

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский ядерный университет «МИФИ»

Обнинский институт атомной энергетики –
филиал федерального государственного автономного образовательного учреждения высшего образования
«Национальный исследовательский ядерный университет «МИФИ»
(ИАТЭ НИЯУ МИФИ)

Утверждено на заседании
УМС ИАТЭ НИЯУ МИФИ
Протокол от 30.08.2021 № 2-
8/2022

Фонд оценочных средств УЧЕБНОЙ ДИСЦИПЛИНЫ

«ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ. ГЛУБОКОЕ ОБУЧЕНИЕ»

название дисциплины

для студентов специальности/направления подготовки

09.04.01 Информатика и вычислительная техника

специализация/профиль:

Большие данные и машинное обучение в задачах атомной энергетики

Форма обучения: очная

г. Обнинск 2022г.

Программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта высшего профессионального образования по направлению подготовки (специальности) 09.04.01 «Информатика и вычислительная техника».

Программу составил: _____ профессор, д.ф.-м.н. С.О. Старков
(ученое звание, степень, фамилия, инициалы)

Рецензент: _____ зав.каф. КСТ Калужского филиала МГТУ им. Н.Э.Баумана, канд. физ.-мат. наук И.В. Чухраев

Программа рассмотрена на заседании ОИКС
(протокол № _____ от «_____» _____ 20__ г.)

Руководитель направления подготовки
090401 «Информатика и
вычислительная техника»

_____ Старков С.О.

«_____» _____ 2022 г.

1. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

1.1. Паспорт фонда оценочных средств по дисциплине

№ п/п	Контролируемые модули, разделы (темы) дисциплины	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
1	Темы 1-8.	ПК-1; СПК-1	Лабораторный практикум. Самостоятельная работа, экзамен.

1.2 Примеры заданий (контрольных вопросов) для оценки качества освоения дисциплины, уровня учебных достижений

1. Структура и свойства искусственного нейрона. Функция активации. Типы функций активации.
2. Принципы построения искусственных нейронных сетей (ИНС). Типы связей. Задачи, решаемые с использованием ИНС
3. Персептрон. Проблема разделения признаков на плоскости, в пространстве.
4. Задача линейного одношагового предсказания на основе нейрона Мак-Каллока-Питца.
5. Принципы обучения нейронных сетей. Обучение с учителем и без учителя. Примеры. Метод соревнования.
6. Формирование обучающих выборок. Пример.
7. Алгоритм обратного распространения ошибок.
8. Логические элементы на основе нейронных сетей.
9. Решение задачи XOR.
10. Многослойный (двухслойный) персептрон. Обучение и области применения.
11. Самоорганизующиеся нейронные сети Кохонена (**SOFM**).
12. Ассоциативная память нейронных сетей. Нейронные сети встречного распространения.
13. Сеть Хопфилда. Принципы обучения. Задачи, решаемые с помощью сетей Хопфилда.
14. Сеть Хэмминга. Обучение и функционирование.
15. Двухнаправленная ассоциативная память.
16. Сети адаптивной резонансной теории.
17. Машина Больцмана. Алгоритм больцмановского обучения (метод «отжига»).
18. Сети с использованием радиальных базисных функций.

19. Сети глубокого обучения.
20. Сверточные сети.
21. Выбор параметров сверточных сетей. Операторы свертки, пулинга.
- 21 Обучение сверточных сетей.
22. Рекуррентные сети.

1.3. Индивидуальные задания для реализации лабораторных работ по дисциплине

Лаб. 1. Построение одношагового предиктора на основе нейронной сети Мак-Каллока-Питца. В качестве функции для предсказания выбираются индивидуальные тригонометрические функции

Студентам предлагается написать программное решение позволяющее производить оценку простейших периодических функций (тип функции – различные комбинации тригонометрических функций)

Лаб. 2. Программная реализация распознавания печатных символов на основе сети Хопфилда. Индивидуально задаются различные печатные символы (буквы кириллицы или латиницы, цифры от 1 до 9)

Лаб. 3. Программная реализация многослойной архитектуры для распознавания буквенных символов. Обучение производится с использованием датасета MNIST. Занятия проводятся в компьютерном классе с доступом к ресурсам библиотек TensorFlou и KERAS.

