

Комплект оценочных материалов по дисциплине
«Нейронные сети»

Задания закрытого типа

Задания закрытого типа на выбор правильного ответа

1. Какой из следующих алгоритмов чаще всего используется для обучения нейронных сетей?

- А) Метод градиентного спуска
- Б) Алгоритм быстрой сортировки
- В) Динамическое программирование
- Г) Алгоритм Дейкстры

Правильный ответ: А

Компетенции (индикаторы): ПК-3

2. Какой тип нейронной сети лучше всего подходит для обработки изображений?

- А) Рекуррентные нейронные сети (RNN)
- Б) Полносвязные нейронные сети (Fully Connected Networks)
- В) Свёрточные нейронные сети (CNN)
- Г) Генеративные состязательные сети (GAN)

Правильный ответ: В

Компетенции (индикаторы): ПК-3

3. Что такое "дропаут" (dropout) в нейронной сети?

- А) Методику увеличения количества данных для обучения
- Б) Метод, предотвращающий переобучение, отключая случайные нейроны на этапе обучения
- В) Способ оптимизации вычислительных ресурсов
- Г) Алгоритм, используемый для инициализации весов

Правильный ответ: Б

Компетенции (индикаторы): ПК-3

4. Какой из следующих параметров не является частью функции `model.compile()` в Keras?

- А) optimizer
- Б) loss
- В) metrics
- Г) layers

Правильный ответ: Г

Компетенции (индикаторы): ПК-3

5. Что делает метод `model.fit()` в Keras?

- А) Оценивает модель на тестовых данных
- Б) Обучает модель на обучающих данных
- В) Предсказывает результаты для новых данных
- Г) Сохраняет модель на диск

Правильный ответ: Б

Компетенции (индикаторы): ПК-3

6. Какой метод используется для оценки модели в Keras?

- А) `model.evaluate()`
- Б) `model.fit()`
- В) `model.compile()`
- Г) `model.summary()`

Правильный ответ: А

Компетенции (индикаторы): ПК-3

7. Какой из следующих методов используется для сохранения модели в Keras?

- А) `model.save()`
- Б) `model.store()`
- В) `model.keep()`
- Г) `model.write()`

Правильный ответ: А

Компетенции (индикаторы): ПК-3

8. Какой из следующих типов нейронных сетей можно создать с помощью Keras?

- А) Полносвязные нейронные сети
- Б) Сверточные нейронные сети
- В) Рекуррентные нейронные сети
- Г) Все вышеперечисленные

Правильный ответ: Г

Компетенции (индикаторы): ПК-3

9. Какой из следующих оптимизаторов использует адаптивные скорости обучения для каждого параметра?

- А) SGD
- Б) Adagrad
- В) Adam
- Г) Все вышеперечисленные

Правильный ответ: В

Компетенции (индикаторы): ПК-3

10. Какой из следующих слоев нельзя добавить в модель с помощью Functional API?

- A) Dense
- Б) Conv2D
- В) Flatten
- Г) Sequential

Правильный ответ: Г

Компетенции (индикаторы): ПК-3

Задания закрытого типа на установление соответствие

1. Установите правильное соответствие.

	Определение	Значение
1)	Сверточная нейронная сеть А) (CNN)	Используется для обработки последовательных данных, таких как текст или временные ряды.
2)	Рекуррентная нейронная сеть Б) (RNN)	Состоит из двух сетей, которые обучаются одновременно: генератора и дискриминатора.
3)	Полносвязная нейронная сеть В) (DNN)	Применяется для извлечения признаков из изображений и обработки изображений.
4)	Генеративно-состязательная сеть Г) (GAN)	Сеть, которая пытается восстановить входные данные, сжимающие их в меньшую размерность.
5)	Автокодировщик (Autoencoder) Д)	Модель, которая использует механизм внимания для акцентирования внимания на определенных частях входных данных.
6)	Сеть с вниманием (Attention E) Network)	Сеть, где каждый нейрон связан с каждым нейроном в следующем слое.

