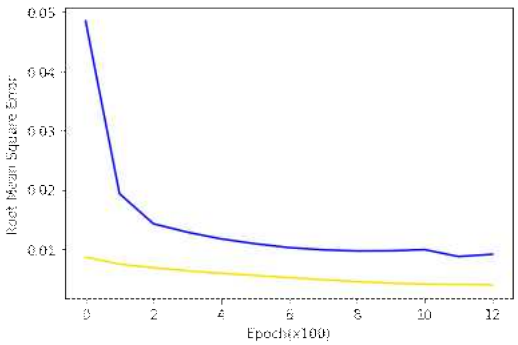
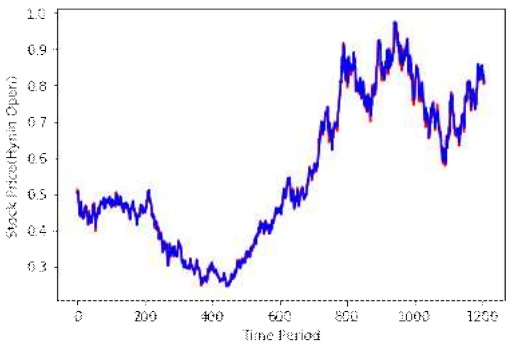
```
---------------------------------------------------------------
DIY 연구소 인공지능 주가 예측 프로그램 (http://cafe.daum.net/diylab)
하이닉스 주식 시가 예측
---------------------------------------------------------------

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4034 entries, 0 to 4033
Data columns (total 7 columns):
Date            4034 non-null object
Open            4034 non-null object
High            4034 non-null object
Low             4034 non-null object
Close           4034 non-null object
Adj Close       4034 non-null object
Volume          4034 non-null object
dtypes: object(7)
memory usage: 220.7+ KB
stock_info.shape:  (4033, 6)
stock_info[0]:  [4.830000e+03 5.040000e+03 4.830000e+03 4.935000e+03 4.935000e+03
 5.104167e+06]
price.shape:  (4033, 5)
price[0]:  [4830. 5040. 4830. 4935. 4935.]
norm_price[0]:  [0.02319222 0.025401   0.02319222 0.02429661 0.02429661]
================================================================================================================
volume.shape:  (4033, 1)
volume[0]:  [5104167.]
norm_volume[0]:  [0.05778133]
================================================================================================================
x.shape:  (4033, 6)
x[0]:  [0.02319222 0.025401   0.02319222 0.02429661 0.02429661 0.05778133]
x[-1]:  [0.81172758 0.81909019 0.80752038 0.81383119 0.81383119 0.02145996]
================================================================================================================
y[0]:  [0.02319222]
y[-1]:  [0.81172758]
[[0.02319222 0.025401   0.02319222 0.02429661 0.02429661 0.05778133]
 [0.02429661 0.025401   0.02319222 0.02319222 0.02319222 0.06090379]
 [0.02208783 0.02429661 0.02098343 0.02319222 0.02319222 0.05946239]
 [0.02319222 0.025401   0.02098343 0.02208783 0.02208783 0.0776658 ]
 [0.02208783 0.02319222 0.02098343 0.02208783 0.02208783 0.03834356]
 [0.02208783 0.02319222 0.02098343 0.02098343 0.02098343 0.03612459]
 [0.02098343 0.02429661 0.02098343 0.02208783 0.02208783 0.09487725]
 [0.02319222 0.02429661 0.02098343 0.02098343 0.02098343 0.08341966]
 [0.02098343 0.02208783 0.01656587 0.01877465 0.01877465 0.12140606]
 [0.01767026 0.01877465 0.01656587 0.01767026 0.01767026 0.05744672]
 [0.01877465 0.01987904 0.01656587 0.01656587 0.01656587 0.06295879]
 [0.01656587 0.01877465 0.01656587 0.01767026 0.01767026 0.05994642]] -> [0.01767026]
X:  Tensor("Placeholder:0", shape=(?, 12, 6), dtype=float32)
Y:  Tensor("Placeholder_1:0", shape=(?, 1), dtype=float32)
targets:  Tensor("Placeholder_2:0", shape=(?, 1), dtype=float32)
predictions:  Tensor("Placeholder_3:0", shape=(?, 1), dtype=float32)

WARNING: The TensorFlow contrib module will not be included in TensorFlow 2.0.
For more information, please see:
  * https://github.com/tensorflow/community/blob/master/rfcs/20180907-contrib-sunset.md
  * https://github.com/tensorflow/addons
If you depend on functionality not listed there, please file an issue.

WARNING:tensorflow:From <ipython-input-1-e79a577e46fa>:166: BasicLSTMCell.__init__ (from tensorflow.python.ops.rnn_cell_impl) is deprecated and will be removed in a future version.
Instructions for updating:
This class is equivalent as tf.keras.layers.LSTMCell, and will be replaced by that in Tensorflow 2.0.
WARNING:tensorflow:From <ipython-input-1-e79a577e46fa>:176: dynamic_rnn (from tensorflow.python.ops.rnn) is deprecated and will be removed in a future version.
Instructions for updating:
Please use `keras.layers.RNN(cell)`, which is equivalent to this API
WARNING:tensorflow:From C:₩Users₩LG₩Anaconda3₩lib₩site-packages₩tensorflow₩python₩ops₩tensor_array_ops.py:162: colocate_with (from tensorflow.python.framework.ops) is deprecated and will be removed in a future version.
Instructions for updating:
```

Colocations handled automatically by placer.
hypothesis: Tensor("rnn/transpose_1:0", shape=(?, 12, 20), dtype=float32)

