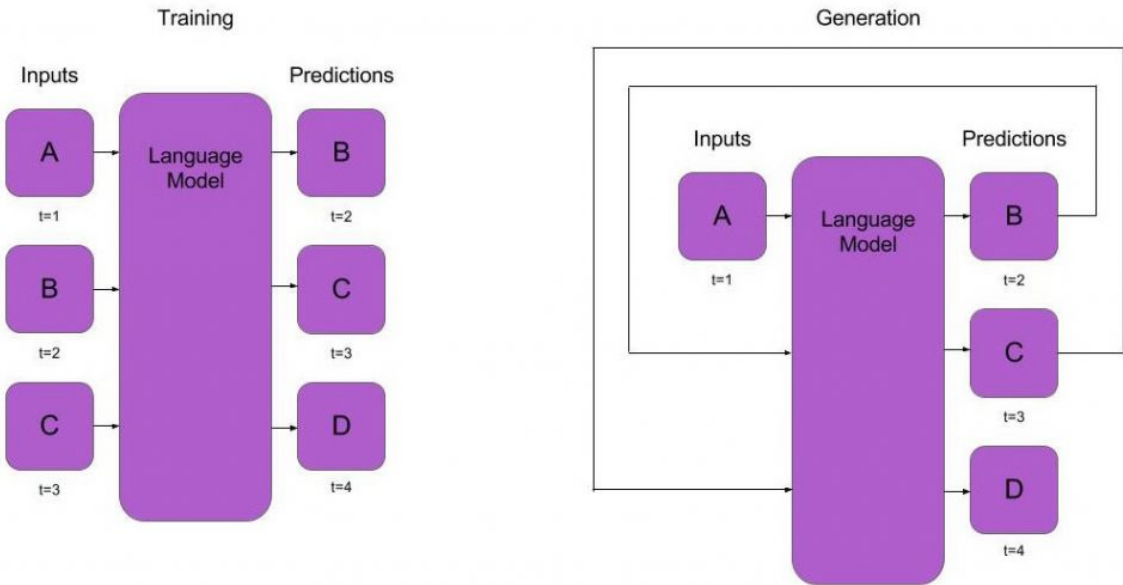


Sequence Modeling With Neural Networks (Part 1): Language & Seq2Seq

APRIL 11, 2016 / MACHINE LEARNING



This blog post is the first in a two part series covering sequence modeling using neural networks. Sequence to sequence problems address areas such as machine translation, where an input sequence in one language is converted into a sequence in another language. In this post we will learn the foundations behind sequence to sequence models and how neural networks can be used to build powerful models capable of analyzing data that varies over time.

Language Models

The foundation of sequence modeling tasks, such as machine translation, is language modelling. At a high level, a language model takes in a sequence of inputs, looks at each element of the sequence and tries to predict the next element of the sequence. We can describe this process by the following equation.

$$Y_t = f(Y_{t-1})$$

Where Y_t is the sequence element at time t , Y_{t-1} is the sequence element at the prior time step, and f is a function mapping the previous element of the sequence to the next element of the sequence. Since we're discussing sequence to sequence models using neural networks, f represents a neural network which predicts the next element of a sequence given the current element of the sequence.

Language models are generative; once trained they can be used to generate sequences of information by feeding their previous outputs back into the model. Below are diagrams showing the training and generation process of a language model.

