Caption generation is a challenging artificial intelligence problem that draws on both computer vision and natural language processing.

The encoder-decoder recurrent neural network architecture has been shown to be effective at this problem. The implementation of this architecture can be distilled into inject and merge based models, and both make different assumptions about the role of the recurrent neural network in addressing the problem.

In this post, you will discover the inject and merge architectures for the encoder-decoder recurrent neural network models on caption generation.

After reading this post, you will know:

- The challenge of caption generation and the use of the encoder-decoder architecture.
- The inject model that combines the encoded image with each word to generate the next word in the caption.
- The merge model that separately encodes the image and description which are decoded in order to generate the next word in the caption.

Let’s get started.

Caption Generation with the Inject and Merge Architectures for the Encoder-Decoder Model
Photo by Bernard Spragg. NZ, some rights reserved.

Image Caption Generation

The problem of image caption generation involves outputting a readable and concise description of the contents of a photograph.

It is a challenging artificial intelligence problem as it requires both techniques from computer vision to interpret the contents of the photograph and techniques from natural language processing to generate the textual description.

Recently, deep learning methods have achieved state-of-the-art results on this challenging problem. The results are so impressive that this problem has become a standard demonstration problem for the capabilities of deep learning.

Encoder-Decoder Architecture

A standard encoder-decoder recurrent neural network architecture is used to address the image caption generation problem.

This involves two elements:

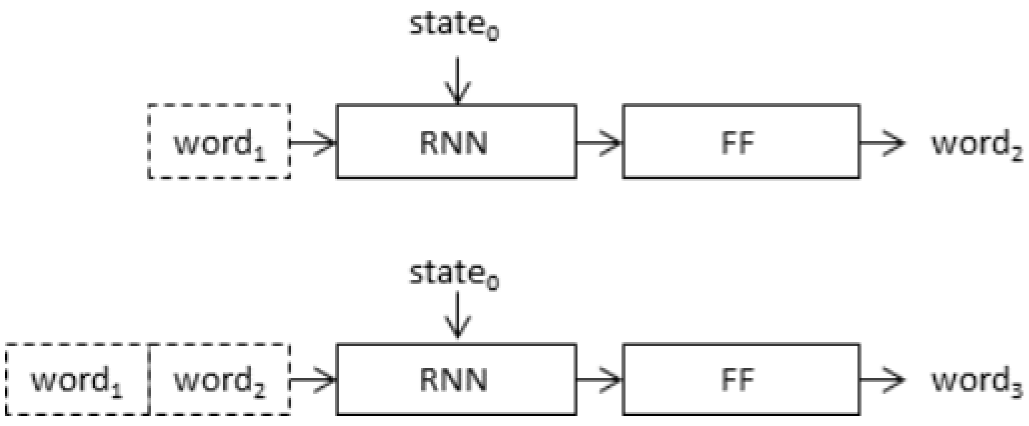
1. **Encoder**: A network model that reads the photograph input and encodes the content into a fixed-length vector using an internal representation.
2. **Decoder**: A network model that reads the encoded photograph and generates the textual description output.

Generally, a convolutional neural network is used to encode the images and a recurrent neural network, such as a Long Short-Term Memory network, is used to either encode the text sequence generated so far, and/or generate the next word in the sequence.

There are many ways to realize this architecture for the problem of caption generation.

It is common to use a pre-trained convolutional neural network model trained on a challenging photograph classification problem to encode the photograph. The pre-trained model can be loaded, the output of the model removed, and the internal representation of the photograph used as the encoding or internal representation of the input image.

It is also common to frame the problem such that the model generates one word of the output textual description, given both the photograph and the description generated so far as input. In this framing, the model is called recursively until the entire output sequence is generated.



Recursive Framing of the Caption Generation Model
Taken from “Where to put the Image in an Image Caption Generator.”

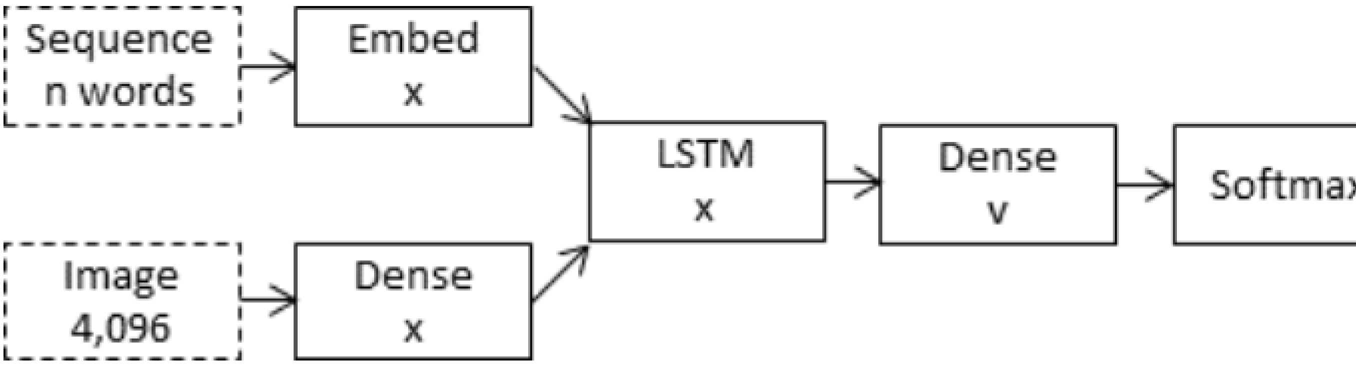
This framing can be implemented using one of two architectures, called by Marc Tanti, et al. as the inject and the merge models.

