
DeepSeries

Deep Learning Models for time series prediction.

Models

- ☒ Seq2Seq / Attention
- ☒ WaveNet
- ☐ Bert / Transformer

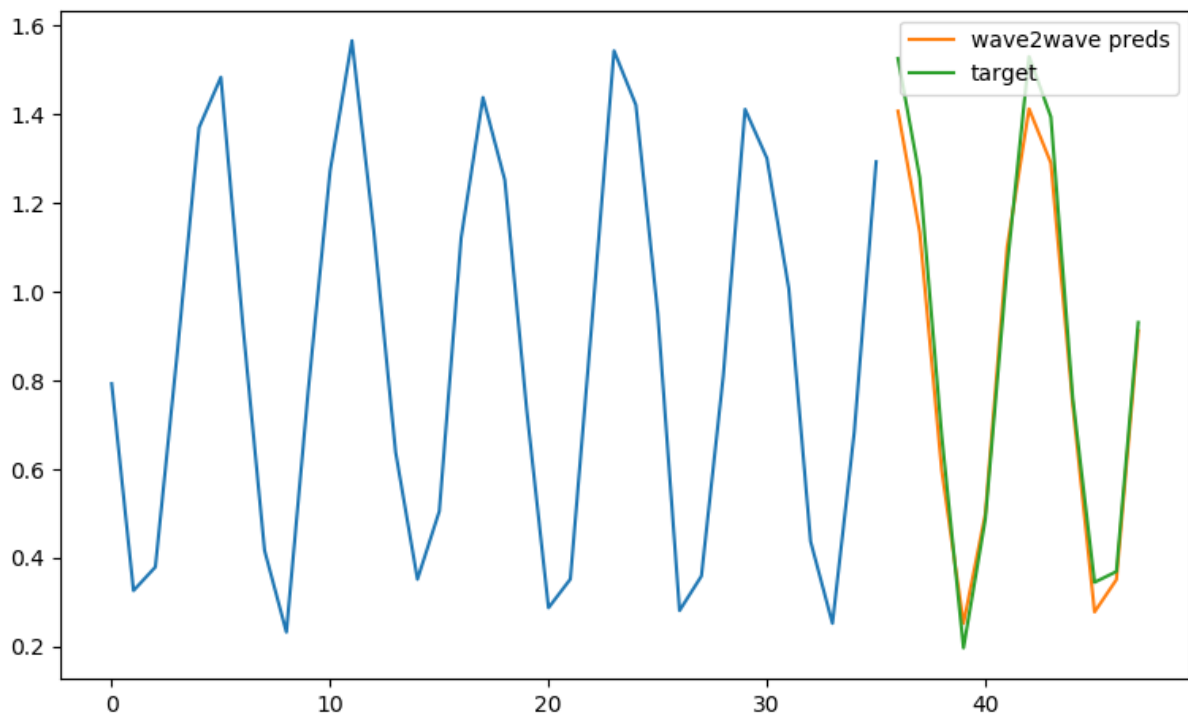
Quick Start

```
1 from deepseries.models import Wave2Wave, RNN2RNN
2 from deepseries.train import Learner
3 from deepseries.data import Value, create_seq2seq_data_loader,
  forward_split
4 from deepseries.nn import RMSE, MSE
5 import deepseries.functional as F
6 import numpy as np
7 import torch
8
9 batch_size = 16
10 enc_len = 36
11 dec_len = 12
12 series_len = 1000
13
14 epoch = 100
15 lr = 0.001
16
17 valid_size = 12
18 test_size = 12
19
20 series = np.sin(np.arange(0, series_len)) + np.random.normal(0, 0.1,
  series_len) + np.log2(np.arange(1, series_len+1))
21 series = series.reshape(1, 1, -1)
22
23 train_idx, valid_idx = forward_split(np.arange(series_len), enc_len=
  enc_len, valid_size=valid_size+test_size)
24 valid_idx, test_idx = forward_split(valid_idx, enc_len, test_size)
25
26 # mask test, will not be used for calculating mean/std.
27 mask = np.zeros_like(series).astype(bool)
28 mask[:, :, test_idx] = False
29 series, mu, std = F.normalize(series, axis=2, fillna=True, mask=mask)
30
31 # create train/valid dataset
```

```

32 train_dl = create_seq2seq_data_loader(series[:, :, train_idx], enc_len,
33                                     dec_len, sampling_rate=0.1,
34                                     batch_size=batch_size, seq_last=
35                                     True, device='cuda')
36
37 valid_dl = create_seq2seq_data_loader(series[:, :, valid_idx], enc_len,
38                                     dec_len,
39                                     batch_size=batch_size, seq_last=
40                                     True, device='cuda')
41
42 # define model
43 wave = Wave2Wave(target_size=1, num_layers=6, num_blocks=1, dropout
44                 =0.1, loss_fn=RMSE())
45 wave.cuda()
46 opt = torch.optim.Adam(wave.parameters(), lr=lr)
47
48 # train model
49 wave_learner = Learner(wave, opt, root_dir="./wave", )
50 wave_learner.fit(max_epochs=epoch, train_dl=train_dl, valid_dl=valid_dl
51                 , early_stopping=True, patient=16)
52
53 # load best model
54 wave_learner.load(wave_learner.best_epoch)
55
56 # predict and show result
57 import matplotlib.pyplot as plt
58 wave_preds = wave_learner.model.predict(torch.tensor(series[:, :,
59 test_idx[: -12]]).float().cuda(), 12).cpu().numpy().reshape(-1)
60
61 plt.plot(series[:, :, -48:-12].reshape(-1))
62 plt.plot(np.arange(36, 48), wave_preds, label="wave2wave preds")
63 plt.plot(np.arange(36, 48), series[:, :, test_idx[-12:]].reshape(-1),
64         label="target")
65 plt.legend()

```



More examples will be update in example folder soon.

Performance

I will test model performance in Kaggle or other data science competition. It will coming soon.

Install

```
1 git clone https://github.com/EvilPsyCho/Deep-Time-Series-Prediction.git
2 cd Deep-Time-Series-Prediction
3 python setup.py install
```

Refs

- WaveNet Keras Totutorial: TimeSeries_Seq2Seq
- WaveNet Kaggle Web Traffic Forecasting Competition RANK 6
- Seq2Seq Kaggle Web Traffic Forecasting Competition RANK 1
- Kaggle: Corporación Favorita Grocery Sales Forecasting Top1 LSTM/LGBM
- Kaggle: Corporación Favorita Grocery Sales Forecasting Top5 LGBM/CNN/Seq2Seq

-
- Temporal Pattern Attention for Multivariate Time Series Forecasting, 2018
 - BahdanauAttention: NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE
 - Effective Approaches to Attention-based Neural Machine Translation
 - BahdanauAttention and LuongAttention