Tuning the knobs of neural networks using Bayesian Optimization

Niall Turbitt

About me

- Machine Learning Engineer @ Argos

- Previously
  - Data Scientist @ Lucidworks
  - Masters in Statistics @ UCD Dublin
  - Mathematics & Economics @ Trinity College Dublin

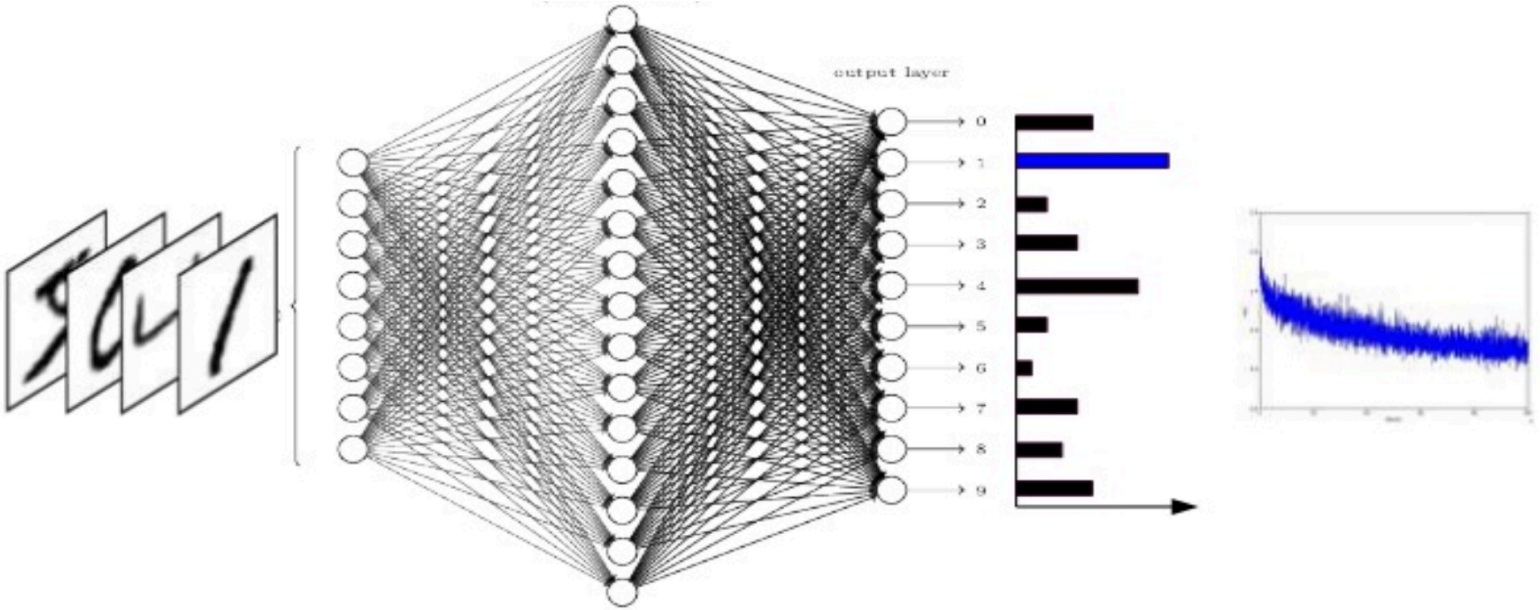
https://github.com/niallturb/pylondinium18

What this talk will cover

- Overview of neural networks
  - Convolutional Neural Networks
- The hyperparameter tuning problem
- Bayesian Opitimization
- Fashion MNIST demo