Правильный ответ:

1	2	3	4	5	6
В	А	Е	Б	Г	Д

Компетенции (индикаторы): ПК-3

2. Установите правильное соответствие.

	Метод	Назначение
1)	Добавление слоя (add)	Используется для добавления нового слоя к последовательной

- | | | |
|--------------------------------|----|---|
| 2) Компиляция модели (compile) | Б) | модели.
Позволяет сохранить архитектуру и веса модели в файл. |
| 3) Обучение модели (fit) | В) | Используется для обучения модели на предоставленных данных. |
| 4) Оценка модели (evaluate) | Г) | Позволяет получить производительность модели на тестовых данных. |
| 5) Предсказание (predict) | Д) | Используется для настройки оптимизатора, функции потерь и метрик перед обучением. |
| 6) Сохранение модели (save) | Е) | Позволяет делать предсказания на новых данных после обучения модели. |

Правильный ответ:

1	2	3	4	5	6
А	Д	В	Г	Е	Б

Компетенции (индикаторы): ПК-3

3. Установите правильное соответствие.

- | Класс | Описание |
|--|--|
| 1) Dense() | А) транспонированные (обратные) светодиодные слои; |
| 2) Conv1D, Conv2D, Conv3D | Б) вспомогательные слои |
| 3) Conv2DTranspose,
Conv3DTranspose | В) рекуррентные слои; |
| 4) SimpleRNN, LSTM, GRU – | Г) сверточные слои; |
| 5) MaxPooling2D, Dropout, BatchNormalization – | Д) полно связанный слой |

Правильный ответ:

1	2	3	4	5
Д	Г	А	В	Б

Компетенции (индикаторы): ПК-3

Задания закрытого типа на установление правильной последовательности

1. Установите следующие шаги в правильной последовательности для создания и обучения модели в Keras.

- А) Компиляция модели
- Б) Определение архитектуры модели

В) Обучение модели на обучающих данных

Г) Создание объекта модели

Д) Оценка модели на тестовых данных

Правильный ответ: Г, Б, А, В, Д

Компетенции (индикаторы): ПК-3

2. Установите правильную последовательность шагов метода обратного распространения ошибки:

А) Обновление весов на основе рассчитанных градиентов.

Б) Вычисление ошибки на выходе сети (разница между предсказанным и истинным значениями).

В) Прямое распространение входных данных через сеть для получения предсказания.

Г) Обратное распространение ошибки через слои сети для вычисления градиентов.

Д) Обновление параметров модели с использованием выбранного оптимизатора.

Правильный ответ: В, Б, Г, А, Д

Компетенции (индикаторы): ПК-3

Задания открытого типа

Задания открытого типа на дополнение

1. _____ – это основная единица нейронной сети, которая принимает входные данные, применяет к ним весовые коэффициенты и передает результат через активационную функцию.

Правильный ответ: Нейрон

Компетенции (индикаторы): ПК-3

2. _____ – это метод, который позволяет улучшить обобщающую способность модели, избегая переобучения, путем добавления шума к весам или отключения некоторых нейронов во время обучения.

Правильный ответ: Регуляризация

Компетенции (индикаторы): ПК-3

3. _____ – представляет собой минимальную значимую единицу данных, используемую в качестве базового элемента для представления, хранения и обработки информации. применяемую при анализе и обработке естественных языков.

Правильный ответ: Токен

Компетенции (индикаторы): ПК-3

Задания открытого типа с кратким свободным ответом

1. Процесс, который нормализует выходы каждого слоя в мини-батче?

Правильный ответ: Batch Normalization

Компетенции (индикаторы): ПК-3

2. Какой метод используется для создания входного слоя в Functional API?

Правильный ответ: Input

Компетенции (индикаторы): ПК-3

3. Какой метод используется для компиляции модели?

Правильный ответ: compile

Компетенции (индикаторы): ПК-3

4. Какой метод используется для обучения модели на предоставленных данных?

