https://doi.org/10.7840/kics.2023.48.1.123

딥러닝 기반 시계열 예측 알고리즘의 모델 크기에 따른 성능 비교 및 분석

최 영 준[◆°], 김 대 근[*]

Comparison and Analysis for the Performance of Deep Learning-Based Time Series Prediction Algorithms According to Increasing Model Size

Youngjoon Choi[◆°], Daekeun Kim[*]

요 약

이 논문은 최신의 딥러닝 기반의 시계열 예측 알고리즘에서 언어 모델의 scaling laws가 적용되는지 실험적으로 확인해 보고자 하였다. 2개의 딥러닝 기반 시계열 예측 알고리즘을 통해 모델의 크기를 기존 알고리즘 대비 최대 24배까지 증가시켰으며, 실험 결과 기존 알고리즘의 state-of-the-art 성능보다 최대 3.6% 좋은 결과를 얻을 수 있었다. 또한, 언어 모델의 scaling laws에서 언급한 모델 크기와 데이터 크기의 관련성에 대해 시계열 예측 모델에서도 유사한 결과를 갖는다는 것을 실험적으로 확인할 수 있었다.

키워드 : 시계열 예측, 딥러닝, 인공지능, 스케일링 법칙, 세이지 메이커, 모델 병렬화
Key Words : Time series prediction, Deep learning, scaling laws, SageMaker, model parallelism

ABSTRACT

This paper deals with empirical research on if scaling laws of language models are be able to be applied to state-of-the-art deep learning-based time series forecasting algorithms. Two deep learning-based time series forecasting algorithms have increased the size of the model by up to 24 times over those created in the original papers, and empirical analysis shows up to 3.6% better than the state-of-the-art performance of one of the algorithms. In addition, similar to the results of the scaling law paper on language models, the results of this paper confirm that model size and data size have a meaningful relationship with the performance of time series forecasting algorithms.

Ⅰ. 서 론

시계열 예측은 통신 산업에서 트래픽 이상 징후 탐지[1] 또는 데이터 트래픽 예측[2] 등을 포함해서 금융, 우주 산업, 물류, 제조 등 다양한 산업에서 활용한다. 특히, 시계열 예측은 시간에 따라 변화하는 방대한 양의 데이터에서 주요 패턴을 찾아내게 되는데, 이런 패턴들 중 추세 패턴은 시간에 따라 선형 또는 지수 등

※ 이번 연구는 AWS의 Amazon SageMaker의 자원을 지원받아 실험을 진행함
◆° First and Corresponding Author : Amazon Web Services, choijoon@amazon.com, 정회원
* Amazon Web Services, daekeun@amazon.com, 정회원
논문번호 : 202210-233-C-RE, Received September 30, 2022; Revised November 9, 2022; Accepted November 9, 2022

으로 변화하는 패턴을 의미하며, 계절성 패턴은 주중, 주말, 월 등의 일자와 관련된 요소에 따라 영향을 받는 패턴이 존재하는 것을 의미한다. 또 다른 패턴들 중 하나인 주기성 패턴은 데이터의 주기적 변동이 불규칙적으로 장기간에 걸쳐서 발생하는 패턴을 의미한다.

과거부터 ARIMA[3], ETS[4], Prophet[5] 등의 전통적인 시계열 알고리즘도 많이 활용되고 있는데, 최근에는 GPU를 포함하여 CPU, 메모리 등의 컴퓨팅 리소스의 성능이 좋아지고, 활용 가능한 시계열 데이터가 증가함에 따라 딥러닝을 이용한 시계열 예측 알고리즘에 대해 많은 연구가 진행되고 있다. 특히, 많은 양의 데이터에 대해서는 딥러닝을 기반한 알고리즘이 전통적인 시계열 알고리즘보다 높은 성능을 나타내고 있으며, 최근 자연어 처리 분야와 컴퓨터 비전 분야에서는 GPT-3[6]와 DALL-E[7] 같은 모델 파라미터 크기를 크게 늘린 초거대 모델이 등장하면서, 모델 크기를 증가하는 것이 모델 성능에도 영향을 줄 수 있다는 실험이 다양하게 진행되고 있다.

