

# Style-Based Transformer for Time Series Forecasting

Dong-Keon Kim<sup>†</sup> · Kwangsu Kim<sup>\*\*</sup>

## ABSTRACT

Time series forecasting refers to predicting future time information based on past time information. Accurately predicting future information is crucial because it is used for establishing strategies or making policy decisions in various fields. Recently, a transformer model has been mainly studied for a time series prediction model. However, the existing transformer model has a limitation in that it has an auto-regressive structure in which the output result is input again when the prediction sequence is output. This limitation causes a problem in that accuracy is lowered when predicting a distant time point. This paper proposes a sequential decoding model focusing on the style transformation technique to handle these problems and make more precise time series forecasting. The proposed model has a structure in which the contents of past data are extracted from the transformer-encoder and reflected in the style-based decoder to generate the predictive sequence. Unlike the decoder structure of the conventional auto-regressive transformer, this structure has the advantage of being able to more accurately predict information from a distant view because the prediction sequence is output all at once. As a result of conducting a prediction experiment with various time series datasets with different data characteristics, it was shown that the model presented in this paper has better prediction accuracy than other existing time series prediction models.

Keywords : Time Series Forecasting, Transformer, Generative Decoder, Style Transfer

# 시계열 예측을 위한 스타일 기반 트랜스포머

김 동 건<sup>†</sup> · 김 광 수<sup>\*\*</sup>

## 요 약

시계열 예측은 과거 시점의 정보를 토대로 미래 시점의 정보를 예측하는 것을 말한다. 향후 시점의 정보를 정확하게 예측하는 것은 다양한 분야 전략 수립, 정책 결정 등을 위해 활용되기 때문에 매우 중요하다. 최근에는 트랜스포머 모델이 시계열 예측 모델로서 주로 연구되고 있다. 그러나 기존의 트랜스포머의 모델은 예측 순차를 출력할 때 출력 결과를 다시 입력하는 자가회귀 구조로 되어 있다는 한계점이 있다. 이 한계점은 멀리 떨어진 시점을 예측할 때 정확도가 떨어진다는 문제점을 초래한다. 본 논문에서는 이러한 문제점을 개선하고 더 정확한 시계열 예측을 위해 스타일 변환 기법에 착안한 순차 디코딩 모델을 제안한다. 제안하는 모델은 트랜스포머-인코더에서 과거 정보의 특성을 추출하고, 이를 스타일-기반 디코더에 반영하여 예측 시계열을 생성하는 구조로 되어 있다. 이 구조는 자가회귀 방식의 기존의 트랜스포머의 디코더 구조와 다르게, 예측 순차를 한꺼번에 출력하기 때문에 더 먼 시점의 정보를 좀 더 정확히 예측할 수 있다는 장점이 있다. 서로 다른 데이터 특성을 가지는 다양한 시계열 데이터셋으로 예측 실험을 진행한 결과, 본 논문에서 제시한 모델이 기존의 다른 시계열 예측 모델보다 예측 정확도가 우수하다는 것을 보인다.

키워드 : 시계열 예측, 트랜스포머, 생성 디코더, 스타일 변환

## 1. 서 론

시계열 예측이란 과거의 시계열 데이터를 토대로 미래의 특정 시점의 데이터를 예측하는 것을 말한다. 시계열 예측은 의료 모니터링, 교통망 분석, 전력 수요 예측 등 다양한 분야에 응용될 수 있어 수십 년 동안 매우 중요한 주제로서 다루

어져 왔다. 그런데 시계열을 예측하는 모델이 정확하게 향후 시점에 대해 예측하는 것은 전술한 응용 분야들의 의사 결정, 전략 수립 등에 매우 큰 영향을 끼친다. 따라서 정교한 시계열 예측 모델을 설계하여 향후 시점에 대한 데이터를 정확하게 예측하는 것은 중요한 연구 주제이다.

딥 러닝을 위시한 인공지능 기술이 발달함에 따라, 이를 활용하는 시계열 예측 모델에 관한 연구도 활발히 진행되고 있다. 초창기의 딥 러닝 기반 시계열 예측 모델 연구들은 ARIMA (Auto-Regressive Integrated Moving Average)와 같은 기존의 회귀 모델이 비정상성(Non-stationary) 데이터를 예측하는 데 한계점이 있다는 것에 주목한다. 그래서 이런 문제를 해결하기 위해 여러 연구에서는 복합적인 특성을 추출할 수 있는 순환 신경망(Recursive Neural Network, RNN), 합성곱

※ 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 지역지능화혁신인재양성(Grand ICT연구센터) 사업의 연구결과로 수행되었음(IITP-2021-2015-0-00742).

† 준 회 원 : 성균관대학교 소프트웨어학과 석사과정

\*\* 정 회 원 : 성균관대학교 소프트웨어학과 교수

Manuscript Received : September 13, 2021

First Revision : October 26, 2021

Accepted : November 3, 2021

\* Corresponding Author : Kwangsu Kim(kim.kwangsu@skku.edu)

신경망 (Convolutional Neural Network, CNN) 등의 인공 신경망 구조 기반의 시계열 예측 프레임워크를 제시한다.

최근에는 RNN 및 CNN 계열의 예측 모델이 가지고 있는 한계점을 보완하기 위해 트랜스포머(Transformer) [1] 기반의 시계열 모델에 관한 연구가 주로 이루어지고 있다. 시계열 예측에 있어 트랜스포머는 기존의 RNN 기반 모델이 가지고 있는 장기 의존성 (Long-term Dependency) 문제를 해결한다. 그뿐만 아니라 트랜스포머는 CNN 모델이 할 수 없는 시간적 해석을 할 수 있어서 시계열 예측의 기반 모델로서의 큰 장점이 있다. 최신 연구들이 제시한 트랜스포머 기반 예측 모델들은 CNN과 RNN 기반의 모델들보다 더 효율적으로 학습하고 정확한 예측 성능을 보인다[2-4].

