Практическое задание №7

Вариант №6

Условие: Необходимо построить рекуррентную нейронную сеть, которая будет прогнозировать значение некоторого периодического сигнала.

Для выполнения задания необходимо:

-Преобразовать последовательность в датасет, который можно подавать на вход нейронной сети (можно использовать функцию gen_data_from_sequence)

-Разбить датасет на обучающую, контрольную и тестовую выборку

-Построить и обучить модель

-Построить график последовательности, предсказанной на тестовой выборке

Вариант:

```
def func(i):
    return (i % 16 + 1) / 16

def gen_sequence(seq_len = 1000):
    seq = [math.cos(i/10) * func(i) + random.normalvariate(0, 0.04) for i in range(seq_len)]
    return np.array(seq)
```

Выполнение:

Была реализована рекуррентная нейронная сеть согласно примеру из методических указаний, последовательность была в датасет пригодный для обработки ИНС, а также разбита на обучающую, контрольную и тестовую выборки.

```
data, res = gen_data_from_sequence()

dataset_size = len(data)
train_size = (dataset_size // 10) * 7
val_size = (dataset_size - train_size) // 2

train_data, train_res = data[:train_size], res[:train_size]
val_data,val_res=data[train_size:train_size+val_size],
                        res[train_size:train_size+val_size]
test_data,test_res=data[train_size+val_size:],res[train_size+val_size:]
```

```
model = Sequential()
model.add(layers.GRU(32,recurrent_activation='sigmoid',input_shape=(Non
e,1),return_sequences=True))
model.add(layers.LSTM(32,activation='relu',input_shape=(None,1),return_
sequences=True,dropout=0.2))
model.add(layers.GRU(32,input_shape=(None,1),recurrent_dropout=0.2))
model.add(layers.Dense(1))

model.compile(optimizer='nadam', loss='mse')
history=model.fit(train_data,train_res,epochs=50,validation_data=
                  (val_data, val_res))
```

Результаты работы ИНС были следующими:

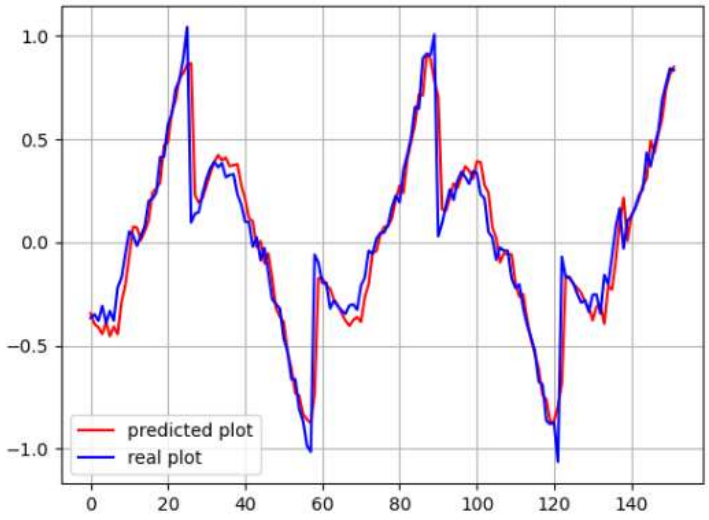

Рисунок 1 – Графики предсказания ИНС1.

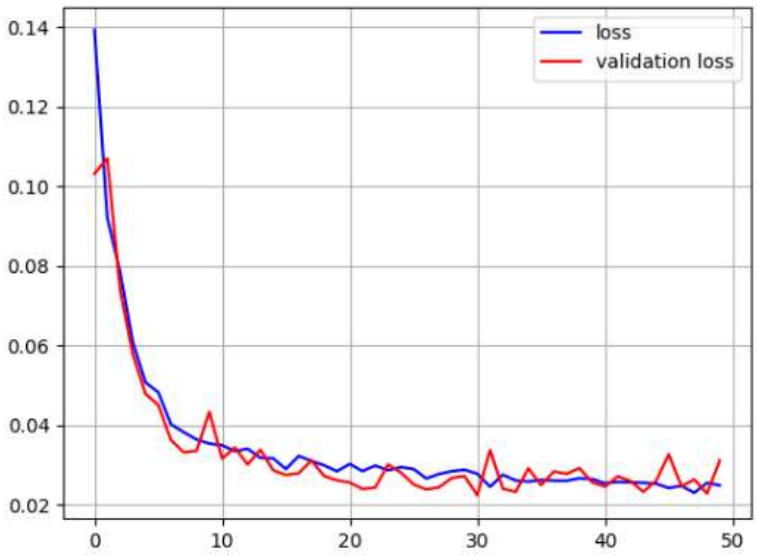

Рисунок 2 – Loss ИНС1.

