



# Прогнозирование генерации электроэнергии солнечными панелями с использованием нейронных сетей

Васильев Степан Петрович

Руководитель:

ассистент

Холодов Александр Сергеевич

## ЦЕЛЬ

- Изучение функционала открытой нейросетевой библиотеки Keras.
- Прогнозирование генерации солнечной электростанции с помощью различных типов рекуррентных нейронных сетей: GRU, SimpleRNN, LSTM.

## АКТУАЛЬНОСТЬ

- С нарастающей тенденцией перехода к децентрализованному энергоснабжению значительно увеличивается доля генерации возобновляемых источников энергии (ВИЭ).
- Солнечные панели – перспективный способ генерации электричества.
- Для удовлетворения спроса и выполнения критерия надежности электроснабжения важно знать объем генерируемой мощности.

## СОЛНЕЧНАЯ ЭНЕРГЕТИКА

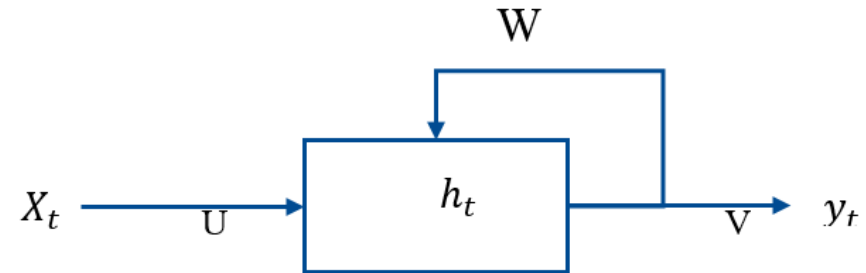


- Рисунок 1 – Карта эффективности использования ВИЭ

\* С.В. Голубев Возобновляемые источники энергии в энергетике газовой отрасли. Перспективы и аспекты применения ВИЭ на объектах ПАО «Газпром» / Газовая промышленность, №12, 2016.

## ТЕОРЕТИЧЕСКАЯ СВОДКА

- Рекуррентные нейронные сети (РНС) – вид нейронных сетей, архитектура которых состоит из трех слоёв: входного, скрытого и выходного. При этом скрытый слой имеет обратную связь сам на себя.
- РНС способны обрабатывать серии событий во времени или последовательные пространственные цепочки.



Модель рекуррентного блока.

$$h_t = \sigma(W h_{t-1} + U x_t + b_h);$$

$$y_t = \sigma(V h_t + b_y),$$

где  $\sigma$  – нелинейная функция;

$x_t$  – входной вектор номер  $t$ ;

$h_t$  (hidden) – состояние скрытого слоя для входа  $x_t$

;

$y_t$  – выход сети для входа  $x_t$  ;

$U, W, V$  – весовые матрицы нейронной сети;

$b_h, b_y$  (bias) – векторы сдвига.

## ИСХОДНЫЕ ДАННЫЕ. ОБРАБОТКА И ПРОВЕРКА КАЧЕСТВА

- В качестве исходных данных (ИД) выступает датасет с различными видами временных рядов в промежутке одного года дискретностью 1 час (60 минут): потребление электроэнергии, тариф на электроэнергию, генерация солнечных панелей.

```
In [3]: ts = pd.read_csv('time_series_60min_singleindex.csv', parse_dates=[0,1])
```

```
In [4]: ts.head(5)
```

```
Out[4]:
```

