

3 조 3 주차 발표자료_Bayesian Optimization

1. Auto ML

머신러닝으로 설계하는 머신러닝

1. Automated Feature Learning
 - 피쳐 엔지니어링
 - 학습 모델에 입력을 그대로 사용하지 않고, 유의미한 피쳐를 추출해서 입력으로 사용하는 방법
 - Automated Feature Learning
 - 최적의 피쳐 추출방법을 학습을 통해 찾는 방법
2. Architecture Search
 - 학습을 통해 최적의 아키텍처를 설계하는 방법
 - 강화학습, 유전 알고리즘 등을 이용한 연구, gradient 기반으로 한 DARTS
3. Hyperparameter Optimization
 - 학습을 시키기 위해 필요한 하이퍼파라미터를 학습을 통해 추정하는 것
 - 학습률, 배치크기 등 학습에 큰 영향을 주는 하이퍼 파라미터들을 기존 수동적 조정에서 나아가 학습을 통한 최적의 하이퍼파라미터 추정

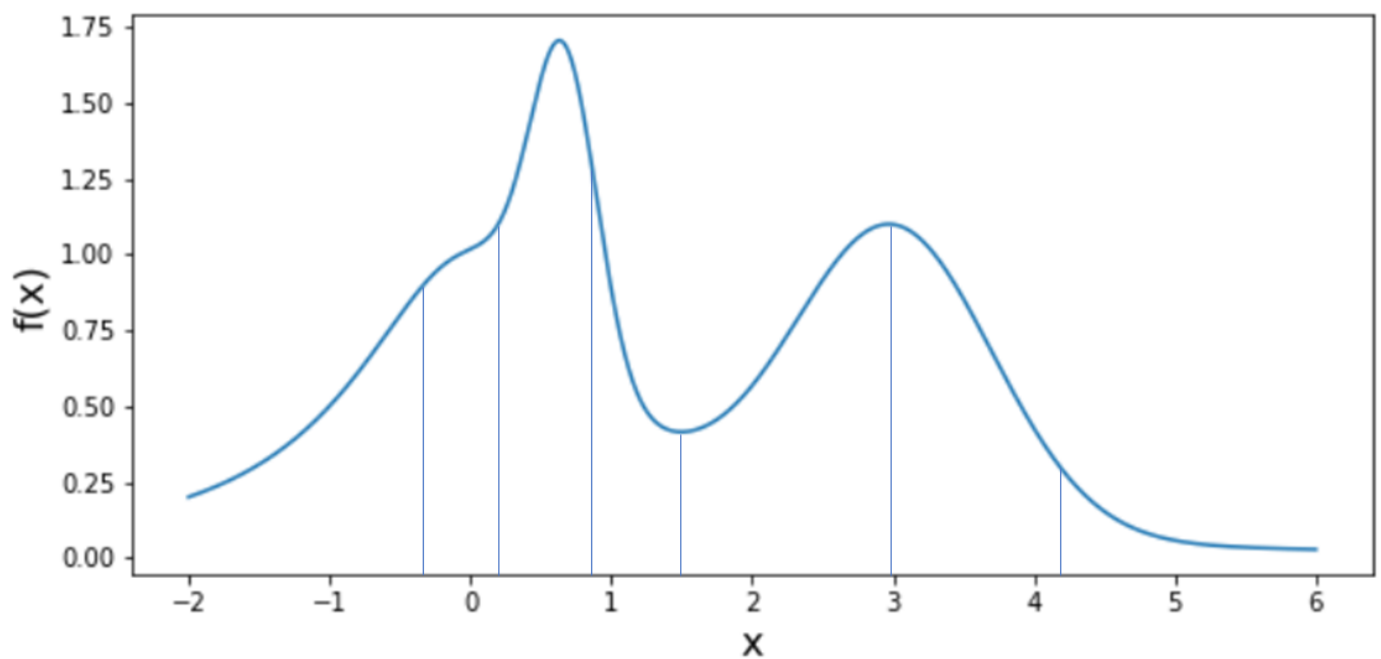
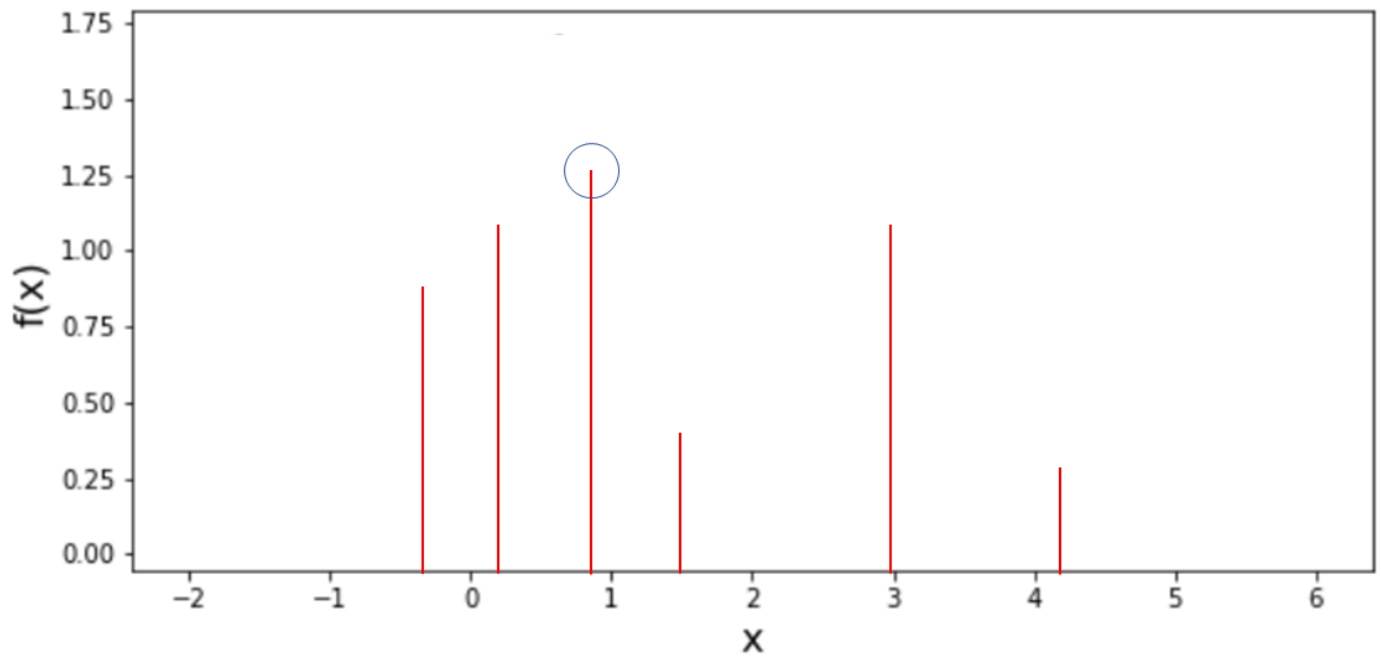
2 Hyperparameter Optimization