Изменим архитектуру ИНС, чтобы добиться лучшего результата. Т.к. ИНС обучается достаточно быстро, заменим слои GRU на слои LST, увеличим количество эпох обучения до 70.

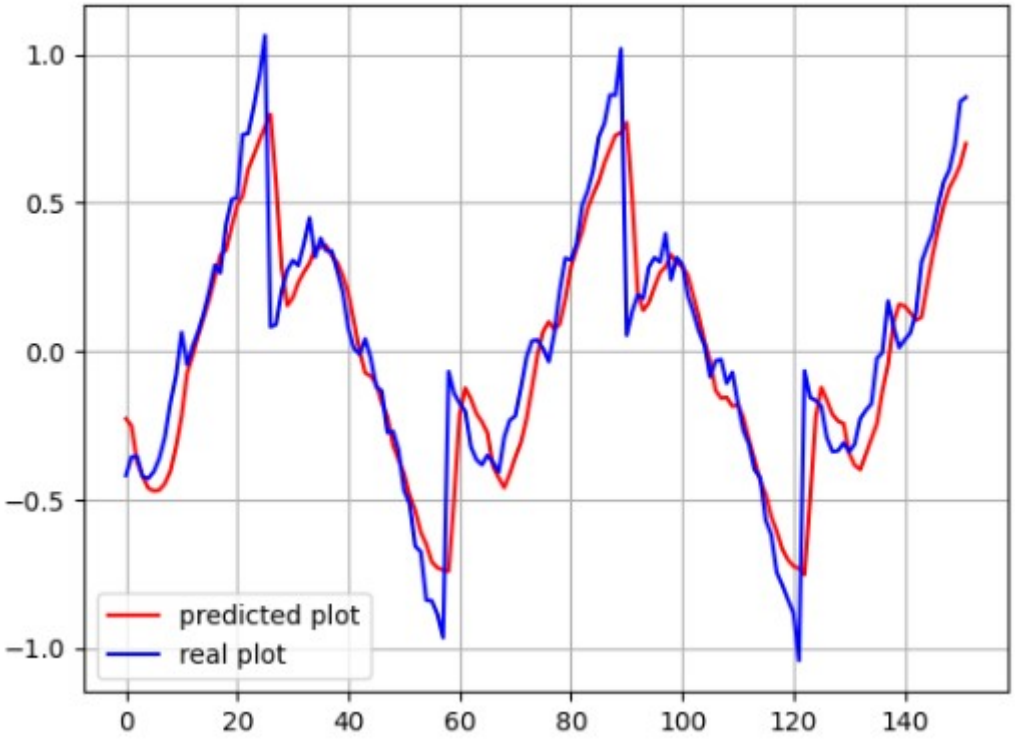

Рисунок 3 – Графики предсказания ИНС2.