이 논문은 이미 검증된 자연어 처리와 컴퓨터 비전 분야에서의 가설이 시계열에서도 적용될 수 있는지를 실험적으로 확인하고자 하는 것으로, 이 논문의 구성은 다음과 같다. II장은 딥러닝을 이용한 시계열 알고리즘에 대한 연구를 조사하고, 자연어 처리에서 실험적으로 수립한 모델 크기의 증가에 따른 모델 성능 관계에 대한 가설을 소개하며, III장은 시계열 데이터와 알고리즘을 이용하여 이런 가설을 실험적으로 확인한 다음, IV장에서 결론 및 성능 개선을 위한 향후 계획을 기술하고 있다.

Ⅱ. 본 론

2.1 딥러닝 기반 시계열 예측 알고리즘

딥러닝 기반의 시계열 예측 알고리즘은 데이터가 시간 축에 따라 의존적인 패턴을 가지고 있다는 측면에서 LSTM (Long Short-Term Memory) 기반의 DeepAR+ [8] 등이 우수한 성능을 내면서 등장하게 되었다. LSTM 기반 알고리즘은 이전 시점의 데이터에서 추출된 정보를 현 시점의 데이터 분석에 함께 활용하면서 시계열 패턴을 찾는 방식이다. 하지만, LSTM 기반의 알고리즘은 시계열의 시점 간격이 커지는 long sequence 시계열 예측에서 과거 시점의 영향력이 약해지는 Long-term dependency problem[9]을 가질 수 있다.

이후, Receptive field의 크기를 크게 가져가면서 Long-term dependency problem 문제를 보완하는 CNN-QR[10] 등의 CNN 기반 알고리즘이 등장하였다. 이 알고리즘은 Dilated Convolution[11]을 이용하면서 Long-term dependency problem을 보완하는 것이 가능해 졌다. 물론, CNN 기반 알고리즘은 시간 축의 데이터에 대해 convolution 필터를 이동시켜 가면서 이를 학습하는 방식으로, 과거와 현재의 시점 정보에 대해 time-invariant 하다는 가정을 기반으로 한다.

그리고, 최근 자연어 처리에서 우수한 성능을 보이고 있는 Transformer[12]를 시계열 데이터에 활용하고 있다. Transformer는 LSTM이나 CNN을 활용하지 않고 attention 매커니즘으로 모든 시점의 가중치를 활용한다. 이는 기존의 난제였던 long sequence 시계열 예측에도 적절하며, Informer[13]와 CoST[14] 등의 여러 연구 결과에서 이를 뒷받침하고 있다.

Transformer는 attention 메커니즘을 사용하는데, 서로 다른 시점의 시계열 데이터 간 관계를 바탕으로 정보의 가중치를 부여한 다음, 중요한 정보를 가진 데이터가 결과에 더욱 크게 반영할 수 있도록 모델 학습을 하게 된다. 하지만, attention 메커니즘은 다른 시점의 시계열 데이터 간 관계를 계산하기 위해 긴 길이의 시계열 데이터들 간의 matrix 연산을 해야 하므로, 연산 복잡도가 높고 방대한 크기의 matrix를 저장하기 위한 메모리가 많이 필요하다. 따라서, 이를 해결하기 위한 연산 복잡도를 낮추는 연구가 진행되면서, 이 연구 중 하나로 Informer 알고리즘이 낮은 연산 복잡도에도 높은 성능을 얻는 결과를 보여 주었다. 이후 추가적인 성능 향상을 위해 시계열 데이터에 대한 contrastive learning[15]을 적용한 CoST 알고리즘이 좋은 성능 결과를 연구를 통해 발표하였다.

2.1.1 Informer

Informer 알고리즘은 Transformer 기반 long sequence의 시계열 데이터에 대한 예측을 수행하는 알고리즘 중 하나이다. 시계열 데이터는 long sequence의 길이가 길어질수록 연산 복잡도가 증가하는 문제가 발생하는데, 이 알고리즘은 시계열 데이터의 특성 상 유의미한 sequence만 선택하여 attention 연산을 수행하는 ProbSparse self-attention 기법을 제시하였고, 이를 통해 시간 복잡도와 메모리 사용량을 줄일 수 있었다. 또한, 하나의 forward step만으로 long sequence output을 얻을 수 있도록 구성하였기에 step-by-step의 추론 과정에서 발생할 수 있는 누적 에러를 방지할 수 있었다.