학습을 수행하기 위해 사전에 설정해야 하는 것인 하이퍼파라미터의 최적값을 탐색하는 문제를 지칭

- 하이퍼파라미터의 최적값 : 학습이 완료된 러닝 모델의 일반화 성능을 최고 수준으로 발휘하도록 하는 하이퍼파라미터의 값

2.1 Manual Search

수동적으로 최적 파라미터 값을 탐색하는 방법

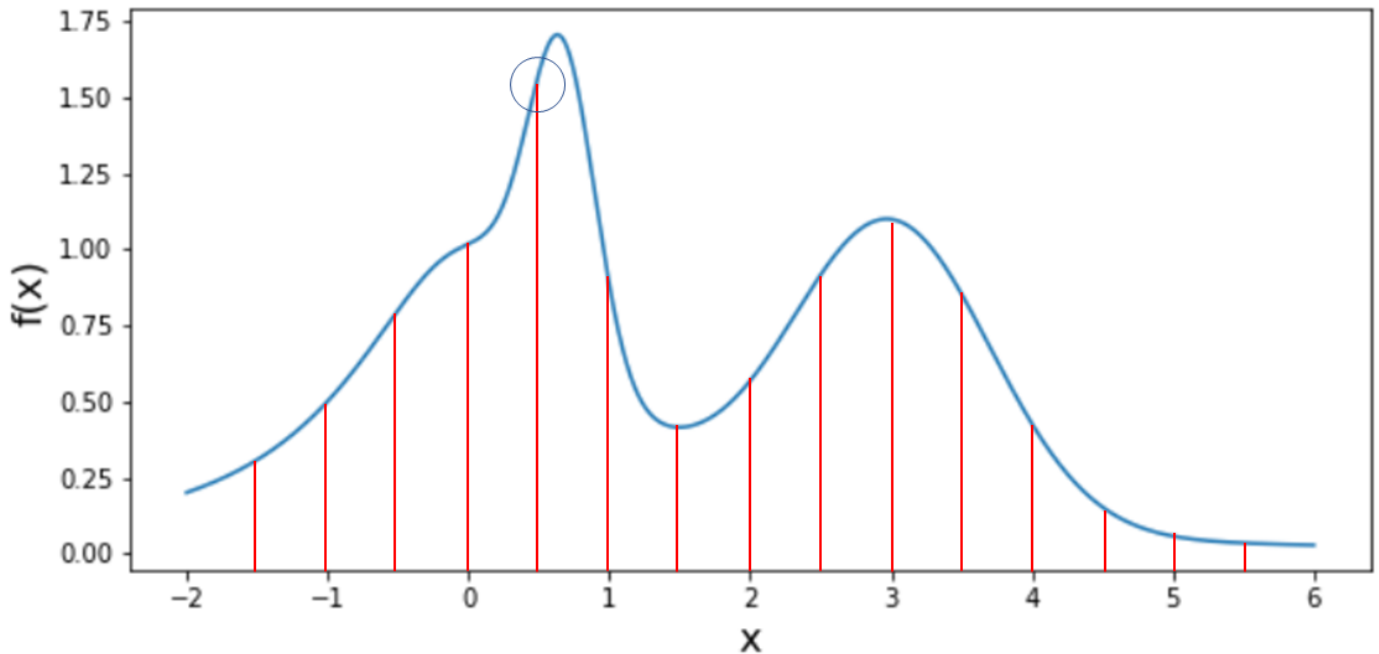


- 주관과 직관에 기반
- 실제 '최적값'인가?
- 한번에 여러 종류의 하이퍼파라미터들을 동시에 탐색할 때 더욱 복잡 (상호 영향 관계)