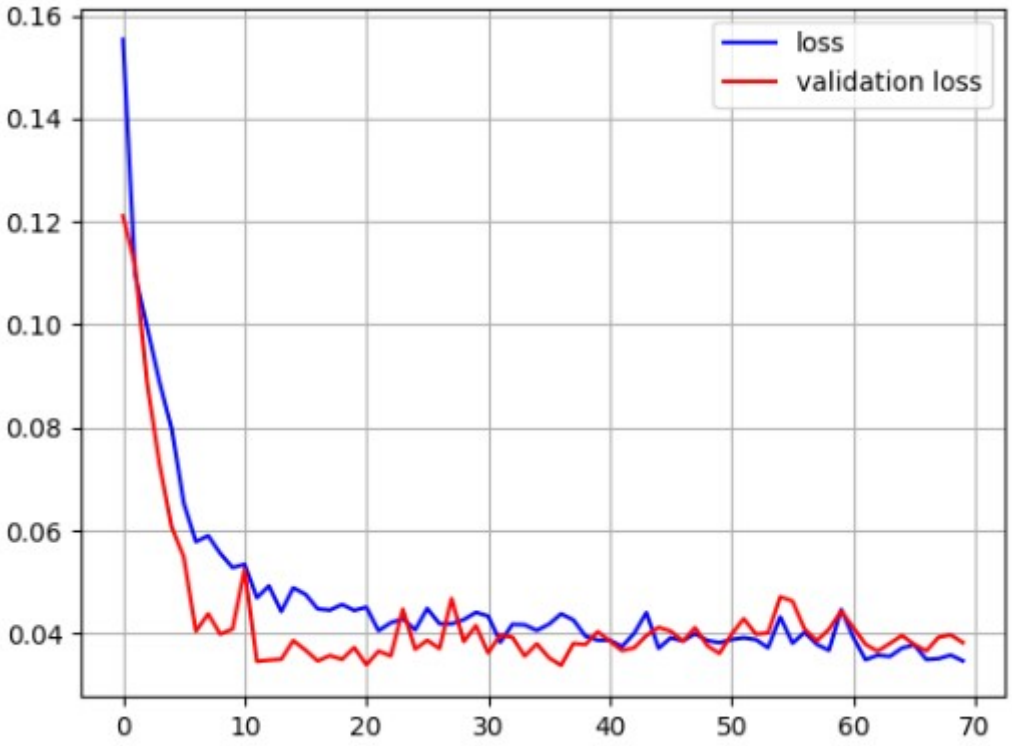

Рисунок 4 – Loss ИНС1.

Добиться повышения точности не удалось, скорее произошло обратное. Добавим слои GRU, изменим функции активации

```
model = Sequential()
model.add(layers.GRU(32,recurrent_activation='sigmoid',input_shape=(None,1),return_sequences=True))
model.add(layers.GRU(64,activation='relu',input_shape=(None,1),return_sequences=True,dropout=0.2))
model.add(layers.GRU(32,activation='sigmoid',input_shape=(None,1),return_sequences=True,dropout=0.2))
model.add(layers.GRU(48,activation='relu',input_shape=(None,1),return_sequences=True,dropout=0.2))
model.add(layers.GRU(32,input_shape=(None,1),recurrent_dropout=0.2))
model.add(layers.Dense(1))

model.compile(optimizer='nadam', loss='mse')
history=model.fit(train_data,train_res,epochs=70,validation_data=(val_data, val_res)
```

