

Практическое задание №7

Вариант №4

Условие:

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

Разбить датасет на обучающую, контрольную и тестовую выборку

Построить и обучить модель

Построить график последовательности, предсказанной на тестовой выборке (пример построения также есть в примере). Данный график необходимо также добавить в `pr`

Также, в файлах с кодом вариантов есть функция `draw_sequence`, которая позволяет нарисовать часть последовательности

Вариант 4

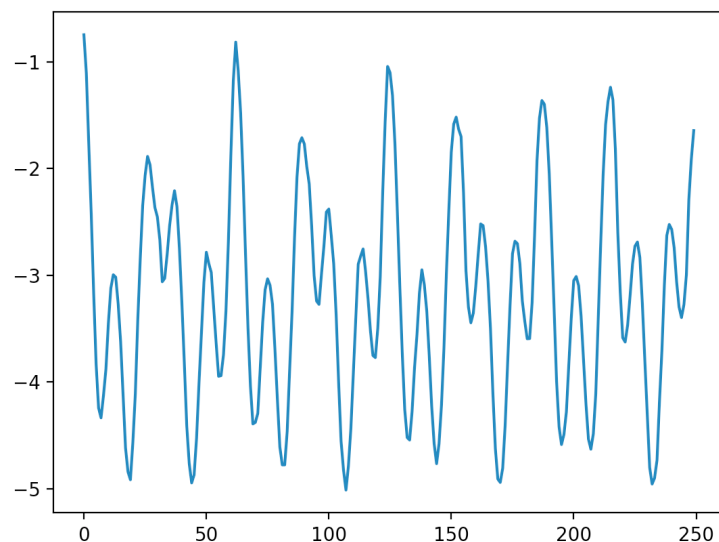
```
def func(i):  
    i = i % 31  
    return ((i-15) ** 2)/100 - 4  
  
def gen_sequence(seq_len = 1000):  
    seq = [math.cos(i/2) + func(i) + random.normalvariate(0, 0.04) for i  
    in range(seq_len)]  
    return np.array(seq)
```

Выполнение:

С помощью функции `gen_data_from_sequence` был сгенерирован датасет: данные были подготовлены и разбиты на обучающие, проверочные и тестовые в соотношении 70:15:15

```
def gen_data_from_sequence(seq_len = 1006, lookback = 10):  
    seq = gen_sequence(seq_len)  
    past = np.array([[seq[j]] for j in range(i, i+lookback)] for i in  
    range(len(seq) - lookback)])  
    future = np.array([[seq[i]] for i in range(lookback, len(seq))])  
    return (past, future)
```

Полученная последовательность:



Была построена модель ИНС1:

Model: "sequential"

Layer (type)	Output Shape	Param #
gru (GRU)	(None, None, 32)	3360
lstm (LSTM)	(None, None, 32)	8320
gru_1 (GRU)	(None, 32)	6336
dense (Dense)	(None, 1)	33

Total params: 18,049

Trainable params: 18,049

Non-trainable params: 0

Данная модель была обучена, результаты следующие:

График потерь.

