

# 베이지안 최적화를 이용한 이동 경로 예측 모델의 성능 개선

송하윤, 남세현  
홍익대학교 컴퓨터공학과

hayoon@hongik.ac.kr, ntommy11@naver.com

## Improving Trajectory Pattern Prediction Model Using Bayesian Optimization

Ha Yoon Song, Sehyun Nam  
Dept. of Computer Engineering, Hongik University

### 요 약

하이퍼파라미터(초매개변수) 최적화란 모델의 학습에 앞서 미리 설정해야 하는 값인 하이퍼파라미터의 최적값을 탐색하는 문제이다. 이때의 최적값은 학습을 끝낸 모델의 성능을 가능한 최대치로 높여주는 값이다. 한편, 최근 모바일 장치를 이용한 포지셔닝 데이터의 대량 수집이 가능해지면서 이를 활용하여 위치 기반 서비스(Location-Based Service)를 위한 데이터 분석 및 예측에 관한 연구가 활발히 이루어졌다. 그중 이동 경로를 이미지로 패턴화하여 국소 지역 내에서 다음 위치를 예측하는 CNN 모델에 대해서 하이퍼파라미터 튜닝을 진행하였다. 결과적으로 베이지안 최적화(Bayesian Optimization)를 통해 모델의 성능을 평균 3.7%, 최대 9.5%까지 개선할 수 있음을 확인하였다.

### 1. 서론

기계학습 모델의 성능은 하이퍼파라미터 값들의 구성에 따라 크게 달라진다. 여기서 하이퍼파라미터란 모델의 학습에 앞서 미리 설정해야 하는 값이다. 대표적으로 학습률, 모멘텀 계수 등이 있다. 그런데 이러한 값들을 설정하는 것은 도메인에 대한 사전 지식과 통찰은 물론이고 많은 시간과 자원이 요구된다. 이를 해결하는 방법에 대한 연구로 Grid Search나 Random Search[1] 등이 제안되었지만, 최적화 과정에서 얻은 값들을 활용하지 않고 매번 새로운 시도를 한다는 문제점이 제기되었다. 베이지안 최적화 [2]는 최적화 과정에서 얻은 값들을 활용하여 목적 함수의 값의 분포를 확률적으로 추론한다. 이에 따라 최적화가 진행되면서 확률상 최적값이 존재할 가능성이 높은 지역을 탐색하게 되고, 더 적은 시간과 비용으로 최적의 하이퍼파라미터 셋을 찾도록 돕는다.

하이퍼파라미터 최적화는 어떠한 도메인의 기계학습 모델에도 적용될 수 있다. 특히 베이지안 최적화는 블랙박스 형태의 목적함수를 대상으로 하기 때문에 다양한 분야에 활용된다. 최근의 연구 동향은 이

미지 처리를 위한 모델에 이를 적용하는 사례가 주를 이루었다. 본 연구에서는 포지셔닝 데이터를 이용해 개인의 이동 경로를 예측하는 모델에 대해서 실험을 진행함으로써 베이지안 최적화의 적용사례를 확장하고자 했다.

### 2. 관련 연구 및 배경

#### 2.1 베이지안 최적화

베이지안 최적화는 표현식을 알지 못하는 미지의 목적함수(objective function)  $f$ 에 대해 그 함수의 값  $f(\lambda)$ 를 최대(또는 최소)로 만드는 최적의 입력값  $\lambda^*$ 를 찾는 방법론이다. 딥러닝 분야에서는 주로 손실 함수(loss function)의 최소화를 목적으로 한다.

$$\lambda^* = \operatorname{argmax}_{\lambda} f(\lambda) \quad (1)$$

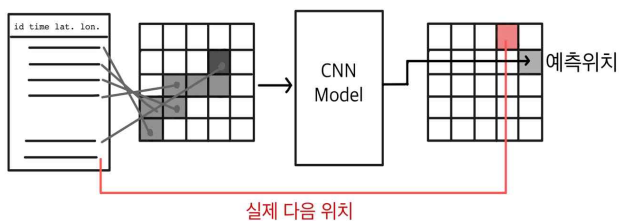
보통은 하나의 입력에 대해  $f$ 의 함수값을 계산하는 것이 오래 걸리는 경우를 상정하기 때문에, 입력값의 후보군을 줄이고 각 후보가 낸 결과들을 활용해 최적해  $\lambda^*$ 를 효율적으로 찾아내는 것이 핵심이다.

