Sequence Networks

Devesh Nath

1 Introduction

Sequence models are a type of machine learning model that are designed to handle sequential data. These models are particularly useful for tasks where the order of the data points is important, such as time series forecasting, natural language processing, and speech recognition. In this document, we will explore various types of sequence models, their architectures, and their applications.

2 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a class of neural networks that are designed to recognize patterns in sequences of data. They are called recurrent because they perform the same task for every element of a sequence, with the output being dependent on the previous computations.

2.1 Vanilla RNN

A Vanilla RNN is the simplest type of RNN. It consists of a single hidden layer where the output from the previous time step is fed back into the network along with the current input. This feedback loop allows the network to maintain a memory of previous inputs, which is useful for tasks where context is important.

2.2 Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks are a type of RNN that are designed to overcome the limitations of Vanilla RNNs, particularly the problem of vanishing gradients. LSTMs introduce a memory cell that can maintain information over long periods of time. They use gates to control the flow of information into and out of the cell, allowing the network to learn which information is important and should be retained.

2.3 Gated Recurrent Unit (GRU)

Gated Recurrent Unit (GRU) networks are a variation of LSTMs that combine the forget and input gates into a single update gate. This simplifies the architecture and makes GRUs computationally more efficient than LSTMs, while still being able to capture long-term dependencies in the data.

2.4 Bidirectional RNN (Bi-RNN)

Bidirectional RNNs (Bi-RNNs) are a type of RNN that process the input sequence in both forward and backward directions. This allows the network to have access to both past and future context, which can be particularly useful for tasks where the entire sequence is available, such as in natural language processing.

3 Natural Language Processing (NLP)

Natural Language Processing (NLP) is a field of artificial intelligence that focuses on the interaction between computers and humans through natural language. NLP involves the application of algorithms to identify and extract the natural language rules such that the unstructured language data is converted into a form that computers can understand.

3.1 Word Embeddings

Word embeddings are a type of word representation that allows words to be represented as vectors in a continuous vector space. These vectors are learned from large corpora of text and capture semantic relationships between words. Words that are similar in meaning are located close to each other in the vector space.

3.1.1 How They Are Learnt and Used

Word embeddings are typically learned using neural network-based models on large text corpora. The most common methods for learning word embeddings are Word2Vec and GloVe. Once learned, these embeddings can be used in various NLP tasks such as sentiment analysis, machine translation, and named entity recognition.

3.2 Word2Vec

Word2Vec is a popular method for learning word embeddings developed by Google. It uses a shallow neural network with one hidden layer to learn word representations. There are two main architectures for Word2Vec: Continuous Bag of Words (CBOW) and Skip-gram. CBOW predicts the target word from the context words, while Skip-gram predicts the context words from the target word.

3.3 Sentiment Classification

Sentiment classification is the task of classifying the sentiment expressed in a piece of text, such as determining whether a movie review is positive or negative. Word embeddings can be used as input features for sentiment classification models, allowing the models to leverage the semantic relationships between words to improve classification performance.

3.4 GloVe

GloVe (Global Vectors for Word Representation) is another method for learning word embeddings developed by Stanford. Unlike Word2Vec, which relies on local context windows, GloVe leverages global word co-occurrence statistics from a corpus to learn word vectors. This allows GloVe to capture both local and global statistical information about words, resulting in more accurate word representations.

4 Attention Mechanism

The attention mechanism is a technique in neural networks that allows the model to focus on specific parts of the input sequence when making predictions. This is particularly useful for tasks where different parts of the input may have varying levels of importance. The attention mechanism has been widely adopted in various sequence modeling tasks, including machine translation, speech recognition, and image captioning.

4.1 The Attention Model

The attention model works by assigning a weight to each element of the input sequence, indicating its importance for the current prediction. These weights are learned during training and are used to create a weighted sum of the input elements, which is then used to make the final prediction. The key components of the attention model are:

- **Query (Q)**: The query represents the current state or the current input for which we want to compute the attention weights.
- **Key (K)**: The key represents the elements of the input sequence that we want to compare with the query.
- **Value (V)**: The value represents the elements of the input sequence that we want to use to compute the weighted sum.

The attention weights are computed by taking the dot product of the query and the keys, followed by a softmax operation to normalize the weights. The weighted sum of the values is then computed using these attention weights.

4.2 Applications of Attention Mechanism

The attention mechanism has been successfully applied to various tasks, including:

4.2.1 Machine Translation

In machine translation, the attention mechanism allows the model to focus on different parts of the source sentence when generating each word of the target sentence. This helps the model to handle long sentences and capture the dependencies between words more effectively.

4.2.2 Speech Recognition

In speech recognition, the attention mechanism helps the model to focus on different parts of the audio signal when transcribing speech to text. This allows the model to handle variations in speech patterns and background noise more effectively.

5 Transformers

Transformers are a type of neural network architecture that has revolutionized the field of natural language processing and other sequence modeling tasks. Unlike traditional RNNs, transformers do not rely on sequential processing of data, which allows them to be more efficient and capable of capturing long-range dependencies in the data.

5.1 Architecture

The transformer architecture consists of an encoder and a decoder, both of which are composed of multiple layers of self-attention and feed-forward neural networks.

5.1.1 Encoder

The encoder is responsible for processing the input sequence and generating a set of hidden representations. Each layer of the encoder consists of two main components:

- **Self-Attention Mechanism**: This mechanism allows the model to weigh the importance of different elements in the input sequence when generating the hidden representations. It computes attention scores for each pair of elements in the sequence and uses these scores to create a weighted sum of the input elements.

- **Feed-Forward Neural Network**: This is a fully connected neural network that processes the output of the self-attention mechanism. It consists of two linear transformations with a ReLU activation function in between.

5.1.2 Decoder

The decoder is responsible for generating the output sequence based on the hidden representations produced by the encoder. Each layer of the decoder consists of three main components:

- **Masked Self-Attention Mechanism**: Similar to the self-attention mechanism in the encoder, but with a masking operation to prevent the model from attending to future positions in the sequence.

- **Encoder-Decoder Attention Mechanism**: This mechanism allows the decoder to attend to the hidden representations generated by the encoder, enabling it to incorporate information from the input sequence when generating the output.

- **Feed-Forward Neural Network**: Similar to the feed-forward network in the encoder, this processes the output of the attention mechanisms.

5.2 Self-Attention Mechanism

The self-attention mechanism is a key component of the transformer architecture. It allows the model to weigh the importance of different elements in the input sequence when generating hidden representations. The self-attention mechanism works as follows:

- Compute the query, key, and value vectors for each element in the input sequence.
- Compute the attention scores by taking the dot product of the query and key vectors, followed by a softmax operation to normalize the scores.
- Compute the weighted sum of the value vectors using the attention scores.

5.3 Applications of Transformers

Transformers have been successfully applied to various tasks, including:

5.3.1 Machine Translation

Transformers have achieved state-of-the-art performance in machine translation tasks, such as translating text from one language to another. The ability to capture long-range dependencies and parallelize computations makes transformers particularly well-suited for this task.

5.3.2 Language Modeling

Transformers have been used to build powerful language models, such as GPT-3, that can generate coherent and contextually relevant text. These models have a wide range of applications, including text generation, dialogue systems, and code generation.