Neural Machine Translation with Sequence to Sequence and Attention

matthew@cinnamon.is

Feb 2019

1 Introduction

In the last assignment, we have dealt with predicting a next token (word or character) based on previous tokens in which we only have one single output required (the next word). However, there're many tasks in NLP required a whole sequence as the output and the length of the output might not be related to the length of the input. For example,

- **Translation:** the input is a sentence in one language while the output is the translated sentence in another language.
- **Summarization:** the input is a long sequence of text and the output is a much shorter summary.
- **Speech recognition:** the system take a sequence of audio wave values as the input and outputting a text sequence.

In this assignment, we hope to introduce another major task in NLP, **Machine Translation** and a common approach for this problem, **sequence-to-sequence**. The rest of the assignment is organized as follow:

- Sec. 2 will give you the background theory behind NMT
- Sec. 3 will give you what is seq2seq and how you use it for NMT
- Sec. 4 will give additional information to build your NMT system, and what should you write in the repot.

2 Neural Machine Translation

In Machine Translation, we need to translate a sequence x from the source language into a sequence y in the target language. Let's say we want to translate from Vietnamese to English, then we have to find the best English sentence $\boldsymbol{y}$ given Vietnamese sentence $\boldsymbol{x}$:

$$argmax_{\boldsymbol{y}} P(\boldsymbol{y}|\boldsymbol{x}) \tag{1}$$

In order to do that in NMT, we have to model the probability distribution $P(\boldsymbol{y}|\boldsymbol{x})$ with our neural networks, which means:

- with an input sentence $\boldsymbol{x} = \{\boldsymbol{x}^1, ..., \boldsymbol{x}^T\}$ where $\boldsymbol{x}^i \in V_x$ and V_x is the Vietnamese vocabulary
- we have to calculate the probability of the output sentence $\boldsymbol{y} = \{\boldsymbol{y}^1, ..., \boldsymbol{y}^{T'}\}$ where $\boldsymbol{y}^i \in V_y$ and V_y is the English vocabulary

Remember how we break down the language model $P(\boldsymbol{x}^1, ..., \boldsymbol{x}^T)$ in the previous assignment?

$$\begin{aligned} P(\boldsymbol{x}^1, ..., \boldsymbol{x}^T) &= P(\boldsymbol{x}^1) \times P(\boldsymbol{x}^2|\boldsymbol{x}^1) \times ... \times P(\boldsymbol{x}^T|\boldsymbol{x}^{T-1}, ..., \boldsymbol{x}^1) \\ &= \prod_{t=1}^{T} P(\boldsymbol{x}^t|\boldsymbol{x}^{t-1}, ..., \boldsymbol{x}^1) \end{aligned} \tag{2}$$

Similarly, we will break down our translation model in to probabilities of each word in the output sentence:

$$\begin{aligned} P(\boldsymbol{y}^1, ..., \boldsymbol{y}^{T'}|\boldsymbol{x}) &= P(\boldsymbol{y}^1|\boldsymbol{x}) \times P(\boldsymbol{y}^2|\boldsymbol{y}^1, \boldsymbol{x}) \times ... \times P(\boldsymbol{y}^{T'}|\boldsymbol{y}^{T'-1}, ..., \boldsymbol{y}^1, \boldsymbol{x}) \\ &= \prod_{t=1}^{T'} P(\boldsymbol{y}^t|\boldsymbol{y}^{t-1}, ..., \boldsymbol{y}^1, \boldsymbol{x}) \\ &= \prod_{t=1}^{T'} P(\boldsymbol{y}^t|\boldsymbol{y}^{t-1}, ..., \boldsymbol{y}^1, \boldsymbol{x}^T, ..., \boldsymbol{x}^1) \end{aligned} \tag{3}$$

Based on Eq. 3 our formula for Machine Translation is pretty similar to the Language Model's formula, as a result, we can use RNN to model the NMT systems in a nearly the same way as LM with few modifications (illustrated in Fig. 1). We will go into the details of this architecture in Sec. 3.