|   | utc_timestamp                | cet_cest_timestamp           | DE_load_actual_entsoe_power_statistics | DE_load_actual_entsoe_transparency | DE_load_forecast_entsoe_transparency | DE_ |
|---|------------------------------|------------------------------|--|------------------------------------|--------------------------------------|-----|
| 0 | 2018-04-08<br>22:00:00+00:00 | 2018-04-09<br>00:00:00+02:00 | 45401.510204                           | 41944                              | 41935.0                              |     |
| 1 | 2018-04-08<br>23:00:00+00:00 | 2018-04-09<br>01:00:00+02:00 | 43654.091837                           | 40544                              | 40497.0                              |     |
| 2 | 2018-04-09<br>00:00:00+00:00 | 2018-04-09<br>02:00:00+02:00 | 42991.785714                           | 40113                              | 39871.0                              |     |
| 3 | 2018-04-09<br>01:00:00+00:00 | 2018-04-09<br>03:00:00+02:00 | 43443.765306                           | 40436                              | 40268.0                              |     |
| 4 | 2018-04-09<br>02:00:00+00:00 | 2018-04-09<br>04:00:00+02:00 | 45393.193878                           | 42321                              | 42044.0                              |     |

```
In [5]: ts.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8784 entries, 0 to 8783
Data columns (total 9 columns):
 utc_timestamp                8784 non-null datetime64[ns, UTC]
 cet_cest_timestamp            8784 non-null object
 DE_load_actual_entsoe_power_statistics  7152 non-null float64
 DE_load_actual_entsoe_transparency      8784 non-null int64
 DE_load_forecast_entsoe_transparency    8712 non-null float64
 DE_price_day_ahead             8638 non-null float64
 DE_solar_capacity              6914 non-null float64
 DE_solar_generation_actual      8784 non-null int64
 DE_solar_profile               6914 non-null float64
dtypes: datetime64[ns, UTC](1), float64(5), int64(2), object(1)
memory usage: 617.8+ KB
```

## ИСХОДНЫЕ ДАННЫЕ. ОБРАБОТКА И ПРОВЕРКА КАЧЕСТВА

- В качестве исходных данных (ИД) выступает датасет с различными видами временных рядов в промежутке одного года дискретностью 1 час (60 минут): потребление электроэнергии, тариф на электроэнергию, генерация солнечных панелей.

```
In [3]: ts = pd.read_csv('time_series_60min_singleindex.csv', parse_dates=[0,1])
```

```
In [4]: ts.head(5)
```