트랜스포머를 활용한 기존 연구들은 자연어 처리 분야에 활용되었던 일반적인 트랜스포머 구조를 빌려 시계열 예측 프레임워크를 구성한다. 자연어 처리에 사용하는 트랜스포머 모델은 자가회귀(Auto-regressive) 구조로 되어 있다. 자가회귀 구조란, 출력한 결과값이 다시 입력으로 들어가는 구조를 말한다. 이 구조는 출력값에 오차가 생기면 다음 예측값을 출력하는 데 영향을 준다는 문제점이 있다. 자연어 처리의 트랜스포머는 단어를 출력하기 때문에 이러한 문제점이 덜 두드러지지만, 시계열 예측의 트랜스포머는 예상되는 수치를 출력하기 때문에 이러한 문제가 더 뚜렷해진다. 그래서 자가회귀 구조 기반의 트랜스포머를 시계열 예측에 활용했을 때 예측하고자 하는 시점이 기존 시점보다 멀수록 예측 정확도가 떨어진다는 구조적인 한계점이 있다.

본 논문에서는 상술한 문제점을 해결하고 좀 더 정확한 시계열 예측을 하기 위해 새로운 구조의 트랜스포머 모델을 제안한다. 제안하는 모델은 기존의 트랜스포머 구조처럼 인코더-디코더(Encoder-Decoder) 구조로 되어 있지만, 디코더가 합성곱 신경망 층을 통해 예측 순차를 한 번에 출력한다는 차이점이 있다. 본 논문에서는 인코더에서 어텐션(Attention)을 통해 추출한 맥락(Context)을 스타일 변환(Style Transfer) [5]을 적용하여 디코더에 반영한다. 디코더는 반영된 스타일을 통해 예측하고자 하는 시계열 순차(Sequence)를 한 번에 출력한다. 제안한 구조를 통해 기존 연구들의 예측 모델이 좋은 결과를 내지 못했던 장기 시점에 대한 예측을 정확하게 하는 것이 본 연구의 주 기여이다.

본 논문의 구성은 다음과 같다. 2장에서는 딥 러닝을 활용한 기존의 시계열 예측 연구를 소개한다. 그리고 3장에서는 본 연구에서 제안하는 트랜스포머 모델의 구조를 부분별로 기술한다. 4장에서는 제안한 예측 프레임워크의 성능을 평가하기 위한 다양한 실험을 진행한 결과를 논한다. 마지막으로 5장에서는 본 논문의 결론 및 향후 연구 방향에 대해 서술한다.

## 2. 이전 연구

딥 러닝 기법이 주목받기 이전에는, 시계열 예측 모델로 ARMA(Auto-Regressive Moving Average)와 같은 회귀

모형을 주로 사용해 왔다[6-8]. 회귀 모형은 주로 데이터 특성이 정상성(Stationary)을 가지고 있는 경우에는 예측 정확도가 높다. 그러나 주기 및 추세 등을 가지고 있는 비정상성(Non-stationary) 시계열 데이터를 예측할 때는 정확도가 낮다는 한계점이 존재한다[9].

기존의 회귀 모형의 문제점들을 해결하기 위해 여러 연구는 딥 러닝 기법을 시계열 예측에 맞게 변형하여 새로운 시계열 예측 모델을 제안한다[10-12]. 그리고 딥 러닝을 활용한 시계열 예측 프레임워크를 다양한 응용 분야에 적용한 연구들이 많이 제시되고 있다. 예를 들어, Tokgöz는 터키의 전력 수요량을 예측하기 위해 자연어 처리에 사용되었던 RNN을 활용한 시계열 예측 모델을 제안한다[13]. 시계열 입력에 대해 시간별로 나타나는 특성을 신경망으로 학습하여 예측 순차를 출력하는 구조를 가진다. Koprinska는 호주에서 생산되는 태양광 발전량을 예측하기 위해 컴퓨터 비전 분야에 사용되었던 CNN 모델을 1차원으로 변형하여 특성을 추출하는 신경망으로 시계열 예측 프레임워크를 고안한다[14]. Huang은 1차원 CNN과 장단기 메모리(Long Short-term Memory)의 장점을 결합한 시계열 예측 모델을 제시한다[15]. 이처럼 초창기의 딥 러닝 기반 시계열 예측 연구들은 주로 RNN, LSTM이나 CNN을 활용하여 모델을 구성한다. 앞서 언급한 딥 러닝 기반의 모델들은 주로 사용했던 회귀 모형보다 예측 정확도 측면에서 더 우수한 성능을 보여준다.

최근까지 주로 활용되었던 RNN, LSTM 및 CNN 모델들은 추세와 주기적 특성을 추출하는 데 있어 장점이 있지만, 다음과 같은 여러 한계점이 있다. RNN, LSTM과 같은 순환 신경망 모델의 경우, 순차적으로 연산을 해서 출력을 하는 구조이기 때문에 병렬적 연산을 할 수 없어 학습 속도가 매우 느리다는 단점이 있다. 그뿐만 아니라 현재 시점으로부터 거리가 먼 과거 데이터에 대한 저장 능력이 떨어지기 때문에, 주기가 매우 긴 데이터를 예측하는 장기 의존성(Long-term Dependency) 문제 같은 경우 예측 정확도가 떨어진다는 한계점이 존재한다[16]. 1차원 CNN 모델의 경우 시간에 대한 정보를 고려하지 않고 합성곱을 한다는 한계점이 있다. 그래서 풀링(Pooling) 등의 과정을 거치면 해당 시간대의 정보가 사라지는 문제점이 있다. 그뿐만 아니라 신경망의 필터(filter) 크기에 매우 큰 영향을 받는다는 문제점이 있다. 그래서 시계열 데이터가 가지는 주기가 필터의 크기보다 큰 경우, 주기를 해석할 수 없어 예측 정확도가 낮아진다는 문제점이 있다.