$$L(W) = \frac{1}{N} \sum_{i=1}^N L_i(f(x_i, W), y_i) + \lambda \cdot R(W)$$

2.2 Grid Search

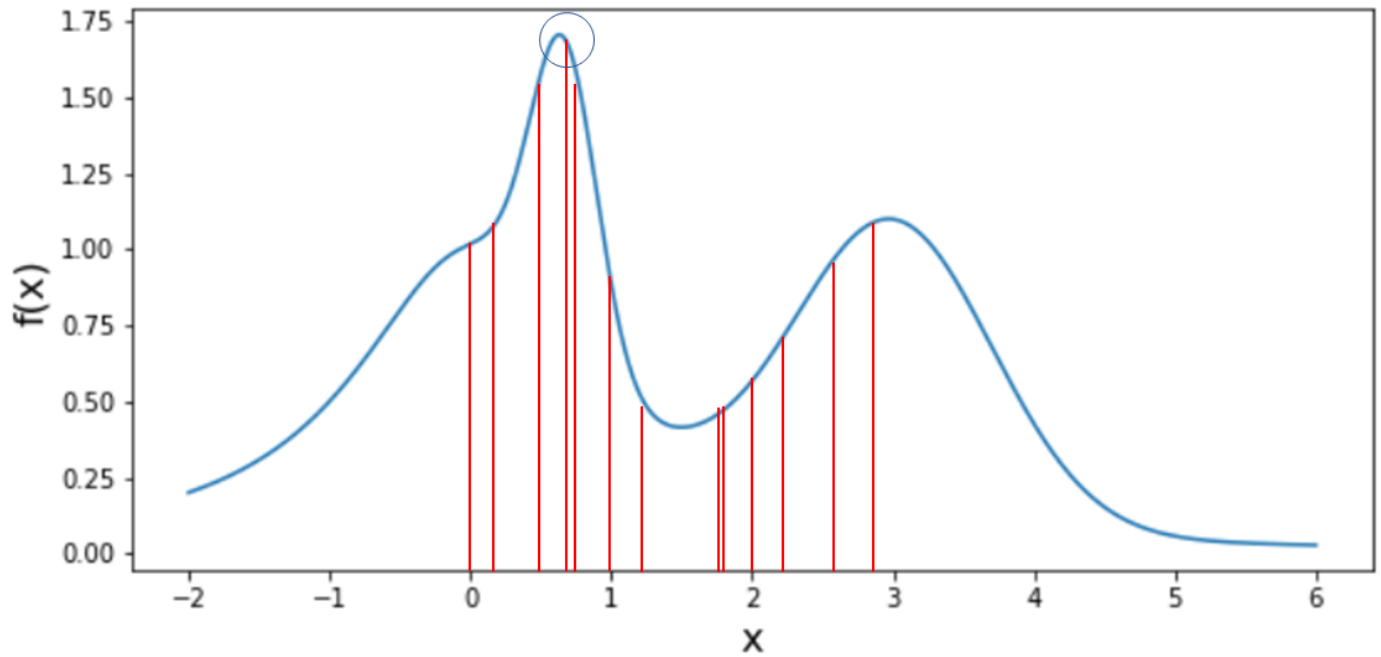
탐색 대상 구간 내 후보 하이퍼파라미터값들을 일정한 간격을 두고 선정, 가장 높은 성능을 발휘했던 하이퍼파라미터 값 최종 선정



- 수동적 요소 : 전체 탐색 대상 구간의 설정 방법, 간격의 길이 설정 방법 등
- 균등하고 전역적인 탐색 가능
- 하이퍼파라미터 개수를 늘리게 되면 탐색 연산 비용 증가(기하급수적으로,,)

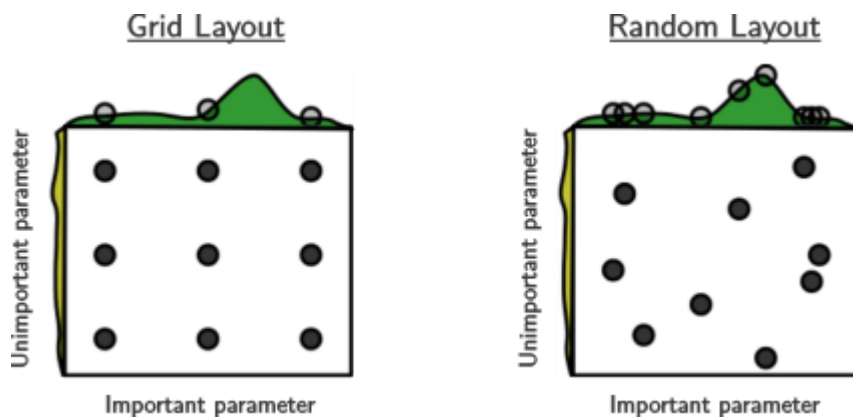
2.3 Random Search

탐색 대상 구간 내 후보 하이퍼파라미터 값들을 랜덤 샘플링을 통해 선정



- Grid Search 에 비해 불필요한 반복 수행 횟수 줄임

참고자료 - Random Search / Grid Search



3. Bayesian Optimization

어느 입력값 x 를 받는 미지의 목적함수 f 를 상정하여, 함수값 $f(x)$ 를 최대로 만드는 최적해 x^* 탐색

- 목적 함수의 표현식을 명시적으로 알지 못하고(예. black-box function)
- 하나의 함수값을 계산하는데 오랜 시간이 소요되는 경우를 가정

