Recurrent Neural Network (RNN) for XOR problem

지능제어 2017 중간고사

발표일: 2017.10.26

메카트로닉스공학과 201760122 홍효성

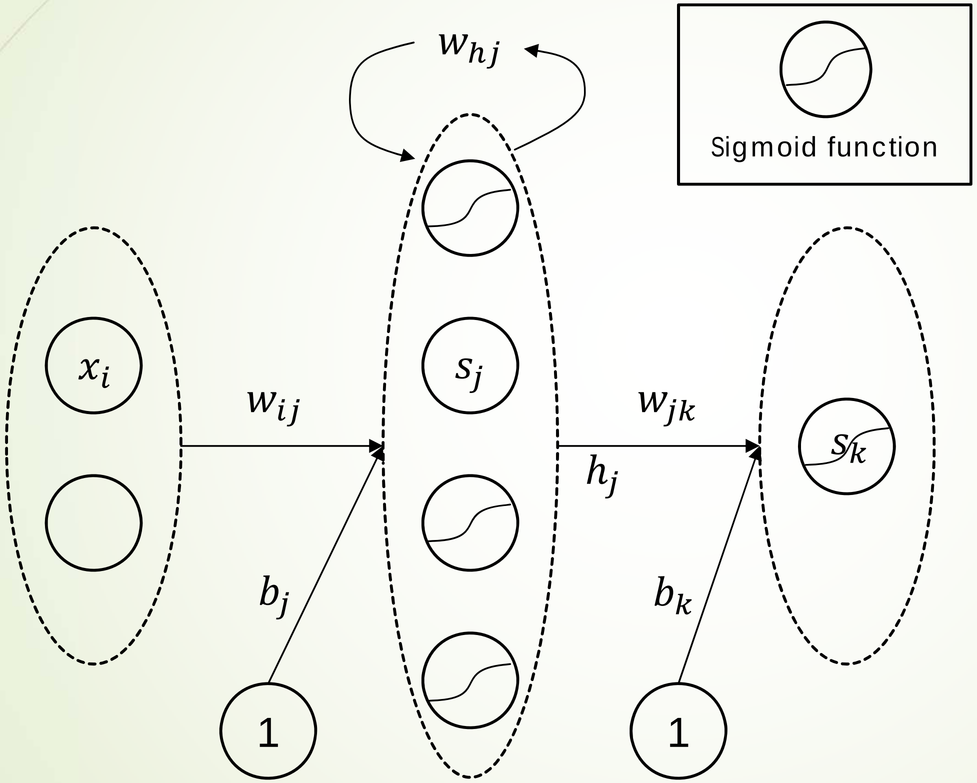
Midterm Exam Take Home
Due on 10/31/2017

1. Try to use the diagonal recurrent neural network with one hidden layer to XOR problem.
2. Derive back-propagation algorithm.
3. Simulate XOR classification.
4. Discuss the results.

Scores are determined in class demonstration.

If you have a problem with dates, we do on10/24/2017.

RNN Architecture



- Forward process

$$s_j = \sum_{i=1}^{N_I}\left(w_{ij}x_i + b_j\right) + \sum_{h=1}^{N_H} w_{hj} h_j^{-1}$$

$$h_j = f\left(s_j\right)$$

$$s_k = \sum_{j=1}^{N_H}\left(w_{jk}h_j + b_k\right)$$

$$y_k = f\left(s_k\right)$$

$$e_k = y_{dk} - y_k$$

$$J = \frac{1}{2}\sum_{k=1}^{N_O} e_k^2$$

(h^{-1}: previous hidden layer)

RNN Backpropagation

- Update variables: $w_{ij}, b_j, w_{hj}, w_{jk}, b_k$

$$\frac{\partial J}{\partial w_{jk}} = \frac{\partial J}{\partial e_k}\frac{\partial e_k}{\partial w_{jk}} = e_k \frac{\partial\left(y_{dk} - y_k\right)}{\partial y_k}\frac{\partial y_k}{\partial w_{jk}} = -e_k \frac{\partial f\left(s_k\right)}{\partial s_k}\frac{\partial s_k}{\partial w_{jk}}$$

$$= -e_k\left(1 - f\left(s_k\right)\right) f\left(s_k\right) \frac{\partial \sum_{j=1}^{N_H}\left(w_{jk}h_j + b_k\right)}{\partial w_{jk}}$$

$$= -e_k\left(1 - f\left(s_k\right)\right) f\left(s_k\right) h_j$$

$$\frac{\partial J}{\partial b_k} = \frac{\partial J}{\partial e_k}\frac{\partial e_k}{\partial b_k} = e_k \frac{\partial\left(y_{dk} - y_k\right)}{\partial y_k}\frac{\partial y_k}{\partial b_k} = -e_k \frac{\partial f\left(s_k\right)}{\partial s_k}\frac{\partial s_k}{\partial b_k}$$