상술한 여러 문제점을 해결하기 위해 최근 연구는 트랜스포머 모델을 시계열 예측 모델로서 활용하고 있다[2-4, 17]. 간단하게 트랜스포머의 어텐션(Attention) 메커니즘을 활용한 연구부터[18] 트랜스포머 모델 전체를 차용하여 예측 프레임워크를 구성하는 연구까지[19] 매우 다양한 트랜스포머 기반 시계열 예측 모델 연구가 진행되고 있다. 트랜스포머 모델은 시간대별로 순차적으로 연산하는 구조가 아닌 병렬 연산이 가능하므로 순환 신경망 기반의 예측 모델들이 해결하기 어려운 장기 의존성 문제를 해결할 수 있다. 그리고 입력

된 시계열이 가지고 있는 시간 정보를 포함하여 특성을 추출하기 때문에 시간적 특성을 고려할 수 없는 CNN과 비교했을 때에도 큰 장점이 있다.

트랜스포머는 어텐션 연산으로 시계열의 특정 시점에 주의를 더 기울이고 이를 콘텐츠(Content)로 반영하여 예측 결과를 순차적으로 출력하는 자가회귀 구조로 되어 있다. 기존의 연구들은 대부분 이와 같은 구조의 트랜스포머를 차용한 시계열 예측 프레임워크를 제안한다. 일반적인 트랜스포머는 이전 단어 배열을 통해 다음 단어를 예측하는 자연어 처리 문제를 해결하기 위해 제시된 구조다. 그러나 단어를 출력하는 자연어 처리와 다르게 시계열 예측 문제는 특정한 수치를 출력한다. 예상되는 수치를 출력하고, 그 값을 다시 입력하는 구조는 예측하고자 하는 순차가 길수록 정확도가 떨어진다는 한계점이 있다[20]. 따라서 기존의 자가회귀 구조식으로 출력하는 트랜스포머 구조는 시계열 예측에는 적합하지 않다.

### 3. 예측 모델

본 장에서는 예측 프레임워크가 해결하고자 하는 시계열 예측 문제를 정의하고, 본 논문에서 소개할 스타일 기반 트랜스포머(Style-Based Transformer) 모델이 구체적으로 어떤 구조로 이루어져 있는지 기술한다. 제안하는 예측 모델의 전체적인 구조는 Fig. 1에 나타나 있다.

#### 3.1 문제 정의

시계열 예측 프레임워크는 일정 구간의 시계열 데이터를 입력으로 받아 특정한 시점이 지난 후의 순차 데이터를 예측

한다. 본 논문에서는 예측 프레임워크에 입력할 시계열 데이터를 Equation (1)과 같이 정의한다.

$$\mathbf{X}_t = \{x_{t-m+1}, x_{t-m+2}, \dots, x_t\} \quad (1)$$

시계열 입력  $\mathbf{x}_t$ 는 시점  $t$ 에 입력으로 들어갈  $m$ 개의 과거 데이터이다. 제안하는 프레임워크는 크기가  $m$ 인 시계열 데이터를 입력으로 받고 직후 시간대의  $n$ 개의 예측 순차  $\hat{\mathbf{y}}_t$ 를 출력한다. 시점  $t$  이후를 예측한 출력 순차  $\hat{\mathbf{y}}_t$ 는 Equation (2)와 같이 정의한다.

$$\hat{\mathbf{Y}}_t = \{\hat{y}_{t+1}, \hat{y}_{t+2}, \dots, \hat{y}_{t+n}\} \quad (2)$$

#### 3.2 어텐션 기반 인코더

본 연구에서 제안하는 시계열 예측 모델은 트랜스포머(Transformer) 구조를 가진다. 트랜스포머는 크게 인코더(Encoder)와 디코더(Decoder)가 여러 개 누적되는 구조를 가진다. 인코더는 멀티 헤드 어텐션(Multi-Head Attention) 층과 순방향 신경망(Feed Forward Neural Network) 층으로 이루어져 있다. 본 논문에서도 이 두 층으로 이루어진 인코더로 트랜스포머를 구성한다.

인코더의 두 층을 설명하기에 앞서, 인코더에 입력할 시계열 데이터를 전처리하는 방법을 설명한다. 본 연구에서는 기존의 트랜스포머가 순차 데이터를 전처리하는 방법인 포지셔널 인코딩(Positional Encoding) 방법을 사용한다. 포지셔널 인코딩은 다차원의 벡터로 표현되는 위치 정보를 순차 데이터에 추가적으로 입력하여 전처리하는 방법이다. 주어진

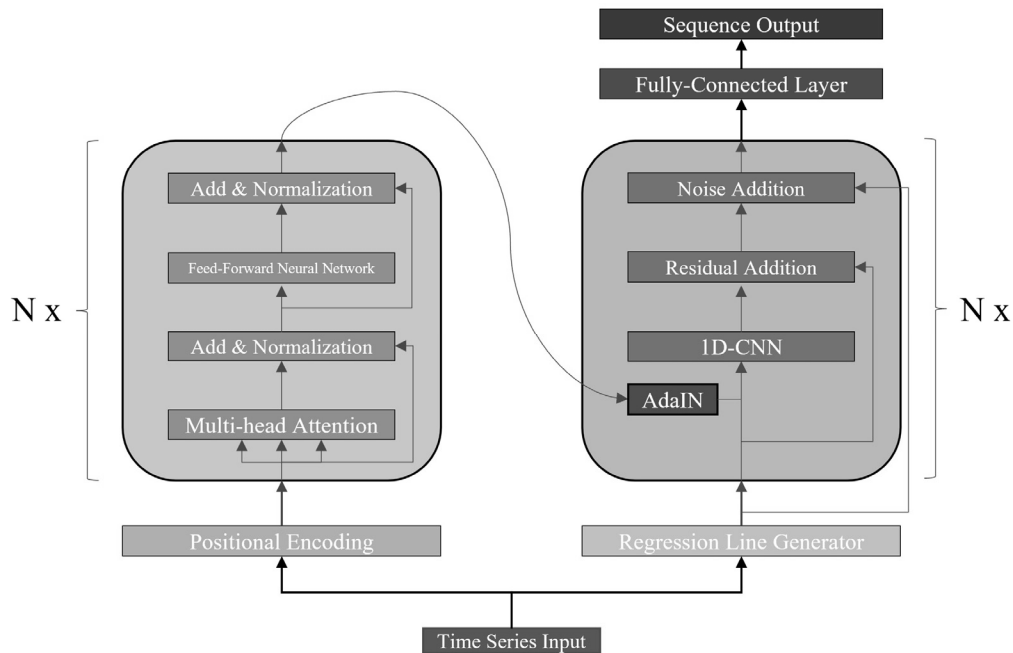


Fig. 1. Overall structure of proposed Style-based Transformer

시계열 데이터 X에 대하여 Equation (3)에 제시된 두 식을 통해 크기가 d인 벡터로 포지셔널 인코딩을 진행한다.