학습을 시작합니다...
epoch: 100, train_error(A): 0.008621325716376305, test_error(B): 0.04846591129899025, B-A: 0.039844587445259094
epoch: 200, train_error(A): 0.007437780033797026, test_error(B): 0.019322605803608894, B-A: 0.011884825304150581
epoch: 300, train_error(A): 0.006831997074186802, test_error(B): 0.014255953952670097, B-A: 0.0074239568784832954
epoch: 400, train_error(A): 0.006300890818238258, test_error(B): 0.012833278626203537, B-A: 0.006532387807965279
epoch: 500, train_error(A): 0.005896895658224821, test_error(B): 0.011700957082211971, B-A: 0.00580406142398715
epoch: 600, train_error(A): 0.005536592099815607, test_error(B): 0.010884909890592098, B-A: 0.005348317790776491
epoch: 700, train_error(A): 0.005164080299437046, test_error(B): 0.010232764296233654, B-A: 0.005068683996796608
epoch: 800, train_error(A): 0.0048005664721131325, test_error(B): 0.009854346513748169, B-A: 0.005053780041635036 5
epoch: 900, train_error(A): 0.00448481272906065, test_error(B): 0.009663633070886135, B-A: 0.005178820341825485
epoch: 1000, train_error(A): 0.004238871857523918, test_error(B): 0.009693573229014874, B-A: 0.005454701371490955
epoch: 1100, train_error(A): 0.0040641482919454575, test_error(B): 0.009892158210277557, B-A: 0.0058280099183321
epoch: 1200, train_error(A): 0.004019417334347963, test_error(B): 0.008720752783119678, B-A: 0.004701335448771715
epoch: 1250, train_error(A): 0.003951387945562601, test_error(B): 0.00910670030862093, B-A: 0.005155312363058329
elapsed_time: 0:01:00.281589
elapsed_time per epoch: 0:00:00.048225
input_data_column_cnt: 6,output_data_column_cnt: 1,seq_length: 12,rnn_cell_hidden_dim: 20,forget_bias: 1.0,num_stacked_layers: 1,keep_prob: 1.0,epoch_num: 1250,learning_rate: 0.01,train_error: 0.003951388,test_error: 0.0091067,min_test_error: 0.008720753





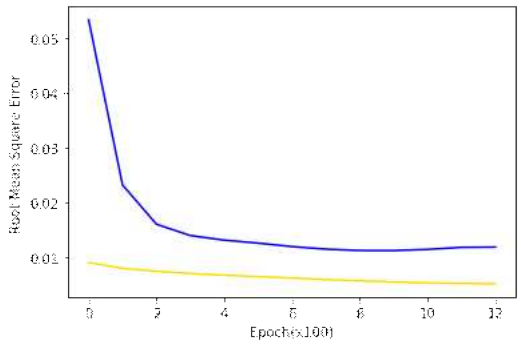
recent_data.shape: (1, 12, 6)
recent_data: [[[0.83381541 0.84854063 0.83381541 0.84328162 0.84328162 0.01564273]
[0.84012622 0.85064423 0.8285564 0.84959243 0.84959243 0.01848774]
[0.83697081 0.84328162 0.8296082 0.83381541 0.83381541 0.0182828]
[0.84538522 0.85064423 0.84012622 0.84854063 0.84854063 0.01931112]
[0.81593479 0.8296082 0.81488299 0.8285564 0.8285564 0.02948099]
[0.82014199 0.84433342 0.81803839 0.83697081 0.83697081 0.02738985]
[0.83276361 0.84012622 0.83171181 0.83486721 0.83486721 0.02656469]
[0.83066001 0.83066001 0.80857218 0.80857218 0.80857218 0.0244853]
[0.82119379 0.83171181 0.81172758 0.81698659 0.81698659 0.01353542]
[0.81803839 0.825401 0.80752038 0.8232974 0.8232974 0.0202467]
[0.80226137 0.80962398 0.79384696 0.80331317 0.80331317 0.04961917]
[0.81172758 0.81909019 0.80752038 0.81383119 0.81383119 0.02145996]]]

