Advanced RNN Topics

(I)

INTELLIGENT SYSTEMS FOR PATTERN RECOGNITION (ISPR)

DAVIDE BACCIU – DIPARTIMENTO DI INFORMATICA - UNIVERSITA’ DI PISA

DAVIDE.BACCIU@UNIPI.IT
A 2 Lectures Outline

○ Dealing with structured/compound data
  ● Sequence-to-sequence
  ● Attention models

○ Dealing with very long-term dependencies
  ● Multiscale networks
  ● Adding memory components

○ Neural reasoners
  ● Neural memories
  ● Differentiable memory read, write, indexing

Extra Lecture
Tomorrow 14/04/2023
– h14 – Aula E
Gated RNN Refresher

LSTM Cell

\[
\begin{align*}
\hat{h}_t &= \sigma(c_t) \\
&= \sigma(\mathbf{x}_t \mathbf{W}_{\text{in}} + \mathbf{h}_{t-1} \mathbf{W}_{\text{hid}}) \\
&= \sigma(\mathbf{x}_t \mathbf{W}_{\text{in}} + \mathbf{h}_{t-1} \mathbf{W}_{\text{hid}}) \\
\end{align*}
\]

GRU Cell

\[
\begin{align*}
\hat{h}_t &= \sigma(c_t) \\
&= \sigma(\mathbf{x}_t \mathbf{W}_{\text{in}} + \mathbf{h}_{t-1} \mathbf{W}_{\text{hid}}) \\
&= \sigma(\mathbf{x}_t \mathbf{W}_{\text{in}} + \mathbf{h}_{t-1} \mathbf{W}_{\text{hid}}) \\
\end{align*}
\]
Use a simplified (and overloaded) graphical notation to represent GRNN layers and stacks of GRNN
Basic Gated RNN (GRNN) Limitations

○ GRNN are excellent to handle size/topology varying data in input
  ● How can we handle size/topology varying outputs?
  ● Sequence-to-sequence

○ Structured data is compound information
  ● Efficient processing needs the ability to focus on certain parts of such information
  ● Attention mechanism

○ GRNN have troubles dealing with very long-range dependencies
  ● Introduce multiscale representation explicitly in the architecture
  ● Introduce external memory components
Sequence-to-sequence and attention
Sequence Transduction

- Input and output are both sequences
- They may have different lengths
- Example: machine translation

The cat is on the table    Il gatto è sul tavolo

How do we model the context here?
Learning to Output Variable Length Sequences

The idea of an unfolded RNN with blank inputs-outputs does not really work well.

The approach is based on an encoder-decoder scheme.
Encoder

Produce a compressed and fixed length representation $c$ of all the input sequence $x_1, \ldots, x_n$

Originally $c = h_n$

Activations of an LSTM/GRU layer of $K$ cells
Decoder

A LSTM/GRU layer of K cells seeded by the context vector $c$

Different approaches to realize this in practice
If we share the parameters between encoder and decoder we can take $s_1 = c$

Or, at least, assume $c$ and $s_1$ have compatible size

We risk to lose memory of $c$ soon
c is contextual information kept throughout output generation
Decoder

\[ s_i = f(c, s_{i-1}, y_{i-1}) \]

It is better to work on a one-step-ahead scheme.
Remember teacher forcing (only) at training time.
Sequence-To-Sequence Learning

Encoder-Decoder can share parameters (but it is uncommon)

Encoder-Decoder can be trained end-to-end or independently

Reversing the input sequence in encoding typically resulted in increased performance (?!)

$\text{Encoder-Decoder can share parameters (but it is uncommon)}$

$\text{Encoder-Decoder can be trained end-to-end or independently}$

$\text{Reversing the input sequence in encoding typically resulted in increased performance (?!)}$
Example - Attention

The cat is on the table

Il gatto è sul tavolo
On the Need of Paying Attention

- Encoder-Decoder scheme assumes the hidden activation of the last input element summarizes sufficient information to generate the output
  - Bias toward most recent past
- Other parts of the input sequence might be very informative for the task
  - Possibly elements appearing very far from sequence end
On the Need of Paying Attention

- Attention mechanisms select which part of the sequence to focus on to obtain a good $c$
Attention Mechanisms – Blackbox View

What’s inside of the box? Attention Module

Aggregated seed \( c \)

Context info \( S \)

Encodings \( h_1, h_2, h_3, \ldots, h_n \)
What’s inside of the box?  

