Master Thesis's Exposé

Name: Muneeb Iqbal Vaiyani

Matrikelnummer: 03719937

Topic: Using Transformer Model for Crypto Currency Price Prediction.

Introduction

A time series is data points indexed over equally spaced points in time and forecasting/predicting has always been a topic of great interest and challenge in machine learning. An example of time series would be the price of stocks and shares.

With recent developments in NLP domain of language modelling, the time series analysis area has advanced a lot. The models which are used in sequential language modelling such as RNN (Recurrent Neural Network) or LSTM (Long Short-Term Memory) to predict the next word(s) can also be used in the time series prediction with few changes in the architecture. Over the year LSTMs have proven to be the better performing in forecasting because they were able to model larger sequence which RNN could not, due to "vanishing" and "exploding" gradient while training. Latest architecture in NLP for the machine translation/language modeling was introduced in [Vaswani et al., 2017] called Transformers. They use the concept of self-attention and positional embedding in the inputs, discarding the sequential nature of input like in LSTM. This led to the possibility of modelling even long-range dependencies and better training time. With the introduction of transformer, the time series community has shown an interest to use this as the possible state of the art model for forecasting.

Our goal is to see if we can use Transformer in forecasting average price of digital assets such as BTC/USD. Is it able to catch the irregular seasonality, high volatility, and frequency better than the LSTM? Bitcoin was the first digital asset/currency in 2009 and currently holds the major market share. The market movement of other coins ("alt coins") is highly correlated with the movement of bitcoin. It also comes with widely accessible data on the internet; hence we chose BTC as our focus, but the aim is to expand the analysis on other assets as well.

Related Research

With the novelty of the domain, the number of research is increasing every year. One example is the work by [Wu, N et al., 2020] which focuses on using the vanilla settings for the forecast of influenza-like-illness cases. There is some work done to alter the architecture and check the performance of the model. [Zhou et al., 2021] proposed to encode timestamps as additional positional encoding. [Zerveas et al., 2021] introduce an embedding layer in Transformer that learn embedding vectors for each position index jointly with other model parameters.

Proposed Methodology

The vanilla architecture [Vaswani et al., 2017] comprises of two distinctive units called encoder and decoder. Both encoder and decoder are composed of multiple identical blocks, the number of blocks is architectural design choice. Each unit of encoder consist of two subunit, Multi-Head

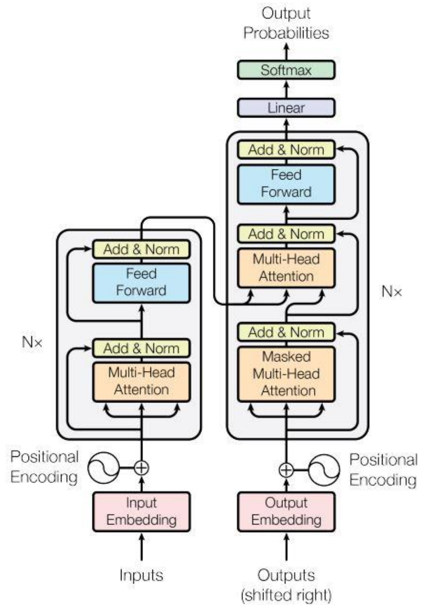

Figure 1 Transformer

Attention and Feed Forward Neural Network. At the end of subunits are the residual connections and normalization layer. The decoder unit consist of three subunits. In addition to Multi-Head Attention and FFNN it also consists of the Masked Multi-Head Attention to make sure that the attention is not use for the data later in the series when decoding it. Every input to the unit is converted to the embeddings. The position encoding is added to give information to the architecture about the relative position of the data points since it processes entire sequence of data unlike sequence aligned models which process data in an ordered sequence manner. At the end of decoder unit is the linear transformation and SoftMax layer which in the original paper gives the probability distribution for the target language vocabulary.

First step is to extract/ download data and manage the size of the data to be processed in the data loader. The higher the granularity the better it is but it increases the data size, so we need to cater this trade off**.** The data need to be analyzed and checked for any anomality. Second and the important step is to modify the architecture. This step consists of two sub-steps:

- The vanilla architecture was used in machine translation, and it used 6 units of encoder and decoder. We do not need both encoder and decoder because our task lies in the domain of regression/predicting price. Research from existing work needs to be done to understand how to modify the transformer for our use case.
- We need to understand how many units of modified architecture best explains the variance in our data. We can stick with default 6 initially and research the existing literature for this parameter, or fine tune this parameter.

Third step is to understand how the inference/training would happen in the transformer in this case. Do we predict one step or a window of steps? In the latter case do we feed the output from one step into the model to predict the next step and so on? Accordingly, the data loader needs to be modelled.

Next step is to train an LSTM on the same data set, this step can also be done before the second step. The LSTM would be the baseline model we compare Transformer with and evaluate the performance.

Possible Extension:

Couple of ways to extend this topic:

1. Modify the positional encoding as done in the [Zhou et al., 2021] using the timestamp and see if any increase in the performance.
2. Convert regression problem to classification?
3. Using same model for multiple coins.

Schedule:

References:

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In NeurIPS, 2017.

Wu, N., Green, B., Ben, X., & O'Banian, S. (2020). Deep Transformer Models for Time Series Forecasting: The Influenza Prevalence Case. International Conference on Machine Learning, Vienna, Austria.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence timeseries forecasting. In AAAI, 2021.

George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A
transformer-based framework for multivariate time series representation learning.