High Integrity Systems Project: Time Series Analysis

Assignment 6

Hellyben Shah

December 3, 2024

What Is Attention? and How Does It Work?

Attention mechanisms in deep learning are inspired by human cognitive function, allowing models to focus on relevant parts of input data.

Rather than processing all inputs identically, this mechanism allows the model to pay different levels of attention to distinct bits of data. It's similar to how our brains prioritize particular elements when processing information, allowing the model to focus on what's important, making it tremendously strong for tasks like interpreting language or identifying patterns in photos.

Before starting to understand attention in detail, let's see some example of attention used with CNN for image captioning which was explained by Prof. Schäfer [1] and originally taken from the research paper [2].

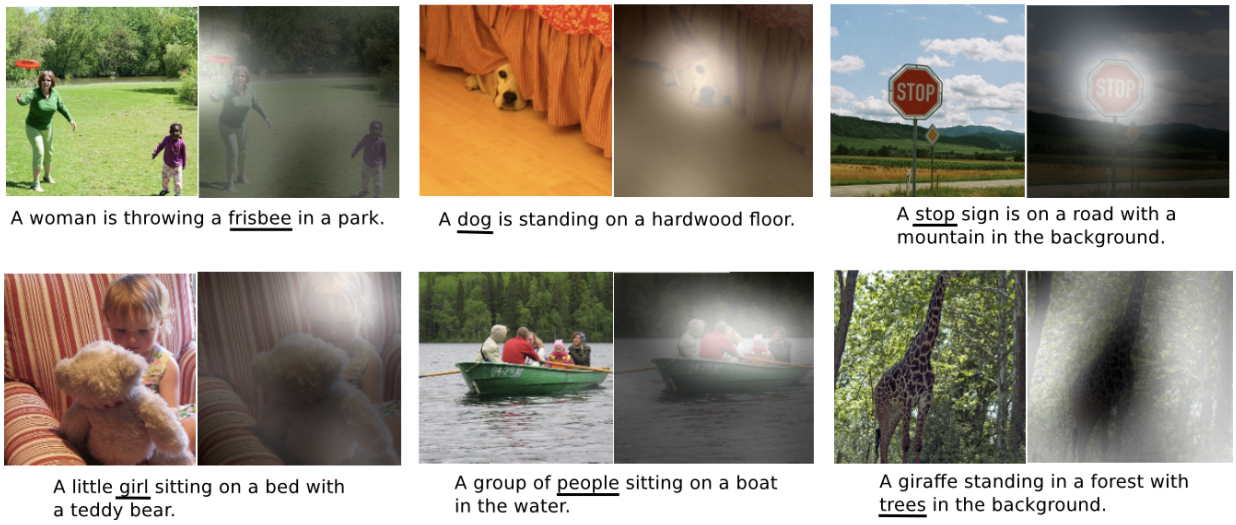

Figure 1: Attention Example [2]

