



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 №1

«Глубокая генеративно-сопоставительная сеть GAN»

ДИСЦИПЛИНА: «Методы глубокого обучения»

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: получение практических навыков построения генеративно-сопоставительной сети.

Постановка задачи:

На основе разобранного примера реализуйте генеративно-сопоставительную сеть. В соответствии с выданным вариантом используйте как объект для реализации:

Вариант 2

Изображение животных.

Результаты выполнения работы

```
def make_generator():
    model = keras.Sequential([
        layers.Input(shape=(latent_dim,)),
        layers.Reshape((1, 1, latent_dim)),
        layers.Conv2DTranspose(512, 4, strides=1, padding='valid', use_bias=False),
        layers.BatchNormalization(),
        layers.ReLU(),
        layers.Conv2DTranspose(256, 4, strides=2, padding='same', use_bias=False),
        layers.BatchNormalization(),
        layers.ReLU(),
        layers.Conv2DTranspose(128, 4, strides=2, padding='same', use_bias=False),
        layers.BatchNormalization(),
        layers.ReLU(),
        layers.Conv2DTranspose(64, 4, strides=2, padding='same', use_bias=False),
        layers.BatchNormalization(),
        layers.ReLU(),
        layers.Conv2DTranspose(3, 4, strides=2, padding='same', activation='tanh')
    ])
    return model
```

Рисунок 1 – Архитектура генератора

```
def (variable) model: Any
    model = keras.Sequential([
        layers.Input(shape=(64, 64, 3)),
        layers.Conv2D(64, 4, strides=2, padding='same'),
        layers.LeakyReLU(0.2),
        layers.Conv2D(128, 4, strides=2, padding='same', use_bias=False),
        layers.BatchNormalization(),
        layers.LeakyReLU(0.2),
        layers.Conv2D(256, 4, strides=2, padding='same', use_bias=False),
        layers.BatchNormalization(),
        layers.LeakyReLU(0.2),
        layers.Conv2D(512, 4, strides=2, padding='same', use_bias=False),
        layers.BatchNormalization(),
        layers.LeakyReLU(0.2),
        layers.Conv2D(1, 4, strides=1, padding='valid'),
        layers.Flatten(),
        layers.Activation('sigmoid')
    ])
    return model
```

