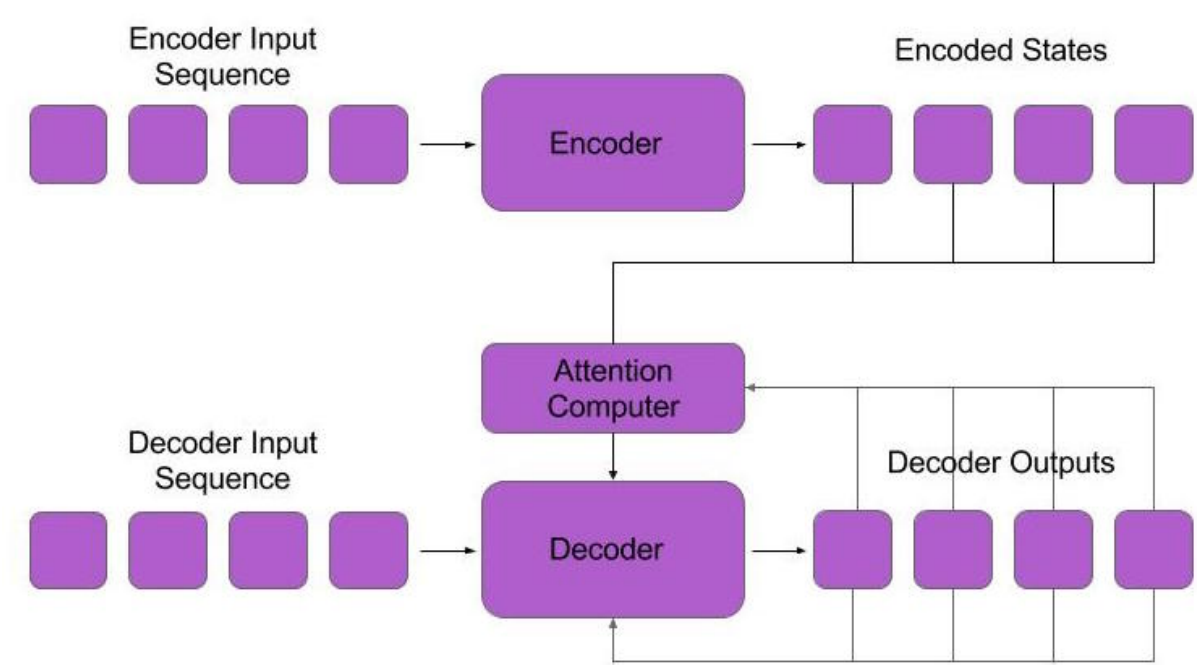


Sequence Modeling with Neural Networks (Part 2): Attention Models

APRIL 18, 2016 / MACHINE LEARNING



Welcome back to our two part series on sequence to sequence models. In the previous post [we saw how language models and sequence to sequence models can be used to handle data that varies over time](#). In this post, we will see how an attention mechanism can be added to the sequence to sequence model to improve its sequence modeling capacity.

What is Attention?

Let’s start things off by thinking about attention in the real world. Every day, humans are bombarded with sensory inputs. Amazingly, our brains are able to reduce the overwhelming amount of signal into useful information, which we then use to make decisions. Recent research has shown that the same processes that allow us to focus on important information while filtering out unnecessary data can be applied to neural networks. This technique, commonly referred to as “attention”, helps us build neural networks that can effectively tackle challenging sequence processing tasks, such as language translation, where simple sequence to sequence models fail.

Implementing Attention

The sequence to sequence model gives us the ability to process input and output sequences. Unfortunately, compressing an entire input sequence into a single fixed vector tends to be quite challenging. This would be like trying to figure out what’s for dinner after smelling all the food at once. You might be able to pick out some dominant scents, but odds are, you’ll be unable to identify the whole dinner menu without smelling each of the foods individually. Moreover, the last state of the encoder contains mostly information from the last element of the encoder sequence. Therefore, the context is biased towards the end of the encoder

AUTHOR

NATHAN LINTZ

OTHER CATEGORIES

- [Announcements](#)
- [artificial intelligence](#)
- [artificial intelligence](#)
- [Ask Slater](#)
- [Business](#)
- [Case Study](#)
- [Citizen Developer](#)
- [Commercial Banking](#)
- [Data Science](#)
- [Developers](#)
- [Featured Writers](#)
- [Financial Services](#)
- [Hackathon Spotlight](#)
- [Image Data Use Case](#)
- [indico](#)
- [Insurance](#)
- [Intelligent Process Automation](#)
- [Investment Banking](#)
- [Life Insurance](#)
- [Machine Learning](#)
- [Opinion Piece](#)
- [Release Notes](#)
- [Robotic Process Automation](#)
- [Text Data Use Case](#)
- [Tutorials](#)
- [Uncategorized](#)
- [Use Case](#)