Neural Networks – Brief Overview

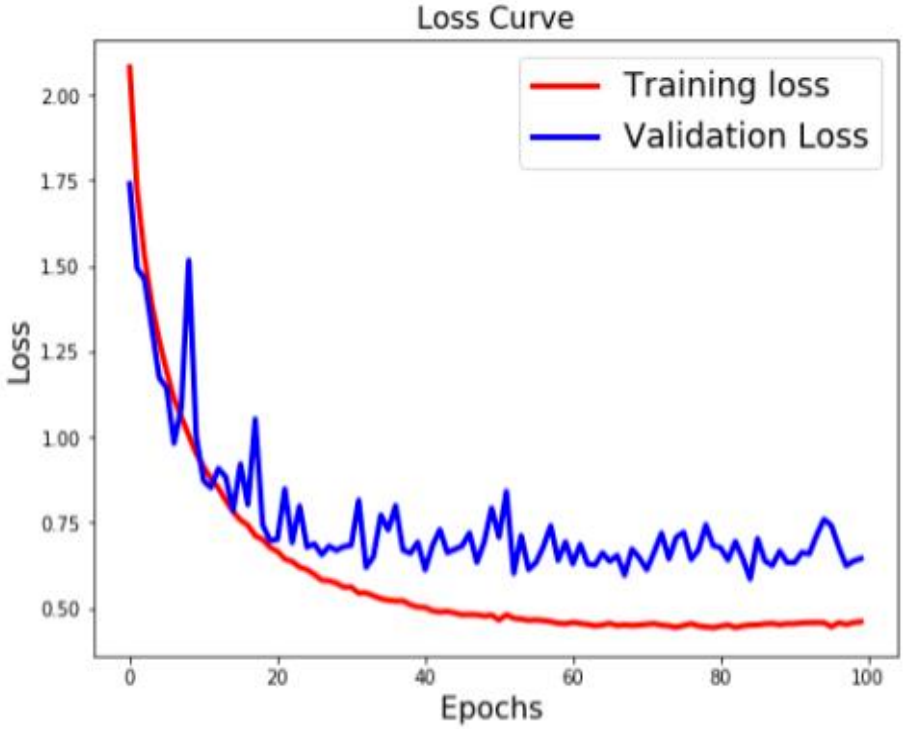
Training a Neural Network

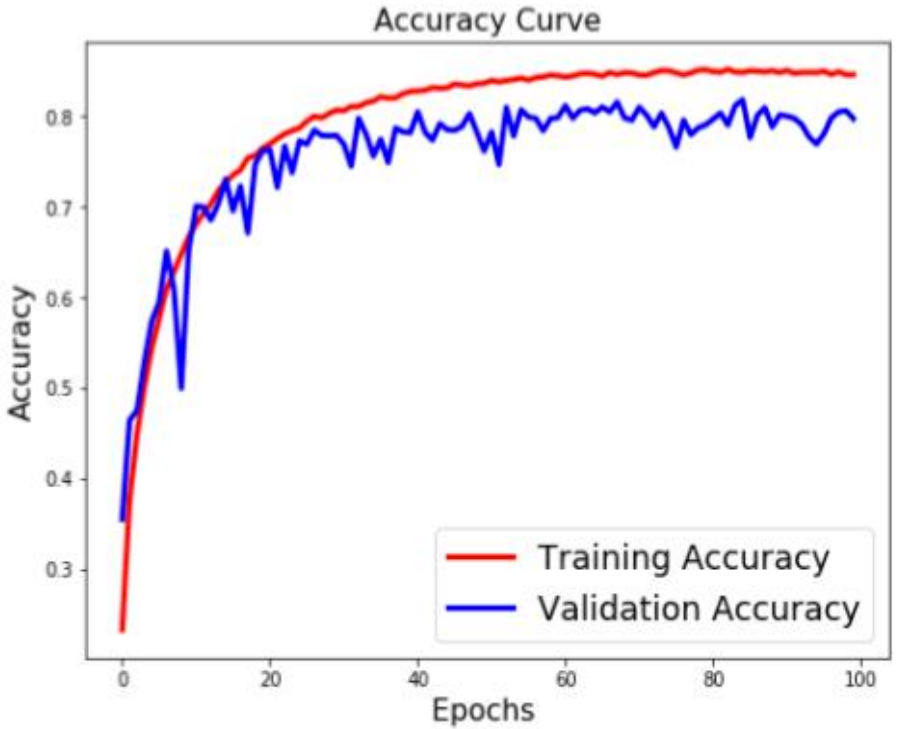


data → "model" / scoring function → loss function