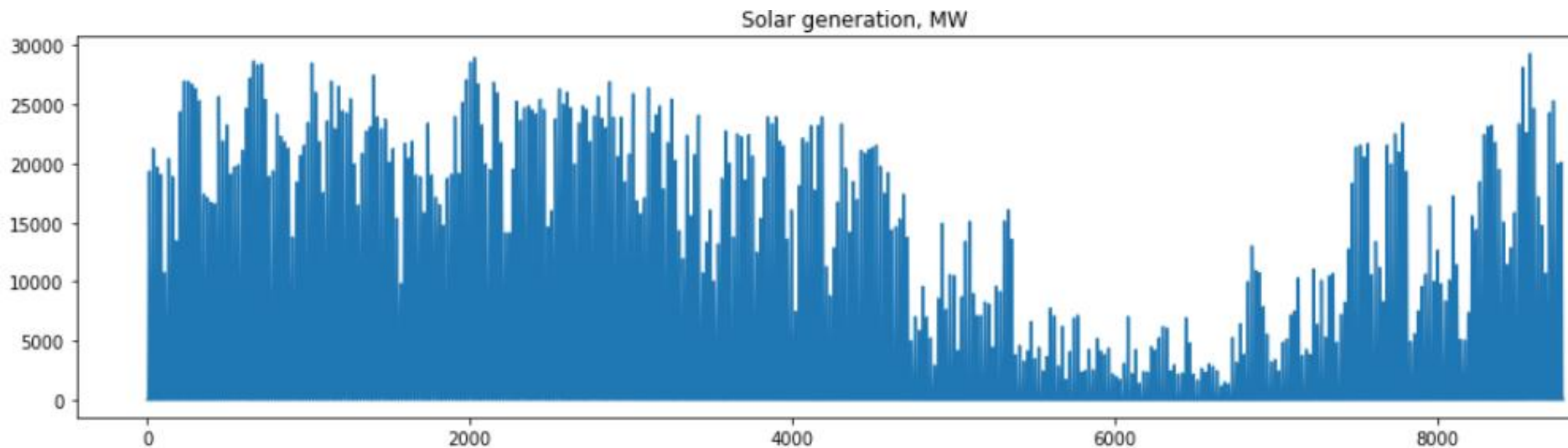
```
Out[4]:
```

|   | utc_timestamp                | cet_cest_timestamp           | DE_load_actual_entsoe_power_statistics | DE_load_actual_entsoe_transparency | DE_load_forecast_entsoe_transparency | DE_ |
|---|------------------------------|------------------------------|--|------------------------------------|--------------------------------------|-----|
| 0 | 2018-04-08<br>22:00:00+00:00 | 2018-04-09<br>00:00:00+02:00 | 45401.510204                           | 41944                              | 41935.0                              |     |
| 1 | 2018-04-08<br>23:00:00+00:00 | 2018-04-09<br>01:00:00+02:00 | 43654.091837                           | 40544                              | 40497.0                              |     |
| 2 | 2018-04-09<br>00:00:00+00:00 | 2018-04-09<br>02:00:00+02:00 | 42991.785714                           | 40113                              | 39871.0                              |     |
| 3 | 2018-04-09<br>01:00:00+00:00 | 2018-04-09<br>03:00:00+02:00 | 43443.765306                           | 40436                              | 40268.0                              |     |
| 4 | 2018-04-09<br>02:00:00+00:00 | 2018-04-09<br>04:00:00+02:00 | 45393.193878                           | 42321                              | 42044.0                              |     |

```
In [5]: ts.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8784 entries, 0 to 8783
Data columns (total 9 columns):
 utc_timestamp                8784 non-null datetime64[ns, UTC]
 cet_cest_timestamp           8784 non-null object
 DE_load_actual_entsoe_power_statistics  7152 non-null float64
 DE_load_actual_entsoe_transparency      8784 non-null int64
 DE_load_forecast_entsoe_transparency    8712 non-null float64
 DE_price_day_ahead            8638 non-null float64
 DE_solar_capacity             6914 non-null float64
 DE_solar_generation_actual      8784 non-null int64
 DE_solar_profile              6914 non-null float64
dtypes: datetime64[ns, UTC](1), float64(5), int64(2), object(1)
memory usage: 617.8+ KB
```

## ИСХОДНЫЕ ДАННЫЕ. ОБРАБОТКА И ПРОВЕРКА КАЧЕСТВА



- Солнечная генерация



## ИСХОДНЫЕ ДАННЫЕ. ОБРАБОТКА И ПРОВЕРКА КАЧЕСТВА

- В качестве проверки ИД на корректность был использован датасет с данными о состоянии окружающей среды, включающий скорость ветра, температуру, данные о солнечной радиации.

```
In [12]: #Позода
weather = pd.read_csv('weather_data.csv', parse_dates=[0])

In [13]: weather.head(5)

Out[13]:
```

|   | utc_timestamp             | DE_windspeed_10m | DE_temperature | DE_radiation_direct_horizontal | DE_radiation_diffuse_horizontal | DE11_windspeed_10m | DE11_temperature |
|---|---------------------------|------------------|----------------|--------------------------------|---------------------------------|--------------------|------------------|
| 0 | 2015-04-09 00:00:00+00:00 | 2.452845         | 4.103          | 0.0                            | 0.0                             | 3.32               | 3.30             |
| 1 | 2015-04-09 01:00:00+00:00 | 2.456084         | 3.715          | 0.0                            | 0.0                             | 3.42               | 2.87             |
| 2 | 2015-04-09 02:00:00+00:00 | 2.569132         | 3.388          | 0.0                            | 0.0                             | 3.45               | 2.50             |
| 3 | 2015-04-09 03:00:00+00:00 | 2.716226         | 3.132          | 0.0                            | 0.0                             | 3.44               | 2.13             |
| 4 | 2015-04-09 04:00:00+00:00 | 2.850298         | 3.023          | 0.0                            | 0.0                             | 3.39               | 1.75             |

