**Google's Neural Translation Machine Project (Residency Day 3)**

Abuchi Godswill Okeke

School of Computer and Information Sciences, University of the Cumberlands

MSAI532 - M51 Natural Language Processing

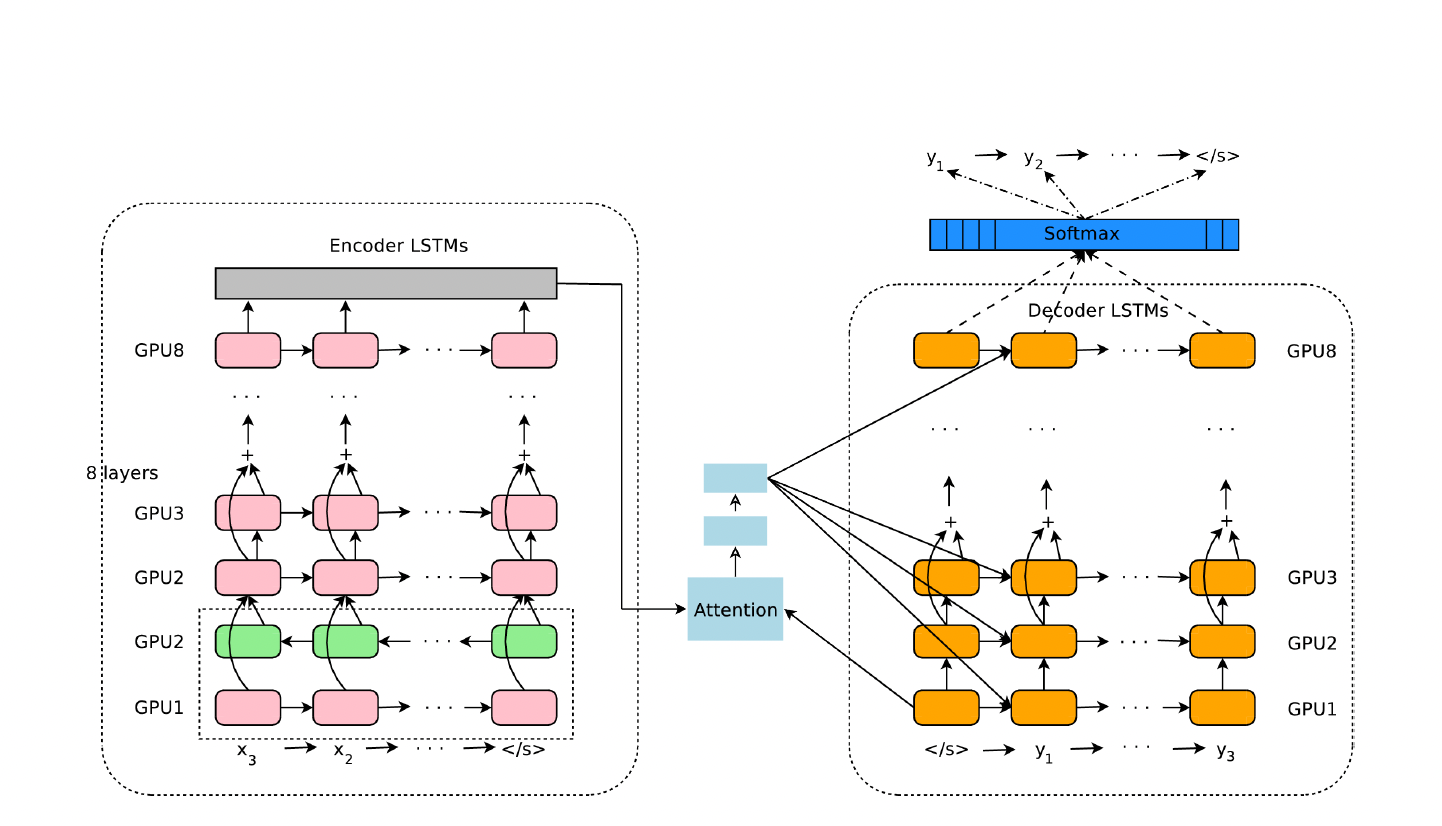
Dr. Toni Farley

Feb 02, 2025

Describing Google’s Neural Machine Translation (GNMT) system model architecture presented in Figure 1.0 below using an example translation process of “I am studying an AI course.” from English sentence to French.