Рисунок 2 – Архитектура дискриминатора

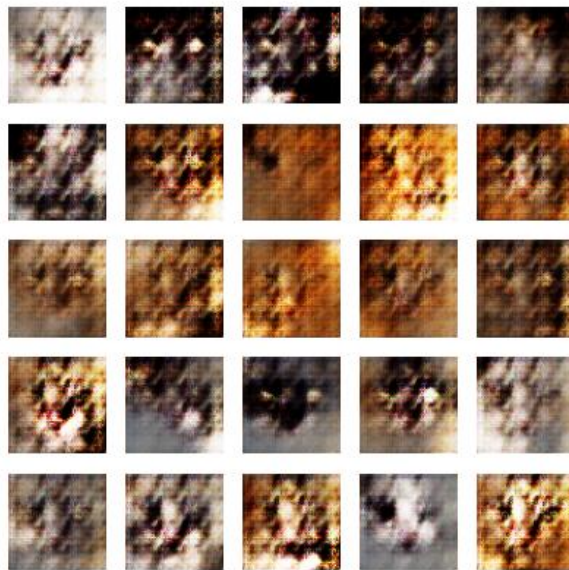


Рисунок 3 – Результат после 10 эпох обучения

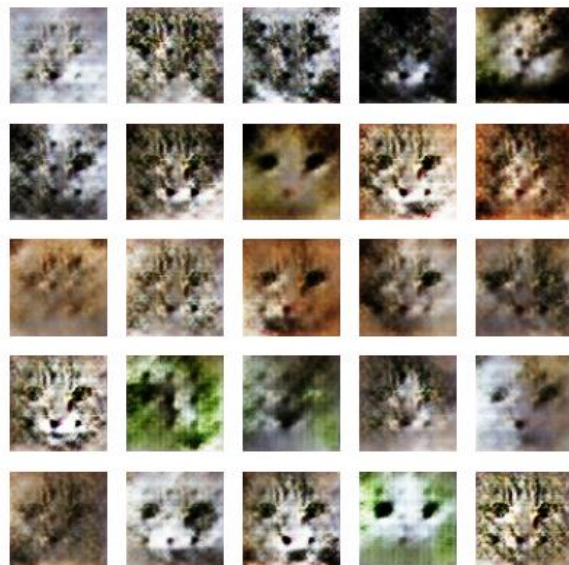


Рисунок 4 – Результат после 20 эпох обучения

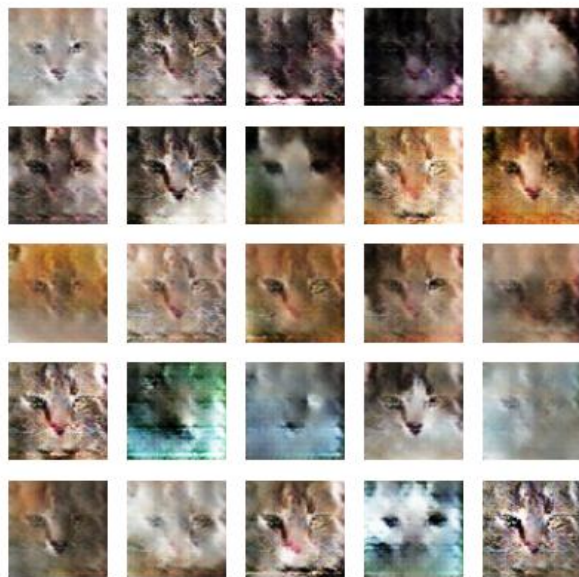


Рисунок 5 – Результат после 30 эпох обучения

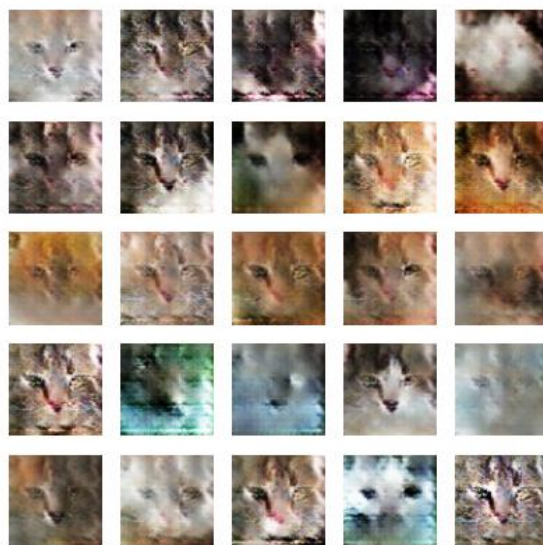


Рисунок 6 – Результат после 40 эпох обучения

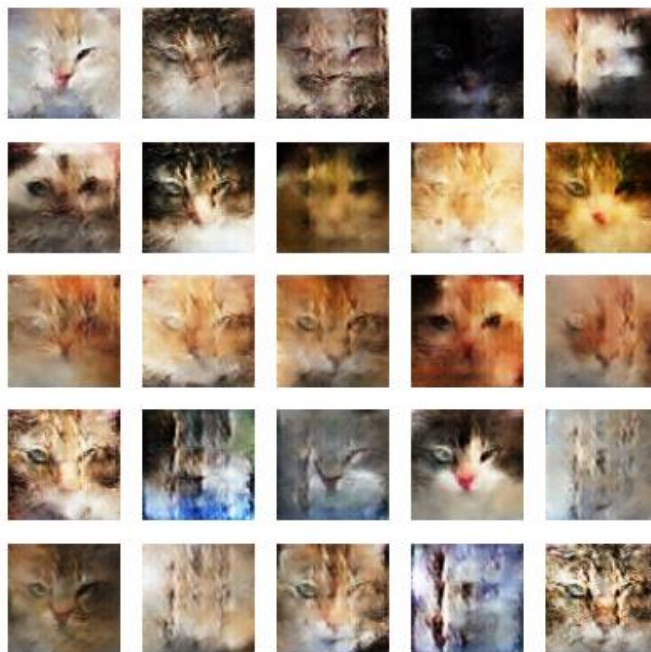


Рисунок 7 – Результат после 50 эпох обучения

Вывод: в ходе выполнения домашней работы были получены практические навыки построения генеративно-сопоставительной сети.