5 rows × 157 columns

```
In [14]: weather.tail()

Out[14]:
```

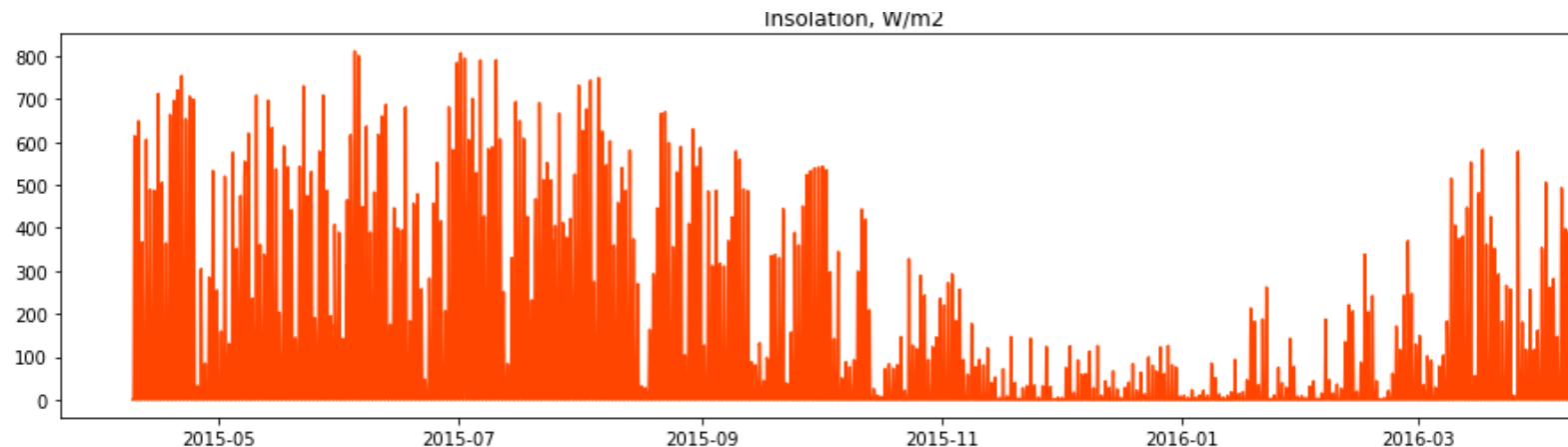
|      | utc_timestamp             | DE_windspeed_10m | DE_temperature | DE_radiation_direct_horizontal | DE_radiation_diffuse_horizontal | DE11_windspeed_10m | DE11_temperat |
|------|---------------------------|------------------|----------------|--------------------------------|---------------------------------|--------------------|---------------|
| 8803 | 2016-04-09 19:00:00+00:00 | 2.494406         | 7.399          | 0.012519                       | 0.204381                        | 1.31               | 6             |
| 8804 | 2016-04-09 20:00:00+00:00 | 2.708159         | 6.899          | 0.000000                       | 0.000000                        | 1.28               | 6             |
| 8805 | 2016-04-09 21:00:00+00:00 | 2.844921         | 6.497          | 0.000000                       | 0.000000                        | 1.14               | 5             |
| 8806 | 2016-04-09 22:00:00+00:00 | 2.908171         | 6.130          | 0.000000                       | 0.000000                        | 1.01               | 5             |
| 8807 | 2016-04-09 23:00:00+00:00 | 2.880794         | 5.789          | 0.000000                       | 0.000000                        | 0.87               | 5             |

