《Analysis of Financial Time Series》

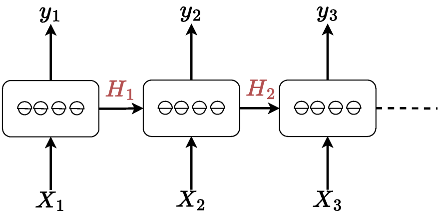
Deep learning models for Time Series Analysis

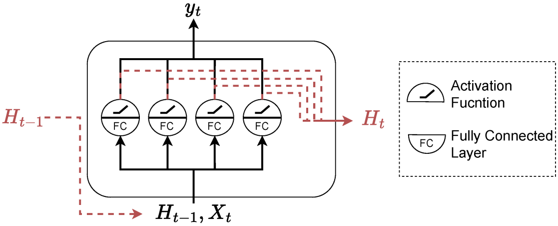
May 26, 2023

Outline

RNN

- Recurrent Neural Network (RNN) is a type of neural network designed to handle sequential data.
- RNN have connections that loop back on themselves, allowing them to maintain a sort of "memory" of past inputs.
- The figure shows the structure of RNN with hidden node size $n_h = 4$.





$$\underset{1\times n_h}{H_t} = \phi(\underset{1\times n_h}{H_{t-1}}\underset{n_h\times n_h}{W_{hh}} + \underset{1\times p}{X_t}\underset{p\times n_h}{W_{xh}} + \underset{1\times n_h}{b_h})$$

$$\underset{}{y_t} = \underset{1\times n_h}{H_t}\underset{n_h\times 1}{W_{hy}} + \underset{1\times 1}{b_y}$$

where ϕ is an activation function, and $W_{hh}, W_{xh}, b_h, W_{hy}, b_y$ are the parameters to be estimated.

Activation functions

- Logistics

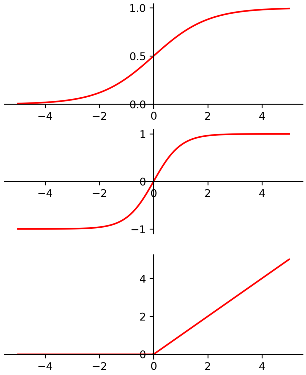
$$\phi(x) := \frac{1}{1 + \exp(-x)}$$

- Hyperbolic Tangent (tanh)

$$\phi(x) := \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}$$

- Rectified Linear Unit (ReLU)

$$\phi(x) := \max(0, x)$$



Both the logistics and hyperbolic tangent functions are examples of sigmoid functions, which have an "S"-shaped curve.

Loss function

- The parameters can be estimated by minimizing a given loss function.
- For continuous response variables, Mean Square Error (MSE) is a common loss function, defined below:

$$\begin{aligned}\mathrm{MSE}(\boldsymbol{W}) &= \frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{y}_t(\boldsymbol{W}))^2 \\ &= \frac{1}{T}\sum_{t=1}^{T} L_t(\boldsymbol{W}),\end{aligned}$$

where $\boldsymbol{W}$ is the parameters that need to estimate, y_t is the actual value and $\hat{y}_i(\boldsymbol{W})$ is the predicted value.

Parameters update

- Gradient descent is an optimization algorithm. In the presence of a non-convex function, its convergence to a local minimum is guaranteed. Its update equation is defined as follows:

$$\boldsymbol{W}^{(k+1)} = \boldsymbol{W}^{(k)} + \eta \nabla \mathsf{MSE}\left(\boldsymbol{W}^{(k)}\right),$$

  where $\nabla \mathsf{MSE}(\boldsymbol{W}) = \left(\frac{\partial \mathsf{MSE}(\boldsymbol{W})}{\partial \boldsymbol{W}_1}, \frac{\partial \mathsf{MSE}(\boldsymbol{W})}{\partial \boldsymbol{W}_2}, \ldots, \frac{\partial \mathsf{MSE}(\boldsymbol{W})}{\partial \boldsymbol{W}_p}\right)'$.
- When the sample size T is large, the computational time required to compute $\nabla \mathsf{MSE}\left(\boldsymbol{W}^{(k)}\right)$ can become significant for one iteration.