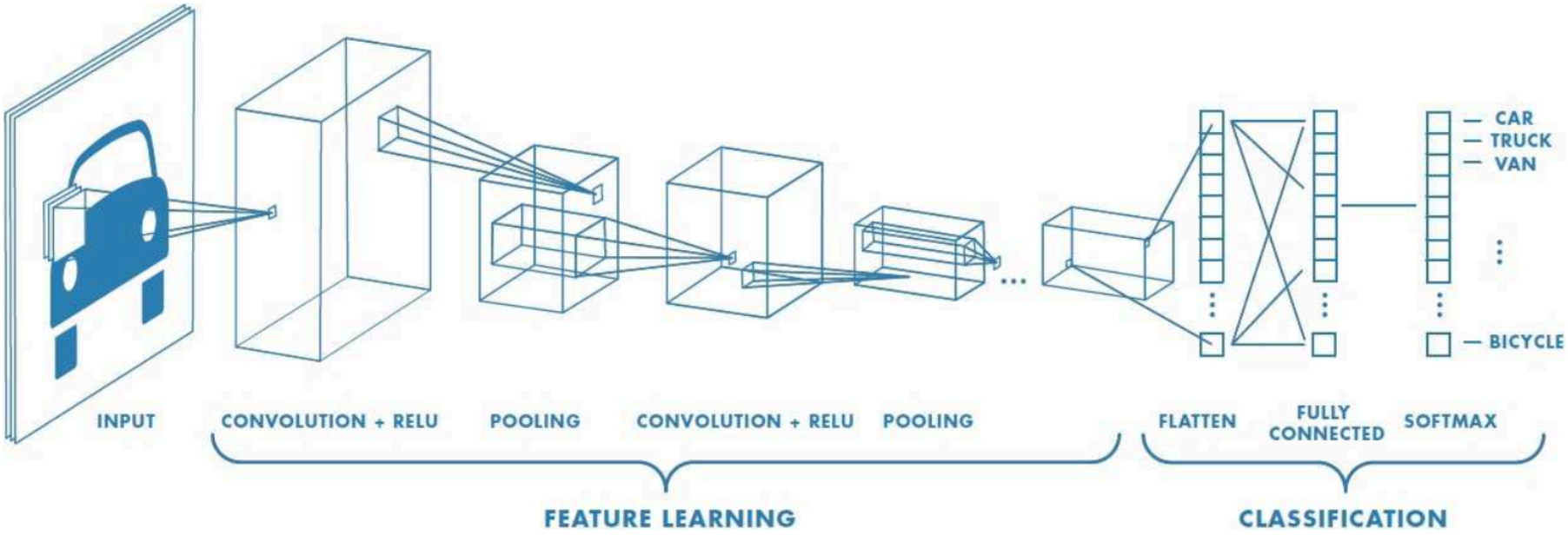
adapt weights (gradient descent)

Training should converge

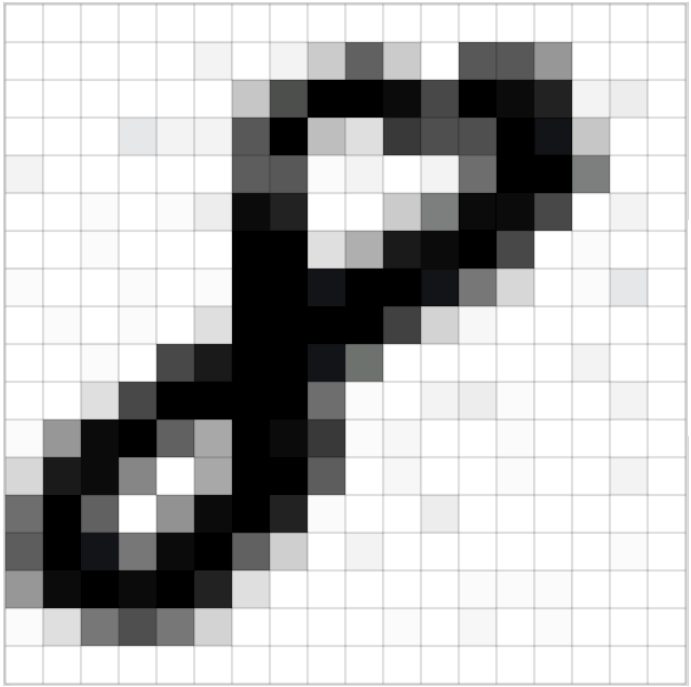




Convolutional Neural Networks (CNNs)



How can we input image data?