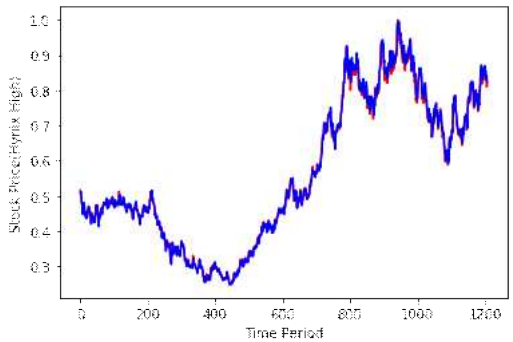
하이닉스 주식 10/14 내일 시가(정규화) test_predict [0.8240604]
하이닉스 주식 10/14 내일 시가(KRW) Tomorrow's stock price [80972.54]

```
-----------------------------------------------------------------
DIY 연구소 인공지능 주가 예측 프로그램 (http://cafe.daum.net/diylab)
하이닉스 주식 고가 예측
-----------------------------------------------------------------

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4034 entries, 0 to 4033
Data columns (total 7 columns):
Date            4034 non-null object
Open            4034 non-null object
High            4034 non-null object
Low             4034 non-null object
Close           4034 non-null object
Adj Close       4034 non-null object
Volume          4034 non-null object
dtypes: object(7)
memory usage: 220.7+ KB
stock_info.shape:  (4033, 6)
stock_info[0]:  [4.830000e+03 5.040000e+03 4.830000e+03 4.935000e+03 4.935000e+03
 5.104167e+06]
price.shape:  (4033, 5)
price[0]:  [4830. 5040. 4830. 4935. 4935.]
norm_price[0]:  [0.02319222 0.025401   0.02319222 0.02429661 0.02429661]
=============================================================================================================
volume.shape:  (4033, 1)
volume[0]:  [5104167.]
norm_volume[0]:  [0.05778133]
=============================================================================================================
x.shape:  (4033, 6)
x[0]:  [0.02319222 0.025401   0.02319222 0.02429661 0.02429661 0.05778133]
x[-1]:  [0.81172758 0.81909019 0.80752038 0.81383119 0.81383119 0.02145996]
=============================================================================================================
y[0]:  [0.025401]
y[-1]:  [0.81909019]
[[0.02319222 0.025401   0.02319222 0.02429661 0.02429661 0.05778133]
 [0.02429661 0.025401   0.02319222 0.02319222 0.02319222 0.06090379]
 [0.02208783 0.02429661 0.02098343 0.02319222 0.02319222 0.05946239]
 [0.02319222 0.025401   0.02098343 0.02208783 0.02208783 0.0776658 ]
 [0.02208783 0.02319222 0.02098343 0.02208783 0.02208783 0.03834356]
 [0.02208783 0.02319222 0.02098343 0.02098343 0.02098343 0.03612459]
 [0.02098343 0.02429661 0.02098343 0.02208783 0.02208783 0.09487725]
 [0.02319222 0.02429661 0.02098343 0.02098343 0.02098343 0.08341966]
 [0.02098343 0.02208783 0.01656587 0.01877465 0.01877465 0.12140606]
 [0.01767026 0.01877465 0.01656587 0.01767026 0.01767026 0.05744672]
 [0.01877465 0.01987904 0.01656587 0.01656587 0.01656587 0.06295879]
 [0.01656587 0.01877465 0.01656587 0.01767026 0.01767026 0.05994642]] -> [0.01767026]
X:  Tensor("Placeholder:0", shape=(?, 12, 6), dtype=float32)
Y:  Tensor("Placeholder_1:0", shape=(?, 1), dtype=float32)
targets:  Tensor("Placeholder_2:0", shape=(?, 1), dtype=float32)
predictions:  Tensor("Placeholder_3:0", shape=(?, 1), dtype=float32)

WARNING: The TensorFlow contrib module will not be included in TensorFlow 2.0.
For more information, please see:
  * https://github.com/tensorflow/community/blob/master/rfcs/20180907-contrib-sunset.md
  * https://github.com/tensorflow/addons
If you depend on functionality not listed there, please file an issue.

WARNING:tensorflow:From <ipython-input-1-413a80cbf260>:166: BasicLSTMCell.__init__ (from tensorflow.python.ops.rnn_cell_impl) is deprecated and will be removed in a future version.
Instructions for updating:
This class is equivalent as tf.keras.layers.LSTMCell, and will be replaced by that in Tensorflow 2.0.
WARNING:tensorflow:From <ipython-input-1-413a80cbf260>:176: dynamic_rnn (from tensorflow.python.ops.rnn) is deprecated and will be removed in a future version.
Instructions for updating:
Please use `keras.layers.RNN(cell)`, which is equivalent to this API
WARNING:tensorflow:From C:₩Users₩LG₩Anaconda3₩lib₩site-packages₩tensorflow₩python₩ops₩tensor_array_ops.py:162: colocate_with (from tensorflow.python.framework.ops) is deprecated and will be removed in a future version.
Instructions for updating:
```

Colocations handled automatically by placer.
hypothesis: Tensor("rnn/transpose_1:0", shape=(?, 12, 20), dtype=float32)

학습을 시작합니다...
epoch: 100, train_error(A): 0.009077072143554688, test_error(B): 0.0533062107861042, B-A: 0.044229138642549515
epoch: 200, train_error(A): 0.007992492988705635, test_error(B): 0.023212706670165062, B-A: 0.015220213681459427
epoch: 300, train_error(A): 0.007477670442312956, test_error(B): 0.016041185706853867, B-A: 0.008563514798879623
epoch: 400, train_error(A): 0.007051472086459398, test_error(B): 0.013974539004266262, B-A: 0.006923066917806864
epoch: 500, train_error(A): 0.006743892561644316, test_error(B): 0.013131145387887955, B-A: 0.006387252826243639
epoch: 600, train_error(A): 0.006487407721579075, test_error(B): 0.012587507255375385, B-A: 0.00610009953379631
epoch: 700, train_error(A): 0.006226883269846439, test_error(B): 0.01196444034576416, B-A: 0.005737557075917721
epoch: 800, train_error(A): 0.0059634381905198 1, test_error(B): 0.011493144556879997, B-A: 0.0055297063663601875
epoch: 900, train_error(A): 0.005717722699046135, test_error(B): 0.011252637952566147, B-A: 0.005534915253520012
epoch: 1000, train_error(A): 0.005506532266736031, test_error(B): 0.01123667974025011, B-A: 0.00573014747351408
epoch: 1100, train_error(A): 0.0053389593958854675, test_error(B): 0.011451669968664646, B-A: 0.006112710572779179
epoch: 1200, train_error(A): 0.005217591766268015, test_error(B): 0.01182760763913393, B-A: 0.006610015872865915
epoch: 1250, train_error(A): 0.005186028312891722, test_error(B): 0.01186993345618248, B-A: 0.006683905143290758
elapsed_time: 0:01:00.184522
elapsed_time per epoch: 0:00:00.048148
input_data_column_cnt: 6,output_data_column_cnt: 1,seq_length: 12,rnn_cell_hidden_dim: 20,forget_bias: 1.0,num_stacked_layers: 1,keep_prob: 1.0,epoch_num: 1250,learning_rate: 0.01,train_error: 0.0051860283,test_error: 0.011869933,min_test_error: 0.01123668