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

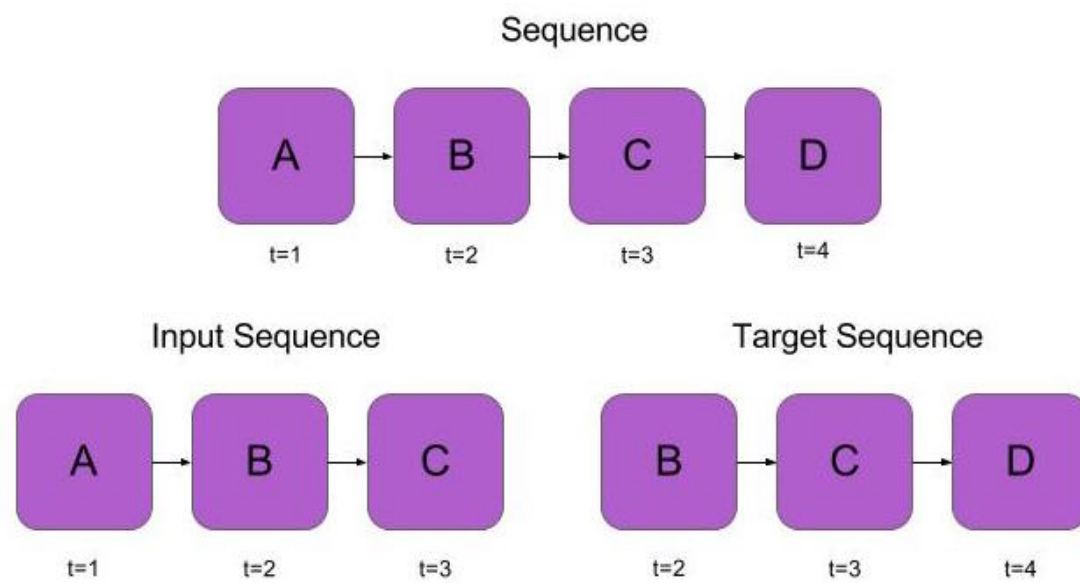


Figure 1: The sequence is ABCD. The input sequence is a slice of the whole sequence up to the last element. The target sequence is a slice of the whole sequence starting from $t=2$.

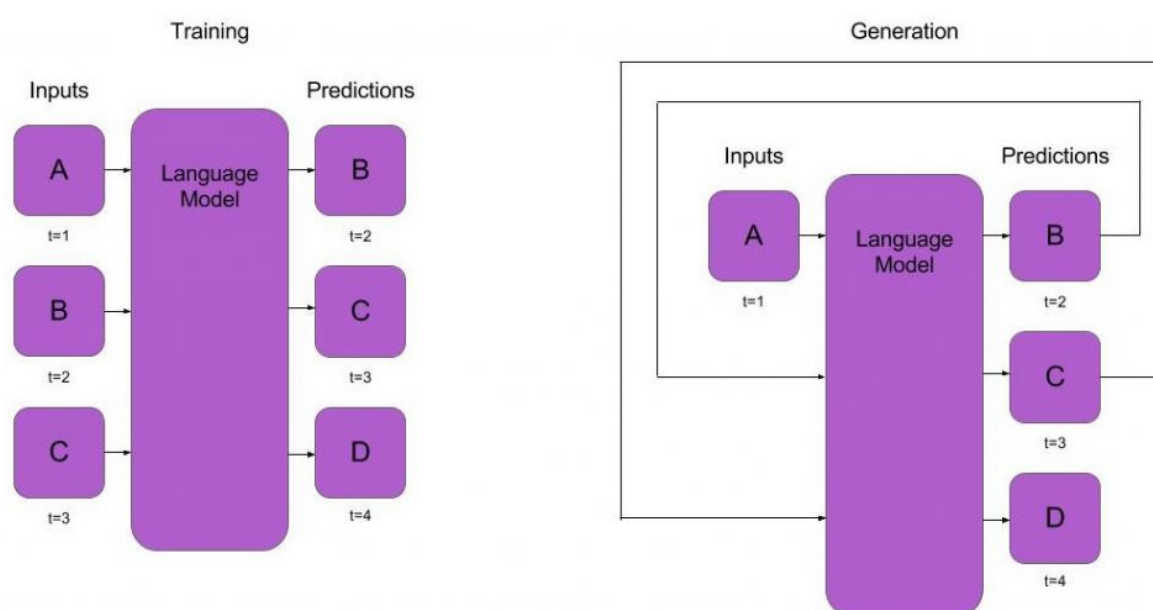


Figure 2: (Left) During training, the model tries to predict the next element of the target sequence given the current element of the target sequence. (Right) During generation, the model passes the result of a single prediction as the input at the next time step.