필수요소

1. Surrogate Model : 현재까지 조사된 입력값-함숫값 점들($(x_1, f(x_1)) (x_2, f(x_2)) \dots$), 을 바탕으로, 미지의 목적 함수의 형태에 대한 확률적 추정 수행하는 모델
→ 이전까지 탐색한 성능 측정 결과(사전 지식) 반영
2. Acquisition Function : 목적 함수에 대한 현재까지의 확률적 추정 결과를 바탕으로 최적 입력값 x^* 를 찾는 데 있어 가장 유용한 만한 다음 입력값 후보를 추천해주는 함수 지칭

수도코드

```
for i=1, 2, 3 ... do

    surrogate model 의 확률적 추적 결과 바탕으로 Acquisition Function를 최대화하는 입력값 후보  $x_{i+1}$  설정

     $f(x_{i+1})$  계산

    surrogate model에  $(x_i, f(x_{i+1}))$  추가하여 확률적 추적 수행

end
```

3.1 Surrogate Model

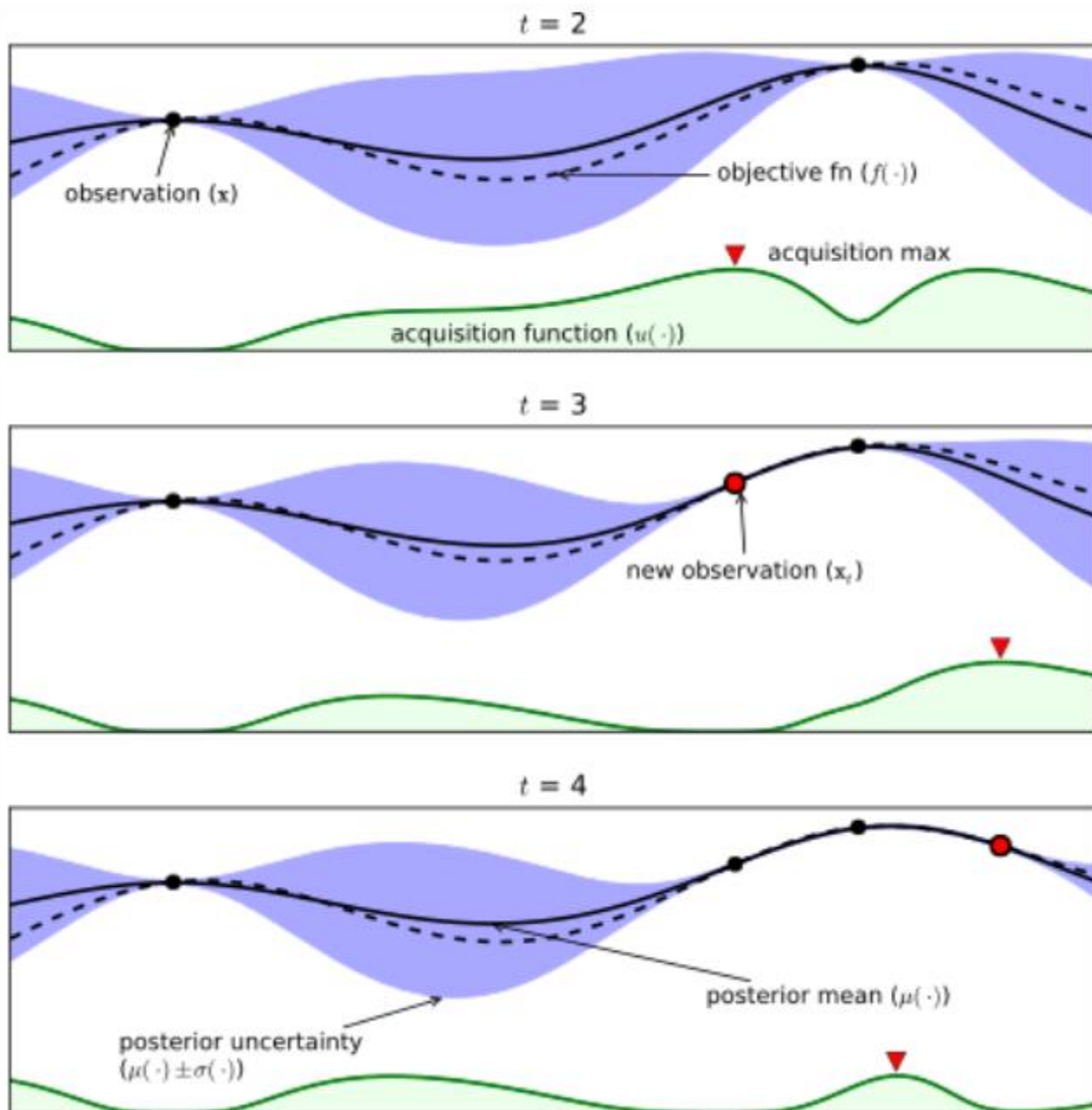
현재까지 조사된 입력값-함수값 점들을 바탕으로 미지의 목적 함수의 대략적인 형태에 대한 확률적인 추적을 수행하는 모델