2.1.2 CoST

CoST 알고리즘은 contrastive learning을 통해 계절성과 추세 패턴을 각각 학습하는 long sequence의 시계열 데이터를 예측하는 모델이다. 계절성 패턴은 Fourier Transform을 수행한 후, learnable Fourier layer를 통해 frequency 영역에서 상호 관계를 학습한 계절성 representation을 얻을 수 있게 된다. 계절성 representation의 강건성을 위해 하나의 데이터 샘플에 대해 scale, shift, jitter등의 데이터 증강을 진행하고, 동종 샘플의 증강은 positive pair로, 이종 샘플에서의 증강은 negative pair로 loss를 설정하여 학습을 수행한다. 추가적으로, 추세 패턴의 경우에는 다중의 1차원 causal convolution으로 구현한 다음, 이를 average pooling으로 결합하여, 다양한 양상의 추세들을 학습할 수 있도록 알고리즘을 구성하였다.

2.2 언어 모델의 scaling laws

한편, 언어 모델은 모델 크기와 모델 성능 관계에 대한 결과인 scaling laws[16]를 실험적으로 보여 주었다. 언어 모델의 성능은 모델의 크기, 데이터의 크기, 그리고 연산 능력에 영향을 받는다 점을 실험적인 결과를 통해 보여주고 있다. 특히, 모델 크기 (즉, 파라미터 개수)가 크게 증가할수록, 모델 예측 성능이 크게 향상되며, GPT-3와 같은 초거대 모델에서 실험을 통해 입증하였다. 물론, 데이터 크기의 증가 없이 모델의 파라미터 수만을 증가하게 되면 오버 피팅에 따라 성능이 오히려 나빠지는 결과도 확인할 수 있다. 따라서, $N^{0.74}/D$ 비율로 모델의 크기 N을 8배 증가할 때에는 데이터의 크기 D는 5배 이상 증가시켜야 오퍼 피팅에 대한 패널티를 피할 수 있다고 언급하고 있다.

Ⅲ. 시계열 알고리즘의 Scaling laws

3.1 실험 접근 방법

이 논문에서는 최근 가장 높은 성능을 보여준 딥러닝 기반 시계열 예측 알고리즘인 Informer와 CoST를 사용하여, 언어 모델 분야에서 실험적으로 입증된 scaling laws가 최신의 시계열 알고리즘에서도 유사한 결과를 나타내는지에 대해 실험적으로 진행해 보려고 합니다. 동등한 비교를 위해 알고리즘 연구에 사용한 동일 데이터 세트를 이용하여, 모델의 크기에 따른 성능 향상 여부를 확인하려고 한다.

3.2 실험 설정

CoST와 Informer 알고리즘에 대해 각각의 모델 크기에 따라 모델의 성능이 향상되는지를 비교하기 위해, 시계열 예측에서 사용하는 벤치마킹용 데이터 세트인 ETT[13] (Electricity Transformer Temperature)를 이용하고자 한다. ETT 데이터 세트는 6개의 features를 이용하여, 목표 값인 오일 온도를 예측하고자 하는 데이터 세트로, 이번 실험에서는 시간 단위의 데이터인 ETTh1를 이용해서 실험을 진행하였다. 데이터 세트는 원 논문과 동일한 설정으로 학습, 검증, 테스트 데이터를 60/20/20으로 시간 축으로 분리하였고, 검증 메트릭은 MSE (Mean Squared Error)와 MAE (Mean Absolute Error)를 사용한다.