**Figure 1.0**

*Google’s Neural Machine Translation system model architecture*

****

**1. Left encoder network**

Google’s Neural Machine Translation (GNMT) system’s first step is the left encoder network. Its job is to understand the input sentence before passing it to the decoder for translation (Wu et al., 2016).

***What are the GPUs layers about: explain inputs/outputs***

The GPU layers in GNMT help speed up translation by breaking down the work across multiple GPUs. Instead of running everything on one machine, the model spreads out different parts of the processing, making it faster and more efficient (Wu et al., 2016).

**Example explaining inputs, encoder, middle attention, decoder output processes, and softmax steps**

**Step 1: Input words**

**The input is** a sequence of source language tokens, represented as embedding vectors.

For example, when you enter the sentence "I am studying an AI course.", the system splits it into smaller pieces among the 8 GPUs:

“I” → “am” → “stud” → “ying” → “an” → “AI” → “course” → “.”

Then each of these word pieces is **turned into numbers** (a vector of 1024 values) understood by the computers for the encoding process.

**Example (word piece** → **vectors**)**:**

"I" → [0.12, 0.34, ..., 0.89]

"am" → [0.23, 0.78, ..., 0.45]

"stud" → [0.56, 0.67, ..., 0.92]

"ying" → [0.44, 0.55, ..., 0.83]

"an" → [0.77, 0.36, ..., 0.21]

"AI" → [0.34, 0.19, ..., 0.90]

"course" → [0.58, 0.72, ..., 0.65]

"." → [0.11, 0.99, ..., 0.08]

***What is Encoder LSTM layer about***

The Encoder LSTM (Long Short-Term Memory) layer is about being responsible for ensuring that the input sentence is processed and converted into a numerical representation that captures meaning and context. Since GNMT uses stacked LSTMs (8 layers deep), each word piece passes through multiple layers of processing, with each layer refining the sentence's representation (Wu et al., 2016).

**Step 2: Encoder Processing on GPUs**

The encoder consists of 8 layers of LSTMs, each running on a separate GPU.

* GPU 1: Processes the first LSTM layer, learning initial patterns.
* GPU 2: Takes output from GPU 1, refining understanding of sentence structure.
* GPU 3–7: Each layer adds more depth, improving how the model understands meaning.
* GPU 8: Produces the final encoded version of the sentence.

**Example: First LSTM Layer (GPU 1 - Bidirectional)**

* This layer reads the sentence forward and backward to capture context from both directions.

Then, For "I" after the first layer:

Forward: [0.58, 0.32, ..., 0.91]

Backward: [0.44, 0.76, ..., 0.39]

Final Output: [0.51, 0.54, ..., 0.65] (1024-dimensional)

The **output of this layer** is passed to **the next LSTM layer.**

**Second LSTM Layer (GPU 2)**

* This layer refines the sentence representation by learning deeper patterns.

Input: [0.51, 0.54, ..., 0.65]

Output: [0.72, 0.28, ..., 0.85]

**Final LSTM Layer (GPU 8)**

After passing through all 8 layers, the final encoder output represents the sentence in a highly meaningful way.

Example final encoding for "I":

[0.91, 0.64, ..., 0.77]

Each word is processed similarly, with context from the whole sentence influencing its final encoding. This final representation is then passed to the attention mechanism (middle attention module).