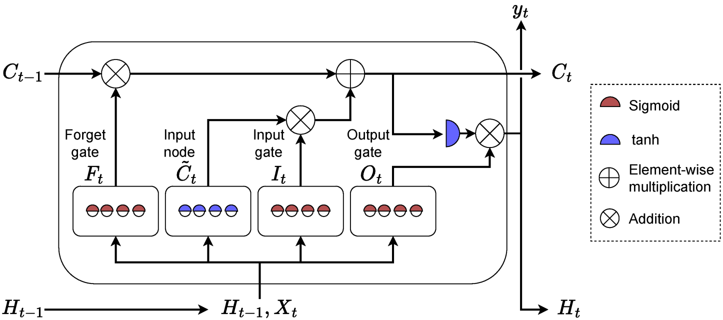
- Stochastic gradient descent (SGD) lowers the computational time by replacing the actual gradient by the gradient of a single observation.

$$\begin{aligned} \boldsymbol{W}^{(k,t)} &= \boldsymbol{W}^{(k,t-1)} + \eta \nabla \mathsf{L}_t \left(\boldsymbol{W}^{(k,t-1)} \right), \quad t = 1, \ldots T \\ \boldsymbol{W}^{(k+1,0)} &= \boldsymbol{W}^{(k,T)} \end{aligned}$$

- There are many other improved optimization algorithm based on SGD, for example, AdaGrad, RMSProp and Adam.

LSTM

- Long short-term memory (LSTM) was first proposed by Hochreiter and Schmidhuber in 1997.
- LSTM includes a module for selectively remembering or forgetting information in its memory cell.
- LSTM has been widely used in handwriting recognition, speech recognition, and translation.
- Notable achievements using LSTM include:
  1. Google researchers used LSTM for speech recognition on Google Voice and reduced transcription errors by 49% in 2015.
  2. Google researchers used LSTMs to reduce translation errors by 60% in 2016.



- The output of a sigmoid function used in LSTM needs to be within the range (0,1).
- The logistic function is a commonly used sigmoid function used in LSTM.

- LSTM has three gates: Forget gate (F_t), input gate (I_t) and output gate (O_t).

$$F_t = \sigma(H_{t-1}W_{hf} + X_tW_{xf} + b_f)$$
$$I_t = \sigma(H_{t-1}W_{hi} + X_tW_{xi} + b_i)$$
$$O_t = \sigma(H_{t-1}W_{ho} + X_tW_{xo} + b_o)$$

- F_t controls the information forgotten from C_{t-1}. I_t controls the information added to C_t from $\tilde{C}_t$ (input node).

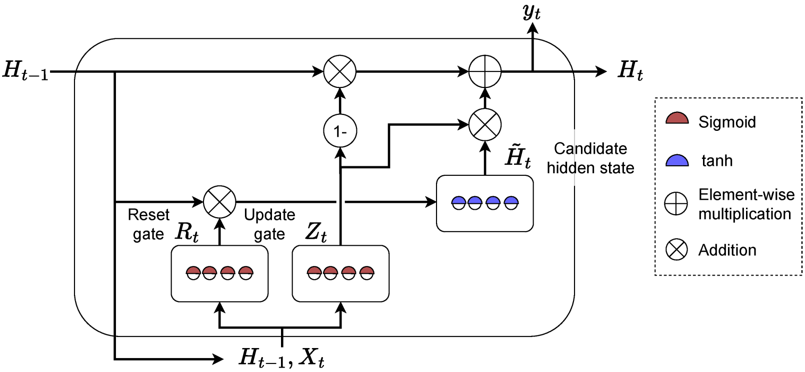
$$\tilde{C}_t = \mathsf{tanh}(H_{t-1}W_{hc} + X_tW_{xc} + b_c)$$
$$C_t = \underbrace{F_t \otimes C_{t-1}}_{\text{Forget information}} + \underbrace{I_t \otimes \tilde{C}_t}_{\text{Input new information}}$$