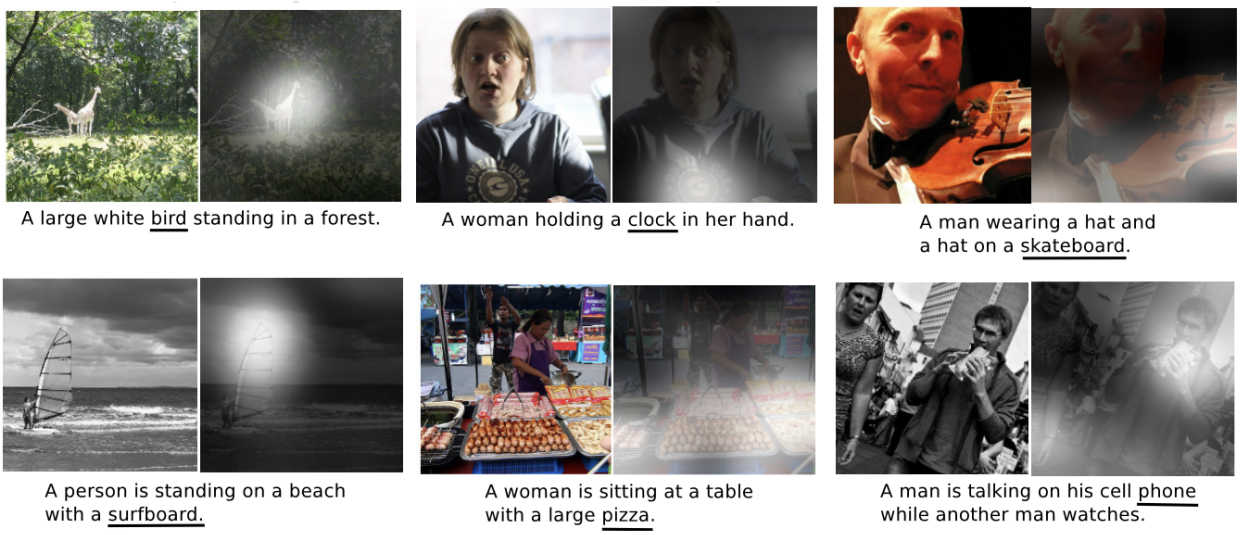

Figure 2: Attention Failure [2]

1 Motivation and Background

Attention mechanisms were introduced to address limitations in traditional sequence-to-sequence (Seq2Seq) models, particularly the information bottleneck created by encoding an entire input sequence into a fixed-length vector. This bottleneck was especially problematic for long sequences, where important information could be lost or diluted.

