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

April 11, 2016 | Machine Learning

← **Back to Blog**

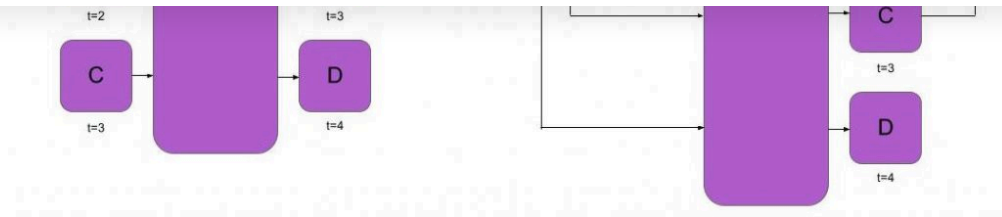This blog post is the first in a two part series covering sequence modeling using neural networks. Sequence to sequence problems address areas such as machine translation, where an input sequence in one language is converted into a sequence in another language. In this post we will learn the foundations 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.

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

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

INDICO DATA

Schedule Demo

WEBINAR: QBE, Everest, and Indico Data discuss how carriers are leveraging LLMs and automation to drive better decisions
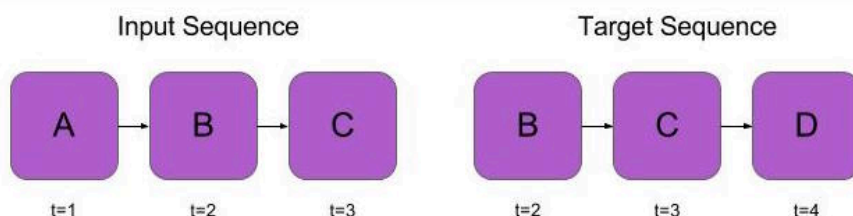
Register Now

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.*
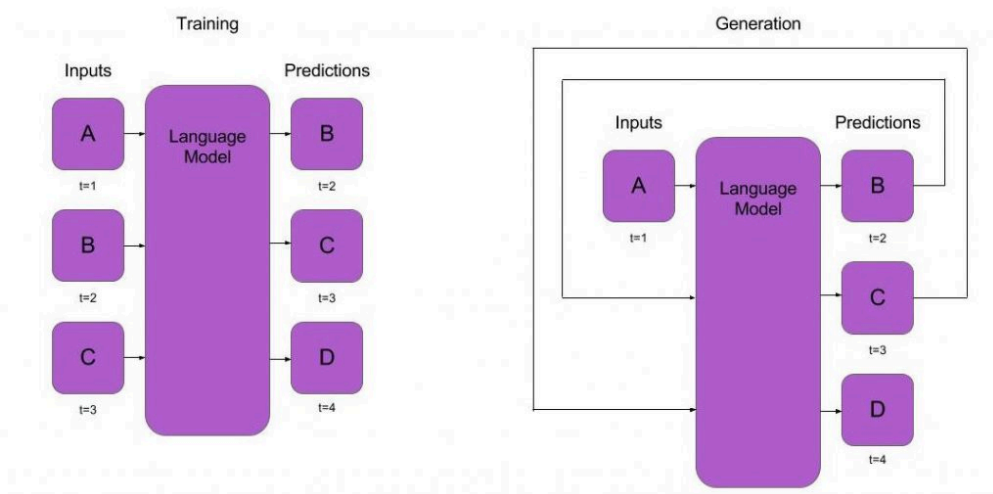

Figure 2: *(Left) During training, the model tries to predict the next element of the target sequence given the current element of the target sequence. (Right) During generation, the model passes the result of a single prediction as the input at the next time step.*

