

Аверина Ольга, 8383

Практическое задание 7

Вариант 1

Необходимо построить рекуррентную нейронную сеть, которая будет прогнозировать значение некоторого периодического сигнала. К каждому варианту предоставляется код, который генерирует последовательность. Для выполнения задания необходимо:

1. Преобразовать последовательность в датасет, который можно подавать на вход нейронной сети (можно использовать функцию `gen_data_from_sequence` из примера)
2. Разбить датасет на обучающую, контрольную и тестовую выборку
3. Построить и обучить модель
4. Построить график последовательности, предсказанной на тестовой выборке (пример построения также есть в примере). Данный график необходимо также добавить в `pr`

Генерация данных.

Был сгенерирован датасет с помощью функции, приведенной в примере:

```
def gen_data_from_sequence(seq_len = 1006, lookback = 10):  
    seq = var1.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
```

При помощи функции `gen_sequence`, содержащейся в файле с вариантом задания, генерируется последовательность. На рис. 1 изображена часть сгенерированной последовательности.

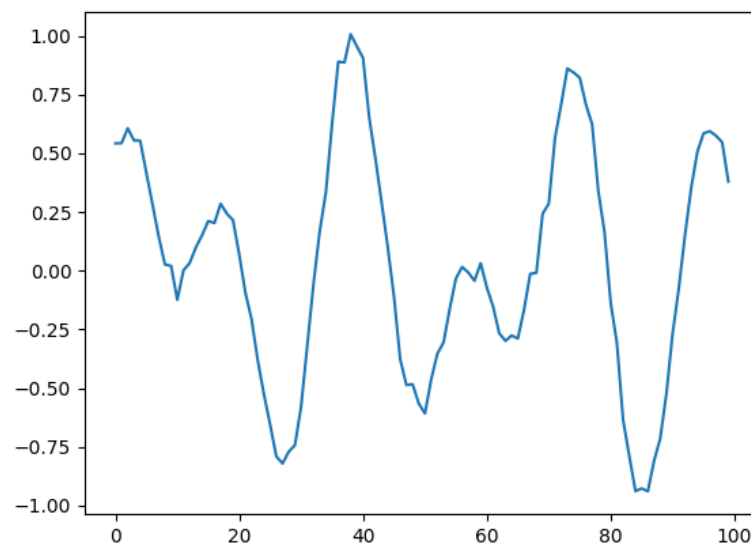


Рис. 1

Разбиение выборки.

Разобьем датасет на обучающую и тестовую выборки, а контрольную выделим уже при обучении модели. Обучающая выборка будет составлять 80% всего набора данных, тестовая – остальные 20%.

```
dataset_size = len(data)
train_size = (dataset_size // 10) * 9
train_data, train_res = data[:train_size], res[:train_size]
test_data, test_res = data[train_size:], res[train_size:]
```

Построение модели.

Построим модель:

```
model = Sequential()
```

```

model.add(layers.GRU(32, recurrent_activation='sigmoid', input_shape=(None, 1), return_sequences=True))
model.add(layers.GRU(32, activation='relu', input_shape=(None, 1), return_sequences=True))
model.add(layers.GRU(32, input_shape=(None, 1), recurrent_dropout=0.2))
model.add(layers.Dense(1))
model.compile(optimizer='nadam', loss='mse')

```

Обучение модели.

Модель была обучена в течение 50 эпох батчами по 32 образца, на контрольную выборку отведено 15% данных. На рис. 2 изображены графики потерь на тренировочной и валидационной выборках.

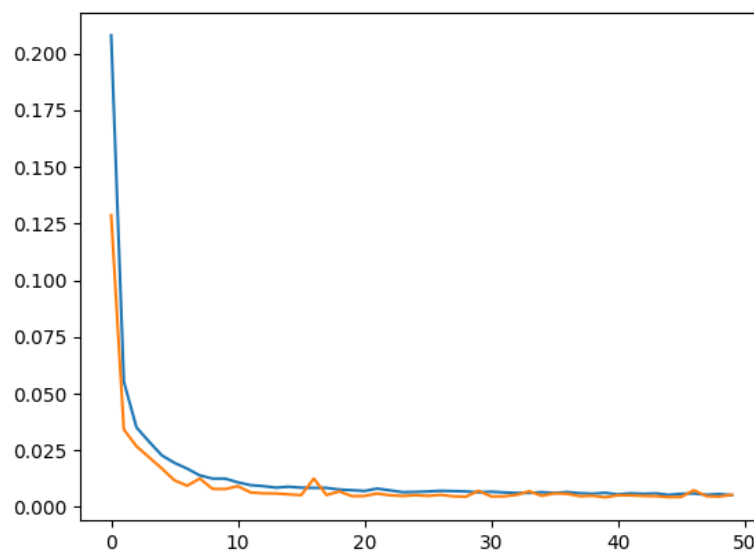


Рис. 2

График предсказанной последовательности.

На рис. 3 изображены график сгенерированной последовательности и график предсказанной последовательности.

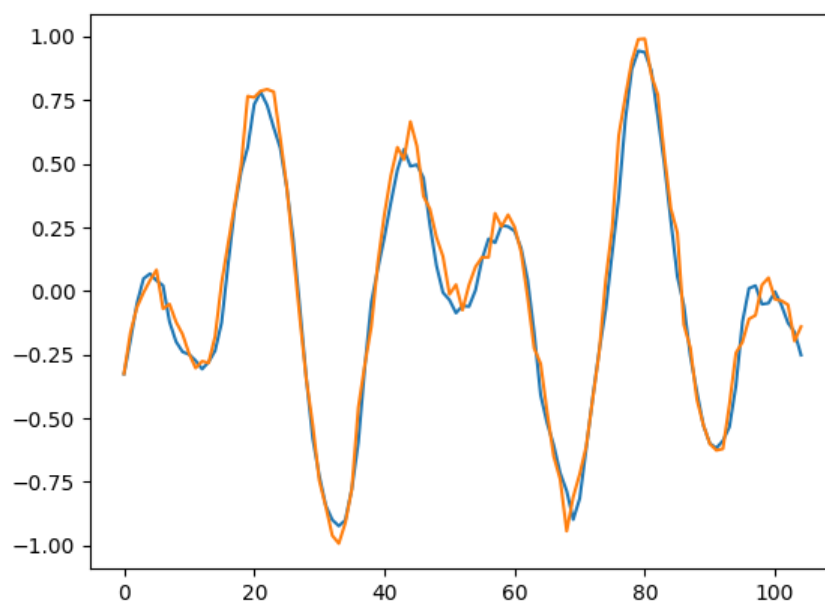


Рис. 3

Как видно из рисунка, значения последовательности предсказываются достаточно точно, но в точках перегиба наблюдаются небольшие отклонения.

Вывод.

В ходе выполнения практической работы была построена рекуррентная нейронная сеть, которая прогнозирует значение некоторого периодического сигнала.