1.4. Критерии оценивания компетенций (результатов):

Успешно освоившими дисциплину считаются студенты, отчитавшиеся по всем формам контроля (два ИДЗ, контрольная работа). Общая оценка за экзамен складывается из баллов за две точки промежуточного контроля в течение семестра и баллов итогового контроля в соответствии с университетской системой рейтингового контроля.

Описание шкалы оценивания:

За успешное выполнение ИДЗ студент получает в соответствии с рейтинговой системой до 30 баллов. Максимальная оценка достигается при полном выполнении задания с демонстрацией работающей программы и контрольными тестами работоспособности самой программы, а также при наличии письменного отчета по работе.

Подробный ответ на вопросы КР на каждый вопрос (всего три вопроса в билете) оценивается в 10% . Дополнительные баллы до 10% выставляются за активность на лекциях и семинарах.

Полученные проценты проставляются в рейтинг. Максимальная оценка составляет 100 баллов.

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

Специализированные компьютерные классы, ауд. 2-510, 2-521 аудиторного фонда ИАТЭ НИЯУ МИФИ.

20 компьютеризованных рабочих мест в ауд. 2-521 ИАТЭ НИЯУ МИФИ.

3. Формы организации самостоятельной работы обучающихся (темы, выносимые для самостоятельного изучения; вопросы для самоконтроля; типовые задания для самопроверки)

Список вопросов для самостоятельной работы

Основные понятия искусственного интеллекта

1. Направления ИИ
2. ИИ в задачах робототехники
3. Машинное обучение. Идеи и подходы.

Архитектуры ИНС

1. Радиально-базисные сети.
2. Аппроксимация с использованием радиально-базисных сетей
3. Обучение без учителя. Примеры
4. Самоорганизующаяся карта Кохонена.
5. Сети адаптивного резонанса (ART-1). Архитектура, функционирование. Обучение.

4. Методические указания и типовые вопросы.

1. Какие бывают слои нейронной сети?

Однослойные нейронные сети:

В однослойных нейронных сетях сигналы с входного слоя сразу подаются на выходной слой.

- входной слой;
- выходной слой.

Многослойные нейронные сети:

- входной слой;
- скрытый слой;
- выходной слой.

2. Каков результат прямого распространения?

Сети с обратными связями (*Recurrent neural network*) — искусственные нейронные сети, в которых выход нейрона может вновь подаваться на его вход. В более общем случае это означает возможность распространения сигнала от выходов к входам. Такие нейросети имеют свойства кратковременной памяти (как у человека).

3. Что такое стохастический градиентный спуск?

Стохастический градиентный спуск (англ. *stochastic gradient descent*) — оптимизационный алгоритм, отличающийся от обычного градиентного спуска тем, что градиент оптимизируемой функции считается на каждом шаге не как сумма градиентов от каждого элемента выборки, а как градиент от одного, случайно выбранного элемента. В этом варианте используется только один случайный обучающий пример для вычисления градиента и обновления параметров на каждой итерации.

4. Для чего нужно обратное распространение ошибки? В чем его суть?

Обратное распространение ошибки — это способ обучения нейронной сети. Цели обратного распространения просты: отрегулировать каждый вес пропорционально тому, насколько он способствует общей ошибке. Если мы будем итеративно уменьшать ошибку каждого веса, в конце концов у нас будет ряд весов, которые дают хорошие прогнозы.

Идея алгоритма **обратного распространения ошибки** состоит в том, чтобы на основе расчетной **ошибки**, полученной на выходном слое нейронной сети, пересчитать значение весов вектора W последнего слоя нейронов. Затем мы переходим к предыдущему слою и так далее, от конца к началу, то **есть**, он состоит из обновления всех весов W слоев, от последнего до достижения входного слоя сети путем **обратного распространения ошибки**, полученной сетью.

5. Как *learning rate* влияет на обучение нейронной сети? Каково типичное значение этого параметра?

Коэффициент скорости обучения — это параметр градиентных алгоритмов обучения нейронных сетей, позволяющий управлять величиной коррекции весов на каждой итерации.