$$PE_{(x_i, 2k)} = x_i + \sin(x_i/10000^{2k/d})$$

$$PE_{(x_i, 2k+1)} = x_i + \cos(x_i/10000^{2k/d}) \quad (3)$$

여기서 시계열 입력  $X_t$ 는 1 x t의 크기를 가지고, 포지셔널 인코딩 결과값은 d x t의 크기를 가진다.  $x_i$ 는 i번째 시점의 시계열 값이다. 총 d 개의 주기 함수로 표현되는 위치 정보를 시계열 입력에 더함으로 각 시계열 데이터에 유일한 위치 정보값을 표현할 수 있다. 이 전처리한 값을 인코더의 입력값으로 활용한다.

인코더의 앞 부분인 멀티 헤드 어텐션 층은 기본적으로 쿼리(Query), 키(Key), 값(Value)의 세 입력을 받는다. 포지셔널 인코딩 결과값  $PE(X_t)$ 와 가중치 행렬  $W$ 를 통해 쿼리, 키, 값을 출력한다. 계산 과정은 Equation (4)와 같다.

$$Q = W_q PE(X_t) \quad K = W_k PE(X_t) \quad V = W_v PE(X_t) \quad (4)$$

위 식에서  $Q, K, V$ 는 각각 쿼리, 키, 값을 의미한다.

구한 쿼리, 키, 값 중 쿼리와 키를 활용하여 입력의 어떤 부분에 더 주의를 기울일지에 대한 어텐션 점수(Attention Score) 값을 계산한다. 계산 과정은 Equation (5)와 같다.

$$Score(Q, K) = \tanh(W'_q Q + W'_k K + b) \quad (5)$$

여기서  $W$ 는 가중치 행렬,  $b$ 는 bias값, tanh는 하이퍼볼릭 탄젠트 함수 (Hyperbolic Tangent)를 의미한다. 구한 Score값은 총 d개의 어텐션 벡터를 가진다.

멀티 헤드 어텐션 층의 최종 출력 값  $C$ 는 구한 어텐션 점수와 앞서 구한 값(Value)을 통해 구한다. 구하는 과정은 Equation (6)의 식과 같다.

$$C_t = Softmax(Score(Q_t, K_t) V_t^T) \quad (6)$$

멀티 헤드 어텐션 층에서 구한  $C$ 를 순방향 신경망에 입력하기에 앞서,  $PE(X_t)$ 와 잔차 연결(Residual Connection)을 한다. 초기 입력값  $PE(X_t)$ 와 잔차 연결을 함으로써 가중치 역전파시 발생하는 기울기 소실(Vanishing Gradient)을 방지할 수 있다. 그리고 잔차 연결에 더해 레이어 정규화(Layer Normalization)[21]를 진행한다. 레이어 정규화를 통해 기울기(Gradient) 값을 안정적으로 구할 수 있다는 장점이 있다. 레이어 정규화는 기존의 트랜스포머의 인코더[1]에서 사용한 방법을 참고한다.

인코더의 뒷 부분은 순방향 신경망이다. 순방향 신경망은 단순한 전연결 층(Fully-connected Layer)로 이루어져 있다. 이 과정에서 주어진 정보를 압축하기 위해 크기가 d x t인 입력값  $(PE(X_t) + C)$ 를  $(d/2) \times t$ 의 모양을 가지는 결과  $E$ 로 출력할 수 있게끔 가중치 행렬  $W$ 를 설정한다. 인코더의

최종 결과는 Equation (7)의 과정으로 구한다.

$$E_t = W_E(PE(X_t) + C_t) \quad (7)$$

하나의 인코더 층 출력값  $E_t$ 는  $(d/2) \times t$ 의 크기를 가진다. 본 연구에서는 기존의 트랜스포머 모델처럼 여러 개의 인코더를 쌓는 구조로 인코더 블록을 구성한다. 일반적인 트랜스포머 모델과 비교했을 때, 제안된 구조에서는 하나의 인코더 층을 지날수록 차원의 수(d)가 절반으로 감소한다는 차이점이 있다. 본 논문에서 제안한 인코더 블록은 3개의 인코더 층을 거쳐 최종적으로 1 x t의 피처를 출력한다. 이 피처를 본 논문에서는 스타일(Style,  $\tau$ )이라고 정의한다.

### 3.3 스타일 기반 디코더

일반적인 트랜스포머 모델은 인코더와 거의 비슷한 모양으로 디코더를 구성한다. 기존의 디코더는 인코더와 마찬가지로 멀티 헤드 어텐션 층과 전연결 신경망 층으로 이루어져 있다. 일반적인 디코더는 마스킹(Mask)된 멀티 헤드 어텐션 층을 통해 현재 위치의 이전 정보들만을 활용하여 현재 정보 하나만을 예측한다. 이 디코더 구조 때문에 트랜스포머는 단일 시점의 결과를 출력하고, 그 결과를 다시 디코더에 입력하는 구조를 통해 순차를 출력한다. 전술하였듯, 이 구조는 예측하고자 하는 시점이 멀수록 정확도가 떨어진다는 문제점이 존재한다.

본 논문에서는 일반적인 트랜스포머 모델의 디코더와 전혀 다른 구조로 디코더를 설계한다. 임의의 초기 시계열 값에 스타일을 디코더 층마다 반영하여 예측하고자 하는 시계열 값을 한 번에 출력하는 방법으로 디코더 블록을 구성한다. 본 연구에서 제안하는 디코더는 스타일 변환(Style Transfer) 층, 1차원 합성곱 신경망(1D-CNN)층, 노이즈 추가(Noise Addition)층으로 이루어져 있다. 기존 트랜스포머의 디코더와 제안하는 디코더의 차이는 Fig. 2와 같다.