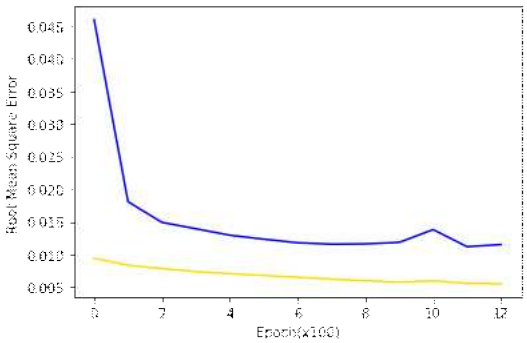
recent_data.shape: (1, 12, 6)
recent_data: [[[0.83381541 0.84854063 0.83381541 0.84328162 0.84328162 0.01564273]
[0.84012622 0.85064423 0.8285564 0.84959243 0.84959243 0.01848774]
[0.83697081 0.84328162 0.8296082 0.83381541 0.83381541 0.0182828]
[0.84538522 0.85064423 0.84012622 0.84854063 0.84854063 0.01931112]
[0.81593479 0.8296082 0.81488299 0.8285564 0.8285564 0.02948099]
[0.82014199 0.84433342 0.81803839 0.83697081 0.83697081 0.02738985]
[0.83276361 0.84012622 0.83171181 0.83486721 0.83486721 0.02656469]
[0.83066001 0.83066001 0.80857218 0.80857218 0.80857218 0.0244853]
[0.82119379 0.83171181 0.81172758 0.81698659 0.81698659 0.01353542]
[0.81803839 0.825401 0.80752038 0.8232974 0.8232974 0.0202467]
[0.80226137 0.80962398 0.79384696 0.80331317 0.80331317 0.04961917]
[0.81172758 0.81909019 0.80752038 0.81383119 0.81383119 0.02145996]]]

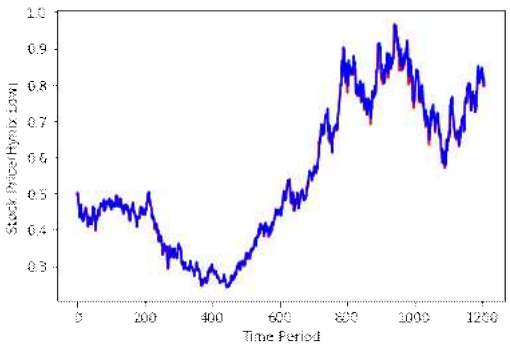
하이닉스 주식 10/14 내일 고가(정규화) test_predict [0.83716446]
하이닉스 주식 10/14 내일 고가(KRW) Tomorrow's stock price [82218.414]

```
---------------------------------------------------------------
DIY 연구소 인공지능 주가 예측 프로그램 (http://cafe.daum.net/diylab)
하이닉스 주식 저가 예측
---------------------------------------------------------------

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4034 entries, 0 to 4033
Data columns (total 7 columns):
Date            4034 non-null object
Open            4034 non-null object
High            4034 non-null object
Low             4034 non-null object
Close           4034 non-null object
Adj Close       4034 non-null object
Volume          4034 non-null object
dtypes: object(7)
memory usage: 220.7+ KB
stock_info.shape:  (4033, 6)
stock_info[0]:  [4.830000e+03 5.040000e+03 4.830000e+03 4.935000e+03 4.935000e+03
 5.104167e+06]
price.shape:  (4033, 5)
price[0]:  [4830. 5040. 4830. 4935. 4935.]
norm_price[0]:  [0.02319222 0.025401   0.02319222 0.02429661 0.02429661]
=============================================================================================================
volume.shape:  (4033, 1)
volume[0]:  [5104167.]
norm_volume[0]:  [0.05778133]
=============================================================================================================
x.shape:  (4033, 6)
x[0]:  [0.02319222 0.025401   0.02319222 0.02429661 0.02429661 0.05778133]
x[-1]:  [0.81172758 0.81909019 0.80752038 0.81383119 0.81383119 0.02145996]
=============================================================================================================
y[0]:  [0.02319222]
y[-1]:  [0.80752038]
[[0.02319222 0.025401   0.02319222 0.02429661 0.02429661 0.05778133]
 [0.02429661 0.025401   0.02319222 0.02319222 0.02319222 0.06090379]
 [0.02208783 0.02429661 0.02098343 0.02319222 0.02319222 0.05946239]
 [0.02319222 0.025401   0.02098343 0.02208783 0.02208783 0.0776658 ]
 [0.02208783 0.02319222 0.02098343 0.02208783 0.02208783 0.03834356]
 [0.02208783 0.02319222 0.02098343 0.02098343 0.02098343 0.03612459]
 [0.02098343 0.02429661 0.02098343 0.02208783 0.02208783 0.09487725]
 [0.02319222 0.02429661 0.02098343 0.02098343 0.02098343 0.08341966]
 [0.02098343 0.02208783 0.01656587 0.01877465 0.01877465 0.12140606]
 [0.01767026 0.01877465 0.01656587 0.01767026 0.01767026 0.05744672]
 [0.01877465 0.01987904 0.01656587 0.01656587 0.01656587 0.06295879]
 [0.01656587 0.01877465 0.01656587 0.01767026 0.01767026 0.05994642]] -> [0.01546148]
X:  Tensor("Placeholder:0", shape=(?, 12, 6), dtype=float32)
Y:  Tensor("Placeholder_1:0", shape=(?, 1), dtype=float32)
targets:  Tensor("Placeholder_2:0", shape=(?, 1), dtype=float32)
predictions:  Tensor("Placeholder_3:0", shape=(?, 1), dtype=float32)

WARNING: The TensorFlow contrib module will not be included in TensorFlow 2.0.
For more information, please see:
  * https://github.com/tensorflow/community/blob/master/rfcs/20180907-contrib-sunset.md
  * https://github.com/tensorflow/addons
If you depend on functionality not listed there, please file an issue.

WARNING:tensorflow:From <ipython-input-1-e2f5fa20a10e>:166: BasicLSTMCell.__init__ (from tensorflow.python.ops.rnn_cell_impl) is deprecated and will be removed in a future version.
Instructions for updating:
This class is equivalent as tf.keras.layers.LSTMCell, and will be replaced by that in Tensorflow 2.0.
WARNING:tensorflow:From <ipython-input-1-e2f5fa20a10e>:176: dynamic_rnn (from tensorflow.python.ops.rnn) is deprecated and will be removed in a future version.
Instructions for updating:
Please use `keras.layers.RNN(cell)`, which is equivalent to this API
WARNING:tensorflow:From C:₩Users₩LG₩Anaconda3₩lib₩site-packages₩tensorflow₩python₩ops₩tensor_array_ops.py:162: colocate_with (from tensorflow.python.framework.ops) is deprecated and will be removed in a future version.
Instructions for updating:
```