- O_t control the information need to output to H_t from C_t.

$$H_t = O_t \otimes \mathsf{tanh}(C_t)$$
$$y_t = H_tW_{hy} + b_y$$

GRU

- The gated recurrent unit (GRU) was first proposed by Kyunghyun Cho et al in 2014.
- The GRU simplified the architecture of LSTM to speed up computation time.
- Compare to LSTM, GRU often achieves comparable performance with faster computation time.


y_t
H_{t-1}
H_t
1-
$\tilde{H}_t$
Candidate
hidden state
Reset
gate
R_t
Update
gate
Z_t
H_{t-1}, X_t
Sigmoid
tanh
Element-wise
multiplication
Addition

- GRU has two gates: Reset gate (R_t), and update gate (Z_t).

$$R_t = \sigma(H_{t-1}W_{hr} + X_tW_{xr} + b_r)$$
$$Z_t = \sigma(H_{t-1}W_{hz} + X_tW_{xz} + b_z)$$

- R_t control the information need to remember from H_{t-1}.

$$\tilde{H}_t = \mathrm{tanh}((R_t \otimes H_{t-1})W_{hh} + X_t W_{xh} + b_h)$$

- Z_t determines the proportion of the multiplication of H_{t-1} and $\tilde{H}_t$ to get H_t.

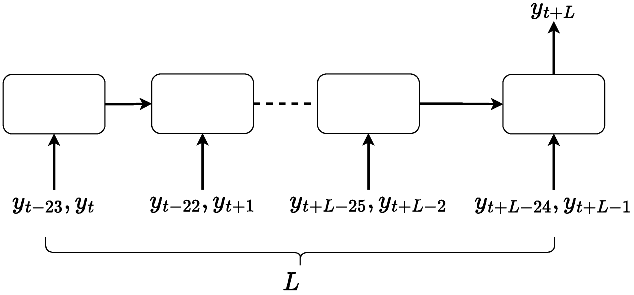
$$H_t = \underbrace{Z_t \otimes H_{t-1}}_{\text{Forget information}} + \underbrace{(1 - Z_t) \otimes \tilde{H}_t}_{\text{Input new information}}$$

- Output

$$y_t = H_t W_{hy} + b_y$$

Real Data: Bike Sharing Demand

- The data records the demand for City Bike rentals in Kaohsiung per hour from January 2017 to March 2019.
- To create data for RNN, LSTM, and GRU models, use the following structure.



The data is split into three sets: training (60%), validation (20%), and test (20%).

- The training set is used to train the model.
- The validation set is used to prevent overfitting and stop the training procedure early.
- The test set is used to evaluate the performance of the model.

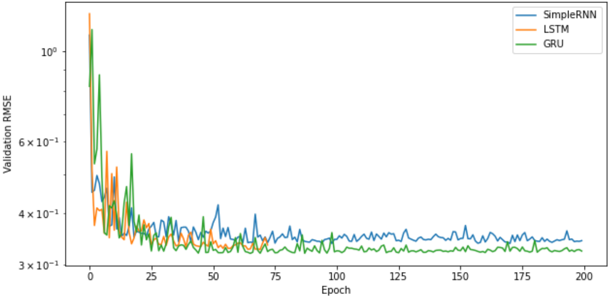


Figure : RMSE of validation set for each epoch.

- The figure shows that, compared to the simple RNN, both LSTM and GRU reach lower RMSE.

Table : RMSE of test set.

	test RMSE (log)	test RMSE
SimpleRNN	0.259	117.222
LSTM	0.267	108.754
GRU	0.241	104.935
SARIMA	0.273	114.618

- The table indicates that the GRU model has the lowest test RMSE among all the models.