную сеть с произвольным числом скрытых слоев.
- Попробуйте добиться эффекта переобучения на небольшом наборе изображений (например, 50). Используйте трехслойную сеть со 100 нейронами на каждом скрытом слое. Попробуйте переобучить сеть, достигнув 100 % accuracy за 20 эпох. Для этого поэкспериментируйте с параметрами `weight_scale` и `learning_rate`.

2. Для реализации собственной модели с помощью Keras Model Subclassing API необходимо выполнить следующие шаги:

- 1) Определить новый класс, который является наследником `tf.keras.Model`.
- 2) В методе `init()` определить все необходимые слои из модуля `tf.keras.layers`
- 3) Реализовать прямой проход в методе `call()` на основе слоев, объявленных в `init()`

Ниже приведен пример использования keras API для определения двухслойной полносвязной сети.

https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/keras

```

class TwoLayerFC(tf.keras.Model):
    def __init__(self, hidden_size, num_classes):
        super(TwoLayerFC, self).__init__()
        initializer = tf.initializers.VarianceScaling(scale=2.0)
        self.fc1 = tf.keras.layers.Dense(hidden_size, activation='relu',
                                         kernel_initializer=initializer)
        self.fc2 = tf.keras.layers.Dense(num_classes, activation='softmax',
                                         kernel_initializer=initializer)
        self.flatten = tf.keras.layers.Flatten()

    def call(self, x, training=False):
        x = self.flatten(x)
        x = self.fc1(x)
        x = self.fc2(x)
        return x

def test_TwoLayerFC():
    """ A small unit test to exercise the TwoLayerFC model above. """
    input_size, hidden_size, num_classes = 50, 42, 10
    x = tf.zeros((64, input_size))
    model = TwoLayerFC(hidden_size, num_classes)
    with tf.device(device):
        scores = model(x)
        print(scores.shape)

test_TwoLayerFC()

```

Реализуйте трехслойную CNN для вашей задачи классификации.

Архитектура сети:

1. Сверточный слой (5 x 5 kernels, zero-padding = 'same')
2. Функция активации ReLU
3. Сверточный слой (3 x 3 kernels, zero-padding = 'same')
4. Функция активации ReLU
5. Полносвязный слой
6. Функция активации Softmax

Обучите трехслойную CNN. В `tf.keras.optimizers.SGD` укажите Nesterov momentum = 0.9 .

https://www.tensorflow.org/versions/r2.0/api_docs/python/tf/optimizers/SGD

Значение accuracy на валидационной выборке после 1 эпохи обучения должно быть > 50% .

1.2. Промежуточная аттестация

Промежуточная аттестация осуществляется в форме зачета

В качестве средств, используемых на промежуточной аттестации предусматривается:

Билеты

1.3. Типовые задания для проведения промежуточной аттестации

Вопросы к зачету

1. Обучение НС- активационные функции, проблемы и решения
2. Обучение НС- препроцессинг данных
3. Обучение НС- инициализация весов сети
4. Обучение НС- Пакетная нормализация
5. Обучение НС- transfer learning
6. Обучение НС- оптимизаторы (от SGD до Adam и далее)
7. Learning rate
8. Переобучение сети и как с ним бороться
9. Аугментация
10. Регуляризация
11. Подбор гиперпараметров
12. Наблюдение за обучением сети (Babysitting DNN)
13. Архитектуры: от LeNet к современным СНС
14. Архитектуры: AlexNet
15. Архитектуры: VGG
16. Архитектуры: GoogLeNet
17. Архитектуры: ResNet
18. Архитектуры: RNN
19. Архитектуры: Attention block
20. Архитектуры: LSTM
21. Генеративно-состязательные сети

2. КРИТЕРИИ ОЦЕНКИ ПО ДИСЦИПЛИНЕ

ШКАЛА И КРИТЕРИИ ОЦЕНИВАНИЯ результатов обучения (РО) по дисциплине				
Оценка	2 (не зачтено)	3 (зачтено)	4 (зачтено)	5 (зачтено)
виды оценочных средств				
Знания (виды оценочных средств: приведены в п. 1.2.)	Отсутствие знаний	Фрагментарные знания	Общие, но не структурированные знания	Сформированные систематические знания
Умения (виды оценочных средств: приведены в п. 1.2.)	Отсутствие умений	В целом успешное, но не систематическое умение	В целом успешное, но содержащее отдельные пробелы умение (допускает неточности не принципиального характера)	Успешное и систематическое умение
Навыки (владения, опыт деятельности) (виды оценочных средств: приведены в п. 1.2..)	Отсутствие навыков (владений, опыта)	Наличие отдельных навыков (наличие фрагментарного опыта)	В целом, сформированные навыки (владения), но используемые не в активной форме	Сформированные навыки (владения), применяемые при решении задач