5 rows × 157 columns

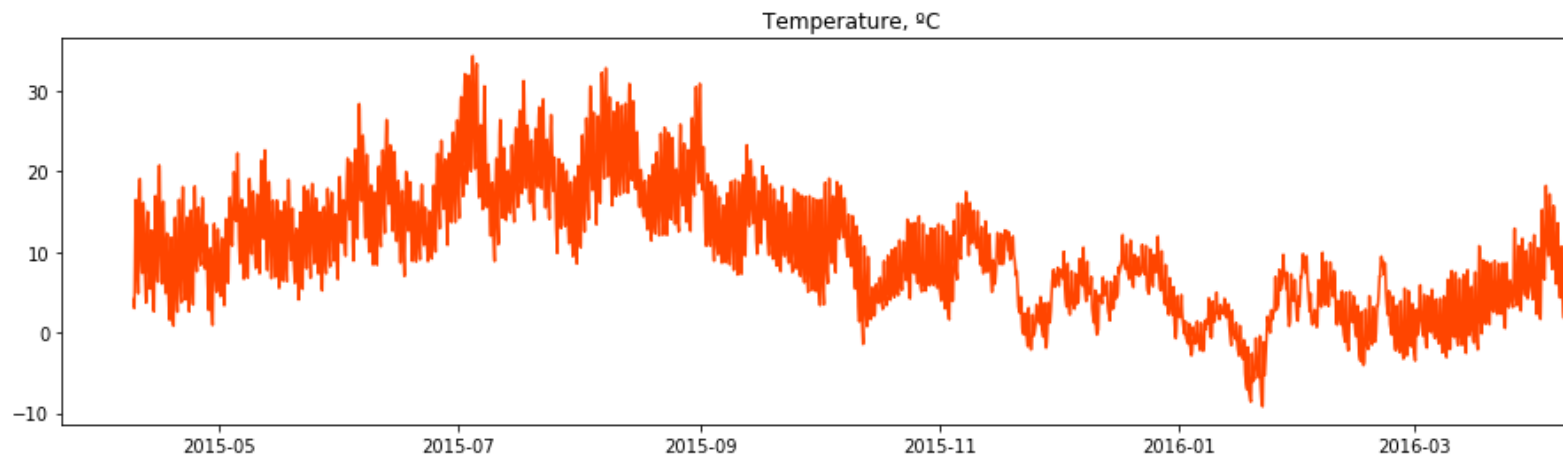
```
In [15]: weather.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8808 entries, 0 to 8807
Columns: 157 entries, utc_timestamp to DE00_radiation_diffuse_horizontal
dtypes: datetime64[ns, UTC](1), float64(156)
memory usage: 10.6 MB
```