실험은 Amazon SageMaker에서 ml.p4d.24xlarge (Nvidia A100 40GB 8장, Intel Xeon Cascade Lake 프로세서)를 사용하여, 각 GPU별로 모델 크기를 조정하면서 병렬로 작업을 수행하였다. Transformer 기반의 거대 모델은 단일 GPU 상에서 모델 학습 시, 학습할 파라미터를 저장하기 위한 GPU 메모리 공간이 부족한 out of memory 에러로 학습을 수행할 수 없다. 따라서, 이러한 거대 모델을 여러 GPU에 분할하여 모델 학습을 수행하는 Gpipe[18], DeepSpeed[19], Megatron[20] 등의 model parallelism 기법들이 연구되고 있다. Amazon SageMaker 또한 자체 model parallelism 기법을 가진 library를 그림 1과 같이 제공하고 있으며, Amazon SageMaker model parallelism[21]을 이용하여 pipeline parallelism[18]과 tensor parallelism[20]을 모두 지원하고 있다.

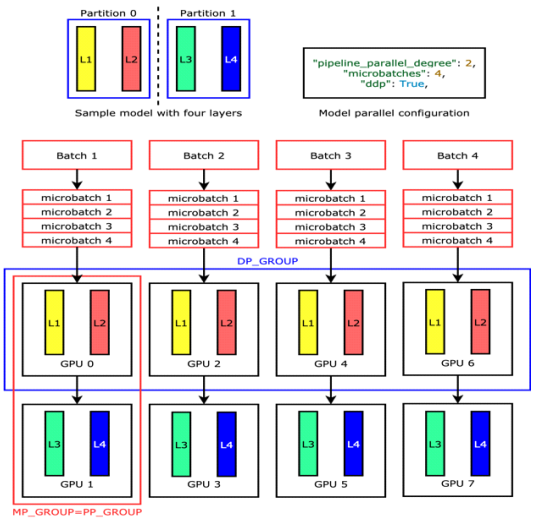


그림 1. Amazon SageMaker 모델 병렬화 구조[17]
Fig. 1. Amazon SageMaker model parallelelism

3.3 실험 결과

3.3.1 모델 크기에 따른 성능 결과

이번 실험에서는 CoST 모델의 입력 차원 크기인 hidden dimension과 입력 차원을 반복되는 횟수인 Depth, 출력 차원 크기인 representation dimension 크기를 변경하여 모델의 파라미터 수를 조정하였다. 이에 따른 표 1의 결과로, 모델의 크기가 증가할수록 MSE와 MAE가 점차 낮아지는 추세를 볼 수 있다. 결과를 보면, 기존 알고리즘의 SOTA (State-Of-The-Art) 성능을 얻은 모델 크기를 기준 (파라미터 비율 : 1.0)으로 파라미터 비율이 작은 경우에는 모델의 성능이 기존 논문보다 나빠졌지만, 반대로 파라미터 비율이 높은 경우에는 성능이 좋아지는 것을 볼 수 있다. 특히, 기존 논문 보다 24배 이상의 모델 파라미터를 사용한 경우에는 예측 길이가 24, 48, 168에 대해 기존 논문의 SOTA 성능보다 향상되었다. 특히, 예측 길이 24에서는 기존 알고리즘 대비 최대 3.6% 성능 향상의 결과를 얻을 수 있었다.

이는 모델 크기가 증가하면서 모델이 데이터의 정보를 더 많이 수용할 수 있는 모델 용량을 가질 수 있다. 또한, CoST모델에서 contrastive learning을 학습할 때 사용하는 representation features의 길이를 함께 증가시켰기에, features에 더 많은 정보가 포함되어 모델 학습 시 반영된 결과로 판단할 수 있다.

표 1. CoST의 모델 크기에 따른 성능 결과
Table 1. Performance results based on model size of CoST