크기가 n인 예측 순차  $\hat{Y}_t$ 를 출력하기 위해 디코더 층은 n 길이를 가진 시계열 값을 입력으로 받는다. 생성적 적대 신경망(Generative Adversarial Networks, GAN)[22]이 잠재

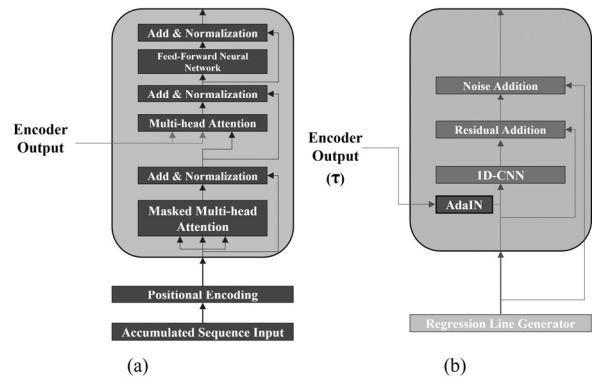


Fig. 2. Structure Comparison of Two Decoder: (a) Typical Decoder, (b) Proposed Decoder

벡터(Latent Vector)에서 데이터를 생성해 내는 방법에 착안하여, 입력 데이터로부터 잠재 벡터를 생성한다. 본 논문에서는 이 잠재 벡터를 초기 시계열 순차라고 하고, 첫 번째 디코더 층의 입력으로 활용한다. 초기 시계열 순차는 입력 시계열 순차와 유사한 추세 및 패턴을 가지고 있다고 가정한다. 언급한 가정에 기인하여, 초기 시계열 순차는 시계열 입력값  $\mathbf{x}_t$ 를 통해 구한 회귀방정식으로부터 추출한다. 초기 시계열 순차  $D_0$ 는 Equation (8)의 과정을 통해 구한다.

$$a = \frac{12}{m^3 + 2m} \sum_{i=1}^m (i - \frac{m}{2})(x_{t-m+i} - \overline{\mathbf{x}}_t)$$

$$b = \overline{\mathbf{x}}_t - a \frac{m}{2} \quad (8)$$

$$\mathbf{x}' = \{m + 1, m + 2, \dots, m + n\}$$

$$D_0 = a\mathbf{x}' + b$$

여기서  $\overline{\mathbf{x}}_t$ 는 시계열 입력  $\mathbf{x}_t$ 의 평균값이다. 구한  $D_0$ 는 첫 번째 디코더 층의 입력으로 들어간다.

디코더 층은 크기가  $n$ 인 순차 입력을 받는다. 디코더는 이 순차 입력에 앞서 인코더에서 추출한 스타일을 입력받는다. 그리고 입력 받은 스타일( $\tau$ )을 순차 입력에 반영한다. 스타일을 입력하는 방법은 Karras et al.[23] 이 고안한 AdaIN(Adaptive Instance Normalization)방법을 참고하여 구성한다. 순차 입력  $D$ 에 스타일  $\tau$ 를 입력하는 구체적인 방법은 Equation (9)의 식과 같다.

$$AdaIN(D, \tau) = \sigma(\tau) \frac{D - \mu(D)}{\sigma(D)} + \mu(\tau) \quad (9)$$

여기서  $\mu$ 는 평균이고,  $\sigma$ 는 표준 편차를 의미한다. 이 방법으로 인코더에서 추출한 스타일을 반영하여 예측하고자 하는 시계열 데이터도 입력한 시계열 데이터의 특성과 유사하게 맞출 수 있다.

스타일 변형된 순차 입력  $AdaIN(D, \tau)$ 을 1차원 CNN층에 반영한다. CNN층은 Karras et al.[23] 이 생성 모델에 포함한 CNN 층과 같이 필터(Filter) 크기를 3으로 하고, 스트라이드(Stride)를 1로 설정한다. 디코더의 1차원 CNN층의 입력과 출력의 크기를 같게 하기 위해서 동일 패딩(Same Padding)을 추가한다. 그리고 기울기 소실을 방지하기 위해 잔차 연결 층을 추가한다. 잔차 연결 층은 1차원 CNN층의 출력과  $AdaIN(D, \tau)$ 을 더한다.

본 연구가 참고한 스타일 기반 생성 모델[23]은 CNN층 다음에 추가적으로 노이즈(Noise)를 더하여 좀 더 강건한 모델 학습이 이루어지게끔 한다. 제안하는 디코더의 노이즈 추가 층에서는 이와 비슷한 방법으로 노이즈를 입력한다. 노이즈는 평균이 0이고, 표준편차가  $\sigma_t$ 인 정규 분포에서 추출한다. 이 때,  $\sigma_t$ 는 시계열 입력  $\mathbf{x}_t$ 의 표준편차이다. 입력한 시계열의 특성(표준편차)을 노이즈로 추가함으로써, 출력 순차에 입력 특성을 반영한다.

전체 디코더는 인코더와 마찬가지로 여러 개의 디코더를 쌓아 하나의 디코더 블록으로 구성한다. 디코더 블록 안에 있는 디코더 층은 모두 똑같은 구조를 가지고 있다. 최종 예측 순차  $\hat{\mathbf{y}}_t$ 는 마지막 디코더 층의 출력에 전연결 층을 통과시켜 출력한다.

## 4. 실험

본 장에서는 제안한 스타일 기반 생성적 트랜스포머를 활용하여 여러 시계열 예측 실험을 진행한 결과를 보인다. 우선 실험에 활용할 시계열 데이터셋을 소개한다. 그리고 3장에서 설명한 예측 모델의 구체적인 구현 방법과 실험 환경, 사용한 평가지표를 설명한다. 마지막으로 예측 모델의 다양한 예측 시점에 대한 실험 결과와 기존 연구에서 제안한 인공신경망 기반 예측 모델과의 비교 결과를 제시한다.