There are many different ways to build a language model, but in this post we'll focus on training recurrent neural network-based language models. Recurrent neural networks (RNNs) are like standard neural networks, but they operate on sequences of data. Basic RNNs take each element of a sequence, multiply the element by a matrix, and then sum the result with the previous output from the network. This is described by the following equation.

$$h_t = \text{activation}(X_t W_x + h_{t-1} W_h)$$

An in-depth explanation of RNNs is out of scope for this post, but if you want to learn more I'd recommend reading Andrej Karpathy's ["The Unreasonable Effectiveness of Recurrent Neural Networks"](#) and Christopher Olah's ["Understanding LSTM Networks"](#).

To use an RNN for a language model, we take the input sequence from $t=1$ to $t=\text{sequence_length} - 1$ and try to predict the same sequence from $t=2$ to $t=\text{sequence_length}$. Since the RNN's output is based on all previous inputs of the sequence, its output can be expressed as $Y_t = g(f(Y_{t-1}, Y_{t-2}, \dots, Y_{t1}))$. The function f gives the next state of the RNN, while the function g maps the state of the RNN to a value in our target vocabulary. Put simply, f gives a hidden state of the network, while g gives the output of the network – much like a softmax layer in a simple neural network.

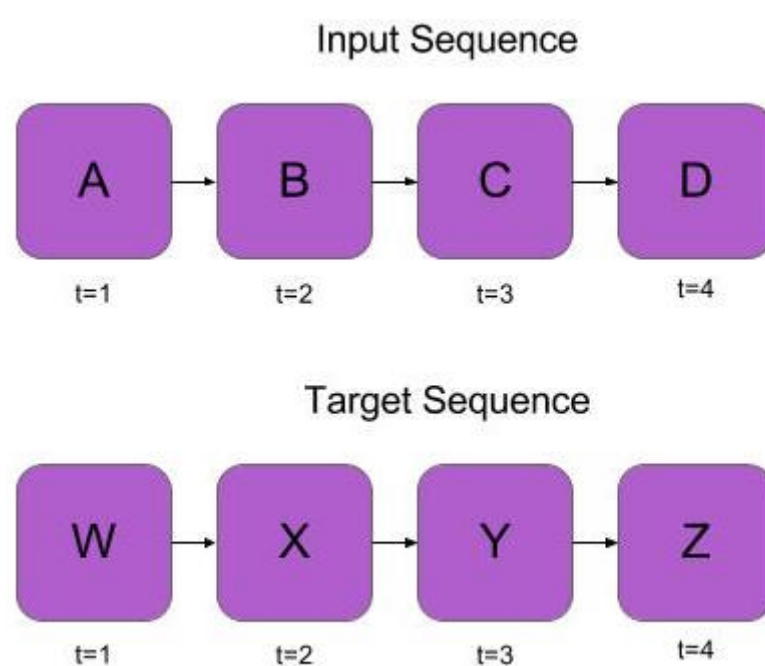
Unlike simple language models that just try to predict a probability for the next

Sequence to Sequence Models

RNNs can be used as language models for predicting future elements of a sequence given prior elements of the sequence. However, we are still missing the components necessary for building translation models since we can only operate on a single sequence, while translation operates on two sequences – the input sequence and the translated sequence.

Sequence to sequence models build on top of language models by adding an encoder step and a decoder step. In the encoder step, a model converts an input sequence (such as an English sentence) into a fixed representation. In the decoder step, a language model is trained on both the output sequence (such as the translated sentence) as well as the fixed representation from the encoder. Since the decoder model sees an encoded representation of the input sequence as well as the translation sequence, it can make more intelligent predictions about future words based on the current word. For example, in a standard language model, we might see the word “crane” and not be sure if the next word should be about the bird or heavy machinery. However, if we also pass an encoder context, the decoder might realize that the input sequence was about construction, not flying animals. Given the context, the decoder can choose the appropriate next word and provide more accurate translations.