Colocations handled automatically by placer.
hypothesis: Tensor("rnn/transpose_1:0", shape=(?, 12, 20), dtype=float32)

학습을 시작합니다...
epoch: 100, train_error(A): 0.009407910518348217, test_error(B): 0.045947711914777756, B-A: 0.036539800465106964
epoch: 200, train_error(A): 0.008351922035217285, test_error(B): 0.01808021403849125, B-A: 0.009728292003273964
epoch: 300, train_error(A): 0.007826634682714939, test_error(B): 0.01491173729300499, B-A: 0.007085102610290505
epoch: 400, train_error(A): 0.007375083398073912, test_error(B): 0.013932507485151291, B-A: 0.006557424087077379
epoch: 500, train_error(A): 0.007052749861031771, test_error(B): 0.012949403375387192, B-A: 0.005896653514355421
epoch: 600, train_error(A): 0.006780439987778664, test_error(B): 0.01234149094671011, B-A: 0.005561050958931446
epoch: 700, train_error(A): 0.006501661613583565, test_error(B): 0.01178803201764822, B-A: 0.005286370404064655
epoch: 800, train_error(A): 0.006222109775990248, test_error(B): 0.011571614071726799, B-A: 0.005349504295736551
epoch: 900, train_error(A): 0.005966630298644304, test_error(B): 0.011618087999522686, B-A: 0.005651457700878382
epoch: 1000, train_error(A): 0.005754465702921152, test_error(B): 0.011848988011479378, B-A: 0.006094522308558226
epoch: 1100, train_error(A): 0.00595815246924758, test_error(B): 0.013802852481603622, B-A: 0.007844699546694756
epoch: 1200, train_error(A): 0.005556986667215824, test_error(B): 0.011184143833816051, B-A: 0.005627157166600227
epoch: 1250, train_error(A): 0.005509804468601942, test_error(B): 0.011519239284098148, B-A: 0.006009434815496206
elapsed_time: 0:00:58.867577
elapsed_time per epoch: 0:00:00.047094
input_data_column_cnt: 6,output_data_column_cnt: 1,seq_length: 12,rnn_cell_hidden_dim: 20,forget_bias: 1.0,num_stacked_layers: 1,keep_prob: 1.0,epoch_num: 1250,learning_rate: 0.01,train_error: 0.0055098045,test_error: 0.011519239,min_test_error: 0.011184144





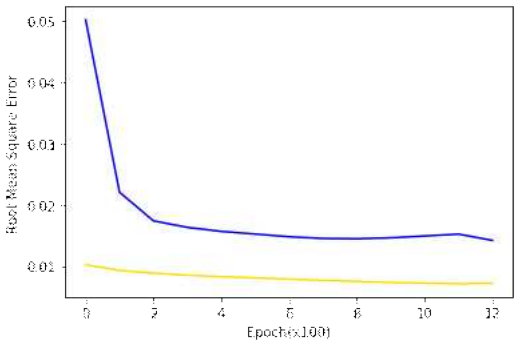
recent_data.shape: (1, 12, 6)
recent_data: [[[0.83381541 0.84854063 0.83381541 0.84328162 0.84328162 0.01564273]
[0.84012622 0.85064423 0.8285564 0.84959243 0.84959243 0.01848774]
[0.83697081 0.84328162 0.8296082 0.83381541 0.83381541 0.0182828]
[0.84538522 0.85064423 0.84012622 0.84854063 0.84854063 0.01931112]
[0.81593479 0.8296082 0.81488299 0.8285564 0.8285564 0.02948099]
[0.82014199 0.84433342 0.81803839 0.83697081 0.83697081 0.02738985]
[0.83276361 0.84012622 0.83171181 0.83486721 0.83486721 0.02656469]
[0.83066001 0.83066001 0.80857218 0.80857218 0.80857218 0.0244853]
[0.82119379 0.83171181 0.81172758 0.81698659 0.81698659 0.01353542]
[0.81803839 0.825401 0.80752038 0.8232974 0.8232974 0.0202467]
[0.80226137 0.80962398 0.79384696 0.80331317 0.80331317 0.04961917]
[0.81172758 0.81909019 0.80752038 0.81383119 0.81383119 0.02145996]]]