Правильный ответ: fit

Компетенции (индикаторы): ПК-3

Задания открытого типа с развернутым ответом

1. Создать модель для классификации изображений из набора данных MNIST, который содержит рукописные цифры от 0 до 9. с использованием Keras.

Привести расширенное решение.

Время выполнения – 45 мин.

Ожидаемый результат:

```
# Импорт необходимых библиотек
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras import layers, models
from tensorflow.keras.datasets import mnist
# 1. Загрузка и подготовка данных
(x_train, y_train), (x_test, y_test) = mnist.load_data()
# Нормализация данных
x_train = x_train.astype('float32') / 255.0
x_test = x_test.astype('float32') / 255.0
# Изменение формы данных для подачи в нейронную сеть
x_train = np.expand_dims(x_train, axis=-1) # Добавляем размерность для
канала
x_test = np.expand_dims(x_test, axis=-1)

# 2. Создание модели
```

```

model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28,
1))) # Свёрточный слой
model.add(layers.MaxPooling2D((2, 2))) # Пуллинговый слой
model.add(layers.Conv2D(64, (3, 3), activation='relu')) # Второй
свёрточный слой
model.add(layers.MaxPooling2D((2, 2))) # Второй пуллинговый слой
model.add(layers.Flatten()) # Преобразование в одномерный вектор
model.add(layers.Dense(64, activation='relu')) # Полносвязный слой
model.add(layers.Dense(10, activation='softmax')) # Выходной слой для
10 классов
# 3. Компиляция модели
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

# 4. Обучение модели
model.fit(x_train, y_train, epochs=5, batch_size=64, validation_split=0.1)

# 5. Оценка модели
test_loss, test_acc = model.evaluate(x_test, y_test)
print(f'\nТестовая точность: {test_acc:.4f}')

# 6. Визуализация некоторых предсказаний
predictions = model.predict(x_test)

# Покажем несколько изображений и их предсказания
num_images = 5
plt.figure(figsize=(10, 2))
for i in range(num_images):
    plt.subplot(1, num_images, i + 1)
    plt.imshow(x_test[i].reshape(28, 28), cmap='gray')
    plt.title(f'Предсказание: {np.argmax(predictions[i])}')
    plt.axis('off')
plt.show()

```

Пояснения:

- Импорт библиотек: импортируем необходимые библиотеки, включая Keras и NumPy.
- Загрузка данных: загружаем набор данных MNIST, который уже разделен на обучающую и тестовую выборки. Данные нормализуются, чтобы значения пикселей находились в диапазоне от 0 до 1.

- Создание модели: создаем последовательную модель, добавляя свёрточные и пуллинговые слои, затем полносвязный слой и выходной слой с функцией активации softmax для многоклассовой классификации.
- Компиляция модели: компилируем модель, указывая оптимизатор (Adam), функцию потерь (sparse categorical crossentropy) и метрику (accuracy).
- Обучение модели: обучаем модель на обучающих данных, используя 5 эпох и размер пакета 64. Также указываем часть данных для валидации.
- Оценка модели: оцениваем модель на тестовых данных и выводим тестовую точность.
- Визуализация предсказаний: показываем несколько изображений из тестового набора и их предсказания.

Критерии оценивания:

- импорт необходимых библиотек;
- загрузка и подготовка данных;
- создание модели;
- компиляция модели;
- обучение модели;
- оценка модели.

Компетенции (индикаторы): ПК-3

2. Создать пример классификации изображения одежды, используя Keras и набор данных "Fashion MNIST". Этот набор данных содержит 70,000 изображений одежды, разделенных на 10 классов (например, обувь, футболки, куртки и т.д.).

Привести расширенное решение.

Время выполнения – 45 мин.

