Reports on Science and Technology, Volume 1, Issue 1
Tuesday, April 17, 2018

Introduction to Recurrent Neural Networks

Abu Ahmad Hania, Tekno Indonesia, Palembang, Indonesia

abuahmad@promotionme.com

Abstract

Recurrent neural networks (RNNs) are popular models that have exhibited a great potential in many machine learning tasks. Despite of their recent popularity, only a small number of resources that thoroughly explain how RNN work, and how to implement them. This paper presents a short tutorial on RNN.

Introduction

The applications of time series are in two stages: First, it allows us to model arbitrary time series on how likely the series to occur in the real data. This gives us a measure of the modeled time series and the actual data. Secondly, a time series allows us to generate new time series.

The idea of the RNN is to exploit the redundancy in a sequential data. We tried to rephrase the explanation of the RNN in [1]. In a traditional neural network, the assumption is that all inputs and outputs are independent of each other. But for many tasks, that is not a good assumption. If we want to predict the next data in a sequence, it is better to know the preceding or the previous data. The RNNs have recurrent name since they perform the same process for every element in a sequence. Another way to describe RNNs is that they have a memory that captures information about what has been calculated so far. In theory, RNNs can make use of information in arbitrary long time series data, but in practice they are limited to looking back only a few steps.

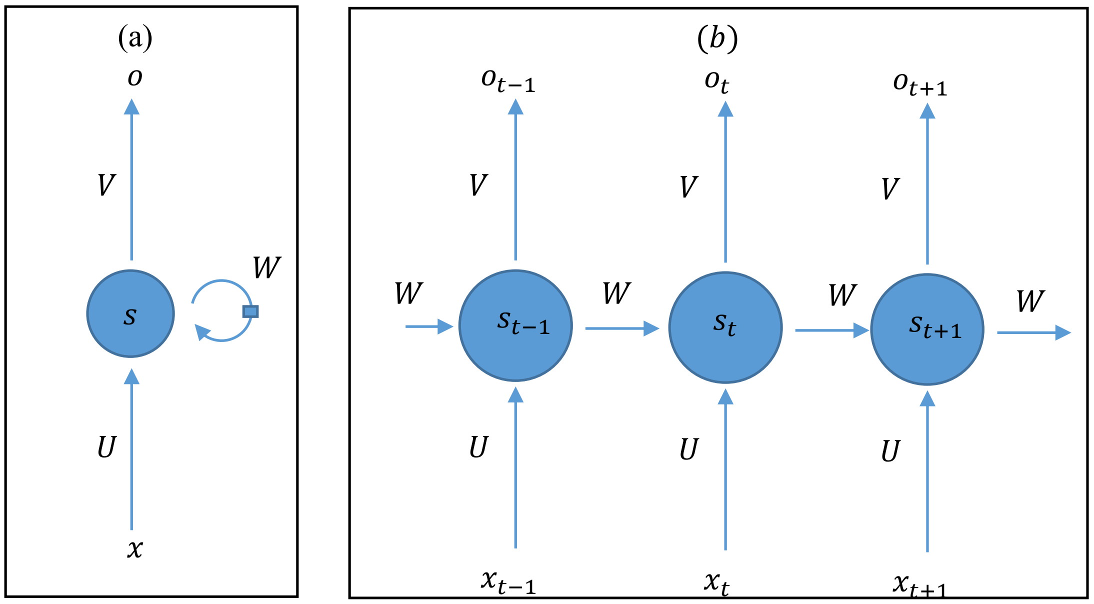

Figure 1 Unfolding an RNN (a) a RNN (b) unfolded RNN

Figure 1 shows a RNN being unfolded into a full time-lagged network. By unfolding we directly mean that we express the network for the complete time sequence. For instance, if the network we want to model is a sequence of five elements, then the network is supposed to be unfolded into five-layer neural networks, one layer for each element.

RNN Algorithm

The algorithm that governs the RNN process is as follows:

a. x_t it the input at time t. For instance, x_1 is a one element in a time series corresponding to third element in a time series.
b. s_t is the hidden state at time t. It represents the memory of the network. s_t is computed based on the previous hidden state and the input at current step $s_t = f(Ux_t + Ws_{t-1})$. The function f is usually a nonlinear function such as tangent hyperbolic or Rectified Linear Unit (ReLU which is required to calculate the first hidden state, is typically set to zero as initial values.
c. o_t is the output at step t. For instance, if we desired to estimate the next value in a time series it would be a vector of probabilities across all values. $o_t = g(Vs_t)$.

Methodology

The hidden state s_t represents the memory of the network. s_t captures information in the previous time steps. The output at step o_t is calculated only based on the memory at time t. In practice, s_t typically cannot capture information from too many steps in the past. Unlike traditional, deep neural network, which uses different parameters at each layer, a

RNN shares the same parameters (U,V, and W) across all time steps. This represents the fact that we are performing the same procedure at each step, just with different inputs. This tremendously reduces the total number of parameters needed to learn.

Figure 1 has outputs at each time step, but depending on the task this may not be necessary. For example, when predicting the value of a time series we may only care about the final output, the value of each element. Similarly, we may not need inputs at each time sample. The main feature of an RNN is its hidden states, which captures some information about a time series.

RNN features

RNN have exhibited great successes in many tasks involving sequential data. At this point, the most common kind of RNNs are the Long Short Term Memory (LSTMs) since they have better performance at capturing long term dependencies. The LSTMs also have different way of calculating the hidden state.

References

[1] Britz, Denny. "Recurrent Neural Networks Tutorial, Part 1 – Introduction to RNNs". 2015. http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/