SUBSCRIBE TO OUR BLOG

Stay up to date with the latest AI, RPA & Intelligent Process Automation content and news

SUBSCRIBE NOW

This website collects personal data and uses cookies to improve services. By using this site, you agree to our Terms & Conditions and Privacy Policy.

I agree

Instead of compressing the entire input sequence into a fixed representation, we can use an attention mechanism. This mechanism will hold onto all states from the encoder and give the decoder a weighted average of the encoder states for each element of the decoder sequence. Now, the decoder can take “glimpses” into the encoder sequence to figure out which element it should output next. Going back to our dinner example, attention is like choosing a dish to smell and predicting its contents instead of smelling everything at once.

Implementing attention is a straightforward modification to our language model. We start by encoding the input sequence with an RNN and hold onto each state it produces. During the decoding phase, we take the state of the decoder network, combine it with the encoder states, and pass this combination to a feedforward network. The feedforward network returns weights for each encoder state. We multiply the encoder inputs by these weights and then compute a weighted average of the encoder states. This resulting context is then passed to the decoder network. Our decoder network can now use different portions of the encoder sequence as context while it’s processing the decoder sequence, instead of using a single fixed representation of the input sequence. This allows the network to focus on the most important parts of the input sequence instead of the whole input sequence, therefore producing smarter predictions for the next word in the decoder sequence. Below is a diagram showing a sequence to sequence model that uses attention during training and generation.

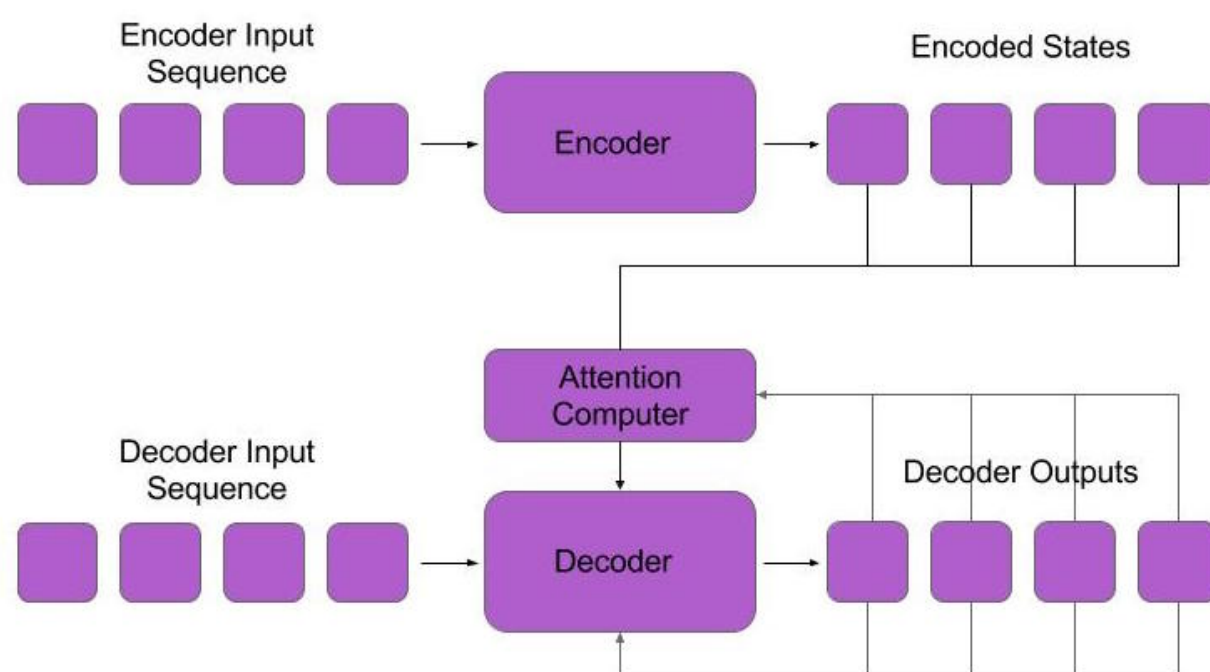


Figure 1. Attention Model: Instead of receiving the last state of the encoder, the attention model uses an attention computer which returns a weighted average of the encoder states.