hyperparameters			ratio of model size	Metrics	prediction length				
hidden dims	depth	repr. dims			24	48	168	336	720
8	5	160	0.25	MSE	0.434	0.476	0.680	0.846	0.999
32	10	320	0.99		0.400	0.446	0.654	0.835	0.986
64	10	320	1.00		0.383	0.435	0.636	**0.813**	0.960
128	15	640	4.02		0.378	0.431	0.639	0.827	0.981
256	20	960	9.31		0.376	0.428	0.632	0.825	0.997
512	25	1280	17.9		0.376	0.431	0.640	0.818	**0.952**
1024	25	1280	24.3		**0.370**	**0.427**	**0.631**	0.814	0.954
8	5	160	0.25	MAE	0.466	0.492	0.601	0.689	0.773
32	10	320	0.99		0.442	0.471	0.590	0.689	0.777
64	10	320	1.00		0.427	0.463	0.581	0.681	0.769
128	15	640	4.02		0.425	0.459	0.580	0.682	0.775
256	20	960	9.31		0.423	0.457	**0.577**	0.680	0.779
512	25	1280	17.9		0.423	0.459	0.584	0.682	**0.768**
1024	25	1280	24.3		**0.417**	**0.456**	**0.577**	**0.678**	**0.768**

표 2. Informer의 모델 크기에 따른 성능 결과
Table 2. Performance results based on model size of Informer

Hyperparameters			ratio of model size	Metrics	prediction length				
no, of encode layers	no, of decode layers	d-model			24	48	168	336	720
2	1	512	1.00	MSE	0.554	**0.649**	1.090	1.394	1.427
2	1	1024	2.88		0.526	0.663	**1.015**	**1.041**	1.392
2	1	1536	5.65		0.620	0.721	1.104	1.132	**1.346**
4	2	2048	19.69		0.645	0.764	1.175	1.352	1.374
2	1	512	1.00	MAE	0.535	**0.610**	0.850	0.983	0.982
2	1	1024	2.88		**0.519**	0.611	**0.801**	**0.814**	0.980
2	1	1536	5.65		0.597	0.647	0.851	0.834	0.967
4	2	2048	19.69		0.564	0.662	0.857	0.940	**0.949**

표 2의 경우, 모델의 encoder 계층의 수와 decoder 계층의 수, 모델의 차원 크기인 d-model을 통해 모델 파라미터 수를 조정하였다. 이에 따른 실험 결과로 기존 알고리즘 성능 대비 2.88 배의 모델 크기를 가질 때, 여러 예측 길이 중 다수에서 가장 좋은 성능 결과를 얻을 수 있었다. 하지만, 이후 모델 크기를 계속 증가하면서 모델 크기가 기존 알고리즘 대비 5배 이상 증가했을 때 성능 결과가 기존 알고리즘의 성능 대비 점차 낮아지게 되며, 파라미터 비율이 19배 이상 되었을 때는 MSE의 경우 모든 예측 길이에서 성능이 저하되는 것을 확인하였다. 이는 데이터의 크기를 늘리지 않은 상황에서 모델의 크기만 늘렸을 때 오버 피팅에 빠졌음을 의미한다.

3.3.2 데이터 크기에 따른 성능 결과

ETTh1 데이터에서 학습에 활용되는 데이터 수는 17,420개이며, 이 중 60%인 8,640개를 학습 데이터로 활용하고 있다. 또 다른 시계열 데이터로는 미국 내 1,600 여개 지역의 11개 기후 특성을 시간 별로 측정한 Weather 데이터 세트가 있으며, 데이터의 크기는 35,064개로 이 중 21,038개를 학습 데이터에 사용한다. Weather 데이터 세트는 ETTh1 데이터 세트 대비 해서 2.4배 큰 데이터의 크기를 가지고 있으며, 이 데이터 세트를 이용하여 CoST와 Informer 알고리즘 중에 대해 예측 길이 별 가장 높은 성능을 보여주는 모델 크기의 비율을 표시하였다.

그림 2와 같이 일부의 예측 길이를 제외하고 대다수에서 데이터 크기가 2.4배가 될 경우 기존 대비 더 큰 모델 파라미터에서 높은 성능을 내는 것을 알 수