Now that we understand the basics of sequence to sequence modeling, we can consider how to build a simple neural translation model. For our encoder, we will use an RNN. The RNN will process the input sequence, then pass its final output to the decoder sequence as a context variable. The decoder will also be an RNN. Its task is to look at the translated input sequence, and then try to predict the next word in the decoder sequence – given the current word in the decoder sequence, as well as the context from the encoder sequence. After training, we can produce translations by encoding the sentence we want to translate and then running the network in generation mode. The sequence to sequence model can be viewed graphically in the diagrams below.



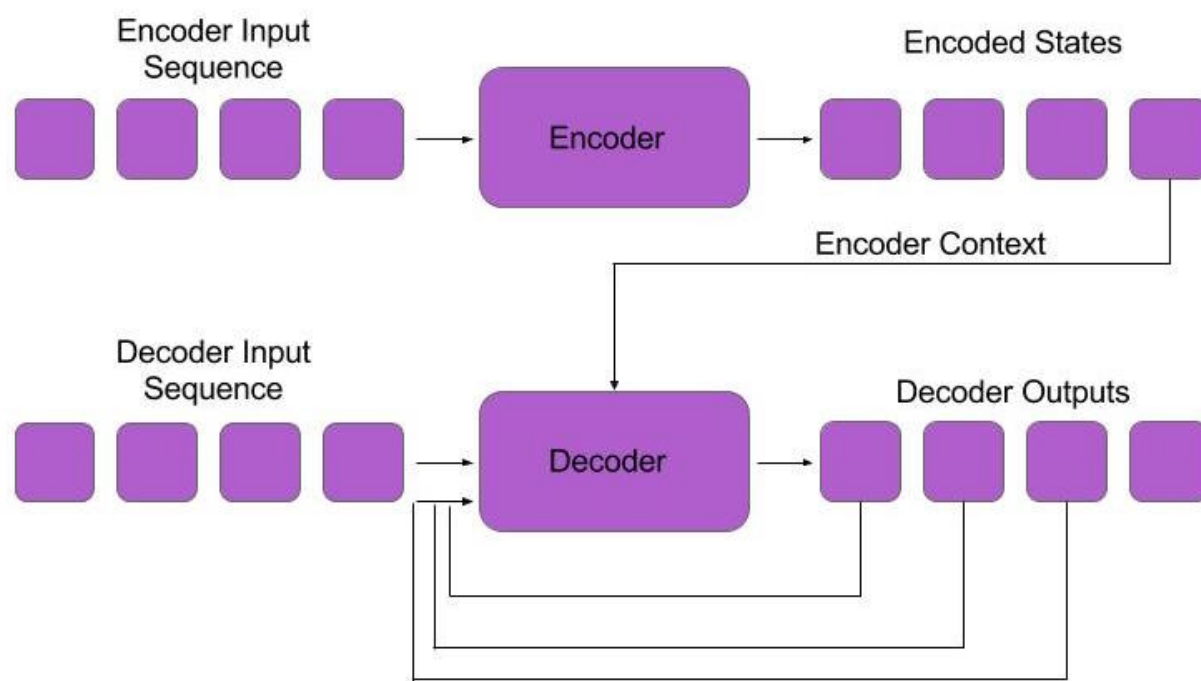


Figure 3: Sequence to Sequence Model – the encoder outputs a sequence of states. The decoder is a language model with an additional parameter for the last state of the encoder.

This concludes Part 1 of our series on sequence to sequence modeling. We saw how simple language models allow us to model simple sequences by predicting the next word in a sequence, given a previous word in the sequence. Additionally, we saw how we can build a more complex model by having a separate step which encodes an input sequence into a context, and by generating an output sequence using a separate neural network. In Part 2, we discuss [attention models and see how such a mechanism can be added to the sequence to sequence model](#) to improve its sequence modeling capacity. 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 [...]



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 [...]



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

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](#)

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