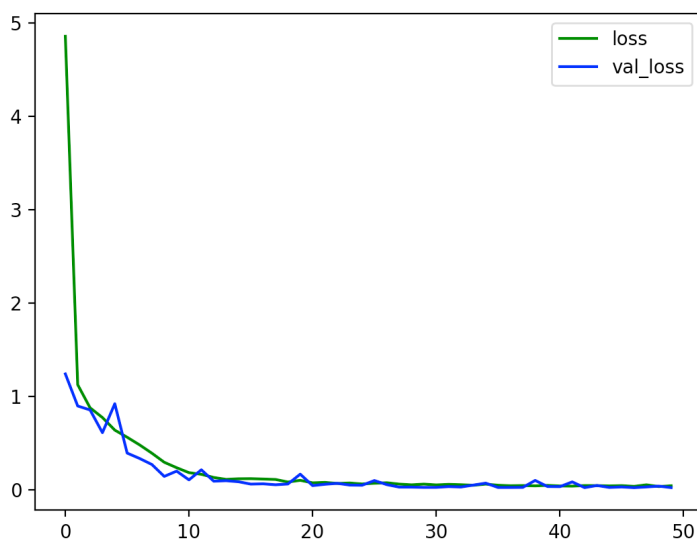
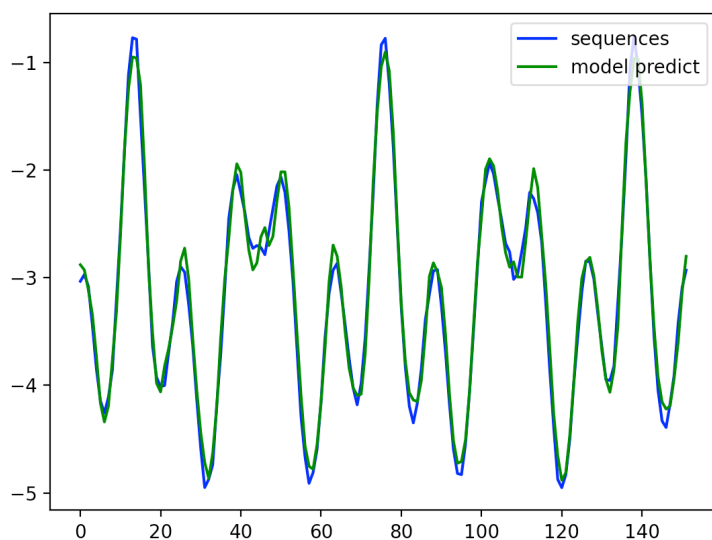


График предсказанной последовательности:



loss: 0.0444 - val_loss: 0.0206.

Видно, что последовательность предсказывается достаточно точно, промахи образуются в местах перегиба.

Была изменена архитектура сети: вместо слоя LSTM поставлен слой GRU с активацией relu и 64 нейронами.

ИНС2:

Model: "sequential"

Layer (type)	Output Shape	Param #
gru (GRU)	(None, None, 32)	3360

gru_1 (GRU)	(None, None, 64)	18816
-------------	------------------	-------

gru_2 (GRU)	(None, 32)	9408
-------------	------------	------

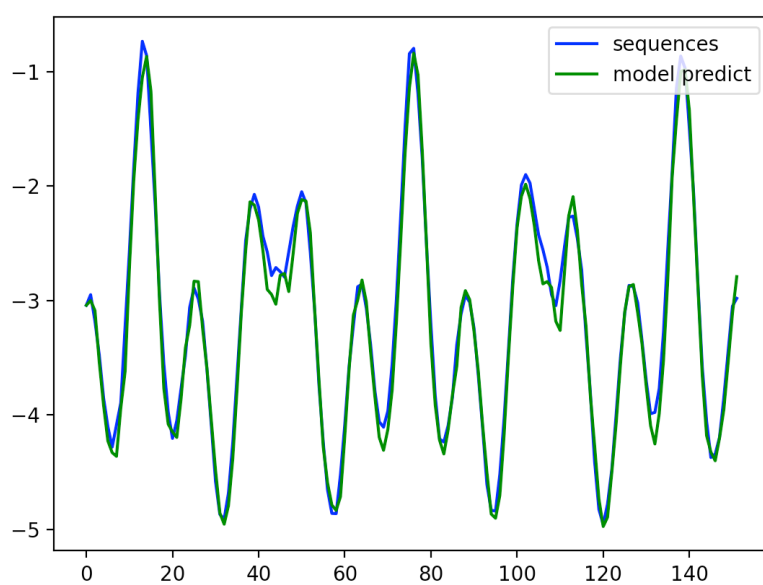
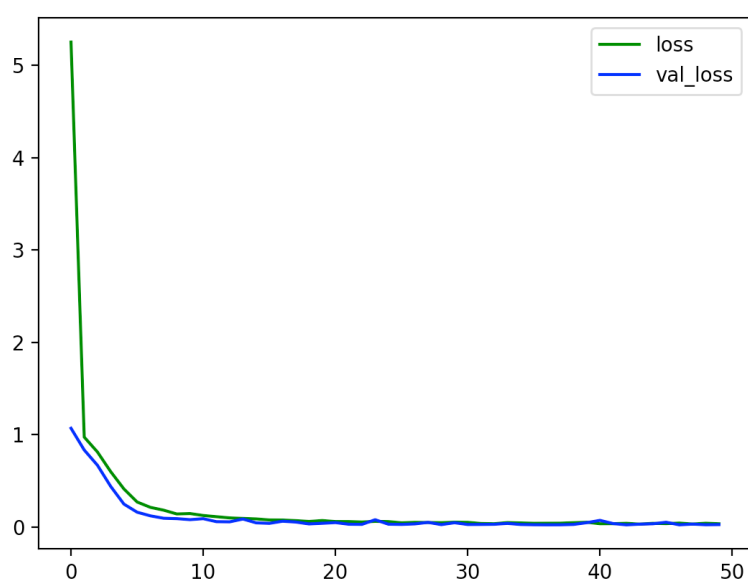
dense (Dense)	(None, 1)	33
---------------	-----------	----

