Neural Networks and Language Models

Revisiting ELIZA and Symbolic AI

In the history of Natural Language Processing (NLP), neural networks have become prominent only fairly recently, even though they were developed around the same time as ELIZA. ELIZA is an example of an attempt at building what is known as "*symbolic AI*". The field of symbolic AI attempted to build rule-based systems grounded in mathematical logic so as to capture conscious thought processes. ELIZA, for instance, responded to any question by drawing from a long list of explicit rules on how to respond — a semi-scripted conversation with many different endpoints.

Neural Networks and Subsymbolic AI

Neuroscience developments in the 1950's suggested that *unconscious processes* play an important role in our brains, informing human perception and cognition, and that this cannot be captured by explicit rule-making. Computer scientists who drew inspiration from this came up with the idea of "*subsymbolic AI*" which seeks to capture these underlying unconscious processes.

In 1958, the psychologist, Frank Rosenblatt, built the "*perceptron*" — a simple program meant to simulate the function of an individual information processing unit within the brain, the neuron. The idea was that by layering networks of these perceptrons, a form of intelligent information processing that brains perform would emerge in these **neural networks**.

Deep Learning

Neural networks went in and out of favor multiple times among AI researchers for nearly fifty years until we arrived at the age of data in the early 2000s. The growth of the internet meant that more data and computing power were available than ever before and neural networks saw a resurgence through the field of **machine learning**.

Machine learning refers to a class of algorithms that learn patterns from vast amounts of data to perform tasks like prediction, inference and generation. Neural networks can be shallow (few layers) or deep (many layers). **Deep learning** is a subset of machine learning methods that use many-layered or deep neural networks.

Birth of the Transformer

There are many different types of neural networks and these networks can be arranged in specific ways, to excel at specific tasks. These different arrangements are often referred to as “neural network architectures”. Convolutional Neural Networks (CNNs) are most commonly used for image data. Recurrent Neural Networks (RNNs) work well with sequential data (like language!) They’re used extensively in NLP tasks like translation and speech recognition since the mid 2000s. The image to the right shows the workflow of a RNN translating a Hindi sentence to English.

In 2014, a specific type of architecture, known as the *transformer* was shown to perform exceptionally well in language-related tasks. Generative Pre-trained Transformers or GPTs are a class of Large Language Models (LLMs) that employ transformers to learn from vast text corpuses to be able to generate coherent “human-sounding” text.

How transformers capture semantic patterns within text is beyond the scope of this lesson, but we can nonetheless understand many important ideas about how LLMs work, their possibilities and limitations based on what we’ve covered in the lesson so far. Let’s dive into these next!

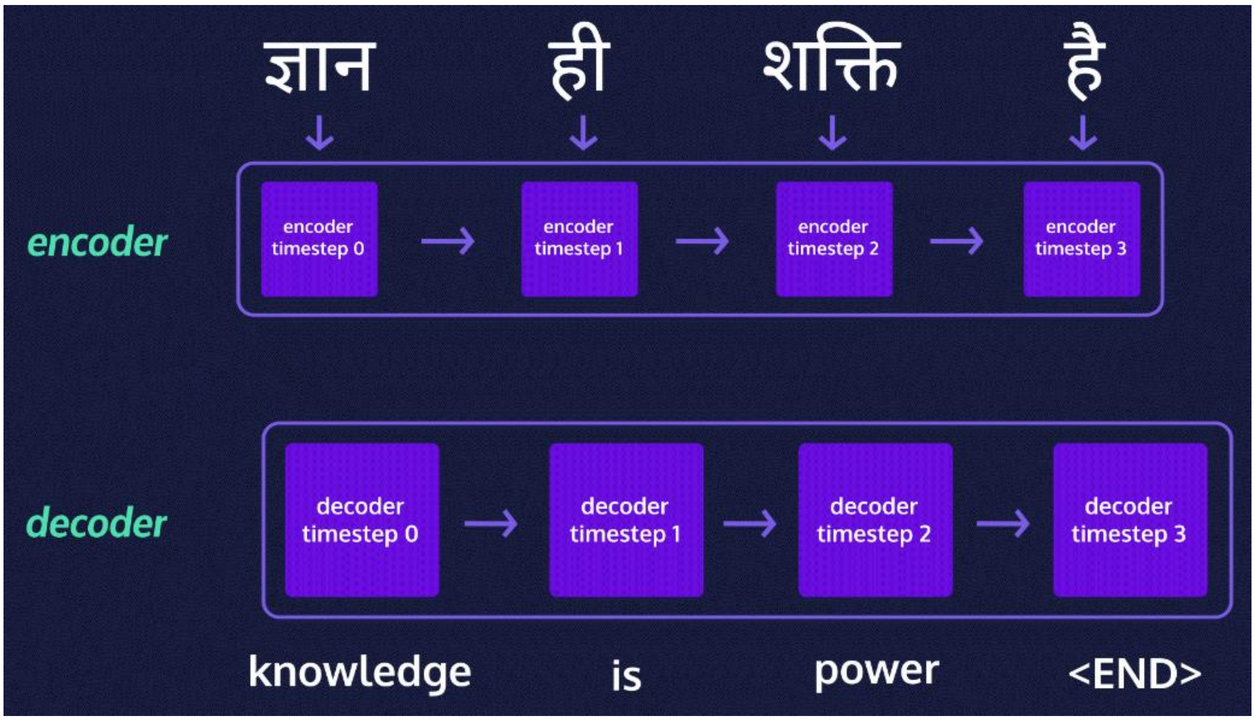
Instructions

The language translation task to the right is typically performed by a type of RNN known as a sequence-to-sequence (referred to as seq-to-seq in short) model. And it is exactly as it sounds - it’s designed to take an input sequence and convert it to an output sequence!

These models are *built on top of a language model* by adding an encoder and decoder step:

- The encoder that maps the Hindi text to a mathematical representation, one word at a time.
- A decoder that maps the mathematical representation to English text.

It’s crucial that the mathematical representation is able to retain the meaning and context of the words as much as possible. Transformer-based language models far outperform previous language models in this regard.


ज्ञान
ही
शक्ति
है
encoder
encoder timestep 0
encoder timestep 1
encoder timestep 2
encoder timestep 3
decoder
decoder timestep 0
decoder timestep 1
decoder timestep 2
decoder timestep 3
knowledge
is
power
<END>