$$= -e_k\left(1 - f\left(s_k\right)\right) f\left(s_k\right) \frac{\partial \sum_{j=1}^{N_H}\left(w_{jk}h_j + b_k\right)}{\partial b_k}$$

$$= -e_k\left(1 - f\left(s_k\right)\right) f\left(s_k\right)$$

RNN Backpropagation

- Update variables: $w_{ij}, b_j, w_{hj}, w_{jk}, b_k$

$$\frac{\partial J}{\partial w_{hj}} = \frac{\partial J}{\partial e_k}\frac{\partial e_k}{\partial w_{hj}} = e_k \frac{\partial\left(y_{dk} - y_k\right)}{\partial y_k}\frac{\partial y_k}{\partial w_{hj}} = -e_k \frac{\partial f\left(s_k\right)}{\partial s_k}\frac{\partial s_k}{\partial w_{hj}}$$

$$= -e_k\left(1 - f\left(s_k\right)\right) f\left(s_k\right) \frac{\partial \sum_{j=1}^{N_H}\left(w_{jk} h_j + b_k\right)}{\partial h_j}\frac{\partial h_j}{\partial w_{hj}}$$

$$= -e_k\left(1 - f\left(s_k\right)\right) f\left(s_k\right) \sum_{j=1}^{N_H}\left(w_{jk}\frac{\partial f\left(s_j\right)}{\partial s_j}\frac{\partial s_j}{\partial w_{hj}}\right)$$

$$= -e_k\left(1 - f\left(s_k\right)\right) f\left(s_k\right) \sum_{j=1}^{N_H}\left(w_{jk}\left(1 - f\left(s_j\right)\right) f\left(s_j\right)\frac{\partial\left(\sum_{i=1}^{N_I}\left(w_{ij} x_i + b_j\right) + \sum_{j=1}^{N_H} w_{hj} h_j^{-1}\right)}{\partial w_{hj}}\right)$$

$$= -e_k\left(1 - f\left(s_k\right)\right) f\left(s_k\right) \sum_{j=1}^{N_H}\left(w_{jk}\left(1 - f\left(s_j\right)\right) f\left(s_j\right) h_j^{-1}\right)$$

RNN Backpropagation

- Update variables: $w_{ij}, b_j, w_{hj}, w_{jk}, b_k$

$$\frac{\partial J}{\partial w_{ij}} = \frac{\partial J}{\partial e_k}\frac{\partial e_k}{\partial w_{ij}} = e_k \frac{\partial\left(y_{dk} - y_k\right)}{\partial y_k}\frac{\partial y_k}{\partial w_{ij}} = -e_k \frac{\partial f\left(s_k\right)}{\partial s_k}\frac{\partial s_k}{\partial w_{ij}}$$

$$= -e_k\left(1 - f\left(s_k\right)\right) f\left(s_k\right)\frac{\partial \sum_{j=1}^{N_H}\left(w_{jk}h_j + b_k\right)}{\partial h_j}\frac{\partial h_j}{\partial w_{ij}}$$

$$= -e_k\left(1 - f\left(s_k\right)\right) f\left(s_k\right)\sum_{j=1}^{N_H}\left(w_{jk}\frac{\partial f\left(s_j\right)}{\partial s_j}\frac{\partial s_j}{\partial w_{ij}}\right)$$

$$= -e_k\left(1 - f\left(s_k\right)\right) f\left(s_k\right)\sum_{j=1}^{N_H}\left(w_{jk}\left(1 - f\left(s_j\right)\right) f\left(s_j\right)\frac{\partial\left(\sum_{i=1}^{N_I}\left(w_{ij}x_i + b_j\right) + \sum_{j=1}^{N_H} w_{hj}h_j^{-1}\right)}{\partial w_{ij}}\right)$$

$$= -e_k\left(1 - f\left(s_k\right)\right) f\left(s_k\right)\sum_{j=1}^{N_H}\left(w_{jk}\left(1 - f\left(s_j\right)\right) f\left(s_j\right)\left(\sum_{i=1}^{N_I} x_i\right)\right)$$