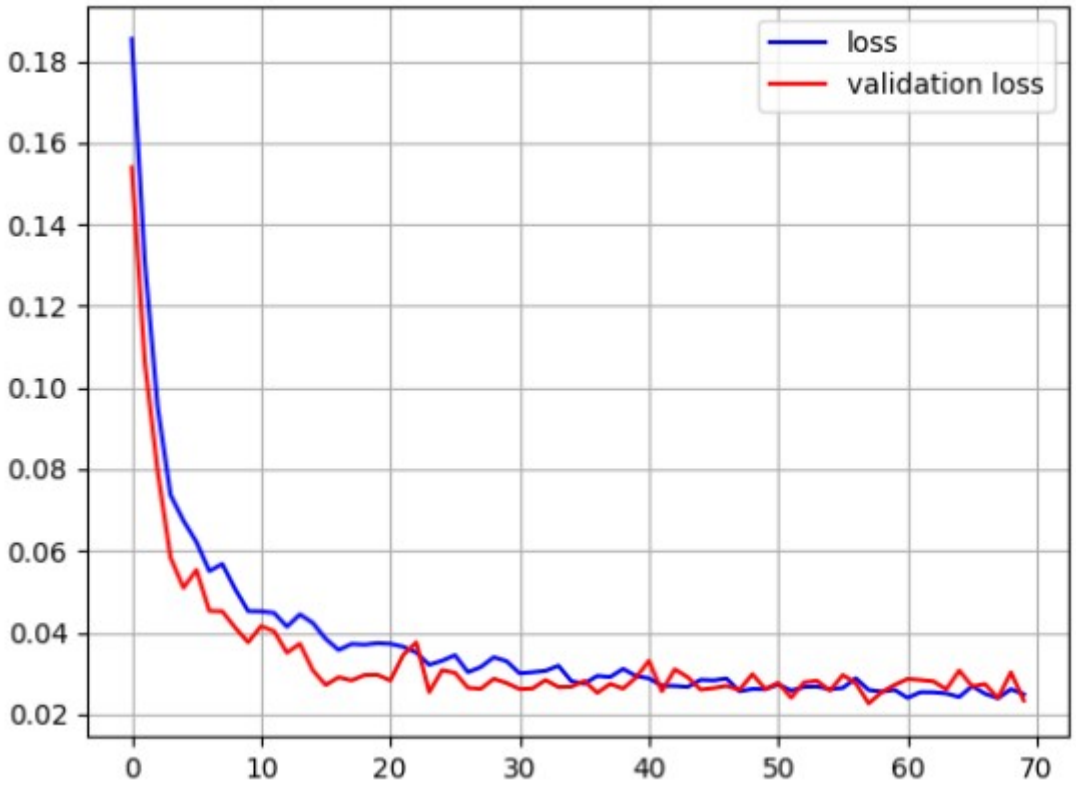

Рисунок 5 – Loss ИНС3.

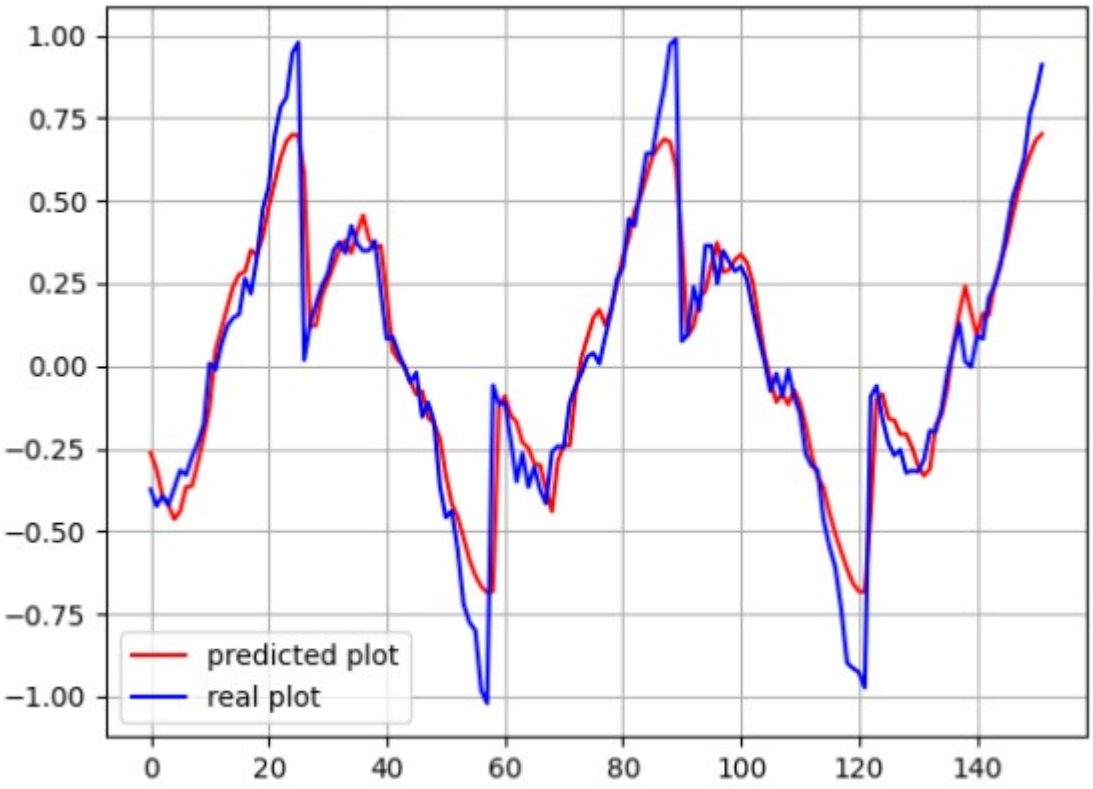

Рисунок 6 – Графики предсказания ИНС3.

ИНС3 показывает loss 0.0233 на проверочных данных, что является лучшим результатом, которого мне удалось достичь.