

Вариант 4 – 5 (номер в зачетке – 11)

$$X \in N(0,10) \quad e \in N(0,0.3)$$

Признак	1	2	3	4	5	6	7
Формула	$\cos(X)+e$	$-X+e$	$\sin(X)*X+e$	$\sqrt{ X }+e$	X^2+e	$- X +4$	$X-(X^2)/5+e$

Выполнение

В функции `generate_data()` были сгенерированы данные, соответствующие варианту:

```
def generate_data(size):
    data = []
    target = []
    for i in range(size):
        X = np.random.normal(0, 10)
        e = np.random.normal(0, 0.3)
        data.append((np.cos(X)+e, -X+e, np.sin(X)*X+e,
np.sqrt(np.fabs(X))+e, -np.fabs(X)+4, X-(X**2)/5+e))
        target.append([X ** 2 + e])
    return np.round(np.array(data), decimals=5), np.round(np.array(target), decimals=5)
```

В массиве `target` находятся данные цели регрессии (результат функции X^2+e), в массиве `data` – результаты остальных функций.

Затем сгенерированные данные были нормализованы:

```
mean = np.mean(train_data, axis=0)
train_data -= mean
std = np.std(train_data, axis=0)
train_data /= std
test_data -= mean
test_data /= std
```

Затем была создана модель с одним входом и двумя выходами (encoded делится на decoded и regress). Структура:

```
encoded = Dense(32, activation='relu')(inputs)
encoded = Dense(32, activation='relu')(encoded)
```

```
encoded = Dense(2, activation='relu')(encoded)

decoded = Dense(64, activation='relu', name='dec1')(encoded)
decoded = Dense(64, activation='relu', name='dec2')(decoded)
decoded = Dense(64, activation='relu', name='dec3')(decoded)
decoded = Dense(6, name='dec4')(decoded)

regress = Dense(32, activation='relu')(encoded)
regress = Dense(64, activation='relu')(regress)
regress = Dense(64, activation='relu')(regress)
regress = Dense(64, activation='relu')(regress)
regress = Dense(1, name='regress')(regress)

encoder = Model(inputs, encoded)
regress_model = Model(inputs, regress)
autoencoder = Model(inputs, decoded)
```

Для компиляции используется оптимизатор Adam, обучение проходит в течении 150 эпох пакетами по 5 образцов.

Модели сохраняются в формате h5.

Данные в файлах decoded_data.csv и test_data.csv почти совпадают, как и в файлах regress_data.csv и test_targets.csv