## ИСХОДНЫЕ ДАННЫЕ. ОБРАБОТКА И ПРОВЕРКА КАЧЕСТВА



- Инсоляция

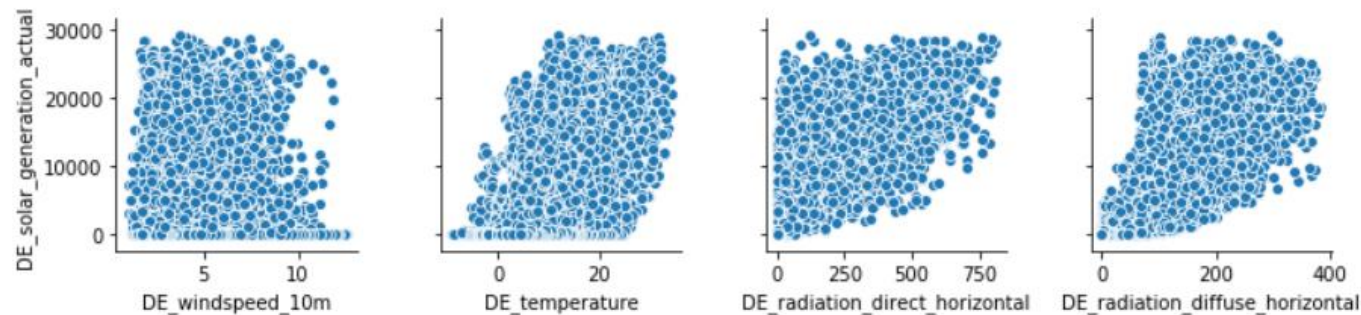


- Температура

## ИСХОДНЫЕ ДАННЫЕ. ОБРАБОТКА И ПРОВЕРКА КАЧЕСТВА

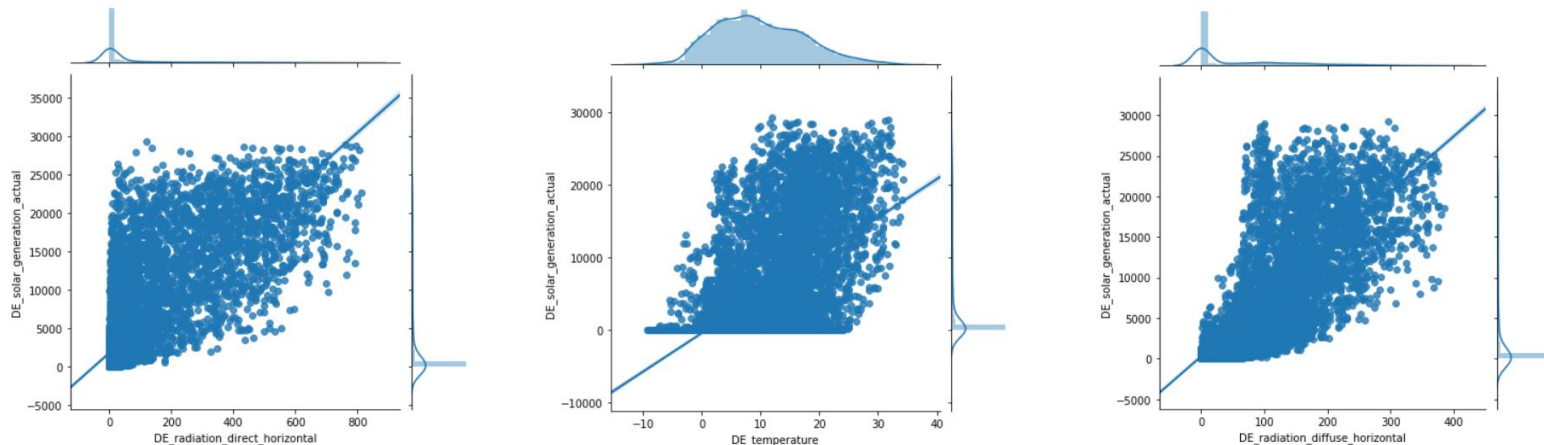
```
In [22]: sns.pairplot(comb, x_vars=['DE_windspeed_10m', 'DE_temperature', 'DE_radiation_direct_horizontal',  
                                     'DE_radiation_diffuse_horizontal'],  
                               y_vars=['DE_solar_generation_actual'])
```

```
Out[22]: <seaborn.axisgrid.PairGrid at 0x7f4a2247fb10>
```

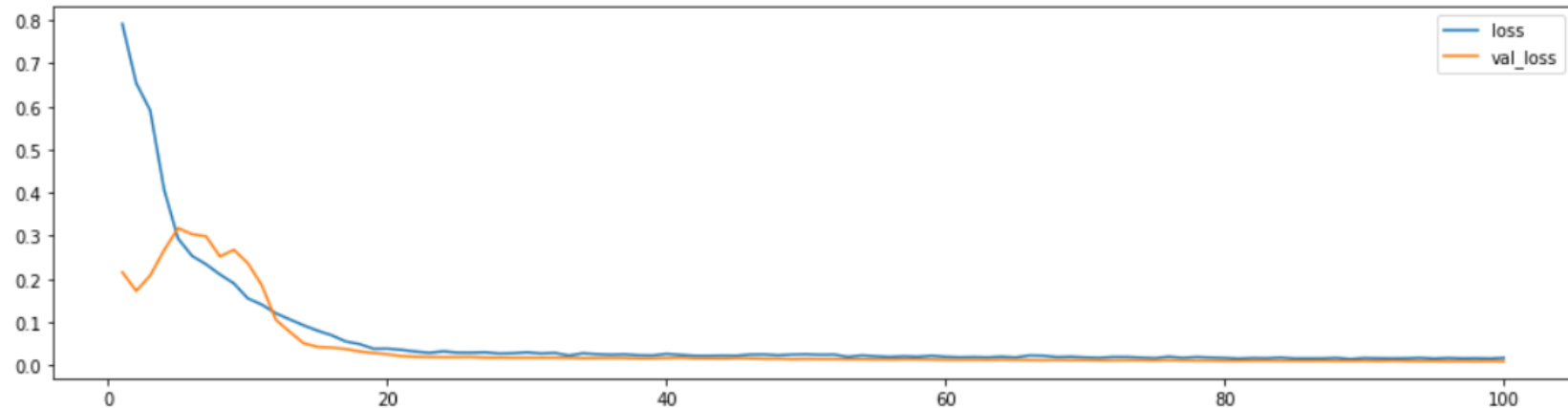


```
In [23]: sns.jointplot(x='DE_radiation_direct_horizontal', y='DE_solar_generation_actual', data=comb, kind='reg')  
sns.jointplot(x='DE_temperature', y='DE_solar_generation_actual', data=comb, kind='reg')  
sns.jointplot(x='DE_radiation_diffuse_horizontal', y='DE_solar_generation_actual', data=comb, kind='reg')
```