Total params: 31,617

Trainable params: 31,617

Non-trainable params: 0

Были получены следующие результаты:



Loss 0.0405 - val_loss: 0.0269

Видно, что особых изменений не произошло, потери возросли не сильно.

Было добавлено еще два слоя GRU

Model: "sequential"

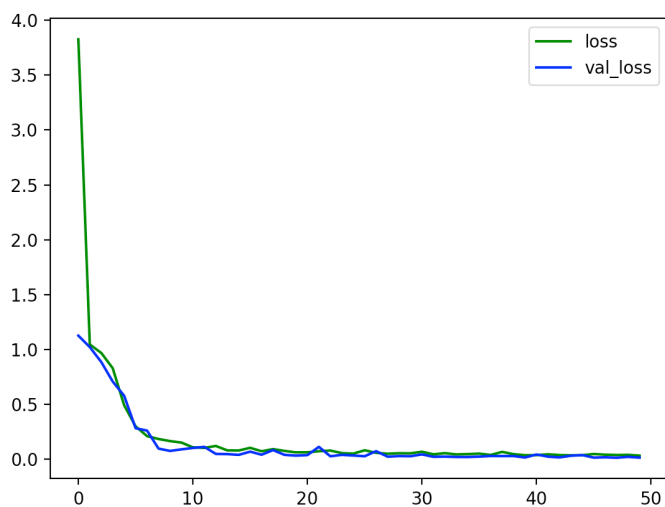
Layer (type)	Output Shape	Param #
gru (GRU)	(None, None, 32)	3360
gru_1 (GRU)	(None, None, 64)	18816
gru_2 (GRU)	(None, None, 32)	9408
gru_3 (GRU)	(None, None, 64)	18816
gru_4 (GRU)	(None, 32)	9408
dense (Dense)	(None, 1)	33

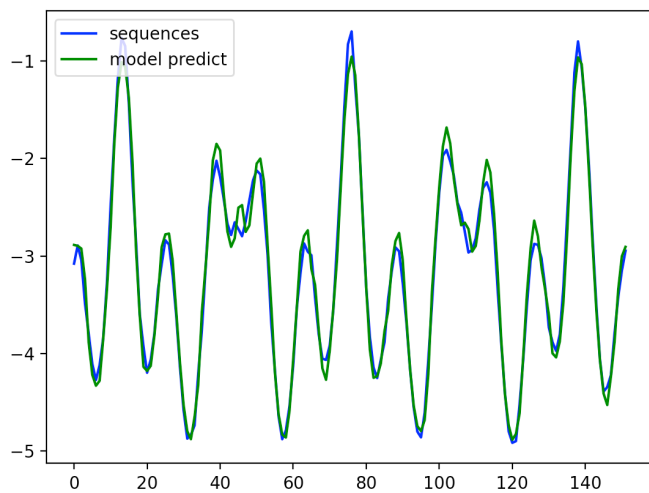
Total params: 59,841

Trainable params: 59,841

Non-trainable params: 0

Полученные результаты:





loss: 0.0339 - val_loss: 0.0155

Потери на обучающих данных возросли, на тестовых же упали. Можно сказать, что к улучшению результатов это не привело.

Наилучшие результаты показала исходная модель.