Practical Deep Neural Networks

GPU computing perspective

Recurrent Neural Networks

Yuhuang Hu   Chu Kiong Loo

Advanced Robotic Lab
Department of Artificial Intelligence
Faculty of Computer Science & IT
University of Malaya

Outline

Outline

Assumed prerequisites

- ☆ Neural Computation (DL book chapter 4)
- ☆ Machine Learning Basics (DL book chapter 5)
- ☆ MLP Networks (DL book chapter 6)

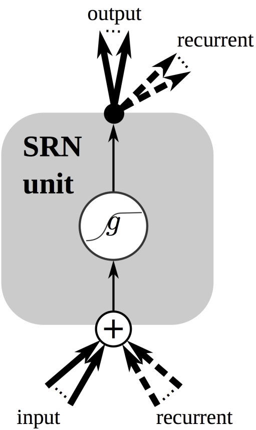
Suggest Readings

- ✍ Deep Learning book Chapter 10: Sequence Modeling: Recurrent Recursive Nets
- ✍ CS224d: GRUs and LSTMs – for machine translation
- ✍ The Unreasonable Effectiveness of Recurrent Neural Networks
- ✍ LSTM: A Search Space Odyssey
- ✍ Supervised Sequence Labelling with Recurrent Neural Networks

Outline

1. Introduction
2. SRN
3. LSTM
4. Sequence Modeling
5. Q&A

SRN architecture



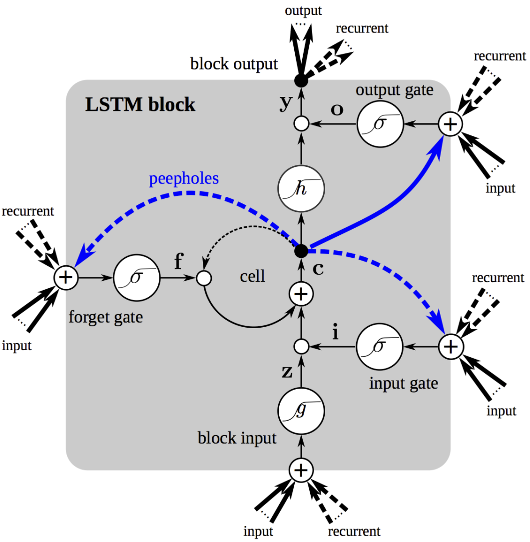
SRN architecture

$$\mathbf{y}_h^t = f_h(\mathbf{W}_i\mathbf{x}^t + \mathbf{W}_h\mathbf{y}^{t-1})$$
$$\mathbf{y}_o^t = f_o(\mathbf{W}_o\mathbf{y}_h^t)$$

where $\mathbf{W}_h$, $\mathbf{W}_i$, $\mathbf{o}$ are the hidden, input and output weight matrices, $\mathbf{x}^t$ is the input vector, and $\mathbf{y}_h^t$ is a vector representing the activation of hidden units at time step t. Functions $f_h(\cdot)$ and $f_o(\cdot)$ are non-linear functions.

