INF721 - Deep Learning
L16: Attention

Prof. Lucas N. Ferreira
Universidade Federal de Viçosa

2024/2

1 Introduction

Attention mechanisms were first introduced in the context of neural machine translation to address limitations of sequence-to-sequence (Seq2Seq) models. The key innovation was enabling the decoder to ”focus” on different parts of the input sequence when generating each output token, rather than relying solely on a fixed-size context vector.

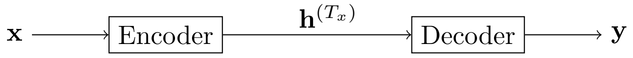
2 Machine Translation Background

Given an input sequence $\mathbf{x} = \{x^{(1)}, x^{(2)}, ..., x^{(T_x)}\}$ in one language, we want to generate an output sequence $\mathbf{y} = \{y^{(1)}, y^{(2)}, ..., y^{(T_y)}\}$ in another language. Key observations:

- Input and output sequences can have different lengths ($T_x \neq T_y$)
- The same input sentence can have multiple valid translations
- Word order may differ between languages

2.1 Traditional Seq2Seq Architecture

If we use a standard Seq2Seq model, the architecture consists of two Recurrent Neural Networks (RNNs) as shown in the Figure below:



- **Encoder**: Processes the input sequence $\mathbf{x}$ and produces a fixed-size context vector $\mathbf{h}^{(T_x)}$
- **Decoder**: Generates the output sequence $\mathbf{y}$ based on the context vector $\mathbf{h}^{(T_x)}$

3 Decoding Strategies

Given the encoder-decoder architecture, we need a strategy to generate the output sequence $\mathbf{y}$. Two common approaches are greedy search and beam search.

3.1 Greedy Search

The simplest decoding strategy that selects the most probable token at each step:

```
Initialize decoder state $\mathbf{s}^{(0)}$ with encoder final state
for t = 1 to T do
    Compute context vector $\mathbf{c}^{(t)}$ using attention
    Compute output probabilities $P(y^{(t)}|\mathbf{y}_{<t}, \mathbf{x})$
    Select $y^{(t)} = \arg\max_w P(w|\mathbf{y}_{<t}, \mathbf{x})$
end for
```

3.2 Beam Search

A more sophisticated strategy that maintains multiple hypothesis:

- Keeps track of b most probable partial translations at each step
- Expands each hypothesis with all possible next tokens
- Selects top b new hypotheses
- Provides better translations at the cost of increased computation

4 The Bottleneck Problem

A fundamental limitation of traditional Seq2Seq models, regardless of the decoding algorithm, is the information bottleneck: all information about the input sequence must be compressed into a fixed-size vector $\mathbf{h}^{(T_x)}$. This creates several challenges:

- Long-range dependencies are difficult to maintain
- Information from early parts of the sequence may be forgotten
- The model struggles with long sequences

5 Attention Mechanism

Instead of relying solely on the final hidden state, attention allows the decoder to:

- Look at all encoder hidden states $\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, ..., \mathbf{h}^{(T_x)}$ at each decoding step
- Assign different importance weights to different parts of the input sequence
- Dynamically focus on relevant information when generating each output token

5.1 Mathematical Formulation

At each decoding time step t, we compute:

$$\mathbf{c}^{(t)} = \sum_{t'=1}^{T_x} \alpha^{(t,t')} \mathbf{h}^{(t')} \tag{1}$$

where:

- $\mathbf{c}^{(t)}$ is the context vector at time t
- $\alpha^{(t,t')}$ are attention weights
- $\mathbf{h}^{(t')}$ are encoder hidden states

The attention weights are computed using:

$$\alpha^{(t,t')} = \frac{\exp(e^{(t,t')})}{\sum_{k=1}^{T_x} \exp(e^{(t,k)})} \tag{2}$$

where $e^{(t,t')}$ is the alignment score:

$$e^{(t,t')} = \tanh(\mathbf{W}_1 \mathbf{h}^{(t')} + \mathbf{W}_2 \mathbf{s}^{(t-1)}) \tag{3}$$

Here, $\mathbf{s}^{(t-1)}$ is the decoder's previous hidden state, and $\mathbf{W}_1$, $\mathbf{W}_2$ are learnable parameters.

After computing the context vector $\mathbf{c}^{(t)}$, we concatenate it with the embedded input token $y^{(t-1)}$ and feed it to the decoder RNN to generate the next token $y^{(t)}$. Formally, the decoding process of each token $y^{(t)}$ can be summarized as:

$$
\begin{aligned}
e^{(t,t')} &= \tanh(\mathbf{W}_1 \mathbf{h}^{(t')} + \mathbf{W}_2 \mathbf{s}^{(t-1)}) && (4) \\
\alpha^{(t,t')} &= \frac{\exp(e^{(t,t')})}{\sum_{k=1}^{T_x} \exp(e^{(t,k)})} && (5) \\
\mathbf{c}^{(t)} &= \sum_{t'=1}^{T_x} \alpha^{(t,t')} \mathbf{h}^{(t')} && (6) \\
\mathbf{s}^{(t)} &= tanh(\mathbf{W}_x [\mathbf{c}^{(t)}, \mathbf{y}^{(t-1)}] + \mathbf{W}_s \mathbf{s}^{(t-1)}) && (7) \\
P(y^{(t)} | \mathbf{y}_{<t}, \mathbf{x}) &= \text{softmax}(\mathbf{W}_y \mathbf{s}^{(t)}) && (8)
\end{aligned}
$$

6 Historical Impact

The introduction of attention mechanisms in 2014 by Bahdanau et al. marked a significant milestone in deep learning:

- Led to significant improvements in machine translation
- Inspired the development of the Transformer architecture
- Influenced modern language models (e.g., BERT, GPT)
- Extended to other domains like computer vision

7 Practical Considerations

7.1 Advantages

- Better handling of long sequences
- Improved gradient flow
- Interpretable attention weights
- Ability to capture long-range dependencies

7.2 Limitations

- Increased computational complexity
- Memory requirements scale with sequence length
- More complex training dynamics

8 Conclusion

Attention mechanisms have revolutionized the field of sequence modeling by enabling models to focus on relevant parts of the input sequence. They have become a fundamental building block in modern deep learning architectures and have led to significant improvements in various tasks, including machine translation, language modeling, and computer vision.

In the next lecture, we will discuss the Transformer architecture, which builds on the concept of attention to achieve state-of-the-art performance in a wide range of tasks.