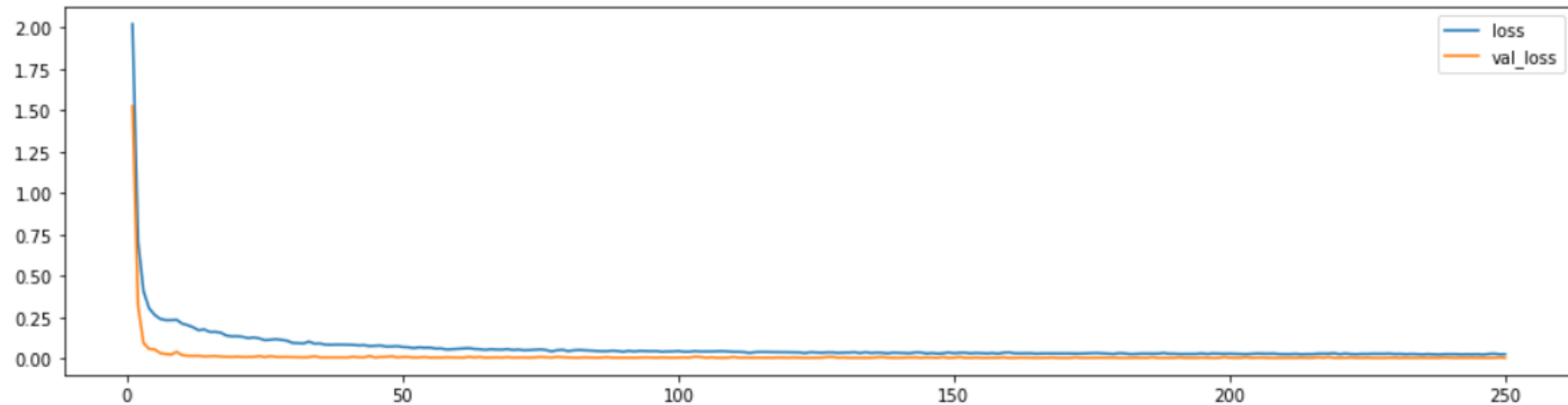
```
Out[23]: <seaborn.axisgrid.JointGrid at 0x7f4a24db3410>
```