3.1.1 Gaussian Processes

가장 많이 사용되는 확률 모델로 모종의 함수들에 대한 확률 분포를 표현

- 보통의 확률 모델 : 어느 특정 변수에 대한 확률 분포를 표현
- 구성 요소 간 결합 분포가 가우시안 분포를 따름
- 평균 함수 μ 와 공분산 함수 k 를 사용하여 함수들에 대한 확률 분포를 표현

$$f(x) \sim \mathcal{GP}(\mu(x), k(x, x')).$$



- 검은색 실선 : 각 x 위치별 평균 평균 $\mu(x)$
- 파란색 음영 : x 위치별 표준편차 $\sigma(x)$
 - 조사된 점으로부터 거리가 먼 x 일수록 이 지점에 대해 추정된 평균값의 불확실성 큼
 - 조사된 점의 개수가 늘어날수록 목적 함수의 추정 결과가 압축됨 → 불확실성 감소

3.2 Acquisition Function

목적 함수에 대해 확률적으로 추정된 현재까지의 결과를 바탕으로, 바로 다음번에 함수값을 조사할 입력값 후보 x_{i+1} 을 추천해주는 함수

- x_{i+1} 은 목적함수의 최적 입력값 x^* 을 찾는 데 있어 가장 유용할 만한 것

exploitation (착취)

현재까지 조사된 점들 중 함수값이 최대인 점 근방을 다음 차례에 시도하는 것

→ 함수값이 가장 큰 점 근방에서 실제 최적 입력값 x^* 를 찾을 가능성이 높지 않을까

exploration (탐색)