The Revenge of the Gates!
Opening the Box

\[ S \rightarrow h_1 \rightarrow \alpha_1 \rightarrow e_1 \Rightarrow \text{softmax} \rightarrow c \]

\[ h_1 \times \alpha_1 + e_1 \]

\[ h_2 \times \alpha_2 + e_2 \]

\[ h_3 \times \alpha_3 + e_3 \]

\[ \cdots \]

\[ h_n \times \alpha_n + e_n \]

\[ c \]
Opening the Box – Relevance

Tanh layer fusing each encoding with current context $s$

$$e_i = a(s, h_i)$$
Opening the Box – Softmax

A differentiable max selector operator

\[ \alpha_i = \frac{\exp(e_i)}{\sum_j \exp(e_j)} \]
Opening the Box – Voting

Aggregated seed by (soft) attention voting

\[ c = \sum_i \alpha_i h_i \]
Attention - Equations

- Relevance: $e_i = a(s, h_i)$
- Normalization: $\alpha_i = \frac{\exp(e_i)}{\sum_j \exp(e_j)}$
- Aggregation: $c = \sum_i \alpha_i h_i$
Attention in Seq2Seq

Context is past output state

Seed considers (subset of) the input states
Learning to Translate with Attention

Bahdanau et al, Show, Neural machine translation by jointly learning to align and translate, ICLR 2015
Seq-to-Seq on Steroids

Yonghui Wu et al. “Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation”,

DAVIDE BACCIU - ISPR COURSE 27
Advanced Attention – Generalize Relevance

This component determines how much each $h$ is correlated/associated with current context $s$.
Advanced Attention – Hard Attention

Sample a single encoding using probability $\alpha_i$
Transformers

- First pure attention-based model
- Self-attention
- No recurrence
- Evolution of GNMT
- SotA in NMT, NLP in general

Encoder-Decoder

- Encoder-decoder architecture
- 6 layer encoder
- 6 layer decoder

Advanced Attention – Self Attention

Each element of an input sequence $X_i$ projects into 3 vectors: query, key and value.

For each element, compute attention over all other vectors

$$SA(Q_i, K, V) = \sum_j \text{softmax}_j \left( \frac{Q_i \cdot K^T}{\sqrt{d_k}} \right) V_j$$

Vaswani et al., Attention Is All You Need, NIPS 2017
Self Attention – K,V,Q Generation

Figure credit to this article
Self Attention – Compute Attention Score

Self-attention

key
value

input #1

1 0 1 0

key
value

input #2

0 2 0 2

key
value

input #3

1 1 1 1
Self Attention – Produce Output
Self Attention – MultiHead

Strubell et al, Linguistically-Informed Self-Attention for Semantic Role Labeling, EMNLP 2018
Multi-Head Self-Attention - Equations

- Self-attention
- Scaled coefficients
- Multi-heads
- Large matrix multiplications
- Optimized for GPU

\[
\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V
\]

\[
\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O
\]

where \( \text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \)
Is self-attention a good mechanism to model temporal dependencies?

What happens if I randomly shuffle some tokens?
Positional Encoding

- Self-attention is order-independent
- But in sequences we need ordering information
- Word embedding + positional embedding

\[
PE(p, 2i) = \sin\left(\frac{p}{10000^{2i/d}}\right) \\
PE(p, 2i + 1) = \cos\left(\frac{p}{10000^{2i/d}}\right)
\]

Frequency

Embedding dimensions \( i \in \{1, d\} \)
Examples of Transformer

Github Copilot

```
function find(arr) {
    var sum = 0;
    var sum2 = 0;
    for (var i = 0; i < arr.length; i++) {
        sum += arr[i];
        sum2 += arr[i];
    }
    if (sum2 === 2020) {
        return sum * sum2;
    } else {
        return 0;
    }
}
```

GPT Series

https://play.aidungeon.io/

There is an ongoing debate in my family about the origin of the word “cowboy.” We know the term was originally used to describe a herder or drover of cattle. Some of my family maintains it was coined in Spain, describing the Mexican or Spanish version of those workers.

My father’s family, on the other hand, insists the term was used by Anglo cowboys to describe the tough but generally peaceful herders or drovers.

The debate came up again a few weeks ago when we went to watch “Lone Ranger” with my father and his wife, my stepmother, Barb. The man riding the white horse was a sheriff, not a cowboy, they insisted.

I finally admitted I didn’t really care where the term came from and we all had a good laugh over it. (from the GPT-3 repo; direct link)
Visualization of Attention

https://distill.pub/2016/augmented-rnns/#attentional-interfaces
Attention-Based Captioning – Focus Shifting

Xu et al, Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, ICML 2015
Attention-Based Captioning - Generation

Learns to correlate textual and visual concepts

Helps understanding why the model fails

Xu et al, Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, ICML 2015
Attention-Based Captioning – The Model

Encodings associated to $n$ image regions

From convolutional layers rather than from fully connected

Xu et al, Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, ICML 2015
The Vision Transformer (ViT)

A. Dosovitskiy et al, ICLR 2021
Take Home Messages

○ Attention.. Attention.. and, again, attention
  ● Soft attention is nice because makes everything fully differentiable
  ● Hard attention is stochastic hence cannot Backprop
  ● Empirical evidences of them being sensitive to different things

○ Encoder-Decoder scheme
  ● A general architecture to compose heterogeneous models and data
  ● Decoding allows sampling complex predictions from an encoding conditioned distribution