INF436 Machine Learning: Lab 6 Recurrent Neural Network

Jae Yun JUN KIM[*]

February 18, 2017

Due: For next lecture/lab session,

Evaluation: code (in group) + (theoretical, practical) questions (individual).

Suppose that you would like to construct a robot that makes a decision based on a signal that it receives through a sensor. This signal consists of 8 bits, and the robot makes decisions based on the sum of these bits. In this Lab, you are asked to construct a binary addition module using a *recurrent neural network (RNN)* with the following input-output relationship:

$$\hat{y} = \sum_{t=1}^{T} x_t, \qquad\qquad \text{where } x_t{\in}\{0,1\} \text{ for all } t,\ T{=}8.$$

1 Exercises

1. Generate the training input data (X) consisting of 30 sequences of 8 binary numbers, following a uniform distribution, where the probability of generating a "0" is the same as that of generating a "1". Make the training output (y) for each sequence be the sum of its elements.

2. Implement the binary addition module using a *recurrent neural network (RNN)* with *resilient propagation.* Because this is a simple linear problem, let us build the RNN with no intercept term, a single hidden neuron, and identity activation function. In your code, you should:
   - 2.a implement the forward steps to find the state values.
   - 2.b compute the gradient of the loss function with respect to the output signal.
   - 2.c implement the backward steps to find the gradient of the loss function with respect to the weights.
   - 2.d update the weights using the resilient propagation approach.

3. Proceed the same steps of question 2, but using this time the *backpropagation approach* (instead of the resilient propagation approach).

4. From your results obtained for question 2 and question 3, comment on the differences that you noticed between the *resilient propagation* approach and the *backpropagation* approach. Justify your observations.

5. Generate test input data consisting of 2 or 3 binary 8-bit sequences (different from the ones generated for training), and verify that the implemented binary addition module works properly.

[*] ECE Paris Graduate School of Engineering, 37 quai de Grenelle CS71520 75 725 Paris 15, France; jaeyun.jun-kim@ece.fr