베이지안 최적화를 진행하기 위해서 두 가지 요소를 정의해야 한다. 첫째는 Surrogate Model로, 시

점  $t$ 까지의 결과들  $(\lambda_1, f(\lambda_1)), \dots, (\lambda_t, f(\lambda_t))$ 을 바탕으로 목적 함수의 생김새를 확률적으로 추정한다. 다음으로 Acquisition Function은 Surrogate Model의 추정 결과를 이용해서 다음 시점에 입력할 후보  $\lambda_{t+1}$ 를 추천하는 역할을 한다. 이 두 가지 요소는 베이지안 최적화의 명세일 뿐이므로 각각의 세부 구현 방식은 다를 수 있다. 일반적으로 Surrogate 모델로는 함수에 대한 확률분포를 추정하는 Gaussian Process 기법이 사용되고, Acquisition Function으로는 Monte-Carlo 기반의 함수가 사용된다. 본 연구에서도 이 방식을 채택했다.

## 2.2 CNN을 이용한 이동 경로 예측

심층신경망을 이용해서 이동패턴을 분석 및 추천하는 몇몇 연구가 발표되었다. 본 연구는 선행 연구 [3]에서 제안된 모델을 대상으로 하이퍼파라미터 최적화를 진행했다. 이 모델은 시계열 위치 데이터로부터 생성된(맵핑된) 패턴 이미지를 입력으로 받고 마지막 시점을 기준으로 다음 위치를 예측한다. 이때 이미지의 국소적 패턴성을 인식하는데 좋은 성능을 보이는 CNN(Convolutional Neural Network) 구조를 이용한다. 특히 신경망 구조의 층수를 5개 이하로 설정했는데, 이처럼 depth가 깊지 않은 모델의 경우 적은 컴퓨팅 자원으로 베이지안 최적화의 성능 향상을 실험할 수 있다는 이점이 있다.



(그림 1) CNN 모델의 예측 및 훈련 과정

## 3. 실험방법

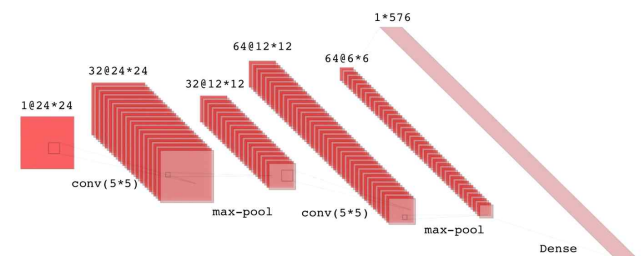
### 3.1 데이터셋

그림 1에서 보이듯, 학습을 위한 패턴 이미지 생성 및 정답(실제 다음 위치)을 레이블링하기 위해서는 시계열 위치 데이터가 필요하다. 이전 연구의 실험[3]에서는 직접 데이터를 수집하여 진행했지만, 실

험 데이터의 일반성을 확보하기 위해 마이크로소프트사에서 연구목적으로 제공하는 GeoLife GPS Trajectories 데이터[4]를 활용했다. 이 자료는 182명의 지원자가 수개월에 걸쳐 수집한 시계열 위치 데이터로 구성되어있다. 여기서 데이터의 양에 따른 개선 효과의 차이를 분석하기 위해 몇몇 지원자를 선별했다. 이후 각 지원자의 이동 경로 패턴 이미지를 생성하고 정답 레이블을 구성해서 전처리가 완료된 데이터셋을 준비한다.