현재까지 추정된 목적 함수 상에서 표준편차가 최대인 점 근방을 다음 차례에 시도하는 것

→ 불확실한 영역에 최적 입력값 x^* 이 존재할 수 있지 않을까

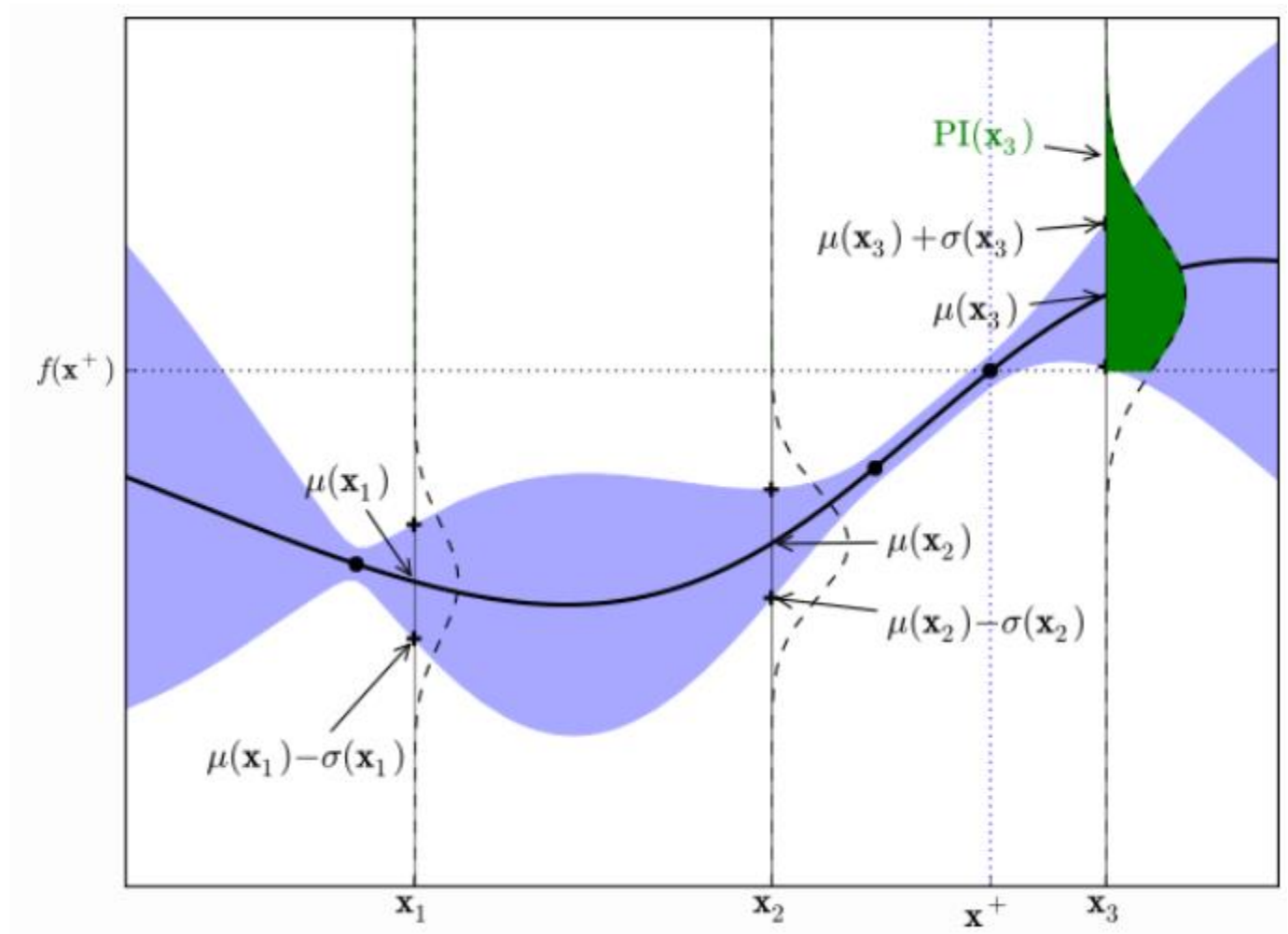
3.2.1 Expected Improvement(EI)

가장 많이 사용되는 Acquisition Function 으로 exploitation 전략과 exploration 전략 모두를 일정 수준 포함하도록 설계된 것

현재까지 추정된 목적 함수를 바탕으로, 어느 후보 입력값 x 에 대해

1. 현재까지 조사된 점들의 함수값 중 최대 함수값 $f(x_+)$ 보다 큰 함수값을 도출할 확률(PI)
2. 그 함수값과 최대 함수값 간 차이값

을 고려하여 x 의 유용성을 반환



- 초록색 음영 : $f(x^+)$ 보다 큰 값에 해당하는 영역
 - 다음 입력값으로 x_3 을 채택했을 시 기존 점들보다 더 큰 함수값을 얻을 가능성이 높음
 - 목적 함수의 최적 입력값 x^* 를 찾는 데 있어 x_3 이 가장 유용할 만한 후보
- PI 값에 함수값 $f(x_3)$ 에 대한 평균과 $f(x_3) - f(x^+)$ 을 가중하여 EI 최종 계산
 - 실제로 $f(x_3)$ 의 값이 $f(x^+)$ 보다 얼마나 더 큰지 반영

$$EI(x) = \mathbb{E}[\max(f(x) - f(x^+), 0)]$$

$$= \begin{cases} (\mu(x) - f(x^+) - \xi)\Phi(Z) + \sigma(x)\phi(Z) & \text{if } \sigma(x) > 0 \\ 0 & \text{if } \sigma(x) = 0 \end{cases}$$

$$Z = \begin{cases} \frac{\mu(x) - f(x^+) - \xi}{\sigma(x)} & \text{if } \sigma(x) > 0 \\ 0 & \text{if } \sigma(x) = 0 \end{cases}$$