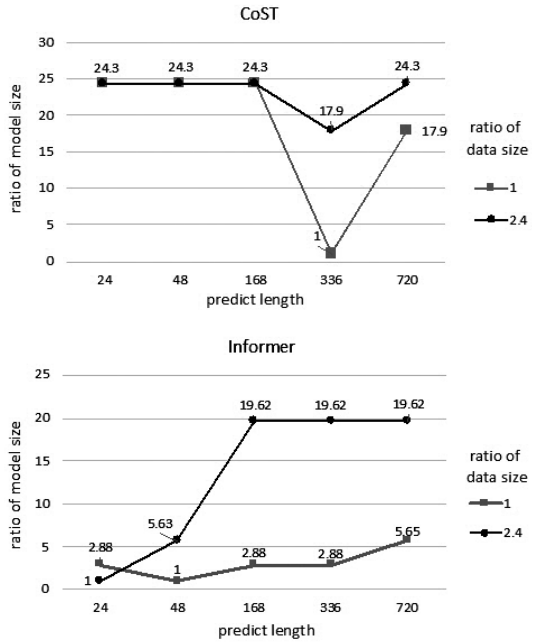


그림 2. 데이터 크기 비율에 따른 최고 성능의 모델 크기 비율
Fig. 2. Highest performance the ratio of model size based on the ratio of data size

있다.

CoST 알고리즘의 경우에는 ETTh1 데이터 세트와 동일하게 Weather 데이터 세트에서도 모델 크기가 큰 경우 더 좋은 성능을 보여주는 것을 실험적으로 확인했다. 특히, 예측 길이가 336일 때 ETTh1에서의 최고 성능은 기존 알고리즘과 동일한 모델 크기일 때였지만, 데이터의 크기가 2.4배 커진 Weather 데이터 세트

표 3. Weather 데이터의 모델 크기에 따른 MSE 결과
Table 3. MSE based on model size for Weather dataset

Model	ratio of model size	prediction length				
		24	48	168	336	720
CoST	0.25	0.323	0.382	0.473	0.500	0.538
	0.99	0.299	0.360	0.466	0.500	0.535
	1.00	0.299	0.361	0.465	0.497	0.532
	4.02	0.294	0.357	0.463	0.495	0.531
	9.31	0.293	0.354	0.462	0.494	0.527
	17.9	0.292	0.353	0.462	**0.492**	0.527
	24.3	**0.292**	**0.352**	**0.462**	0.493	**0.526**
Informer	1.00	**0.327**	0.399	0.645	0.676	0.653
	2.88	0.330	0.415	0.636	0.671	0.658
	5.63	0.336	**0.405**	0.624	0.684	0.671
	19.62	0.349	0.421	**0.609**	**0.643**	**0.651**

의 경우에는 모델의 크기가 기존 알고리즘 대비 17.9배 커진 경우에 가장 높은 성능을 보여주고 있다는 점에서 모델 크기와 데이터 크기를 높일 경우 모델의 성능을 더욱 향상할 수 있다는 것을 알게 되었다.

추가로 Informer 알고리즘의 경우 데이터 크기가 2.4배 큰 Weather 데이터 세트에서 예측 길이가 24일 때를 제외하고 모두 모델 크기가 큰 경우 더 좋은 성능을 보여주는 것을 확인했다. 표 3은 Weather 데이터 세트에 대해 CoST와 Informer 알고리즘에 대한 Mean Squared Error의 값을 보여주고 있으며, 각 모델 크기 별 예측 길이에 대한 상세한 성능 결과를 볼 수 있다.

Ⅳ. 결 론

본 논문에서는 딥러닝 기반의 시계열 예측 모델에 대해 기존 알고리즘 대비 모델 크기를 증가시켜서 더 좋은 성능을 얻을 수 있다는 것과 함께, 언어 모델에서 나타난 scaling laws를 시계열 예측 모델에서도 유사하게 나타난다는 것을 실험적으로 확인해 보았다. 추가적으로, 데이터 크기를 증가했을 때 더 큰 모델 크기를 가진 시계열 예측 모델에서 더 좋은 성능을 보여준다는 실험적 결과를 얻을 수 있었다. 그리고, 이를 통해 딥러닝 기반의 시계열 예측 모델 또한 언어 모델과 유사한 scaling laws의 경향을 보인다는 점을 알게 되었다.

향후, 다양한 시계열 예측 모델에 대한 유사 실험을 추가적인 진행하여 가설에 대한 증명을 계속 진행할 예정이며, 특히 데이터가 부족한 경우 모델 크기를 키웠을 때 학습을 안정적으로 진행할 수 있는 기법들에 대한 추가적인 연구를 앞으로 진행할 계획이다.