### 4.1 실험 데이터셋

본 논문에서는 시계열 예측 분야에서 벤치마크 데이터셋으로 자주 활용되는 Traffic, Exchange Rate, Electricity, Solar Energy의 4개의 데이터셋을 활용한다[24]. 데이터셋에 대한 세부 사항은 Table 1에 명시되어 있다.

언급한 벤치마크 데이터셋들은 단일 변수의 시계열 데이터가 아닌 여러 변수의 시계열 데이터를 모은 실제 데이터셋이다. 본 실험에서는 하나의 데이터셋 안에 있는 개별 변수들에 대해 각각 시계열 예측 실험을 진행한다. 데이터셋이 가진 시계열 데이터의 전체 길이(Total Length)중 앞부분 80%는 훈련에 사용하고, 나머지 뒷부분 20%는 테스트 데이터로 사용한다.

### 4.2 실험 준비

본 실험에서는 3장에서 설명한 스타일 기반 생성적 트랜스포머 모델을 활용한다. 인코더 블록은 2개의 인코더 층을 쌓아서 구성하고, 디코더 블록은 6개의 디코더 층을 쌓아서 구성한다. 3장에서 언급한 포지셔널 인코딩(Positional Encoding)의 차원수  $d$ 는 4로 설정한다.

구성한 예측 모델의 옵티마이저(Optimizer)는 Adam[25]을 사용한다. 학습률은 0.001로 사용하고, 배치(Batch) 크기는 1024로 설정하고, 에폭(Epoch)은 300으로 한다. 손실 함수는 평균 제곱 오차(Mean Squared Error)로 설정한다.

Table 1. Details of Experiment Datasets

Dataset	Total Length	The number of variables	Time Unit
Traffic	17,544	862	1 hour
Exchange Rate	7,588	8	1 day
Electricity	26,304	321	1 hour
Solar Energy	52,560	137	10 minutes

### 4.3 평가지표

평가지표는 기존의 시계열 예측 연구들에서 지표로써 주로 사용하는 평균 제곱근 편차(Rooted Mean Squared Error, RMSE)와 상관계수(Correlation Coefficient, CORR)로 한다. RMSE와 CORR을 구하는 식은 Equation (10)과 같다.

$$RMSE = \sqrt{\frac{1}{T_{test}} \sum_{i=1}^{T_{test}} (y_i - \hat{y}_i)^2}$$

$$CORR = \frac{1}{T_{test}} \sum_{i=1}^{T_{test}} \frac{(y_i - \mu(Y))(\hat{y}_i - \mu(\hat{Y}))}{\sqrt{(y_i - \mu(Y))^2(\hat{y}_i - \mu(\hat{Y}))^2}} \quad (10)$$

위 식에서  $T_{test}$ 는 테스트 데이터의 총 길이,  $y_i$ 는  $i$  시점에 서의 실제 값,  $\hat{y}_i$ 는  $i$  시점에서의 예측값을 의미한다. 그리고  $Y$ 는 테스트 기간 전체의 실제 값 집합,  $\hat{Y}$ 는 테스트 기간 전체의 예측값 집합을 뜻한다.

### 4.4 시계열 예측 실험

본 논문에서 제안하는 예측 모델의 성능평가 실험을 진행

한다. 입력하는 시계열 길이  $m$ 은 30, 출력하는 시계열 길이 또한 30으로 고정한다. 앞서 언급한 4개의 데이터셋에 대해 시계열 예측 실험을 진행하고, 결과를 그래프 및 표로 나타낸다. 그래프는 실제로 모델이 어떻게 값을 예측하는지 시각화 하여 보여주고, 표는 언급한 성능평가 지표를 활용하여 구체 적으로 어느 정도의 성능을 내는지 보여준다. 결과 그래프는 Fig. 3에, 결과표는 Table 2에 나타나 있다.

Fig. 3으로 표현한 그래프는 각 데이터셋의 1번째 변수에 대한 마지막 1,000개의 데이터에 대해 실제값과 예측값을 도시한 그래프이다. 예측 결과는 예측하기 제일 어려운 예측 순차의 맨 마지막 값에 대해 나타낸다. Traffic 데이터와 Electricity 데이터에 관한 결과를 보면, 급격하게 상승 또는 하강하는 값에 대해서는 다소 부정확하게 예측하는 것을 볼 수 있다. 그러나 4개의 데이터셋의 예측 결과를 보았을 때 전체적인 추세는 매우 정확하게 따라가는 것을 볼 수 있는데, 이는 실제적인 수치 지표인 Table 2에서도 나타난다. Table 2는 테스트 데이터 전체에 대해 평가지표로 나타난 값들을 보여준다. 후술할 비교 결과에서도 확인할 수 있듯, 길이가 긴 순차에 대해서 정확도 높게 예측하는 것을 볼 수 있다.

