Attention Mechanism in Image Captioning: Show, Attend, and Tell paper explained

Contents

1 Introduction

This document explains the attention mechanism in image captioning, combining mathematical rigor and step-by-step examples. Attention allows a model to focus on the most relevant parts of an image while generating captions. We will define key symbols, explain the mathematical formulations, and provide a worked-out example.

—

2 Symbol Definitions

- $\mathbf{a} = \{\mathbf{a}_1, \ldots, \mathbf{a}_L\}$: A set (or list) of **image feature vectors** extracted from a CNN (Convolutional Neural Network). Each $\mathbf{a}_i \in \mathbb{R}^D$ represents a patch (region) of the image.
  - L: Total number of patches (e.g., $14 \times 14 = 196$).
  - D: Dimensionality of each feature vector (e.g., 512 or 1024).
- $e_{t,i}$: **Attention energy** for patch i at time t. A scalar score indicating how relevant patch i is for predicting the next word at time t.
- $\alpha_{t,i}$: **Attention weight** for patch i at time t. A scalar between 0 and 1 (after applying softmax) that indicates the focus on patch i when generating the next word.
- $\mathbf{h}_t$: The **hidden state** of the LSTM at time t. It represents the memory and context of what has been generated so far.
- f_{att}: A small neural network (often an MLP) that computes $e_{t,i}$ from $\mathbf{a}_i$ and $\mathbf{h}_{t-1}$.
- $\mathbf{z}_t$: The **context vector** at time t, a weighted combination of the image features $\mathbf{a}_i$ using the attention weights $\alpha_{t,i}$.
- $\mathbf{x}_t$: The **embedding** of the previously generated word at time t, fed into the LSTM along with $\mathbf{z}_t$.
- $p(y_t|y_{1:t-1}, \mathbf{a})$: The probability of predicting the next word y_t given all previous words $y_{1:t-1}$ and the image features $\mathbf{a}$.
- W_o, b_o: Trainable parameters (weight matrix and bias vector) used to transform $\mathbf{h}_t$ into a vocabulary-sized vector for the softmax output layer.
- θ: All **trainable parameters** in the model, including CNN parameters, LSTM parameters, and attention MLP parameters.

—

3 Mathematical Details

The attention mechanism is a series of steps that calculate attention scores, convert them into probabilities, and use these probabilities to form a weighted context vector. These steps are as follows:

3.1 Step 1: Compute Attention Energies

$$e_{t,i} = f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1})$$

Here, f_{att} is typically defined as:

$$e_{t,i} = \mathbf{v}_a^\top \tanh(W_a \mathbf{a}_i + W_h \mathbf{h}_{t-1}),$$

where:

- $W_a \in \mathbb{R}^{d_h \times D}$ and $W_h \in \mathbb{R}^{d_h \times d_h}$ are weight matrices.
- tanh introduces non-linearity.
- $\mathbf{v}_a \in \mathbb{R}^{d_h}$ is a learnable vector that projects the transformed features to a scalar.

3.2 Step 2: Compute Attention Weights

Attention energies $e_{t,i}$ are converted into probabilities using the softmax function:

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{k=1}^{L} \exp(e_{t,k})}.$$

The weights $\alpha_{t,i}$ represent how much focus is placed on each patch.

3.3 Step 3: Form the Context Vector

The context vector $\mathbf{z}_t$ is a weighted combination of the image feature vectors:

$$\mathbf{z}_t = \sum_{i=1}^{L} \alpha_{t,i}\, \mathbf{a}_i.$$

3.4 Step 4: Update the LSTM State and Predict the Next Word

The LSTM state is updated using the context vector $\mathbf{z}_t$, the embedding of the previous word $\mathbf{x}_t$, and the previous hidden state $\mathbf{h}_{t-1}$:

$$\mathbf{h}_t = \text{LSTM}(\mathbf{h}_{t-1}, \mathbf{x}_t, \mathbf{z}_t).$$

The probability of the next word is computed as:

$$p(y_t \mid y_{1:t-1}, \mathbf{a}) = \text{Softmax}(W_o\, \mathbf{h}_t + b_o).$$

—

4 Worked-Out Example: Describing an Image

Let's describe an image that shows *"a cat sitting on a couch"*.

4.1 Image Features

Suppose the image is divided into $L = 4$ patches ($L = 4$ for simplicity). The CNN extracts feature vectors $\mathbf{a} = \{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4\}$, where each $\mathbf{a}_i \in \mathbb{R}^3$ (e.g., $D = 3$).

Example values for $\mathbf{a}_i$:

$$\mathbf{a}_1 = [1, 0, 2], \quad \mathbf{a}_2 = [0, 2, 1], \quad \mathbf{a}_3 = [3, 1, 0], \quad \mathbf{a}_4 = [1, 1, 1].$$

4.2 At $t = 1$: Generating the Word "a"

- Compute $e_{t,i}$ using $f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1})$:

$$e_{t,i} = \mathbf{v}_a^\top \tanh(W_a \mathbf{a}_i + W_h \mathbf{h}_{t-1}).$$

Assume the following:

$$W_a = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 1 \end{bmatrix}, \quad W_h = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{v}_a = \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

- Apply softmax to calculate $\alpha_{t,i}$.
- Compute $\mathbf{z}_t = \sum_{i=1}^{L} \alpha_{t,i} \mathbf{a}_i$.

4.3 At $t = 2$: Generating the Word "cat"

Repeat the same steps, focusing more on patches corresponding to the cat.

—

5 Why Attention Helps

The attention mechanism enables the model to focus dynamically on relevant image regions for each word, improving caption accuracy and interpretability.

—