



```

future = np.array([[seq[i]] for i in range(lookback,
len(seq))])
return past, future

```

Датасет был разбит на обучающую, контрольную и тестовую выборку (листинг 3). Обучающие данные используют 70% от датасета, контрольные – 50% от датасета за вычетом обучающих данных, и тестовые – оставшиеся данные.

### Листинг 3 – Разбиение данных

```

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:]

```

Затем была построена и обучена модель (листинг 4).

### Листинг 4 – Построение и обучение модели

```

model = Sequential()
model.add(layers.GRU(32, recurrent_activation='sigmoid',
input_shape=(None, 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))

```

После этого, были построены графики последовательности, предсказанной на тестовой выборке (листинг 5).

### Листинг 5 – Построение графиков

```

loss = history.history['loss']
val_loss = history.history['val_loss']
plt.plot(range(len(loss)), loss)

```

```
plt.plot(range(len(val_loss)), val_loss)
plt.show()
```

```
predicted_res = model.predict(test_data)
pred_length = range(len(predicted_res))
plt.plot(pred_length, predicted_res)
plt.plot(pred_length, test_res)
plt.show()
```

Результат выполнения программы:

Epoch 50/50

22/22 [=====] - 0s 8ms/step - loss: 0.0502

- val\_loss: 0.0474

Потери на обучающих данных составляют 0.05, на контрольных – 0.047.

Графики представлены на рис. 1.

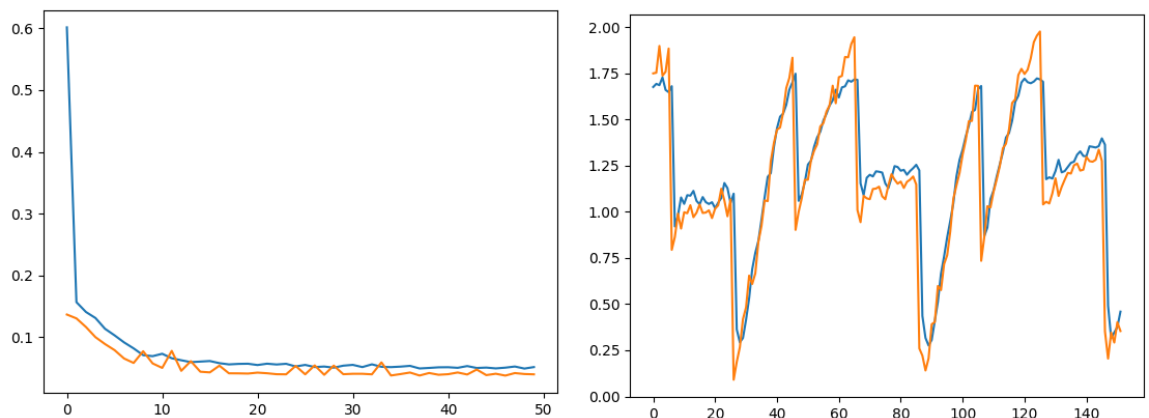


Рисунок 1 – Графики

Первый график – график ошибок на тренировочных (синяя кривая) и на контрольных данных (оранжевая). На втором графике показано предсказание некоторого периодического сигнала (синий график) и то, как сигнал должен выглядеть (оранжевый).

Если добавить нормализацию данных (листинг 5), то потери на тренировочных упадут до 0.043, а на контрольных – до 0.037. Графики см. на рис. 2.

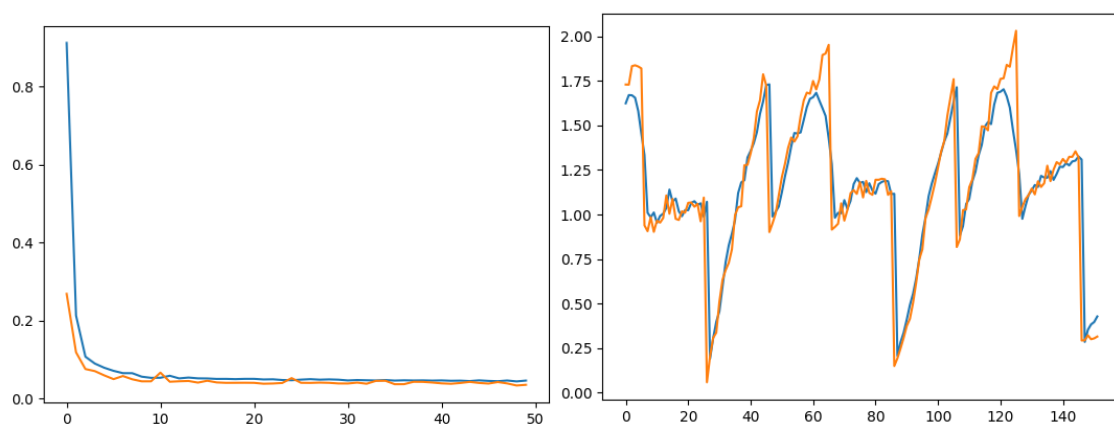


Рисунок 2 – Графики