Выбирается в диапазоне от 0 до 1. Нуль указывать бессмысленно, поскольку в этом случае корректировка весов вообще производиться не будет. Большие значения (0,7 — 1) будут соответствовать большому значению шага коррекции. При этом алгоритм будет работать быстрее (т.е. для поиска минимума функции ошибки потребуется меньше итераций). Однако может снизиться точность настройки модели на минимум функции ошибки, что потенциально увеличит ошибку обучения.

Малые значения коэффициента (0,1 — 0,3) соответствуют меньшему шагу коррекции весов. В этом случае число шагов (или эпох) обучения, требуемое для поиска экстремума, как правило, увеличивается, но возрастает и точность настройки алгоритма на минимум функции ошибки, что потенциально уменьшает ошибку обучения. На практике коэффициент скорости обучения обычно подбирают экспериментально.

Скорость обучения влияет на скорость, с которой обучается ваша нейронная сеть, а также проявлением обучения является изменение веса, а скорость обучения влияет на способ этого изменения.

1. Какие бывают слои нейронной сети?

Традиционно нейронные сети имели только три типа слоев: скрытый, входной и выходной. Это все действительно один и тот же тип слоя, если вы просто учтете, что входные слои загружаются из внешних данных (а не предыдущего слоя) и выходные данные канала во внешнее место назначения (а не следующий слой). Эти три слоя теперь обычно называют плотными слоями. Это связано с тем, что каждый нейрон в этом слое полностью связан со следующим слоем. В случае выходного слоя нейроны являются просто держателями, прямых связей нет. Современные нейронные сети имеют много дополнительных типов слоев.

2. Каков результат прямого распространения?

Для прямого распространения, независимо от размера, процесс можно выразить следующей формулой:

$$a_2 = \sigma(z_2) = \sigma(a_1 * W_2 + b_2)$$

Среди них верхний индекс представляет количество слоев, звездочка представляет свертку, b представляет термин смещения смещения, σ представляет функцию активации.

3. Что такое стохастический градиентный спуск?

Стохастический градиентный спуск (SGD) — это простой, но очень эффективный подход к подгонке линейных классификаторов и регрессоров под выпуклые функции потерь, такие как (линейные) [Метод опорных векторов](#) и [логистическая регрессия](#). Несмотря на то, что SGD существует в сообществе машинного обучения уже давно, совсем недавно он привлек значительное внимание в контексте крупномасштабного обучения.

SGD успешно применяется для решения крупномасштабных и разреженных задач машинного обучения, часто встречающихся при классификации текста и обработке естественного языка. Учитывая, что данные немногочисленны, классификаторы в этом модуле легко масштабируются для решения задач с более чем 105 обучающими примерами и более чем 105 функциями.

Строго говоря, SGD — это просто метод оптимизации и не соответствует конкретному семейству моделей машинного обучения. Это всего лишь *способ* обучить модель.

4. Для чего нужно обратное распространение ошибки? В чем его суть?

Обратное распространение ошибки — это способ обучения [нейронной сети](#). Цели обратного распространения просты: отрегулировать каждый вес пропорционально тому, насколько он способствует общей ошибке. Если мы будем итеративно уменьшать ошибку каждого веса, в конце концов у нас будет ряд весов, которые дают хорошие прогнозы.

5. Как learning rate влияет на обучение нейронной сети? Каково типичное значение этого параметра?

Коэффициент скорости [обучения](#) — это параметр градиентных алгоритмов обучения [нейронных сетей](#), позволяющий управлять величиной коррекции весов на каждой итерации.

Выбирается в диапазоне от 0 до 1. Ноль указывать бессмысленно, поскольку в этом случае корректировка весов вообще производиться не будет.

Выбор параметра противоречив. Большие значения (0,7 — 1) будут соответствовать большому значению шага коррекции. При этом алгоритм будет работать быстрее (т.е. для поиска минимума [функции ошибки](#) потребуется меньше итераций). Однако может снизиться точность настройки модели на минимум функции ошибки, что потенциально увеличит [ошибку обучения](#).