- Φ : 표준정규분포의 누적분포함수(CDF)
- ϕ : 표준정규분포의 확률분포함수(PDF)를 나타내며,
- ξ : exploration 과 exploitation 간의 상대적 강도를 조절해 주는 파라미터
 - 클수록 exploration 의 강도가 높아짐
 - 작을수록 exploitation 의 강도가 높아짐

3.3 Bayesian Optimization 수행 과정

1. 입력값, 목적 함수 및 그 외 설정값들 정의
 - 입력값 x : 학습률
 - 목적 함수 $f(x)$: 성능 함수 (e.g. 정확도)
 - 입력값 x 의 탐색 대상 구간: (a,b) .
 - 맨 처음에 조사할 입력값-함숫값 점들의 갯수: n
 - 맨 마지막 차례까지 조사할 입력값-함숫값 점들의 최대 갯수: N
2. 설정한 탐색 대상 구간내에서 처음 n 개의 입력값들을 랜덤하게 샘플링하여 선택
3. 선택한 n 개의 입력값 x_1, x_2, \dots, x_n 을 각각 학습률 값으로 설정
4. 딥러닝 모델을 학습한 뒤, 검증 데이터셋을 사용하여 학습이 완료된 모델의 성능 결과 수치를 계산
 - 이들을 각각 함숫값 $f(x_1), f(x_2), \dots, f(x_n)$ 으로 간주
5. 입력값-함숫값 점들의 모음에 대하여 Surrogate Model로 확률적 추정 수행
6. 조사된 입력값-함숫값 점들이 총 N 개에 도달할 때까지 반복
 - 기존 입력값-함숫값 점들의 모음에 대한 Surrogate Model의 확률적 추정 결과를 바탕으로, 입력값 구간 (a,b) 내에서의 TI의 값을 계산
 - 그 값이 가장 큰 점을 다음 입력값 후보 x_{t+1} 로 선정
 - 다음 입력값 후보 x_{t+1} 을 학습률 값으로 설정하여 딥러닝 모델을 학습한 뒤, 검증 데이터셋을 사용하여 학습이 완료된 모델의 성능 결과 수치 계산 $\rightarrow f(x_{t+1})$ 값
 - 새로운 점 $(x_{t+1}, f(x_{t+1}))$ 을 기존 입력값-함숫값 점들의 모음에 추가
 - 갱신된 점들의 모음에 대하여 Surrogate Model로 확률적 추정을 다시 수행

7. 총 N 개의 입력값-함숫값 점들에 대하여 확률적으로 추정된 목적 함수 결과물을 바탕으로, 평균 함수 $\mu(x)$ 을 최대로 만드는 최적해 x^* 를 최종 선택
 8. 해당 x^* 값을 학습률로 사용하여 딥러닝 모델을 학습하면, 일반화 성능이 극대화된 모델을 얻을 수 있음
-

참고 자료

- [Bayesian Optimization 개요: 딥러닝 모델의 효과적인 hyperparameter 탐색 방법론](#)
- Shahriari et al., Taking the human out of the loop: A review of bayesian optimization.
 - [Shahriari, Bobak, et al. "Taking the human out of the loop: A review of bayesian optimization." Proceedings of the IEEE 104.1 \(2016\): 148–175.](#)
- Brochu et al., A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning.
 - [Brochu, Eric, Vlad M. Cora, and Nando De Freitas. "A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning." arXiv preprint arXiv:1012.2599 \(2010\).](#)