### 3.2 모델 선정

학습을 위한 모델로는 기존 연구 [3]에서 가장 우수한 성능을 보였던 Conv(5×5)-MP - Conv(5×5)-MP - FNN(576) 구조를 선택했다. 이 모델은 24×24 크기의 패턴 이미지를 입력으로 받고 24×24 크기의 전결합층을 출력으로 한다. 이 출력에 대한 Cross-Entropy 함수를 통해 예측값과 실제값의 차를 계산하고 손실을 측정하게 된다. 이 손실 함수가 최적화의 대상인 목적함수  $f$ 가 된다.



(그림 2) 실험에 이용된 CNN 모델 아키텍처

### 3.3 베이지안 최적화 알고리즘의 적용

베이지안 최적화에 앞서 튜닝할 하이퍼파라미터를 선정해야 한다. 본 연구에서는 신경망 모델의 훈련에 있어서 가장 중요한 요소인 학습률(learning rate)과 관성계수(momentum)을 대상으로 실험을 진행했다. 이때 하이퍼파라미터 집합  $\lambda$ 를 식 (2)와 같이 설정했다.

$$\lambda = \{lr[10^{-6} : 0.4], momentum[0.0 : 1.0]\} \quad (2)$$

각각의 하이퍼파라미터는 미리 지정한 범위를 갖는다. 학습률과 모멘텀 계수의 경우 바운드 값을 지정하게 되는데 최적화 과정에서 이 구간 내의 연속된 범위를 탐색하게 된다.

다음으로 목적함수  $f$ 에 대해 조사할 점  $(\lambda, f(\lambda))$

**Algorithm 1** Bayesian Optimization's algorithm

```

1: procedure BAYESIANOPTIMIZER( $f, \lambda; N$ )
2:   set  $\lambda_1$  and compute  $f(\lambda_1)$ 
3:   for  $i = 1, \dots, N$  do
4:      $\text{SurrogateModel}(f, \{(\lambda_1, f(\lambda_1)), \dots, (\lambda_i, f(\lambda_i))\})$ 
5:      $\lambda_{i+1} \leftarrow \text{AcquisitionFunction}(\text{SurrogateModel})$ 
6:     compute  $f(\lambda_{i+1})$ 
7:   end for
8:    $\lambda^* \leftarrow \arg \max_{\lambda} (f(\lambda))$ 
9:   return  $\lambda^*$ 
10: end procedure

```

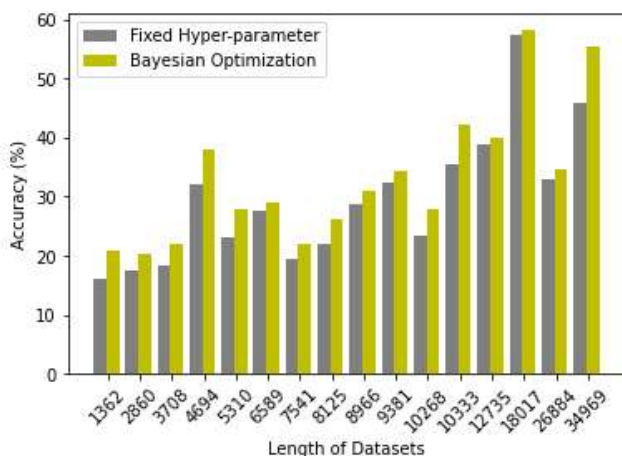
(의사코드 1) 베이지안 최적화 알고리즘

의 개수  $N$ 을 지정한다. 본 연구에서는 조사할 최대 점의 개수를 20으로 정하였다. 즉, 첫 번째 점부터  $(\lambda_1, f(\lambda_1)), \dots, (\lambda_i, f(\lambda_i))$ 에 대하여 Surrogate Model의 추정을 진행하고 이를 바탕으로 Acquisition Function을 통해  $\lambda_{i+1}$ 의 값을 얻어낸다. 이후 Surrogate Model에  $(\lambda_{i+1}, f(\lambda_{i+1}))$ 을 추가하고  $i = 20$ 이 될 때까지 이 과정을 반복한다. 이 과정은 의사코드 1로 표현될 수 있다.