## РЕЗУЛЬТАТЫ РАБОТЫ МОДЕЛЕЙ НЕЙРОННЫХ СЕТЕЙ

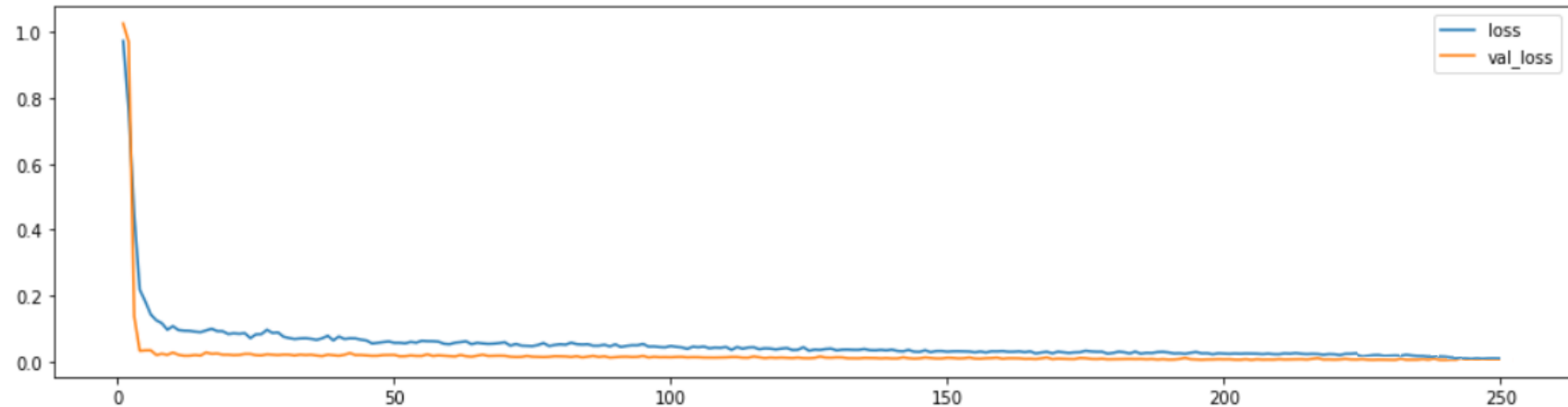


- Графики потерь GRU



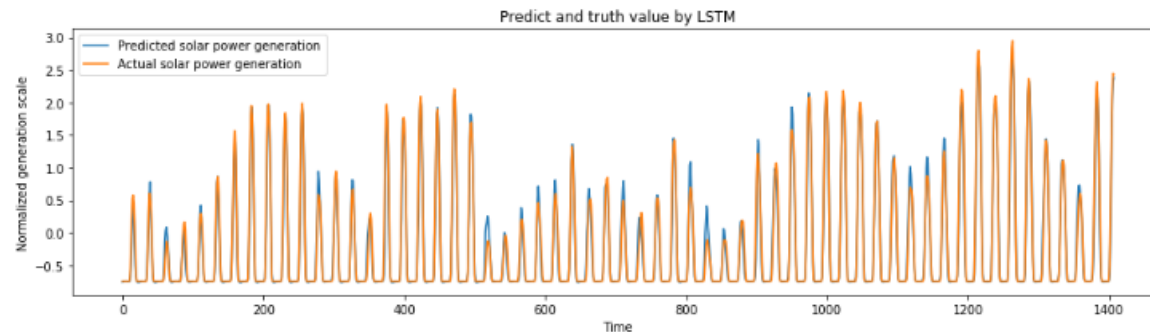
- Графики потерь SimpleRNN

## РЕЗУЛЬТАТЫ РАБОТЫ МОДЕЛЕЙ НЕЙРОННЫХ СЕТЕЙ



- Графики потерь LSTM

```
In [65]: plt.figure(figsize=(16,4))
plt.plot(predLSTM_Test1[0:], label = 'Predicted solar power generation')
plt.plot(truthLSTM_Test1[0:], label = 'Actual solar power generation')
plt.title('Predict and truth value by LSTM')
plt.xlabel('Time')
plt.ylabel('Normalized generation scale')
plt.legend();
```



## РЕЗУЛЬТАТЫ ПРОГНОЗИРОВАНИЯ СОЛНЕЧНОЙ ГЕНЕРАЦИИ

| <b>Рекуррентная<br/>нейросеть</b> | <b>Среднее абсолютное<br/>отклонение</b> | <b>Среднее абсолютное<br/>отклонение в<br/>именованных<br/>параметрах, МВт</b> |
|-----------------------------------|--|--|
| <b>GRU</b>                        | 0,09070427                               | 718,40   |
| <b>SimpleRNN</b>                  | 0,097224064                              | 770,04   |
| <b>LSTM</b>                       | 0,06989208                               | 553,56   |

## ЗАКЛЮЧЕНИЕ

1. Освоены основные принципы использования НС для реализации моделей прогнозирования временных последовательностей.
2. Произведено прогнозирование генерации солнечных панелей с помощью различных типов рекуррентной нейронной сети. Лучшие результаты по итогам показала PHC LSTM.



**СПАСИБО ЗА ВНИМАНИЕ!**

Васильев Степан Петрович  
[wardenwalle@gmail.com](mailto:wardenwalle@gmail.com)  
+7(916)929-70-64