Ожидаемый результат:

```
# Импорт необходимых библиотек
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras import layers, models
from tensorflow.keras.datasets import fashion_mnist

# 1. Загрузка и подготовка данных
(x_train, y_train), (x_test, y_test) = fashion_mnist.load_data()

# Нормализация данных
x_train = x_train.astype('float32') / 255.0
x_test = x_test.astype('float32') / 255.0

# Изменение формы данных для подачи в нейронную сеть
```

```

x_train = np.expand_dims(x_train, axis=-1) # Добавляем размерность для
канала
x_test = np.expand_dims(x_test, axis=-1)
# 2. Создание модели
model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28,
1))) # Свёрточный слой
model.add(layers.MaxPooling2D((2, 2))) # Пуллинговый слой
model.add(layers.Conv2D(64, (3, 3), activation='relu')) # Второй
свёрточный слой
model.add(layers.MaxPooling2D((2, 2))) # Второй пуллинговый слой
model.add(layers.Flatten()) # Преобразование в одномерный вектор
model.add(layers.Dense(64, activation='relu')) # Полносвязный слой
model.add(layers.Dense(10, activation='softmax')) # Выходной слой для
10 классов
# 3. Компиляция модели
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
# 4. Обучение модели
model.fit(x_train, y_train, epochs=10, batch_size=64, validation_split=0.1)

# 5. Оценка модели
test_loss, test_acc = model.evaluate(x_test, y_test)
print(f'\nТестовая точность: {test_acc:.4f}')
# 6. Визуализация предсказаний
predictions = model.predict(x_test)
# Покажем несколько изображений и их предсказания
num_images = 5
plt.figure(figsize=(10, 2))
for i in range(num_images):
    plt.subplot(1, num_images, i + 1)
    plt.imshow(x_test[i].reshape(28, 28), cmap='gray')
    plt.title(f'Предсказание: {np.argmax(predictions[i])}')
    plt.axis('off')
plt.show()
Пояснения:
1. Импорт библиотек: импортируем необходимые библиотеки, включая
Keras, NumPy и Matplotlib.
2. Загрузка данных: загружаем набор данных Fashion MNIST, который
уже разделен на обучающую и тестовую выборки. Данные нормализуются,
чтобы значения пикселей находились в диапазоне от 0 до 1.

```

3. Подготовка данных: изменяем форму данных, добавляя размерность для канала, чтобы они соответствовали требованиям свёрточной нейронной сети.

4. Создание модели: создаем последовательную модель, добавляя свёрточные и пулинговые слои, затем полносвязный слой и выходной слой с функцией активации softmax для многоклассовой классификации.

5. Компиляция модели: компилируем модель, указывая оптимизатор (Adam), функцию потерь (sparse categorical crossentropy) и метрику (accuracy).

6. Обучение модели: обучаем модель на обучающих данных, используя 10 эпох и размер пакета 64. Также указываем часть данных для валидации.

7. Оценка модели: оцениваем модель на тестовых данных и выводим тестовую точность.

Критерии оценивания:

- импорт необходимых библиотек;
- загрузка и подготовка данных;
- создание модели;
- компиляция модели;
- обучение модели;
- оценка модели;
- визуализация результатов.

Компетенции (индикаторы): ПК-3

Экспертное заключение

Представленный комплект оценочных материалов по дисциплине «Нейронные сети» соответствует требованиям ФГОС ВО.

Предлагаемые оценочные материалы адекватны целям и задачам реализации основной профессиональной образовательной программы по направлению подготовки 09.04.04 Программная инженерия.

Виды оценочных средств, включенные в представленный фонд, отвечают основным принципам формирования ФОС.

Разработанные и представленные для экспертизы оценочные материалы рекомендуются к использованию в процессе подготовки обучающихся по указанному направлению.

Председатель учебно-методической комиссии
института компьютерных систем
и информационных технологий

Ветрова Н.Н.

Лист изменений и дополнений

№ п/п	Виды дополнений и изменений	Дата и номер протокола заседания кафедры (кафедр), на котором были рассмотрены и одобрены изменения и дополнения	Подпись (с расшифровкой) заведующего кафедрой (заведующих кафедрами)
1.	Дополнен комплектом оценочных материалов	протокол заседания кафедры компьютерных систем и сетей № <u>8</u> от <u>10.03.2025</u>	 С.В. Попов