Practical 7 – Recurrent Neural Networks

CS5062 – Machine Learning

Overview

In this exercise, you will work on the recurrent neural networks (RNN) and use Python to implement the RNN algorithm introduced in the Lecture.

Theoretical Exercises

As we learnt in previous lecture, we saw feedforward artificial neural networks, which do not contain any cycles and for which the nodes do not maintain a persistent state over several runs. This exercise considers artificial neural networks with nodes that maintain a persistent state that can be updated. This kind of neural network is called a recurrent neural networks (RNN). As an example, consider the following RNN with

$$y_t = W_{x_t} + V_{s_t}$$
$$s_{t+1} = y_t$$

from some initial state s_0, where t denotes the t-th call of the RNN, i.e., x_t is the t-th input.

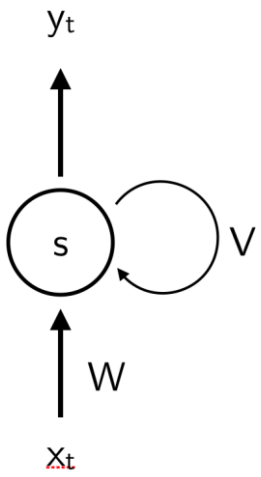


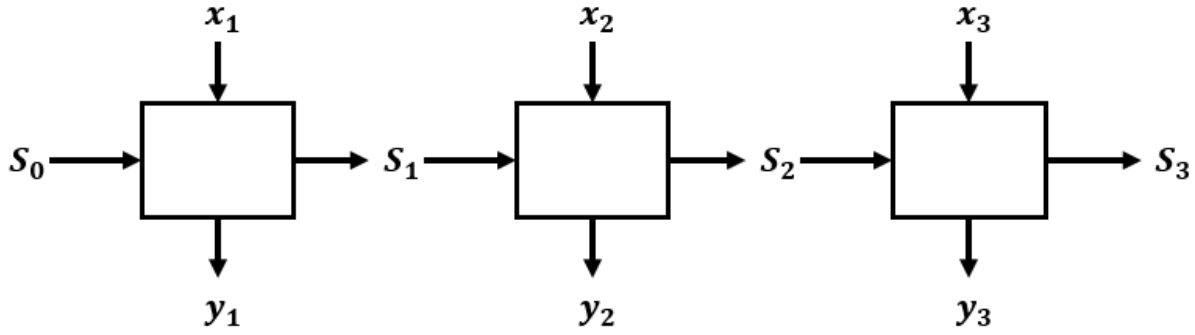
Figure 1

1) What is the recurrent state in the RNN from Figure 1? Name one example that can be more naturally modelled with RNNs than with feedforward neural networks?
   Solution: The recurrent state is denoted s. In this case it conincides with the output. Reccurent models are used to model data with temporal structure, e.g. speech recognition, machine translation.
2) As the state of an RNN changes over different runs of the RNN, the loss functions that we use for feedforward neural networks do not yield consistent results. For given dataset X, please propose a loss function (based on the mean square loss function) for RNNs and justify why you chose this loss function.

Solution: We have a data $X = \{(x_t, y_t)\}$, where we assume that the data is ordered temporally. Thus, we define the loss function to be $L(U, W, s_0) = \sum_{t=1}^{T}\big(y(t) - f(x_t, s_{t-1}(U, W), U, W)\big)^2/T$, where s_{t-1} is the previous recurrent state. The initial state s_0 needs to be specified and the problem depends on it as well.

3) For a dataset $X \coloneqq (x_t, y_t)_1^k$ (for some $k \in \mathbb{N}$), show how information is propagated by drawing a feedforward neural network that corresponds to the RNN from Figure 1 for $k = 3$. Recall that a feedforward neural network does not contain nodes with a persistent state. (Hint: unfold the RNN.)

Solution:



Programming Exercises

The task of this exercise is to write Python code to implement RNN algorithm on a stack price prediction:

1) We use "Google stack price" data for this tutorial. The training and testing sets of the dataset can be downloaded from myAberdeen.
2) Your code to implement RNN algorithm should include the following functions:
    a. Load training and testing data set;
    b. Get the column of "open" price;
    c. Normalize the features;
    d. Create a data structure with 60-time steps and 1 output. That is, the previous 60 time-steps are features and the 61-th time-step is the label;
    e. Initialise the RNN;
    f. Specify your own RNN;
    g. Compile and fit the RNN for training set;
3) Apply your RNN algorithm to the testing set of "Google stack price" and predict the predicted prices and actual prices along with computing mean square error.

Solution: The source codes of the solution is provided in "Recurrent_Neural_Network.ipynb" file, which can be downloaded from myAberdeen.