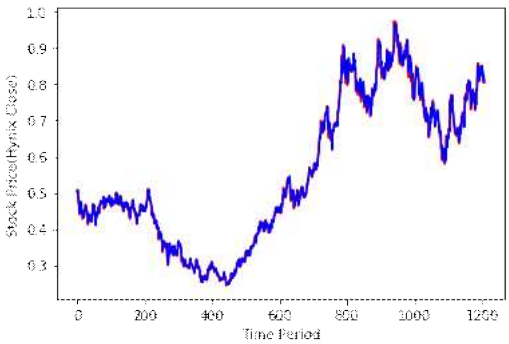
하이닉스 주식 10/14 내일 저가(정규화) test_predict [0.8154211]
하이닉스 주식 10/14 내일 저가(KRW) Tomorrow's stock price [80151.164]

```
-------------------------------------------------------------
DIY 연구소 인공지능 주가 예측 프로그램 (http://cafe.daum.net/diylab)
하이닉스 주식 종가 예측
-------------------------------------------------------------

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4034 entries, 0 to 4033
Data columns (total 7 columns):
Date            4034 non-null object
Open            4034 non-null object
High            4034 non-null object
Low             4034 non-null object
Close           4034 non-null object
Adj Close       4034 non-null object
Volume          4034 non-null object
dtypes: object(7)
memory usage: 220.7+ KB
stock_info.shape:  (4033, 6)
stock_info[0]:  [4.830000e+03 5.040000e+03 4.830000e+03 4.935000e+03 4.935000e+03
 5.104167e+06]
price.shape:  (4033, 5)
price[0]:  [4830. 5040. 4830. 4935. 4935.]
norm_price[0]:  [0.02319222 0.025401   0.02319222 0.02429661 0.02429661]
=============================================================================================================
volume.shape:  (4033, 1)
volume[0]:  [5104167.]
norm_volume[0]:  [0.05778133]
=============================================================================================================
x.shape:  (4033, 6)
x[0]:  [0.02319222 0.025401   0.02319222 0.02429661 0.02429661 0.05778133]
x[-1]:  [0.81172758 0.81909019 0.80752038 0.81383119 0.81383119 0.02145996]
=============================================================================================================
y[0]:  [0.02429661]
y[-1]:  [0.81383119]
[[0.02319222 0.025401   0.02319222 0.02429661 0.02429661 0.05778133]
 [0.02429661 0.025401   0.02319222 0.02319222 0.02319222 0.06090379]
 [0.02208783 0.02429661 0.02098343 0.02319222 0.02319222 0.05946239]
 [0.02319222 0.025401   0.02098343 0.02208783 0.02208783 0.0776658 ]
 [0.02208783 0.02319222 0.02098343 0.02208783 0.02208783 0.03834356]
 [0.02208783 0.02319222 0.02098343 0.02098343 0.02098343 0.03612459]
 [0.02098343 0.02429661 0.02098343 0.02208783 0.02208783 0.09487725]
 [0.02319222 0.02429661 0.02098343 0.02098343 0.02098343 0.08341966]
 [0.02098343 0.02208783 0.01656587 0.01877465 0.01877465 0.12140606]
 [0.01767026 0.01877465 0.01656587 0.01767026 0.01767026 0.05744672]
 [0.01877465 0.01987904 0.01656587 0.01656587 0.01656587 0.06295879]
 [0.01656587 0.01877465 0.01656587 0.01767026 0.01767026 0.05994642]] -> [0.01546148]
X:  Tensor("Placeholder:0", shape=(?, 12, 6), dtype=float32)
Y:  Tensor("Placeholder_1:0", shape=(?, 1), dtype=float32)
targets:  Tensor("Placeholder_2:0", shape=(?, 1), dtype=float32)
predictions:  Tensor("Placeholder_3:0", shape=(?, 1), dtype=float32)

WARNING: The TensorFlow contrib module will not be included in TensorFlow 2.0.
For more information, please see:
  * https://github.com/tensorflow/community/blob/master/rfcs/20180907-contrib-sunset.md
  * https://github.com/tensorflow/addons
If you depend on functionality not listed there, please file an issue.

WARNING:tensorflow:From <ipython-input-1-c655935b88ae>:166: BasicLSTMCell.__init__ (from tensorflow.python.ops.rnn_cell_impl) is deprecated and will be removed in a future version.
Instructions for updating:
This class is equivalent as tf.keras.layers.LSTMCell, and will be replaced by that in Tensorflow 2.0.
WARNING:tensorflow:From <ipython-input-1-c655935b88ae>:176: dynamic_rnn (from tensorflow.python.ops.rnn) is deprecated and will be removed in a future version.
Instructions for updating:
Please use `keras.layers.RNN(cell)`, which is equivalent to this API
WARNING:tensorflow:From C:₩Users₩LG₩Anaconda3₩lib₩site-packages₩tensorflow₩python₩ops₩tensor_array_ops.py:162: colocate_with (from tensorflow.python.framework.ops) is deprecated and will be removed in a future version.
Instructions for updating:
```

Colocations handled automatically by placer.
hypothesis: Tensor("rnn/transpose_1:0", shape=(?, 12, 20), dtype=float32)

학습을 시작합니다...
epoch: 100, train_error(A): 0.010200385004281998, test_error(B): 0.05023644119501114, B-A: 0.04003605619072914
epoch: 200, train_error(A): 0.009261084720492363, test_error(B): 0.02204902097582817, B-A: 0.012787936255335808
epoch: 300, train_error(A): 0.008833080530166626, test_error(B): 0.017370907589793205, B-A: 0.00853782705962658
epoch: 400, train_error(A): 0.008483902551233768, test_error(B): 0.016299553215503693, B-A: 0.007815650664269924
epoch: 500, train_error(A): 0.00824100710451603, test_error(B): 0.015647832304239273, B-A: 0.007406825199723244
epoch: 600, train_error(A): 0.00804463867098093, test_error(B): 0.015210730955004692, B-A: 0.007166092284023762
epoch: 700, train_error(A): 0.007847809232771397, test_error(B): 0.014770220965147018, B-A: 0.006922411732375622
epoch: 800, train_error(A): 0.007649016565745735, test_error(B): 0.014494887553155422, B-A: 0.006845870986580849
epoch: 900, train_error(A): 0.00746251130476594, test_error(B): 0.014440013095736504, B-A: 0.006977501790970564
epoch: 1000, train_error(A): 0.007301111239939928, test_error(B): 0.014594899490475655, B-A: 0.007293788250535726 5
epoch: 1100, train_error(A): 0.007174298632889986, test_error(B): 0.014893933199346066, B-A: 0.007719634566456079 5
epoch: 1200, train_error(A): 0.007086732890456915, test_error(B): 0.015191539190709591, B-A: 0.008104806765913963
epoch: 1250, train_error(A): 0.007171241566538811, test_error(B): 0.014172709546983242, B-A: 0.007001467980444431
elapsed_time: 0:00:57.897907
elapsed_time per epoch: 0:00:00.046318
input_data_column_cnt: 6,output_data_column_cnt: 1,seq_length: 12,rnn_cell_hidden_dim: 20,forget_bias: 1.0,num_stacked_layers: 1,keep_prob: 1.0,epoch_num: 1250,learning_rate: 0.01,train_error: 0.0071712416,test_error: 0.01417271,min_test_error: 0.01417271