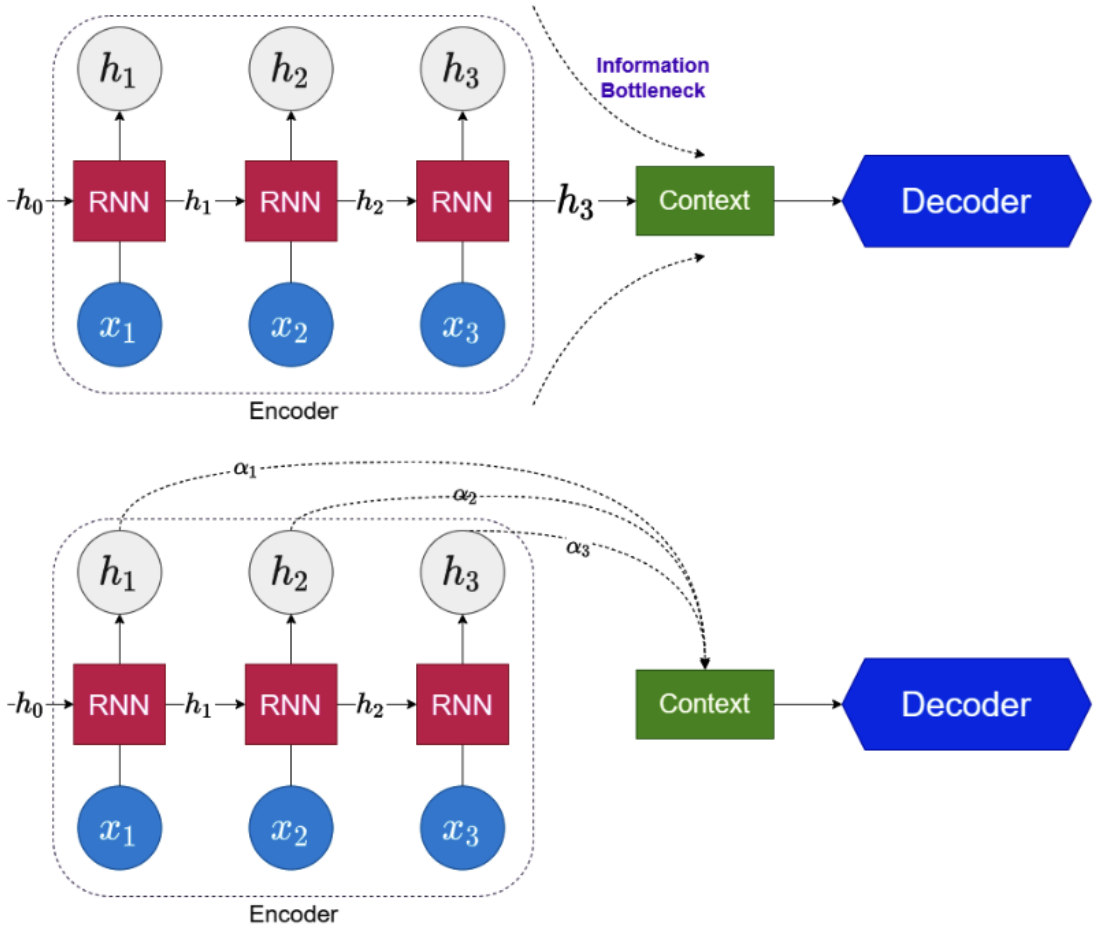

Figure 3: Seq2Seq vs Attention [3]

2 Mathematical Formulation

The generalized attention model can be expressed as:

$$A(q, K, V) = \sum p(a(k, q)) \cdot v_i$$

Where:

- q is the query vector
- K is the set of key vectors
- V is the set of value vectors

- a(k, q) is the alignment function
- p is the distribution function

3 Core Components of Attention

3.1 Query, Key, and Value Vectors Calculation

- **Query (Q)**: Represents the current decoder state or what we're looking for.
- **Key (K)**: Represents the encoder states, used to match against the query.
- **Value (V)**: The actual content of the encoder states, used to compute the context.

These vectors are typically created by linear transformations of the input:

$$Q = X \cdot W_q$$

$$K = X \cdot W_k$$

$$V = X \cdot W_v$$

Where X is the input, and W_q, W_k, W_v are learnable weight matrices.

3.2 Alignment Score Calculation with Functions

The alignment function computes a similarity score between the query and each key. Common alignment functions include:

1. **Dot Product**:
$$a(k_i, q) = q^T \cdot k_i$$
2. **Scaled Dot Product**:
$$a(k_i, q) = \frac{1}{\sqrt{d_k}} q^T \cdot k_i$$
Where d_k is the dimension of the key vectors.
3. **General Attention**:
$$a(k_i, q) = q^T \cdot W \cdot k_i$$
Where W is a learnable weight matrix.
4. **Additive/Concat Attention**:
$$a(k_i, q) = v_{imp}^T \cdot \tanh(W_q \cdot q^T + W_k \cdot k_i + b)$$
Where v_{imp}, W_q, W_k, and b are learnable parameters.

The various alignment functions used in attention mechanisms have different advantages and disadvantages.

1. **Dot Product Attention**: Simple and computationally efficient.
2. **Scaled Dot Product Attention**: Addresses the issue of vanishing gradients in large input dimensions.
3. **General Attention**: Introduces a learnable weight matrix for more flexibility.
4. **Additive/Concat Attention**: Allows for learning of the similarity function.

3.3 Distribution Functions

The distribution function converts alignment scores into attention weights. The most common is the softmax function:

$$\alpha_i = \frac{\exp(e_i)}{\sum_j \exp(e_j)}$$

Where e_i are the alignment scores.

4 Detailed Attention Process

1. **Input Processing**:
   - Encoder processes the input sequence, generating hidden states $(h_1, h_2, ..., h_T)$. previously denoted with k_i
   - Decoder generates a query vector q at each decoding step.
2. **Alignment Score Calculation**:
   - For each encoder hidden state h_i and the current query q, compute the alignment score $e_i = a(h_i, q)$.
   - This results in a vector of alignment scores $[e_1, e_2, ..., e_T]$.
3. **Attention Weight Calculation**:
   - Apply the distribution function (e.g., softmax) to the alignment scores:

     $$\alpha_i = \text{softmax}(e_i)$$

   - This produces attention weights $[\alpha_1, \alpha_2, ..., \alpha_T]$ that sum to 1.
4. **Context Vector Computation**:
   - Compute the weighted sum of encoder hidden states:

     $$c = \sum_i \alpha_i \cdot h_i$$

5. **Output Generation**:

- Combine the context vector c with the current decoder state s_t:

$$\hat{y}_t = f(c, s_t)$$

Where f is typically a feed-forward neural network.