3 Sequence to Sequence

The sequence to sequence model aims to map a fixed length input with a fixed length output where the length of the input and output might differ. As illustrated in Fig. 1, the sequence to sequence model is consists of 2 main components:

- **Encoder** is a RNN layer (or a stack of several RNN layers) which takes the input sentence and *encode* its information into a fixed length *context vector*. This representation is expected to be a good summary of the whole source sequence's meaning. The most basic form of the encoder-decoder architecture uses the last hidden state of the encoder as the context vector.

$$\boldsymbol{h}_e^t = \sigma(\boldsymbol{W}_{eh}\boldsymbol{h}_e^{t-1} + \boldsymbol{W}_x\boldsymbol{x}^t) \tag{4}$$

$$\boldsymbol{h}_e^0 \textit{ is randomly initialized} \tag{5}$$

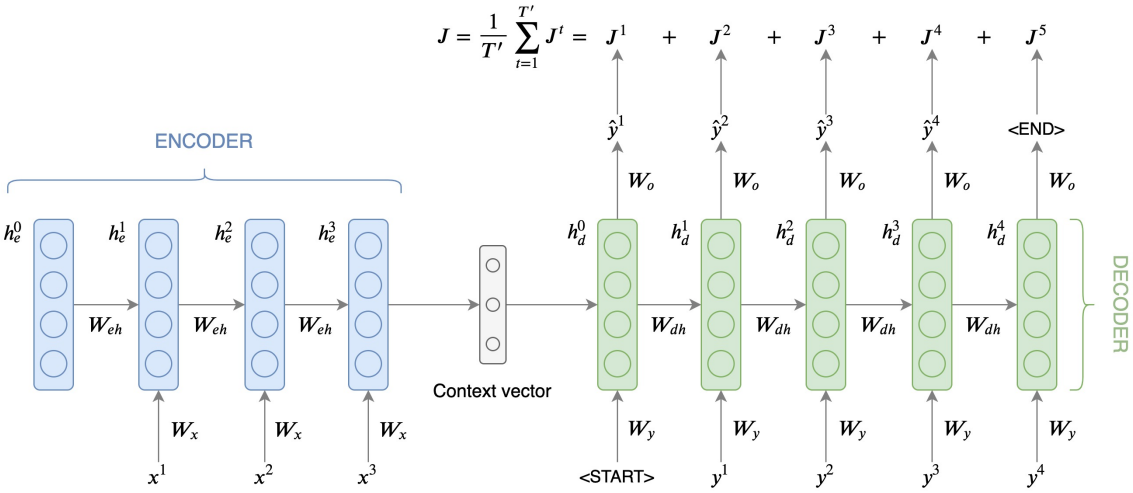

Figure 1: Neural Machine Translation with Seq2Seq

- **Decoder** is also a RNN to predict an output $\hat{\boldsymbol{y}}^t$ at each decoding time step t. Each recurrent unit accepts a hidden state from the previous unit and produces output as well as it own hidden state. At the start of the decoding process, the context vector is fed to the first recurrent cell of the decoder.

$$\boldsymbol{h}_d^t = \sigma(\boldsymbol{W}_{dh}\boldsymbol{h}_d^{t-1} + \boldsymbol{W}_y\boldsymbol{y}^t) \tag{6}$$

$$\hat{\boldsymbol{y}}^t = softmax(\boldsymbol{W}_o\boldsymbol{h}_d^t) \tag{7}$$

$$\boldsymbol{h}_d^0 = \boldsymbol{h}_e^T \tag{8}$$

4 Build A NMT system

In this assignment, your task is to build a simple NMT system with seq2seq which is capable of translating from Vietnamese to English.

4.1 Inference

First, we will need to know how to get the output text from the model discussed above first. Remember our criteria to get the translated sentence in Eq. 1? $argmax_{\boldsymbol{y}} P(\boldsymbol{y}|\boldsymbol{x})$. One naive way to acquire the translation is just get the word which have the best probability at a decoding timestep.

$$\boldsymbol{y}^t = argmax(\hat{\boldsymbol{y}}^t) \tag{9}$$

It's not the optimal way to get the translation, but let's settle with this method for now, we will learn how to improve it later.

4.2 Data

In this assignment, we will use the bilingual dataset Vietnamese-English at `https://nlp.stanford.edu/projects/nmt/data/iwslt15.en-vi/`

4.3 Report

In your report, go briefly about how you prepared training data, your configurations on model's hyperparameters, training settings and your model's initial results. Additionally, you should answer the following questions:

- What is the loss function for training Seq2seq model in the case of NMT? (How you calculate J^t)
- We're going to use BLEU (Bilingual Evaluation Understudy) for evaluating our NMT system, explain briefly how it work?