**2. Middle Attention Module**

The attention module in Google’s Neural Machine Translation (GNMT) system helps the model focus on the most relevant words while translating a sentence. Instead of treating the entire input as a single block of information, attention allows the model to dynamically shift focus to different words at each step of the translation process (Bahdanau et al., 2014).

**Step 3: Attention Mechanism**

Continuing with the example "I am studying an AI course.",

Before translating, the attention mechanism helps decide which words are important at each step. Translating into French, the model focuses on "AI course" when translating "studying." If the French translation requires a different word order, attention adjusts the focus dynamically (Wu et al., 2016).

At each step, the model assigns **importance scores** to different words, making sure the right words contribute more to the translation. For instance, when translating "studying", the attention might distribute as follows:

"stud" → 70% focus

"ying" → 20% focus

"course" → 10% focus

These probabilities ensure that the right words influence the translation at the right time, making the output more natural.

**3. Right Decoder Network**

The GNMT right decoder network system is responsible for generating the translated sentence step by step. It works closely with the attention module, which helps it focus on relevant words from the encoder’s output (Wu et al., 2016).

***Explain Softmax with an example***

Continuing with the example “I am studying an AI course.”,

The **Softmax function** is the final step of the decoder. It converts the model’s predictions into probabilities, allowing it to pick the most likely next word.

**Final Step: Softmax Prediction**

After processing the sentence, the decoder is at the stage where it needs to predict the first word. The model generates scores for possible words:

"Je" → Score: 2.8

"Moi" → Score: 1.3

"Il" → Score: 0.6

Then, at every step, the model uses a Softmax function to pick the best possible next word. For “I am” → “Je suis” it might consider: “Je” with 85% probability, “Moi” with 10% probability, and “Il” with 5% probability. Since “Je” has the highest probability, the model chooses it as the first word of the translation. This process repeats for each word until the entire sentence is generated.

***What is Decoder LSTM layer about***

The decoder LSTM works like a language model that generates words one at a time. At each step, it takes the previous word’s embedding (starting with a special <SOS> token), the context vector from the attention module, and the hidden state from the previous LSTM step.

Then, it processes this information and produces a new hidden state, which helps predict the next word (Sutskever et al., 2014).

**Step 4: Decoder Processing/Output translation**

Continuing with the example “I am studying an AI course.”, this step is performed and then scores are passed down to Softmax prediction for probability assignment and word selection.

The decoder takes the encoded sentence and predicts the translation one word at a time. Each decoder layer runs on a separate GPU, just like the encoder.

1. First GPU (Layer 1) takes the start token (<SOS>) and begins generating. Then predicts "Je" for the French translation.
2. The second GPU (Layer 2) takes "Je" and refines the sentence structure. Then predicts "suis" (matching the tense of "am studying").
3. By the last GPU (Layer 8), the full sentence might be predicted as: "Je suis en train d’étudier un cours d’IA."

Each word is predicted step by step, using previous outputs + context from attention. These steps run through 8 layers of LSTMs, with each layer refining the prediction further (Bahdanau et al., 2014). The Decoder is important because it generates fluent sentences instead of translating word-for-word, relies on attention to stay focused on the correct words, and the stacked LSTMs refine each prediction, making translations more accurate. The Softmax function ensures the best word is chosen at every step, leading to a smooth and natural translation.

In summary, Google's Neural Machine Translation (GNMT) architecture system works like a well-coordinated team of networks, breaking down the translation process into three key parts: the encoder, attention module, and decoder. The encoder LSTM layers, running on multiple GPUs, help the model understand the meaning of the input sentence by converting words into numbers. The attention module ensures that the model focuses on the right words at the right time, rather than translating blindly. Finally, the decoder LSTM generates the translated sentence step by step, using Softmax to choose the best words. This entire process makes translations faster, smoother, and more accurate, producing natural and fluent results instead of awkward, word-for-word conversions.

**References**

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. Advances in Neural Information Processing Systems, 27, 3104-3112.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., ... & Dean, J. (2016). Google’s neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144.