5 Types of Attention Mechanisms

5.1 Global vs. Local Attention

- **Global Attention**: Considers all encoder hidden states for each decoding step.
- **Local Attention**: Focuses on a subset of encoder states, reducing computational complexity.

5.2 Self-Attention

- Used in models like Transformers.
- Allows each position in a sequence to attend to all positions in the same sequence.
- Computed as: $\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}}) \cdot V$

5.3 Multi-Head Attention

- Extends self-attention by applying multiple attention "heads" in parallel.
- Each head can focus on different aspects of the input.
- Computed as: $\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h) \cdot W_o$ Where each $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$

6 Implementation Considerations

- **Masking**: Used in self-attention to prevent positions from attending to subsequent positions (for tasks like language modeling).
- **Positional Encoding**: Added to input embeddings to retain sequence order information in models like Transformers.
- **Attention Visualization**: Attention weights can be visualized to interpret model decisions.

7 Advantages & Disadvantages[4]

Advantages

1. **Selective Information Processing:** Focuses on important parts of input, improving pattern recognition.

2. **Improved Model Interpretability:** Reveals which input elements are relevant for predictions.
3. **Capturing Long-Range Dependencies:** Connects distant elements in sequential data.
4. **Transfer Learning Capabilities:** Enhances adaptability across different tasks and domains.
5. **Efficient Information Processing:** Enables selective processing, improving scalability and efficiency.

Disadvantages

1. **Computational Complexity:** Increases resource requirements, especially for long sequences.
2. **Dependency on Model Architecture:** Effectiveness varies based on overall model design and task.
3. **Overfitting Risks:** Can lead to memorization instead of generalization, particularly with many attention heads.
4. **Attention to Noise:** May focus on irrelevant or noisy parts of input, requiring careful tuning.

8 Integration of attention mechanisms into sequence-to-sequence (Seq2Seq) models for forecasting tasks

8.1 Key Differences

1. **Decoder Design:** Unlike traditional Seq2Seq models that use a fully connected layer as the decoder, the attention-based model uses attention to improve the decoding process.
2. **Attention Types:** The model supports different types of attention, such as those proposed by Luong et al. and Bahdanau et al., which differ in how they incorporate context vectors during decoding.

8.2 Attention Mechanisms

Luong's Approach

- Uses the current decoder hidden state to compute similarity with all encoder hidden states.
- The context vector is concatenated with the decoder hidden state before passing through a linear layer.

Bahdanau's Approach

- Uses the previous decoder hidden state to compute similarity with encoder hidden states.
- The context vector is concatenated with the input to the current decoding step.

References

[1] J. Schäfer, "Neural networks – iii," https://campuas.frankfurt-university.de/pluginfile.php/303826/mod_folder/content/0/LfD12.pdf?forcedownload=1, 2024, department of Computer Sciences, Summer Semester 2024.

[2] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," *CoRR*, vol. abs/1502.03044, 2015. [Online]. Available: http://arxiv.org/abs/1502.03044

[3] M. Joseph, *Modern Time Series Forecasting with Python: Explore industry-ready time series forecasting using modern machine learning and deep learning.* Packt Publishing Ltd, 2022.

[4] FreeCodeCamp. (2024) What are attention mechanisms in deep learning? [Online; accessed 3-December-2024]. [Online]. Available: https://www.freecodecamp.org/news/what-are-attention-mechanisms-in-deep-learning/