Inject Model

The inject model combines the encoded form of the image with each word from the text description generated so-far.

The approach uses the recurrent neural network as a text generation model that uses a sequence of both image and word information as input in order to generate the next word in the sequence.

> In these ‘inject’ architectures, the image vector (usually derived from the activation values of a hidden layer in a convolutional neural network) is injected into the RNN, for example by treating the image vector on a par with a ‘word’ and including it as part of the caption prefix.
>
> — Where to put the Image in an Image Caption Generator, 2017.



Inject Architecture for Encoder-Decoder Model
Taken from “What is the Role of Recurrent Neural Networks (RNNs) in an Image Caption Generator?”.

This model combines the concerns of the image with each input word, requiring the encoder to develop an encoding that incorporates both visual and linguistic information together.

In an inject model, the RNN is trained to predict sequences based on histories consisting of both linguistic and perceptual features. Hence, in this model, the RNN is primarily responsible for image-conditioned language generation.

Merge Model

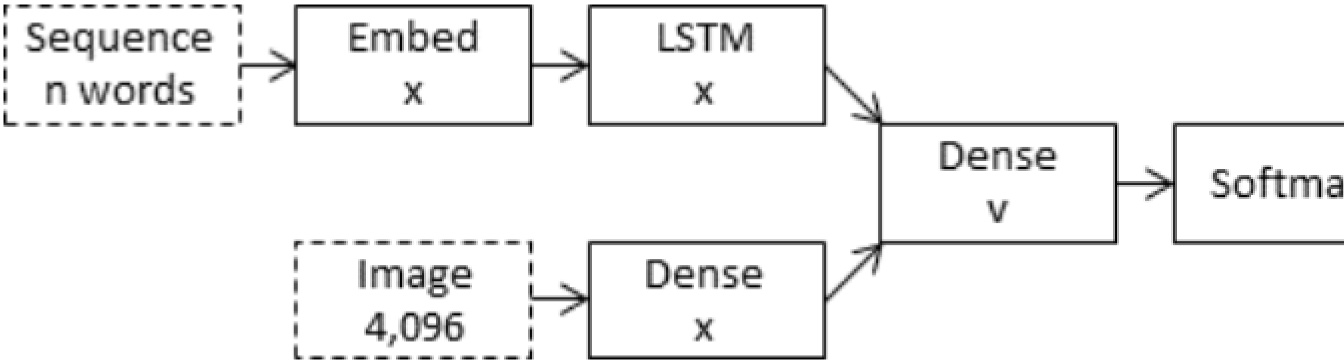
The merge model combines both the encoded form of the image input with the encoded form of the text description generated so far.

The combination of these two encoded inputs is then used by a very simple decoder model to generate the next word in the sequence.

The approach uses the recurrent neural network only to encode the text generated so far.

In the case of ‘merge’ architectures, the image is left out of the RNN subnetwork, such that the RNN handles only the caption prefix, that is, handles only purely linguistic information. After the prefix has been vectorised, the image vector is then merged with the prefix vector in a separate ‘multimodal layer’ which comes after the RNN subnetwork

— Where to put the Image in an Image Caption Generator, 2017.



Merge Architecture for Encoder-Decoder Model
Taken from “What is the Role of Recurrent Neural Networks (RNNs) in an Image Caption Generator?”.

This separates the concern of modeling the image input, the text input and the combining and interpretation of the encoded inputs.

As mentioned, it is common to use a pre-trained model for encoding the image, but similarly, this architecture also permits a pre-trained language model to be used to encode the caption text input.

… in the merge architecture, RNNs in effect encode linguistic representations, which themselves constitute the input to a later prediction stage that comes after a multimodal layer. It is only at this late stage that image features are used to condition predictions

— What is the Role of Recurrent Neural Networks (RNNs) in an Image Caption Generator?, 2017.

There are multiple ways to combine the two encoded inputs, such as concatenation, multiplication, and addition, although experiments by Marc Tanti, et al. have shown addition to work better.

Generally, Marc Tanti, et al. found the merge architecture to be more effective compared to the inject approach.

Overall, the evidence suggests that delaying the merging of image features with linguistic encodings to a late stage in the architecture may be advantageous […] results suggest that a merge architecture has a higher capacity than an inject architecture and can generate better quality captions with smaller layers.

— What is the Role of Recurrent Neural Networks (RNNs) in an Image Caption Generator?, 2017.

More on the Merge Model

The success of the merge model for the encoder-decoder architecture suggests that the role of the recurrent neural network is to encode input rather than generate output.

This is a departure from the common understanding where it is believed that the contribution of the recurrent neural network is that of a generative model.

If the RNN had the primary role of generating captions, then it would need to have access to the image in order to know what to generate. This does not seem to be the case as including the image into the RNN is not generally beneficial to its performance as a caption generator.

— What is the Role of Recurrent Neural Networks (RNNs) in an Image Caption Generator?, 2017.

The explicit comparison of the inject and merge models, and the success of merge over inject for caption generation, raises the question of whether this approach translates to related sequence-to-sequence generation problems.

Instead of pre-trained models used to encode images, pre-trained language models could be used to encode source text in problems such as text summarization, question answering, and machine translation.

We would like to investigate whether similar changes in architecture would work in sequence-to-sequence tasks such as machine translation, where instead of conditioning a language model on an image we are conditioning a target language model on sentences in a source language.

— What is the Role of Recurrent Neural Networks (RNNs) in an Image Caption Generator?, 2017.