1.Abstract

The application of neural networks to finance has received much of attention because no assumption about a suitable mathematical model has to be made prior to forecasting and they are capable of extracting useful information from large sets of data, which is required to describe nonlinear input-output relations of financial time series. The report presents a model of keras LSTM where 4 layered neural network is trained with weekly, daily data and another LSTM model for comparison to predict gold index.

2. Data

Model 1 uses GLD, SPY, Bond, DXY, WTI indexes as input, Model 2 uses GLD, SPY, TLT data as input. The number of features of model 1 is 21. The number of features of model 2 is 16.

[Model 1 – GLD, SPY, Bond, DXY, WTI]
Input data: GLD, close change rate of GLD, SPY, Bond 3, 10 year, DXY, WTI (Crude Oil)
Output data: GLD (Open, High, Low, Close)

Daily data
GLD + GLD change rate + SPY + Bond 3, 10 yr + DXY + WTI from 11/18/2004 to 12/14/2019
Look back = 15 days, columns = 21 (GLD 5+ GLD change rate 1+ SPY 5 + Bond 2+ DXY 4 + WTI 4)

Weekly data
GLD + GLD change rate + SPY + Bond 3, 10 yr + DXY + WTI from 11/18/2004 to 12/14/2019
Look back = 3 weeks, columns = 21 (GLD 5+ GLD change rate 1+ SPY 5 + Bond 2+ DXY 4+ WTI 4)

Trained on 3294 samples, validated on 470 samples.

[Model 2 – GLD, SPY, TLT]
Input data: GLD, SPY, TLT
Output data: GLD, SPY, TLT

Daily data
GLD + SPY + TLT from 11/18/2004 to 12/14/2019
Look back = 15 days, columns = 15 (GLD 5 + SPY 5 + TLT 5)

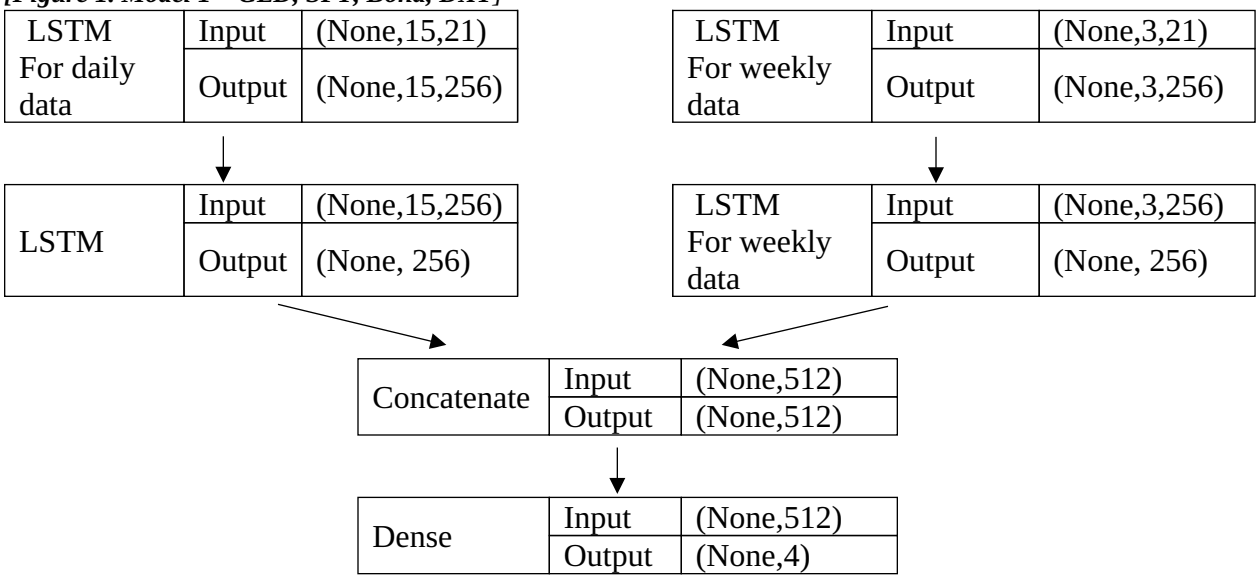
Trained on 3556 samples, validated on 180 samples.

3. Architecture

3.1 Model 1

Model1 is consisted of 2 stacked coupled LSTM being trained with daily, weekly data.
The original LSTM model is comprised of a single hidden LSTM layer followed by a standard feedforward output layer. The Stacked LSTM is an extension to this model that has multiple hidden LSTM layers where each layer contains multiple memory cells. Stacked LSTMs are now a stable technique for challenging sequence prediction problems. A Stacked LSTM architecture can be defined as an LSTM model comprised of multiple LSTM layers. An LSTM layer above provides a sequence output rather than a single value output to the LSTM layer below. Specifically, one output per input time step, rather than one output time step for all input time steps.Given that LSTMs operate on sequence data, it means that the addition of layers adds levels of abstraction of input observations over time. In effect, chunking observations over time or representing the problem at different time scales.
Building a deep RNN by stacking multiple recurrent hidden states on top of each other. This approach potentially allows the hidden state at each level to operate at different timescale. Stacked LSTMs or Deep LSTMs were introduced by Graves, et al. in their application of LSTMs to speech recognition, beating a benchmark on a challenging standard problem. RNNs are inherently deep in time, since their hidden state is a function of all previous hidden states. The question that inspired this paper was whether RNNs could also benefit from depth in space; that is from stacking multiple recurrent hidden layers on top of each other, just as feedforward layers are stacked in conventional deep networks.[1]
Loss function is mean square error which measures the average of the squares of the errors that is, the average squared difference between the estimated values and the actual value. Optimizer is Adamx which is set of learning rate for 0.001, beta_1 for 0.9, beta_2 for 0.999. L2 Regularization is applied which has value 0.0001. Scaled input data from -1 to 1 because when change rate (-1 to 1) is used as input to model, the prediction is worse than scaled change rate (0 to 1). Early stop monitors training error, it stops when 20 times increase of training error which is set of 20 as patience.

[Figure 1. Model 1 – GLD, SPY, Bond, DXY]



3.2 Model 2

Model 2 is consisted of stacked LSTM being trained with daily data.
Loss function is mean square error which measures the average of the squares of the errors that is, the average squared difference between the estimated values and the actual value.
Optimizer is Adam which is set of learning rate for 0.001, beta_1 for 0.9, beta_2 for 0.999.
L2 Regularization is applied which has value 0.0001.
Scaled input data from 0 to 1.
Early stop monitors training error, it stops when 20 times increase of training error which is set of 20 as patience.

[Figure 2. Model 2 – GLD, SPY, TLT]

LSTM	Input	(None,15,16)
	Output	(None,15,128)

↓

LSTM	Input	(None,15,128)
	Output	(None,15,128)

↓

Timedistributed (Dense)	Input	(None,15,128)
	Output	(None,15,16)

4. Analysis of Model

[Table 1. Error and Accuracy of Model 1]

	Train	Validation
Error	0.00035525	0.00018396
Accuracy	0.6721	0.9596

[Table 2. Error and Accuracy of Model 2]

	Train	Validation
Error	0.0020	0.0126
Accuracy	0.7328	0.3422

Model 1 prediction was able to be improved by using DXY and WTI index as well as weekly data as input. Applying optimizer adamx helps accuracy for model 1. The test errors of both models are bigger than train or validation error which can be regarded as over fit but it shows better prediction than low fitted model.