Outline

LSTM architecture



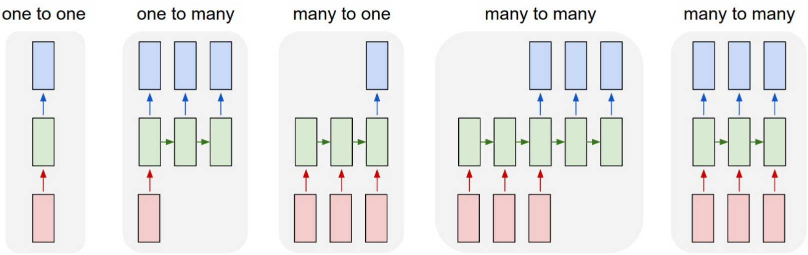
LSTM architecture

$$
\begin{aligned}
\mathbf{z}^t &= g(\mathbf{W}_z\mathbf{x}^t + \mathbf{R}_z\mathbf{y}^{t-1} + \mathbf{b}_z) && \textit{block input}\\
\mathbf{i}^t &= \sigma(\mathbf{W}_i\mathbf{x}^t + \mathbf{R}_i\mathbf{y}^{t-1} + \mathbf{p}_i \odot \mathbf{c}^{t-1} + \mathbf{b}_i) && \textit{input gate}\\
\mathbf{f}^t &= \sigma(\mathbf{W}_f\mathbf{x}^t + \mathbf{R}_f\mathbf{y}^{t-1} + \mathbf{p}_f \odot \mathbf{c}^{t-1} + \mathbf{b}_\mathbf{f}) && \textit{forget gate}\\
\mathbf{c}^t &= \mathbf{i}^t \odot \mathbf{z}^t + \mathbf{f} \odot \mathbf{c}^{t-1} && \textit{cell state}\\
\mathbf{o}^t &= \sigma(\mathbf{W}_o\mathbf{x}^t + \mathbf{R}_o\mathbf{y}^{t-1} + \mathbf{p}_o \odot \mathbf{c}^{t} + \mathbf{b}_o) && \textit{output gate}\\
\mathbf{y}^t &= \mathbf{o}^t \cdot h(\mathbf{c}^t) && \textit{block output}
\end{aligned}
$$

Here $\mathbf{x}^t$ is the input vector at time t, the $\mathbf{W}$ are rectangular matrices, the $\mathbf{R}$ are square recurrent weight matrices, the $\mathbf{p}$ are peehole weights vectors and $\mathbf{b}$ are bias vectors. Functions σ, g and h are point-wise non-linear activation functions: *logistic sigmoid* is used for as activation function of the gates and *hyperbolic tangent* is usually used as the block input and output activation function. The point-wise multiplication of two vectors is denoted with $\odot$

Outline

Modes of Processing



Left to right: **(a)** fixed-size input to fixed-size output (*e.g.* image classification); **(b)** sequence output (*e.g.* image captioning); **(c)** sequence input (*e.g.* sentiment analysis); **(d)** sequence input and sequence output (*e.g.* machine translation); **(e)** synced sequence input and output (*e.g.* video classification)

Example: character prediction

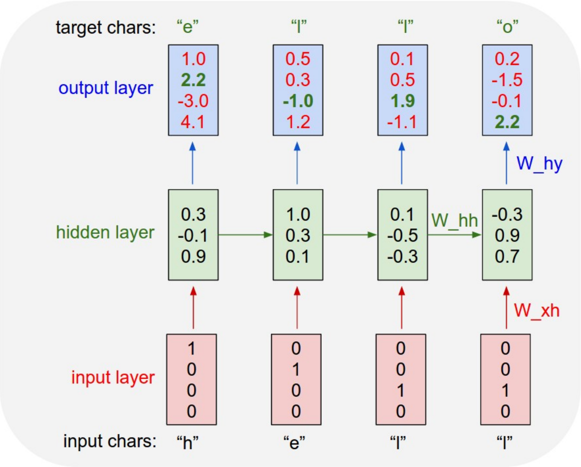
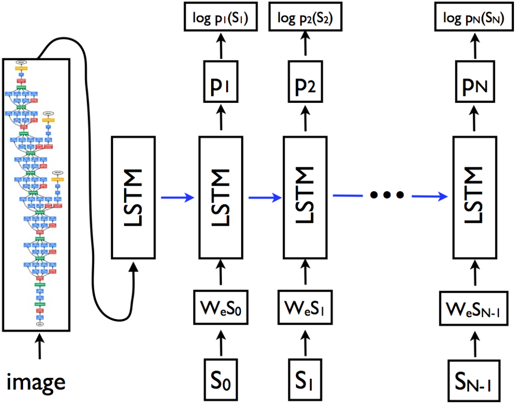


Figure: Predict "hello"

Example: image captioning



Outline

Q&A