References

[1] K. H. Cho and D. H. Lee, “A study on traffic anomaly detection scheme based time series model,” *J. KICS*, vol. 33, no. 5, pp. 304-309, May 2008.

[2] Q. H. Do, et al., “Prediction of data traffic in telecom networks based on deep neural networks,” *J. Computer Sci.,* vol. 16, no. 9, pp. 1268-1277, 2020.

[3] George E. P. Box, et al., “*Time series analysis: Forecasting and control*,” John Wiley & Sons, May 2015.

[4] E. S. Gardner Jr., "Exponential smoothing: The state of the art," *J. Forecasting*, vol. 4, no. 1, pp. 1-28, 1985.

[5] S. J. Taylor and B. Letham, "Forecasting at scale," *The Am. Statistician*, vol. 72, no. 1, pp. 37-45, Jan. 2018.

[6] T. Brown, et al., "Language models are few-shot learners," *Advances in NIPS*, vol. 33, pp. 1877-1901, 2020.

[7] A. Ramesh, et al., "Zero-shot text-to-image generation," *Int. Conf. Mach. Learn.,* PMLR, pp. 8821-8831, Jul. 2021.

[8] D. Salinas, et al., "DeepAR: Probabilistic forecasting with autoregressive recurrent networks," *Int. J. Forecasting,* vol. 36, no. 3, pp. 1181-1191, 2020.

[9] Y. Bengio, P. Frasconi, and P. Simard, "The problem of learning long-term dependencies in recurrent networks," *IEEE Int. Conf. Neural Netw.*, vol. 3, pp. 1183-1188, Mar. 1993.

[10] R. Wen, et al., "A multi-horizon quantile recurrent forecaster," *arXiv preprint arXiv: 1711.11053*, Nov. 2017.

[11] I. K. Hassani, T. Pellegrini, and T. Masquelier, "Dilated convolution with learnable spacings," *arXiv preprint arXiv: 2112.03740*, 2021.

[12] A. Vaswani, et al., "Attention is all you need," *Advances in NIPS*, vol. 30, 2017.

[13] H. Zhou, et al., "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proc. AAAI*, 2021.

[14] G. Woo, et al., "CoST: Contrastive learning of disentangled seasonal-trend representations for time series forecasting," *arXiv preprint arXiv: 2202.01575,* 2022.

[15] E. Eldele, et al., "Time-series representation learning via temporal and contextual contrasting," *arXiv preprint arXiv:2106.14112*, 2021.

[16] J. Kaplan, et al., "Scaling laws for neural language models," *arXiv preprint arXiv: 2001.08361,* 2020.

[17] Guide, "Introduction to Model Parallelism," *Aamzon SageMaker*, 2021.

[18] Y. Huang, et al., "Gpipe: Efficient training of giant neural networks using pipeline parallelism," *Advances in NIPS*, vol. 32, 2019.

[19] J. Rasley, et al., "Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters," in *Proc. 26th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining*, pp. 3505-3506, 2020.

[20] M. Shoeybi, et al., "Megatron-lm: Training multi-billion parameter language models using model parallelism," *arXiv preprint arXiv: 1909.08053*, 2019.

[21] C. Karakus, et al., "Amazon SageMaker model parallelism: A general and flexible framework for large model training," *arXiv preprint arXiv:2111.05972,* 2021.

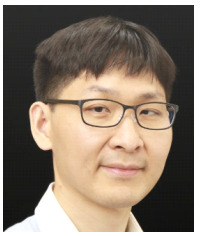
최 영 준 (Youngjoon Choi)

2003년 2월 : 고려대학교 기계공학과 졸업

2019년 8월 : 고려대학교 빅데이터융합과 석사

<관심분야> 인공지능, 로봇공학, 시계열예측

김 대 근 (Daekeun Kim)



2007년 12월 : 위스콘신대학교 컴퓨터과학과, 수학과 졸업

2012년 4월 : 미시간대학교 컴퓨터과학과 석사

<관심분야> 인공지능, 컴퓨터비전, 자연어처리