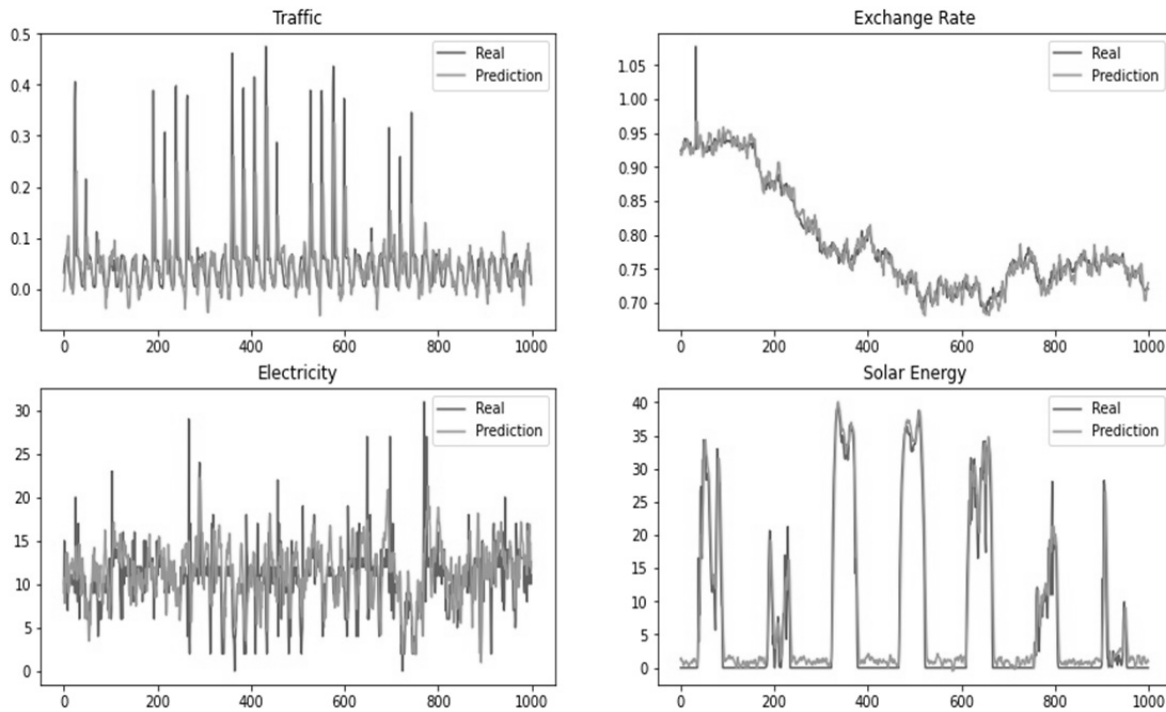


Fig 3. Prediction results of 4 experimental datasets

Table 2. Performance Metrics of Experimental Results

Results	Traffic	Exchange	Electricity	Solar
RMSE	0.7389	0.0353	0.4318	0.8804
CORR	0.9012	0.9539	0.9188	0.9557

4.5 기존의 연구들과의 비교

본 논문에서 제안하는 예측 모델의 우수성을 입증하기 위해 다른 기존의 연구들에서 제안한 시계열 예측 모델과의 비교 실험을 진행한다. 비교 모델은 회귀 모형인 AR(Auto-Regressive), 인공신경망 모델인 LSTM, 일반 트랜스포머, 그리고 최근 연구인 LSTnet[24], SpringNet[19]으로 선정한다. 입력하는 시계열 길이 m은 30으로 고정하고, 출력 순차 값 n은 각 10, 20, 30을 사용한다. 예측 결과는 Table 3에 나

타나 있다. 본 논문에서 제시한 시계열 예측 모델은 Table 3에 Ours로 표기되어 있다. 각 실험에서 제일 좋은 성능을 낸 지표는 굵은 글씨로 표기되어 있다.

예측하고자 하는 순차의 길이가 짧은 경우 (n=10), 최근에 제시된 SpringNet이나 일반적인 Transformer의 성능이 다른 비교 모델들보다 제일 우수한 것을 볼 수 있다. 그러나 예측 순차의 길이가 긴 실험에서는 (n=20, 30) 본 연구에서 제안한 예측 모델의 다른 비교 모델들보다 성능이 더 우수한 것을 볼 수 있다.

Table 3. Comparison Results of Other Existing Prediction Models

RMSE	n	Traffic	Exchange	Electricity	Solar
AR	10	0.6189	0.0349	0.1217	0.6017
	20	0.8918	0.0407	<b>0.2365</b>	0.8329
	30	1.1152	0.0694	0.4379	0.9782
LSTM	10	0.5867	0.0357	0.1019	0.5905
	20	0.7988	0.0419	0.2571	0.8063
	30	0.8901	0.0584	0.4979	0.9487
Transformer	10	0.5381	0.0302	0.0985	0.6008
	20	0.8018	0.0386	0.2106	0.7902
	30	0.8842	0.0499	0.4868	0.8586
LSTnet	10	0.4912	0.0298	0.0879	0.5819
	20	0.6879	0.0417	0.2315	0.8013
	30	0.7596	0.0595	0.4908	0.9376
SpringNet	10	<b>0.4819</b>	<b>0.0237</b>	<b>0.0858</b>	<b>0.5797</b>
	20	0.7011	0.0308	0.2719	0.7998
	30	0.7892	0.0395	0.4552	0.9405
Ours	10	0.5017	0.0245	0.0910	0.5909
	20	<b>0.6237</b>	<b>0.0299</b>	0.2702	<b>0.7805</b>
	30	<b>0.7389</b>	<b>0.0353</b>	<b>0.4318</b>	<b>0.8804</b>
CORR	n	Traffic	Exchange	Electricity	Solar
AR	10	0.7550	0.9530	0.8602	0.8661
	20	0.7518	0.9368	0.8596	0.5465
	30	0.7498	0.9291	0.8504	0.4909
LSTM	10	0.9085	0.9547	0.8576	0.9173
	20	0.8818	0.9321	0.8407	0.8911
	30	0.8632	0.9291	0.8231	0.8708
Transformer	10	<b>0.9192</b>	<b>0.9600</b>	0.8712	0.9598
	20	0.8908	0.9418	0.8691	0.9418
	30	0.8717	0.9377	0.8349	0.9281
LSTnet	10	0.8658	0.9524	0.9101	0.9518
	20	0.8597	0.9338	0.9098	0.9421
	30	0.8438	0.9217	0.9035	0.9304
SpringNet	10	0.8719	0.9612	<b>0.9204</b>	<b>0.9623</b>
	20	0.8589	0.9409	0.9107	0.9420
	30	0.8317	0.9312	0.9026	0.9378
Ours	10	0.9098	0.9578	0.9197	0.9605
	20	<b>0.9075</b>	<b>0.9554</b>	<b>0.9195</b>	<b>0.9582</b>
	30	<b>0.9012</b>	<b>0.9539</b>	<b>0.9188</b>	<b>0.9557</b>

5. 결 론

본 논문에서는 정확한 시계열 예측을 위해 예측 순차 데이터를 한 번에 출력하는 새로운 구조의 트랜스포머 모델을 제시한다. 트랜스포머의 인코더에서 출력되는 콘텐츠(Content)를 입력한 시계열의 스타일(Style)로 해석한다. 기존의 연구들이 활용한 트랜스포머와 같이 디코더를 인코더와 비슷한 어텐션 기반 구조로 설계하는 것이 아닌, 스타일을 반영하는 생성 모델로 디코더를 설계한다. 인코더에서 추출한 스타일을 변환하여 디코더에 반영하고, 디코더는 여러 층을 통해 예측하고자 하는 시계열 순차를 출력한다.

