Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и управление»

КАФЕДРА ИУК4 «Программная инженерия»

Домашняя работа №2

«Архитектурные особенности нейронных сетей. Библиотека Tensorflow. Автокодировщики»

ДИСЦИПЛИНА: «Интеллектуальные информационные системы анализа данных»

Выполнил: студент гр. ИУК4-21М _____ (Сафронов Н.С.)
(подпись) (Ф.И.О.)

Проверил: _____ (Белов Ю.С.)
(подпись) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:
- Оценка:

Калуга, 2025

Цель работы: приобретение практических навыков по реализации автокодировщика с использованием различных сверточных слоев.

Постановка задачи:

Разработать шумоподавляющий автокодировщик (используя библиотеку TensorFlow), который будет работать с набором данных MNIST. Параметры автокодировщика указаны в варианте. Выполнение домашней работы осуществляется на языке программирования Python с использованием окружения Anaconda и библиотек Scikit – Learn и TensorFlow. Использовать сторонние библиотеки (кроме Scikit–Learn, Matplotlib и TensorFlow), реализующие заявленную функциональность, запрещено.

Вариант 7

Автокодировщик: 4 слоя, функция активации hard_sigmoid. Сравнить результаты работы (визуально) автокодировщика для различных типов слоев (Conv2D, Conv2DTranspose, SeparableConv2D). Количество эпох обучения равно 7.

Результаты выполнения работы

```python
def build_conv2d_model():
    input_img = layers.Input(shape=(28, 28, 1))

    # Энкодер
    x = layers.Conv2D(32, (3,3), activation='hard_sigmoid', padding='same', strides=2)(input_img)
    x = layers.Conv2D(64, (3,3), activation='hard_sigmoid', padding='same', strides=2)(x)

    # Декодер
    x = layers.UpSampling2D((2, 2))(x)
    x = layers.Conv2D(64, (3, 3), activation='hard_sigmoid', padding='same')(x)
    x = layers.UpSampling2D((2, 2))(x)
    decoded = layers.Conv2D(1, (3, 3), activation='sigmoid', padding='same')(x)

    autoencoder = models.Model(input_img, decoded)
    autoencoder.compile(optimizer='adam', loss='mse')
    return autoencoder
```

Рисунок 1 – Архитектура сети с блоками Conv2D

```python
def build_transposed_model():
    input_img = layers.Input(shape=(28, 28, 1))

    # Энкодер
    x = layers.Conv2D(32, (3, 3), activation='hard_sigmoid', padding='same', strides=2)(input_img)
    x = layers.Conv2D(64, (3, 3), activation='hard_sigmoid', padding='same', strides=2)(x)

    # Декодер
    x = layers.Conv2DTranspose(64, (3, 3), strides=2, padding='same', activation='hard_sigmoid')(x)
    decoded = layers.Conv2DTranspose(1, (3, 3), strides=2, padding='same', activation='hard_sigmoid')(x)

    autoencoder = models.Model(input_img, decoded)
    autoencoder.compile(optimizer='adam', loss='mse')
    return autoencoder
```

Рисунок 2 – Архитектура сети с блоками Conv2DTranspose

```python
def build_separable_model():
    input_img = layers.Input(shape=(28, 28, 1))

    # Энкодер
    x = layers.SeparableConv2D(32, (3,3), activation='hard_sigmoid', padding='same', strides=2)(input_img)
    x = layers.SeparableConv2D(64, (3,3), activation='hard_sigmoid', padding='same', strides=2)(x)

    # Декодер
    x = layers.UpSampling2D((2, 2))(x)
    x = layers.Conv2D(64, (3, 3), activation='hard_sigmoid', padding='same')(x)
    x = layers.UpSampling2D((2, 2))(x)
    decoded = layers.Conv2D(1, (3, 3), activation='sigmoid', padding='same')(x)

    autoencoder = models.Model(input_img, decoded)
    autoencoder.compile(optimizer='adam', loss='mse')
    return autoencoder
```