Малые значения коэффициента (0,1 — 0,3) соответствуют меньшему шагу коррекции весов. В этом случае число шагов (или эпох) обучения, требуемое для поиска экстремума, как правило, увеличивается, но

возрастает и точность настройки алгоритма на минимум функции ошибки, что потенциально уменьшает ошибку обучения. На практике коэффициент скорости обучения обычно подбирают экспериментально.

1. Почему мы должны преобразовывать целочисленные входные данные в значения с плавающей точкой во время обучения?

Потому что выходные данные не являются числом а являются тензором

2. Каковы методы изменения формы тензорного объекта?

Библиотека Pytorch

Использование метода reshape()

Этот метод используется для придания данному тензору заданной формы (изменения размеров)

Использование метода flatten()

flatten() используется для преобразования N-мерного тензора в одномерный тензор.

Использование метода view()

view() используется для изменения тензора в двумерном формате, т.Е. строк и столбцов. Мы должны указать количество строк и количество столбцов для просмотра.

Использование метода resize()

Это используется для изменения размеров данного тензора.

Использование метода unsqueeze()

Это используется для изменения формы тензора путем добавления новых измерений в заданных положениях.

3. Почему вычисления с тензорами выполняются быстрее, чем с массивами NumPy?

TensorFlow - это библиотека глубокого обучения, которая разработана для наилучшей работы на графических процессорах. Графический процессор установлен на шине PCIe, и связь с этой шиной намного медленнее, чем связь между процессором и системной памятью.

Итак, у нас есть преимущества в использовании графического процессора только в том случае, если для данных нужно выполнить так много вычислений, что время передачи системного графического процессора становится незначительным по отношению к фактическому времени вычислений.

4. Как извлекаются значения промежуточного уровня нейронной сети?

Значения скрытых слоёв определяются как:

$$\mathbf{h}_t^1 = f_h(\mathbf{x}_t, \mathbf{h}_{t-1}^1)$$

$$\mathbf{h}_t^2 = f_h(\mathbf{h}_t^1, \mathbf{h}_{t-1}^2)$$

5. Как последовательный метод помогает упростить определение архитектуры нейронной сети?

В этом случае сначала подаётся один сигнал, затем оценивается результат работы нейросети. Если результат неудовлетворительный, начинается настройка сети методом обратного распространения ошибки. И лишь после того как удастся настроить нейросеть оптимальным образом, происходит переход к следующему сигналу.

Достоинство такого метода – сравнительно небольшое время обучения. А также то, что если многие сигналы похожи или одинаковы, последовательный метод лучше себя оправдывает.

6. Контрольные вопросы.

1. Искусственный интеллект. Возможные определения. Тест Тьюринга
2. Модель нейрона Мак-Каллока и Питтса
3. Функции активации, примеры наиболее распространенных функций активации
4. Обучение нейронных сетей (с Учителем и Без Учителя, с подкреплением). Обучающие выборки. Пример
5. Модель простейшего Персептрона
6. Архитектуры нейронных сетей.
7. Сети прямого распространения. Многослойный персептрон.
8. Обучение сетей. Метод обратного распространения ошибки. Обучение выходного слоя.
9. Обучение сетей. Метод обратного распространения ошибки. Обучение скрытых слоев.
10. Алгоритм метода обратного распространения ошибки. Выбор параметров обучения.
11. Модели логических элементов на основе нейронных сетей: «И», «ИЛИ», «Исключающее ИЛИ»
12. Вероятностные нейронные сети.
13. Классификатор на основе ИНС. Сети Хэмминга. Обучение сетей Хэмминга (Обучение без Учителя).
14. Метод «отжига» для обучения ИНС
15. Алгоритм обучения с использованием машины Больцмана.
16. Полносвязные сети. Сеть Хопфилда.

17. Автоассоциативная память на основе сети Хопфилда.
18. Обучение сети Хопфилда.
19. Радиально-базисные сети.
20. Задача аппроксимации с использованием радиально-базисных сетей.
21. ИНС для задач кластеризации. Возможные типы ИНС.
22. ИНС для задач распознавания. Возможные архитектуры ИНС
23. ИНС для задач аппроксимации. Возможные архитектуры ИНС.
24. ИНС для задач предсказания временных трендов. Возможные архитектуры ИНС.
25. Сети адаптивного резонанса (ART-1). Архитектура, функционирование. Обучение.