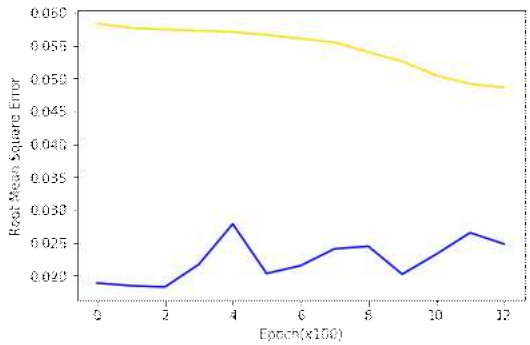
recent_data.shape: (1, 12, 6)
recent_data: [[[0.83381541 0.84854063 0.83381541 0.84328162 0.84328162 0.01564273]
[0.84012622 0.85064423 0.8285564 0.84959243 0.84959243 0.01848774]
[0.83697081 0.84328162 0.8296082 0.83381541 0.83381541 0.0182828]
[0.84538522 0.85064423 0.84012622 0.84854063 0.84854063 0.01931112]
[0.81593479 0.8296082 0.81488299 0.8285564 0.8285564 0.02948099]
[0.82014199 0.84433342 0.81803839 0.83697081 0.83697081 0.02738985]
[0.83276361 0.84012622 0.83171181 0.83486721 0.83486721 0.02656469]
[0.83066001 0.83066001 0.80857218 0.80857218 0.80857218 0.0244853]
[0.82119379 0.83171181 0.81172758 0.81698659 0.81698659 0.01353542]
[0.81803839 0.825401 0.80752038 0.8232974 0.8232974 0.0202467]
[0.80226137 0.80962398 0.79384696 0.80331317 0.80331317 0.04961917]
[0.81172758 0.81909019 0.80752038 0.81383119 0.81383119 0.02145996]]]

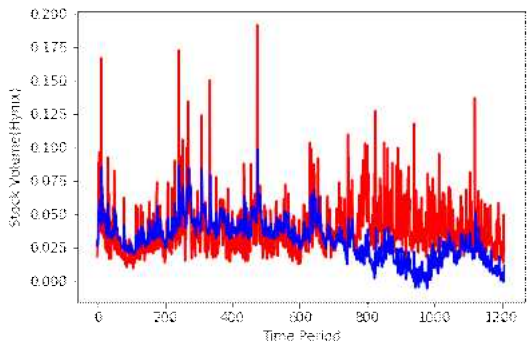
하이닉스 주식 10/14 내일 종가(정규화) test_predict [0.82134825]
하이닉스 주식 10/14 내일 종가(KRW) Tomorrow's stock price [80714.69]

```
---------------------------------------------------------------
DIY 연구소 인공지능 주가 예측 프로그램 (http://cafe.daum.net/diylab)
하이닉스 주식 거래량 예측
---------------------------------------------------------------

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4034 entries, 0 to 4033
Data columns (total 7 columns):
Date            4034 non-null object
Open            4034 non-null object
High            4034 non-null object
Low             4034 non-null object
Close           4034 non-null object
Adj Close       4034 non-null object
Volume          4034 non-null object
dtypes: object(7)
memory usage: 220.7+ KB
stock_info.shape:  (4033, 6)
stock_info[0]:  [4.830000e+03 5.040000e+03 4.830000e+03 4.935000e+03 4.935000e+03
 5.104167e+06]
price.shape:  (4033, 5)
price[0]:  [4830. 5040. 4830. 4935. 4935.]
norm_price[0]:  [0.02319222 0.025401   0.02319222 0.02429661 0.02429661]
================================================================================================================
volume.shape:  (4033, 1)
volume[0]:  [5104167.]
norm_volume[0]:  [0.05778133]
================================================================================================================
x.shape:  (4033, 6)
x[0]:  [0.02319222 0.025401   0.02319222 0.02429661 0.02429661 0.05778133]
x[-1]:  [0.81172758 0.81909019 0.80752038 0.81383119 0.81383119 0.02145996]
================================================================================================================
y[0]:  [0.05778133]
y[-1]:  [0.02145996]
[[0.02319222 0.025401   0.02319222 0.02429661 0.02429661 0.05778133]
 [0.02429661 0.025401   0.02319222 0.02319222 0.02319222 0.06090379]
 [0.02208783 0.02429661 0.02098343 0.02319222 0.02319222 0.05946239]
 [0.02319222 0.025401   0.02098343 0.02208783 0.02208783 0.0776658 ]
 [0.02208783 0.02319222 0.02098343 0.02208783 0.02208783 0.03834356]
 [0.02208783 0.02319222 0.02098343 0.02098343 0.02098343 0.03612459]
 [0.02098343 0.02429661 0.02098343 0.02208783 0.02208783 0.09487725]
 [0.02319222 0.02429661 0.02098343 0.02098343 0.02098343 0.08341966]
 [0.02098343 0.02208783 0.01656587 0.01877465 0.01877465 0.12140606]
 [0.01767026 0.01877465 0.01656587 0.01767026 0.01767026 0.05744672]
 [0.01877465 0.01987904 0.01656587 0.01656587 0.01656587 0.06295879]
 [0.01656587 0.01877465 0.01656587 0.01767026 0.01767026 0.05994642]] -> [0.0397917]
X:  Tensor("Placeholder:0", shape=(?, 12, 6), dtype=float32)
Y:  Tensor("Placeholder_1:0", shape=(?, 1), dtype=float32)
targets:  Tensor("Placeholder_2:0", shape=(?, 1), dtype=float32)
predictions:  Tensor("Placeholder_3:0", shape=(?, 1), dtype=float32)

WARNING: The TensorFlow contrib module will not be included in TensorFlow 2.0.
For more information, please see:
  * https://github.com/tensorflow/community/blob/master/rfcs/20180907-contrib-sunset.md
  * https://github.com/tensorflow/addons
If you depend on functionality not listed there, please file an issue.

WARNING:tensorflow:From <ipython-input-1-fcf3cf177dfb>:166: BasicLSTMCell.__init__ (from tensorflow.python.ops.rnn_cell_impl) is deprecated and will be removed in a future version.
Instructions for updating:
This class is equivalent as tf.keras.layers.LSTMCell, and will be replaced by that in Tensorflow 2.0.
WARNING:tensorflow:From <ipython-input-1-fcf3cf177dfb>:176: dynamic_rnn (from tensorflow.python.ops.rnn) is deprecated and will be removed in a future version.
Instructions for updating:
Please use `keras.layers.RNN(cell)`, which is equivalent to this API
WARNING:tensorflow:From C:₩Users₩LG₩Anaconda3₩lib₩site-packages₩tensorflow₩python₩ops₩tensor_array_ops.py:162: colocate_with (from tensorflow.python.framework.ops) is deprecated and will be removed in a future version.
Instructions for updating:
```

