Hyperparameter Optimization Overview

Hyperparameter Optimization

There are too many hyperparameters to be set by expert

- Learning rate, Momentum rate, Dropout,Normalization,Number of layers, number of nodes

Is there any efficient way to set them?

- No

We can regard it as function optimization(예시로)

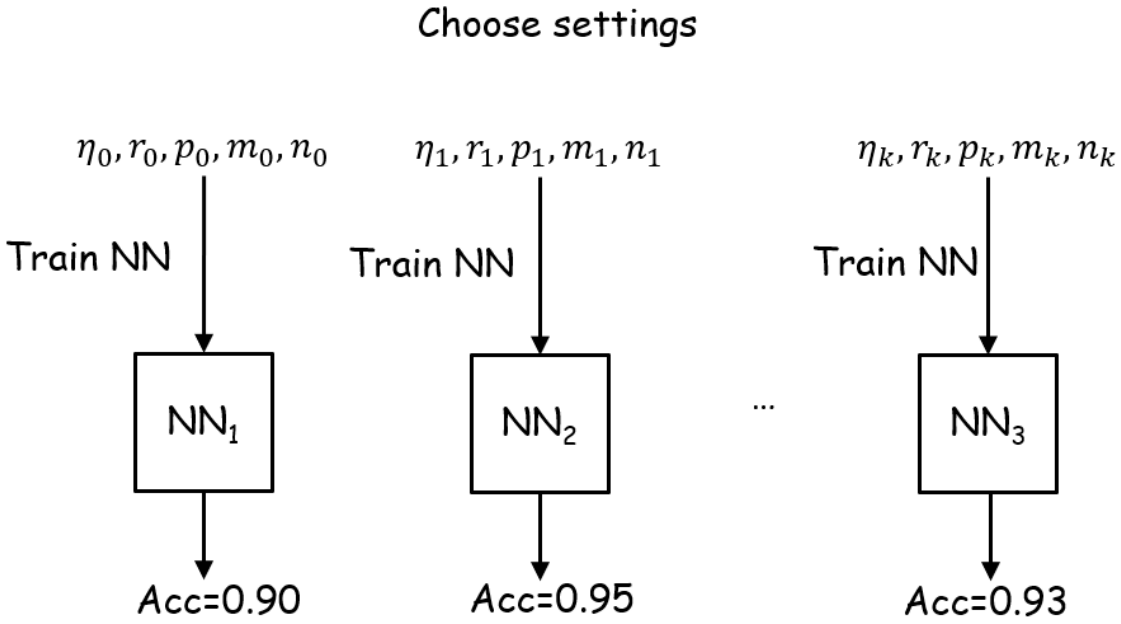
- I will use 3 - layer perceptron
- I want to find out the optimal hyperparameters
  - Learning rate : η
  - Momentum : r
  - Dropout probability : p
  - Number of nodes in the first hidden layer : m
  - Number of nodes in the second hidden layer : n
- Then, the accuracy pf my NN is a function of η, r, p, m, n

$$Accuracy = f(\eta, r, p, m, n)$$

- I nee to solve

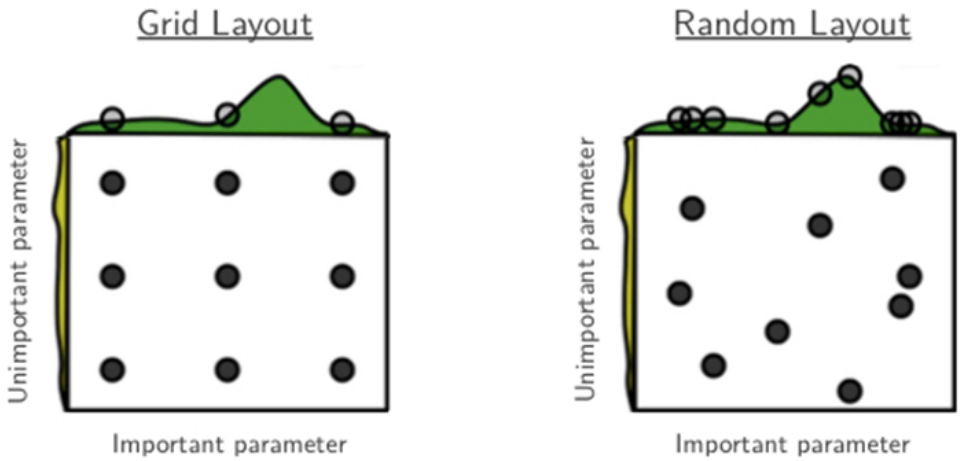
$$\underset{\eta,r,p,m,n}{\operatorname{argmax}} f(\eta, r, p, m, n)$$

- 그러나 문제가 있다.
  - I do not know what f(η, r, p, m, n) is
  - But I can query(질문). That is, for a some setting we can evaluate f
  - However, the evaluation is **very expensive** because we need to train a neural network!!



Some Simple Search Algorithm

하이퍼파라미터를 찾는 간단한 search algorithm이 있다.



It is not efficient

- It does not utilize the previous tries.
- It would be better to less search the area with low potential and more search the area with high potential

—> Can we choose a better netx point based on previous search results?

It is too costly

- To evaluate how a set of hyperparameters is good, we need to train a neural network
  - We need to train a large neural network every time we try a new set of parameters

—> Can we gradually train NNs as search goes on?

Bayesian Optimization

- Definition

$$\arg\max_{x} f(x)$$

- You don't know anything about f(x)
- You can query but it is very expensive

- ANy good idea??

Any Good Idea???

- No information on f(x)...
- First choose a random point, x1, and evaluate f(x1)
- Guess the shape of f(x) based on (x1,f(x1))

- Based on the guess, choose the next point, x2, and evaluate f(x2)
- Guess the shape of f(x) based on{(x1,f(x1)),(x2,f(x2))}

- Repeat those steps

Overall Description