Results

Now that we understand the attention modification to sequence to sequence tasks, let’s look at an example translation task that uses attention. For this task, we will take in an input sequence of characters and try to translate it into a sequence of characters where each word of the input sequence is reversed. For example, if the input sequence is “the cat sat on the mat”, the sequence to sequence model will try to predict “eht tac tas no eht tam.”

While we could solve this problem with a simple sequence to sequence model, it would be quite challenging to figure out what the reverse of the input sequence is based on a single output from the encoder network. With the attention mechanism, the decoder can focus on the previous character of the input sequence when trying to predict the next element of the output sequence. I have plotted the weighted averages from the attention mechanism to visualize which portion of the input sequence the output decoder is looking at when we make a prediction. We can see that the decoder’s attention is focused on predicting the

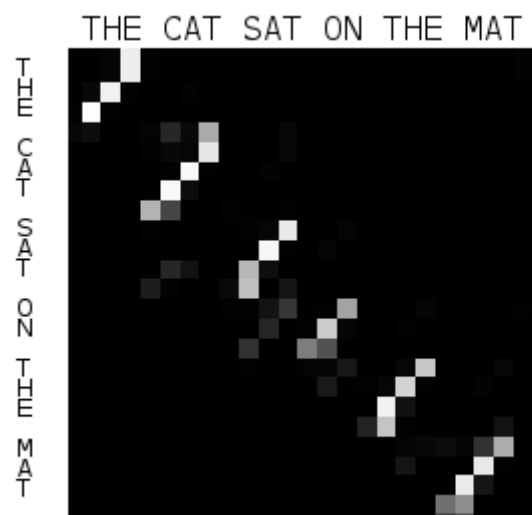


Figure 2. Attention Weights: the horizontal axis corresponds to the element of the decoder sequence the decoder is currently operating on. The vertical axis corresponds to the weighted average of the encoder states. Lighter pixels correspond to higher attention at that state from the encoder.

In this blog post series we saw how to use neural networks for processing and generating sequences. Hopefully, you now understand why attention helps neural networks make smarter predictions, in the same way that human attention helps us focus on important information while discarding unnecessary signals. If you have any questions feel free to reach out to us at contact@indico.io!

Related Posts



ARTIFICIAL INTELLIGENCE, INTELLIGENT PROCESS AUTOMATION, MACHINE LEARNING, ROBOTIC PROCESS AUTOMATION

indico | November 13, 2020

3 Essential Factors in the Intelligent Automation Build vs. Buy Decision

Like most any company in the artificial intelligence space, a question we often deal with from potential customers looking at [...]

[VIEW NOW](#)




CITIZEN DEVELOPER, DATA SCIENCE, INTELLIGENT PROCESS AUTOMATION, MACHINE LEARNING

indico | August 24, 2020

How Intelligent Process Automation Enables Citizen-led AI Development

A longstanding problem that has plagued artificial intelligence projects, including document process automation, is complexity. Essentially, the issue is there's [...]

[VIEW NOW](#)



ARTIFICIAL INTELLIGENCE, INTELLIGENT PROCESS AUTOMATION, MACHINE LEARNING, UNCATEGORIZED

indico | June 29, 2020

Not All Intelligent Process Automation Requires Million-dollar Hardware

While the artificial intelligence market is unquestionably enjoying rapid growth, cost is a gating factor that gives some companies pause. [...]

VIEW NOW

Don't Miss a Post

Get our best content on Intelligent Process Automation sent to your inbox weekly

SUBSCRIBE

Intelligent Process Automation solutions for unstructured content workflows.

WE'RE HIRING



About us

- Our Story
- Leadership
- Contact Us
- Approach
- The Indico Advantage
- What is IPA

News & Events

Support

Use Cases

- Sales & Support
- Finance & Operations
- Legal & Compliance
- Roles
- CIO/CTO/IT
- SME/Business Analyst
- Data Scientist
- Application Developer

Resources

- Blog
- Documentation
- Platform
- Product
- Implementation
- Support & Services
- Careers

Copyright © 2021 Indico Data Solutions, INC.

[Terms & Conditions](#)

This website collects personal data and uses cookies to improve services. By using this site, you agree to our [Terms & Conditions](#) and [Privacy Policy](#).

I agree