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
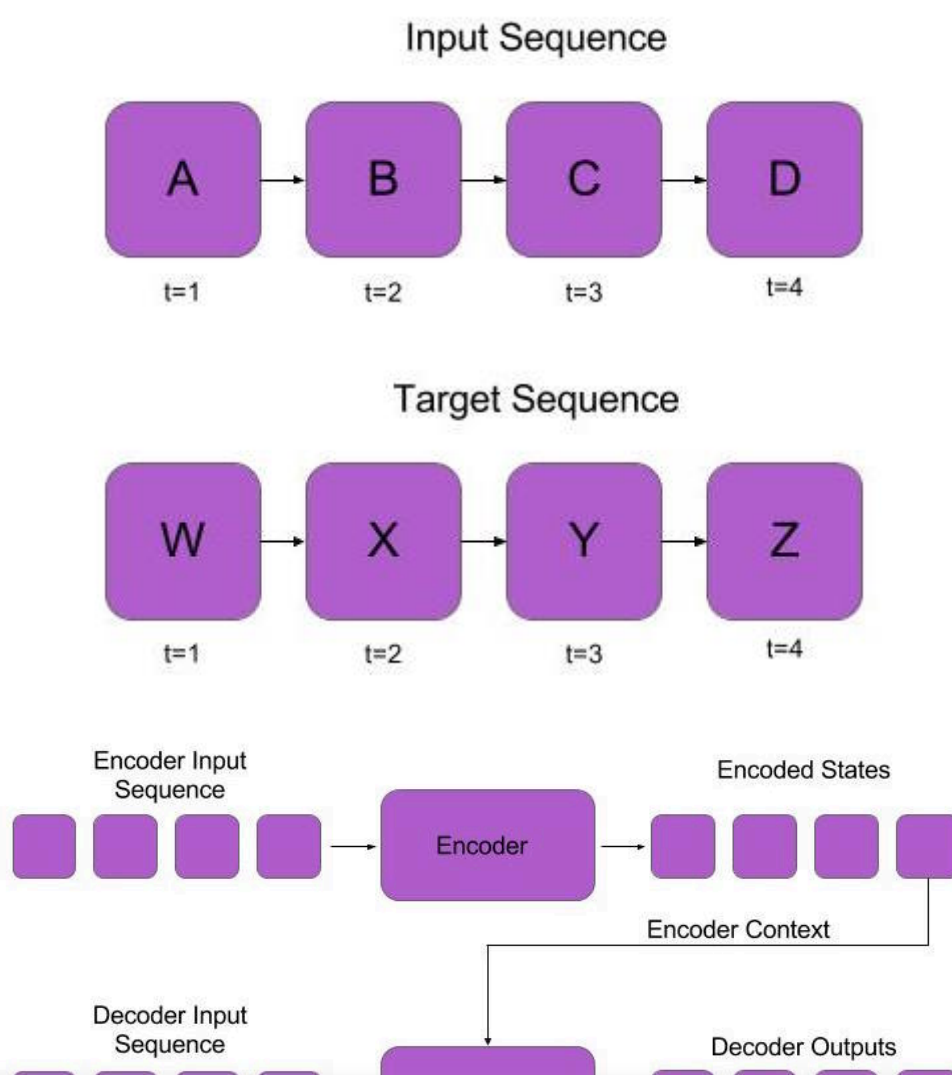
Schedule Demo

WEBINAR: QBE, Everest, and Indico Data discuss how carriers are leveraging LLMs and automation to drive better decisions

Register Now

sequence from *t=2* to *t=sequence_length*. Since the RNN's output is based on all previous inputs of the sequence, its output can be be expressed as $Y_t = g \left(f \left(Y_{t-1}, Y_{t-2}, \ldots Y_{t1} \right) \right)$. 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 word given the current word, RNN models capture the entire context of the input sequence. Therefore, RNNs predict the probability of generating the next word based on the current word, as well as all previous words.

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

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.

Input Sequence

A	B	C	D
t=1	t=2	t=3	t=4

Target Sequence

W	X	Y	Z
t=1	t=2	t=3	t=4

Encoder Input Sequence → Encoder → Encoded States

Encoder Context

Decoder Input Sequence → Decoder Outputs

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.

[addtoany]

Increase intake capacity. Drive top line revenue growth.

Schedule Demo

Related Posts

WEBINAR: QBE, Everest, and Indico Data discuss how carriers are leveraging LLMs and automation to drive better decisions

Register Now

Machine Learning, Release Notes

Release Notes – Indico Unstructured Data Platform v5.3

Read More →

Citizen Developer, Machine Learning

Overcome the complexity of machine

WEBINAR: QBE, Everest, and Indico Data discuss how carriers are leveraging LLMs and automation to drive better decisions

Register Now

WEBINAR: QBE, Everest, and Indico Data discuss how carriers are leveraging LLMs and automation to drive better decisions

Register Now

View All

Get started with Indico

SCHEDULE

1-1 Demo

Schedule

WEBINAR: QBE, Everest, and Indico Data discuss how carriers are leveraging LLMs and automation to drive better decisions

Register Now

Solutions

Insurance

Financial Services

Commercial Real Estate

Shared Services

Resources

Overview

Intelligent Intake Blog

Knowledge Base

Developer Documentation

Partners

Overview

Company

Leadership

Press & Events

Careers at Indico Data

Contact

Bug Bounty Policy

Privacy Policy

Subscribe

Copyright © 2024 Indico Data Solutions, Inc.