Рисунок 3 – Архитектура сети с блоками SeparableConv2D

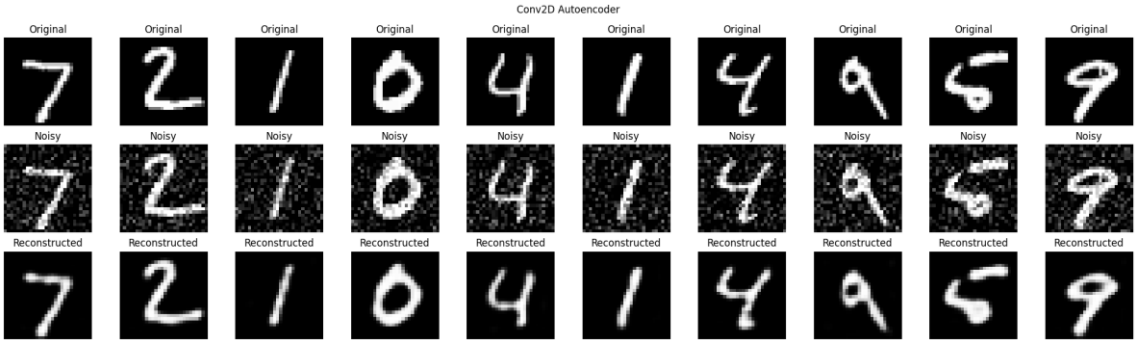

Рисунок 4 – Результат сети с блоками Conv2D

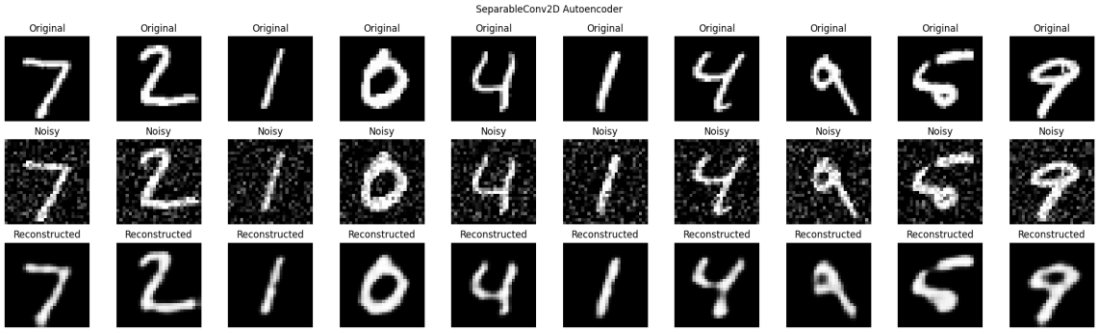

Рисунок 5 – Результат сети с блоками SeparableConv2D

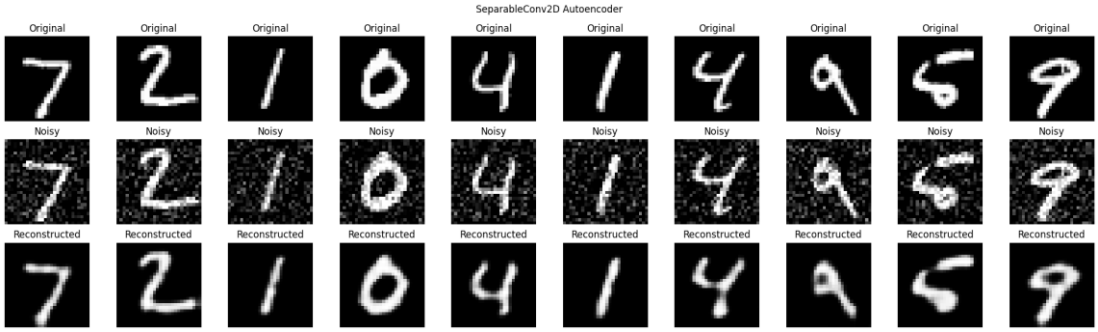

Рисунок 5 – Результат сети с блоками SeparableConv2D

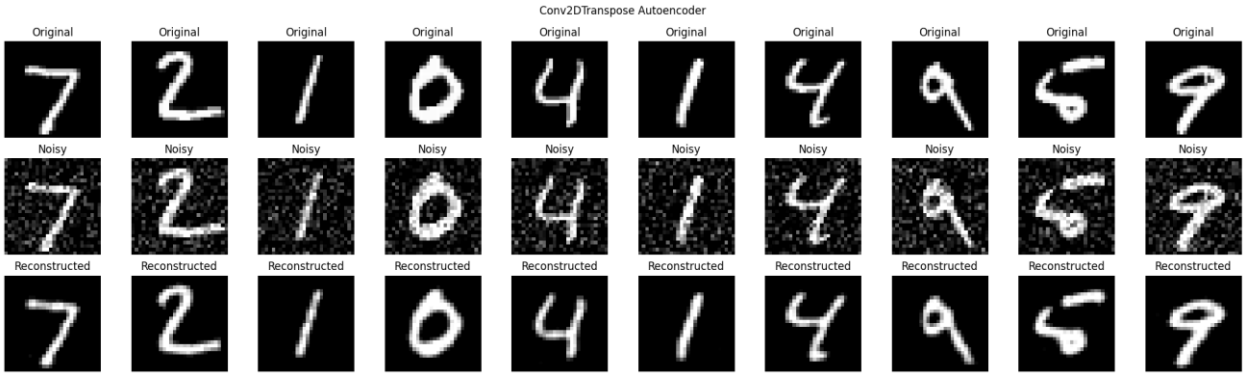

Рисунок 5 – Результат сети с блоками Conv2DTranspose

Вывод: в ходе выполнения домашней работы были получены практические навыки по реализации автокодировщика с использованием различных сверточных слоев.