RNN Backpropagation

- Update variables: $w_{ij}, b_j, w_{hj}, w_{jk}, b_k$

$$\frac{\partial J}{\partial b_j} = \frac{\partial J}{\partial e_k}\frac{\partial e_k}{\partial b_j} = e_k \frac{\partial\left(y_{dk} - y_k\right)}{\partial y_k}\frac{\partial y_k}{\partial b_j} = -e_k \frac{\partial f\left(s_k\right)}{\partial s_k}\frac{\partial s_k}{\partial b_j}$$

$$= -e_k\left(1 - f\left(s_k\right)\right) f\left(s_k\right) \frac{\partial \sum_{j=1}^{N_H}\left(w_{jk} h_j + b_k\right)}{\partial h_j}\frac{\partial h_j}{\partial b_j}$$

$$= -e_k\left(1 - f\left(s_k\right)\right) f\left(s_k\right) \sum_{j=1}^{N_H}\left(w_{jk} \frac{\partial f\left(s_j\right)}{\partial s_j}\frac{\partial s_j}{\partial b_j}\right)$$

$$= -e_k\left(1 - f\left(s_k\right)\right) f\left(s_k\right) \sum_{j=1}^{N_H}\left(w_{jk}\left(1 - f\left(s_j\right)\right) f\left(s_j\right) \frac{\partial\left(\sum_{i=1}^{N_I}\left(w_{ij} x_i + b_j\right) + \sum_{j=1}^{N_H} w_{hj} h_j^{-1}\right)}{\partial b_j}\right)$$

$$= -e_k\left(1 - f\left(s_k\right)\right) f\left(s_k\right) \sum_{j=1}^{N_H}\left(w_{jk}\left(1 - f\left(s_j\right)\right) f\left(s_j\right)\right)$$

RNN Backpropagation

- Summary

$$\frac{\partial J}{\partial w_{ij}} = -e_k\left(1-f\left(s_k\right)\right)f\left(s_k\right)\sum_{j=1}^{N_H}\left(w_{jk}\left(1-f\left(s_j\right)\right)f\left(s_j\right)\left(\sum_{i=1}^{N_I}x_i\right)\right)$$

$$\frac{\partial J}{\partial b_j} = -e_k\left(1-f\left(s_k\right)\right)f\left(s_k\right)\sum_{j=1}^{N_H}\left(w_{jk}\left(1-f\left(s_j\right)\right)f\left(s_j\right)\right)$$

$$\frac{\partial J}{\partial w_{hj}} = -e_k\left(1-f\left(s_k\right)\right)f\left(s_k\right)\sum_{j=1}^{N_H}\left(w_{jk}\left(1-f\left(s_j\right)\right)f\left(s_j\right)h_j^{-1}\right)$$

$$\frac{\partial J}{\partial w_{jk}} = -e_k\left(1-f\left(s_k\right)\right)f\left(s_k\right)h_j$$

$$\frac{\partial J}{\partial b_k} = -e_k\left(1-f\left(s_k\right)\right)f\left(s_k\right)$$

RNN Backpropagation

- Gradient descent

$$w_{ij} \leftarrow w_{ij} - \eta \nabla w_{ij}$$
$$b_j \leftarrow b_j - \eta \nabla b_j$$
$$w_{hj} \leftarrow w_{hj} - \eta \nabla w_{hj}$$
$$w_{jk} \leftarrow w_{jk} - \eta \nabla w_{jk}$$
$$b_k \leftarrow b_k - \eta \nabla b_k$$

XOR Problem Test Result

- XOR data definition

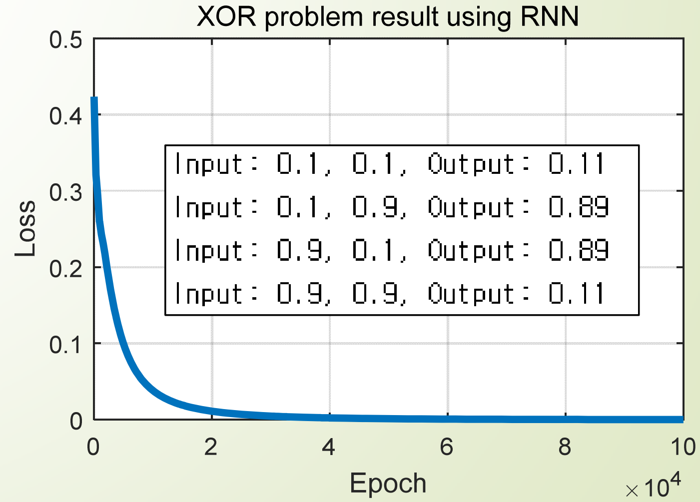
$$x_1 \otimes x_2 = y$$

x_1	x_2	y
0.1	0.1	0.1
0.1	0.9	0.9
0.9	0.1	0.9
0.9	0.9	0.1

- Since the sigmoid function that has the range of 0<f(x)<1 was used for activation function, input data was set to have 0.1 and 0.9 rather than 0 and 1.

- XOR Test Result on MATLAB#

Parameter	Value
Input_size	2
Hidden_size	20
Output_size	1
Sequence_length	2
Epoch_size	100,000
Learning_rate(η)	0.5



My source code can be downloaded from: https://github.com/peytonhong/RNN/blob/master/RNN_XOR.m