본 연구가 제시한 시계열 예측 프레임워크는 기존의 CNN, RNN, LSTM과 같은 인공신경망을 기반으로 하는 시계열 예측 모델보다 정확도 측면에서 훨씬 우수한 성능을 보인다. 그리고 최근 연구에서 제시된 트랜스포머 기반의 예측 모델들보다 더 정확하게 장기 시점을 예측한다.

향후 연구에서는 인코더에서 시계열적 콘텐츠(Content)를 더 잘 추출할 수 있는 새로운 구조의 트랜스포머 모델을 설계하거나, 단변수 시계열 예측(Univariate Time Series Forecasting)이 아닌 다변수 시계열 예측(Multivariate Time Series Forecasting)에 활용될 수 있는 트랜스포머 모델을 설계하는 방향으로 본 논문을 확장할 예정이다.

References

- [1] A. Vaswani, et al., "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.
- [2] S. Li, et al., "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," *Advances in Neural Information Processing Systems*, Vol.32, pp.5243-5253, 2019.
- [3] L. Cai, K. Janowicz, G. Mai, B. Yan, and R. Zhu, "Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting," *Transactions in GIS*, Vol.24., No.3, pp.736-755, 2020.
- [4] F. Giuliani, I. Hasan, M. Cristani, and F. Galasso, "Transformer networks for trajectory forecasting," *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021.

- [5] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- [6] B. L. Bowerman and R. T. O'connell, "Time series and forecasting," North Scituate, MA: Duxbury Press, 1979.
- [7] J. Contreras, R. Espinola, F. J. Nogales, and A. J. Conejo, "ARIMA models to predict next-day electricity prices," *IEEE Transactions on Power Systems*, Vol.18, No.3, pp.1014-1020, 2003.
- [8] A. I. McLeod, "Diagnostic checking of periodic autoregression models with application," *Journal of Time Series Analysis*, Vol.15, No.2, pp.221-233, 1994.
- [9] R. B. Darlington and A. F. Hayes, "Regression analysis and linear models," New York, NY: Guilford, 2017.
- [10] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," *IEEE Transactions on Neural Networks*, Vol.5, No.2, pp.240-254, 1994.
- [11] R. J. Frank, N. Davey, and S. P. Hunt, "Time series prediction and neural networks," *Journal of Intelligent and Robotic Systems*, Vol.31, No.1, pp.91-103, 2001.
- [12] A. B. Geva, "ScaleNet-multiscale neural-network architecture for time series prediction," *IEEE Transactions on Neural Networks*, Vol.9, No.6, pp.1471-1482, 1998.
- [13] A. Tokgöz and G. Ünal, "A RNN based time series approach for forecasting turkish electricity load," *2018 26th Signal Processing and Communications Applications Conference (SIU)*, IEEE, 2018.
- [14] I. Koprinska, D. Wu, and Z. Wang, "Convolutional neural networks for energy time series forecasting," *2018 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2018.
- [15] C.-J. Huang and P.-H. Kuo, "A deep cnn-lstm model for particulate matter (PM<sub>2.5</sub>) forecasting in smart cities," *Sensors*, Vol.18, No.7, pp.2220, 2018.
- [16] J. Pamina and B. Raja, "Survey on deep learning algorithms," *International Journal of Emerging Technology and Innovative Engineering*, Vol.5, No.1, 2019.
- [17] J. J. Dabrowski, Y. Zhang, and A. Rahman, "ForecastNet: A time-variant deep feed-forward neural network architecture for multi-step-ahead time-series forecasting," *International Conference on Neural Information Processing*, Springer, Cham, 2020.
- [18] Y. G. Cinar, H. Mirisae, P. Goswami, E. Gaussier, A. Ait-Bachir, and V. Strijov, "Position-based content attention for time series forecasting with sequence-to-sequence rnns," *International Conference on neural Information Processing*, Springer, Cham, 2017.
- [19] Y. Lin, I. Koprinska, and M. Rana, "SpringNet: Transformer and spring DTW for time series forecasting," *International Conference on Neural Information Processing*, Springer, Cham, 2020.
- [20] N. Chen, S. Watanabe, J. Villalba, P. Želasko, and N. Dehak, "Non-autoregressive transformer for speech recognition," *IEEE Signal Processing Letters*, Vol.28, pp.121-125, 2020.
- [21] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.
- [22] I. Goodfellow, et al., "Generative adversarial nets," *Advances in Neural Information Processing Systems*, Vol.27, 2014.
- [23] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.
- [24] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long-and short-term temporal patterns with deep neural networks," *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018.
- [25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.



**김 동 건**

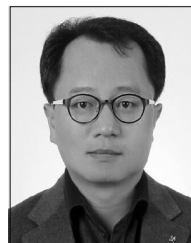
<https://orcid.org/0000-0001-5128-4689>

e-mail : kdk1996@skku.edu

2020년 성균관대학교 시스템경영공학과 (학사)

2020년 성균관대학교 전자전기공학부 (학사)

2020년 ~ 현 재 성균관대학교 소프트웨어학과 석사과정  
 관심분야 : Computer Vision, Machine Learning, Time Series Forecasting



**김 광 수**

<https://orcid.org/0000-0002-1097-8112>

e-mail : kwangsu.kim@skku.edu

1994년 한양대학교 전자공학과(학사)

2007년 University of Southern California, Computer Science(박사)

1994년 ~ 2019년 과학기술정보통신부 국장  
 2019년 ~ 현 재 성균관대학교 소프트웨어학과 교수  
 2019년 ~ 현 재 성균관대학교 성균인공지능융합원 원장  
 관심분야 : Computer Vision, Domain Adaptation, Federated Learning, AI Application