ECE599/692 - Deep Learning

Lecture 14 – Recurrent Neural Network (RNN)

Hairong Qi, Gonzalez Family Professor
Electrical Engineering and Computer Science
University of Tennessee, Knoxville
http://www.eecs.utk.edu/faculty/qi
Email: hqi@utk.edu

Outline

• Introduction
• LSTM vs. GRU
• Applications and Implementations
• References:

A friendly introduction to NN [1]
A friendly introduction to RNN

A more complicated case

A more complicated case (cont’d)
A more complicated case (cont’d)

The children’s book example from Brandon Rohrer [2]

Doug saw Jane.
Jane saw Spot.
Spot saw Doug.

Dictionary = {Doug, Jane, Spot, saw, .}

Potential mistakes:
Doug saw Doug.
Doug saw Jane saw Spot saw Doug…

The standard RNN module

\[ h_t = f(Ux_t + Wh_{t-1} + b) \]

Gradient vanishing issue: by the end of the RNN, the data from the first timestep has very little impact on the output of the RNN. An example of word prediction, “I grew up in France… I speak fluent French.”
The long-short term memory (LSTM) module

LSTMs are explicitly designed to avoid the long-term dependency problem.

Two keys of LSTM

“Cell state” which works like a conveyor belt runs straight down the entire chain, easy for information to flow along without changes.

“Gates” which control or decide what kind of information could go or throw away from the cell state.

LSTM – forget gate

Take the example of a language model trying to predict the next word based on all the previous ones. In such a problem, the cell state might include the gender of the present subject, so that the correct pronouns can be used. When we see a new subject, we want to forget the gender of the old subject.

\[ f_t = \sigma (W_f [h_{t-1}, x_t] + b_f) \]
It actually drops the information about the old subject’s gender and add the new information, as we decided in the previous steps.

$LSTM – output gate$

de$s = \sigma(W_s [h_{t-1}, x_t] + b_s)$

$h_t = de_t \cdot \tanh(C_t)$
The gated recurrent units (GRUs) module

\[ z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \]
\[ r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \]
\[ \tilde{h}_t = \tanh(W \cdot [r_t \odot h_{t-1}, x_t]) \]
\[ h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \]

Similar with LSTM but with only two gates and less parameters. The “update gate” determines how much of previous memory to be kept. The “reset gate” determines how to combine the new input with the previous memory.

Comparison of the gating mechanism

\[ f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \]
\[ i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \]
\[ C_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \]
\[ C_t = C_t \odot i_t + C_{t-1} \odot (1 - i_t) \]
\[ h_t = C_t \odot \sigma(b) + h_{t-1} \odot (1 - \sigma(b)) \]
**LSTM vs. GRU**

\[
\begin{align*}
    f_t &= \sigma(W_f [h_{t-1}, x_t] + b_f) \\
    i_t &= \sigma(W_i [h_{t-1}, x_t] + b_i) \\
    C_t &= \tanh(W_c [h_{t-1}, x_t] + b_c) \\
    o_t &= \sigma(W_o [h_{t-1}, x_t] + b_o) \\
    h_t &= o_t \cdot \tanh(C_t)
\end{align*}
\]

\[
\begin{align*}
    f_t &= \sigma(W_f [h_{t-1}, x_t] + b_f) \\
    r_t &= \sigma(W_r [h_{t-1}, x_t] + b_r) \\
    h_t &= \tanh(W \cdot [r_t \cdot h_{t-1}, x_t] + b_h)
\end{align*}
\]

**Application example: The talking face**

Goal: Given an arbitrary audio clip and a face image, automatically generate realistic and smooth face video with accurate lip sync.

![Image](image.png)

(Suwajanakorn et al., 2017)

Application: Face animation, entertainment, video bandwidth reduction, etc.

**The proposed framework**

The proposed method: conditional video generation
The proposed framework

\[
E_G = R_{\text{inc}} \cdot \ln(1 - DL/u) + R_{\text{em}} \cdot \ln(1 - DL/u)\]  \hspace{1cm} (2)

\[
E_D = R_{\text{in}} \cdot \ln(1 - DL/u)\]  \hspace{1cm} (3)

The final objective function is

\[
\min_{\theta} \sum \left[ \alpha(t) - \hat{\alpha}(t) \right] \]  \hspace{1cm} (4)

where \( F(G, E, R, D) = \sum \alpha(t) + \lambda_{\text{text}} + \lambda_{\text{sec}} + \lambda_{\text{sec}} \).