Colocations handled automatically by placer.
hypothesis: Tensor("rnn/transpose_1:0", shape=(?, 12, 20), dtype=float32)

학습을 시작합니다...
epoch: 100, train_error(A): 0.05830257758498192, test_error(B): 0.018841316923499107, B-A: -0.03946126252412796
epoch: 200, train_error(A): 0.05765319988131523, test_error(B): 0.01842484250664711, B-A: -0.03922835737466812
epoch: 300, train_error(A): 0.05738981440663338, test_error(B): 0.018241697922348976, B-A: -0.03914811462163925
epoch: 400, train_error(A): 0.05724354460835457, test_error(B): 0.02167440392076969, B-A: -0.03556913882493973
epoch: 500, train_error(A): 0.05703761801123619, test_error(B): 0.027823617681860924, B-A: -0.029214000329375267
epoch: 600, train_error(A): 0.05658892169594765, test_error(B): 0.020292000845074654, B-A: -0.036296918988227844
epoch: 700, train_error(A): 0.05601595342159271, test_error(B): 0.021475790068507195, B-A: -0.03454016149044037
epoch: 800, train_error(A): 0.055460408329963684, test_error(B): 0.024045711383223534, B-A: -0.031414695084095
epoch: 900, train_error(A): 0.05397447571158409, test_error(B): 0.024424413219094276, B-A: -0.029550062492489815
epoch: 1000, train_error(A): 0.052560072392225266, test_error(B): 0.020182093605399132, B-A: -0.03237798064947128
epoch: 1100, train_error(A): 0.050413183867931366, test_error(B): 0.023213354870676994, B-A: -0.02719982899725437
epoch: 1200, train_error(A): 0.04914474859833717, test_error(B): 0.026491452008485794, B-A: -0.02265329658985138
epoch: 1250, train_error(A): 0.04856440797448158, test_error(B): 0.02478068135678768, B-A: -0.0237837266176939
elapsed_time: 0:01:09.258933
elapsed_time per epoch: 0:00:00.055407
input_data_column_cnt: 6,output_data_column_cnt: 1,seq_length: 12,rnn_cell_hidden_dim: 20,forget_bias: 1.0,num_stacked_layers: 1,keep_prob: 1.0,epoch_num: 1250,learning_rate: 0.01,train_error: 0.048564408,test_error: 0.024780681,min_test_error: 0.018241698





recent_data.shape: (1, 12, 6)
recent_data: [[[0.83381541 0.84854063 0.83381541 0.84328162 0.84328162 0.01564273]
[0.84012622 0.85064423 0.8285564 0.84959243 0.84959243 0.01848774]
[0.83697081 0.84328162 0.8296082 0.83381541 0.83381541 0.0182828]
[0.84538522 0.85064423 0.84012622 0.84854063 0.84854063 0.01931112]
[0.81593479 0.8296082 0.81488299 0.8285564 0.8285564 0.02948099]
[0.82014199 0.84433342 0.81803839 0.83697081 0.83697081 0.02738985]
[0.83276361 0.84012622 0.83171181 0.83486721 0.83486721 0.02656469]
[0.83066001 0.83066001 0.80857218 0.80857218 0.80857218 0.0244853]
[0.82119379 0.83171181 0.81172758 0.81698659 0.81698659 0.01353542]
[0.81803839 0.825401 0.80752038 0.8232974 0.8232974 0.0202467]
[0.80226137 0.80962398 0.79384696 0.80331317 0.80331317 0.04961917]
[0.81172758 0.81909019 0.80752038 0.81383119 0.81383119 0.02145996]]]

하이닉스 주식 10/14 내일 거래량(정규화) test_predict [0.00225231]
하이닉스 주식 10/14 내일 거래량(KRW) Tomorrow's stock volume [2839.1387]