**4. 결과 및 분석**

베이지안 최적화의 성능 향상을 보이기 위해 고정된 값의 하이퍼파라미터 집합  $\lambda_{\text{fixed}}$ 을 대조군으로 설정하였다. 식 (3)의 하이퍼파라미터 값은 신경망 학습 시 일반적으로 시도되는 값으로, 학습률은 0.001, 모멘텀 계수는 0.0이다.

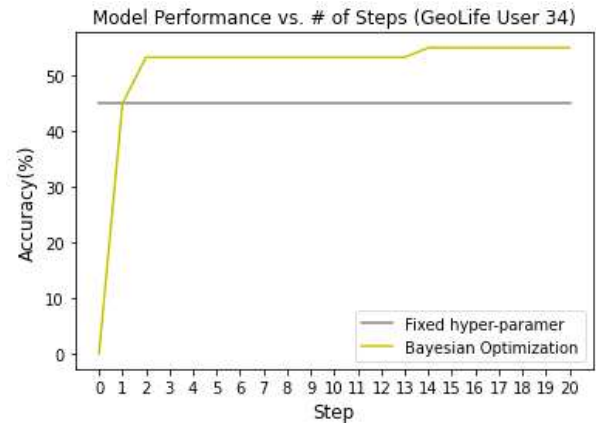
$$\lambda_{\text{fixed}} = \{lr = 0.001, \text{momentum} = 0.0\} \quad (3)$$



(그림 3) 베이지안 최적화에 따른 성능 향상

일반적으로 학습 데이터셋의 크기가 커질수록 정확도도 상승하는 경향이 있었으나, 베이지안 최적화

의 성능 향상 폭은 그 크기와는 관련이 없는 결과를 보였다. 이는 최적화의 알고리즘 자체가 목적 함수 추론 과정에서 부분적으로 무작위성(randomness)을 내포하고 있기 때문에 각 사례에 대해 서로 다른 성능 향상을 보인 것으로 해석된다. 결과적으로 베이지안 최적화는 모델의 성능(정확도)을 평균 3.7%, 최대 9.5%까지 개선할 수 있었다.



(그림 4) 최적화 단계별 모델의 성능

또한, 그림 4에서 보이듯 베이지안 최적화의 단계수를 나타내는 매개변수  $N$ 에 도달하기 훨씬 이전에 최적값에 가까운 하이퍼파라미터 집합을 찾는 경우들을 발견할 수 있었다. 즉,  $N$ 처럼 하이퍼파라미터 최적화에 사용되는 매개변수 역시 최적화의 대상이 될 수 있고, 향후 연구의 필요성이 제기된다.

**5. 결론**

본 논문은 심층신경망 모델의 하이퍼파라미터 튜닝 방법으로 제안된 베이지안 최적화를 위치 기반 서비스 영역에 적용하려는 시도이다. 이를 위해 이동 경로 패턴에 대해 예측을 수행하는 신경망 모델을 대상으로 최적화를 진행하였고, 평균 3.7%, 최대 9.5%의 성능 향상을 확인했다. 이후 해당 영역의 연구에서 이와 같은 최적화 방법이 계속해서 적용될 수 있을 것으로 기대된다.

**사사**

이 연구는 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행됨(NRF-2019R1F1A1056123).

### 참고문헌

- [1] James Bergstra, Yoshua Bengio. “Random Search for HyperParameter Optimization”, Journal of Machine Learning Research 13 (2012) 281-305.
- [2] Peter I. Frazier. “A Tutorial on Bayesian Optimization”, arXiv: 1807.02811v1 [stat.ML] 8 Jul 2018.
- [3] Dabin You, Ha Yoon Song. “Trajectory Pattern Construction and Next Location Prediction of Individual Human Mobility with Deep Learning Models“, Journal of Computing Science and Engineering, Vol. 14, No. 2, June 2020, pp. 52-65
- [4] Microsoft, <https://www.microsoft.com/en-us/research/publication/geolife-gps-trajectory-dataset-user-guide/>