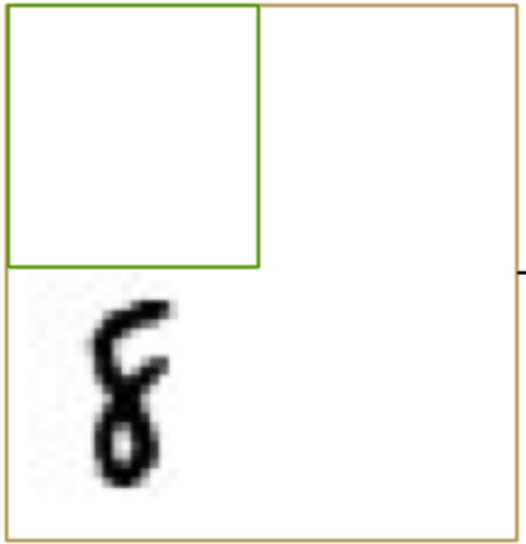


=

```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 12, 0, 11, 39, 137, 37, 0, 152, 147, 84, 0, 0, 0, 0,
0, 1, 0, 0, 0, 41, 160, 250, 255, 235, 162, 255, 238, 206, 11, 13, 0, 0, 0, 0, 16, 9, 9, 150, 251, 45, 21, 184, 159, 154, 2
55, 233, 40, 0, 0, 10, 0, 0, 0, 0, 0, 145, 146, 3, 10, 0, 11, 124, 253, 255, 107, 0, 0, 0, 0, 3, 0, 4, 15, 236, 216, 0, 0,
38, 109, 247, 240, 169, 0, 11, 0, 1, 0, 2, 0, 0, 0, 253, 253, 23, 62, 224, 241, 255, 164, 0, 5, 0, 0, 6, 0, 0, 4, 0, 3, 252
, 250, 228, 255, 255, 234, 112, 28, 0, 2, 17, 0, 0, 2, 1, 4, 0, 21, 255, 253, 251, 255, 172, 31, 8, 0, 1, 0, 0, 0, 0, 0, 4,
 0, 163, 225, 251, 255, 229, 120, 0, 0, 0, 0, 0, 11, 0, 0, 0, 0, 21, 162, 255, 255, 254, 255, 126, 6, 0, 10, 14, 6, 0, 0, 9
, 0, 3, 79, 242, 255, 141, 66, 255, 245, 189, 7, 8, 0, 0, 5, 0, 0, 0, 0, 26, 221, 237, 98, 0, 67, 251, 255, 144, 0, 8, 0, 0
, 7, 0, 0, 11, 0, 125, 255, 141, 0, 87, 244, 255, 208, 3, 0, 0, 13, 0, 1, 0, 1, 0, 0, 145, 248, 228, 116, 235, 255, 141, 34
, 0, 11, 0, 1, 0, 0, 0, 1, 3, 0, 85, 237, 253, 246, 255, 210, 21, 1, 0, 1, 0, 0, 6, 2, 4, 0, 0, 0, 6, 23, 112, 157, 114, 32
, 0, 0, 0, 0, 2, 0, 8, 0, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Tunnel Vision

Test Image

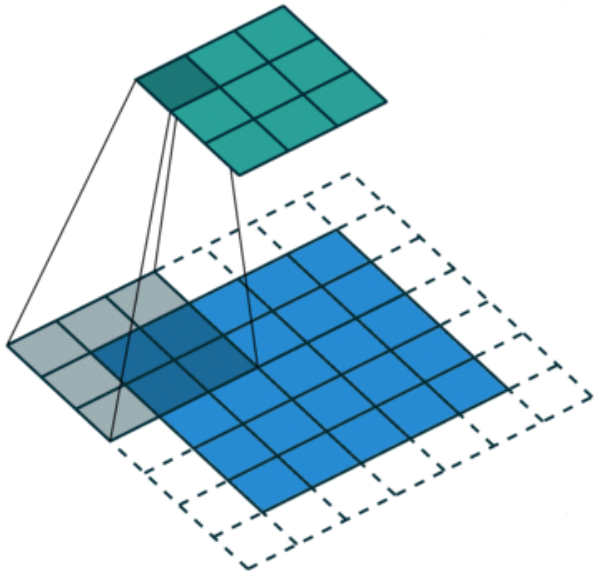


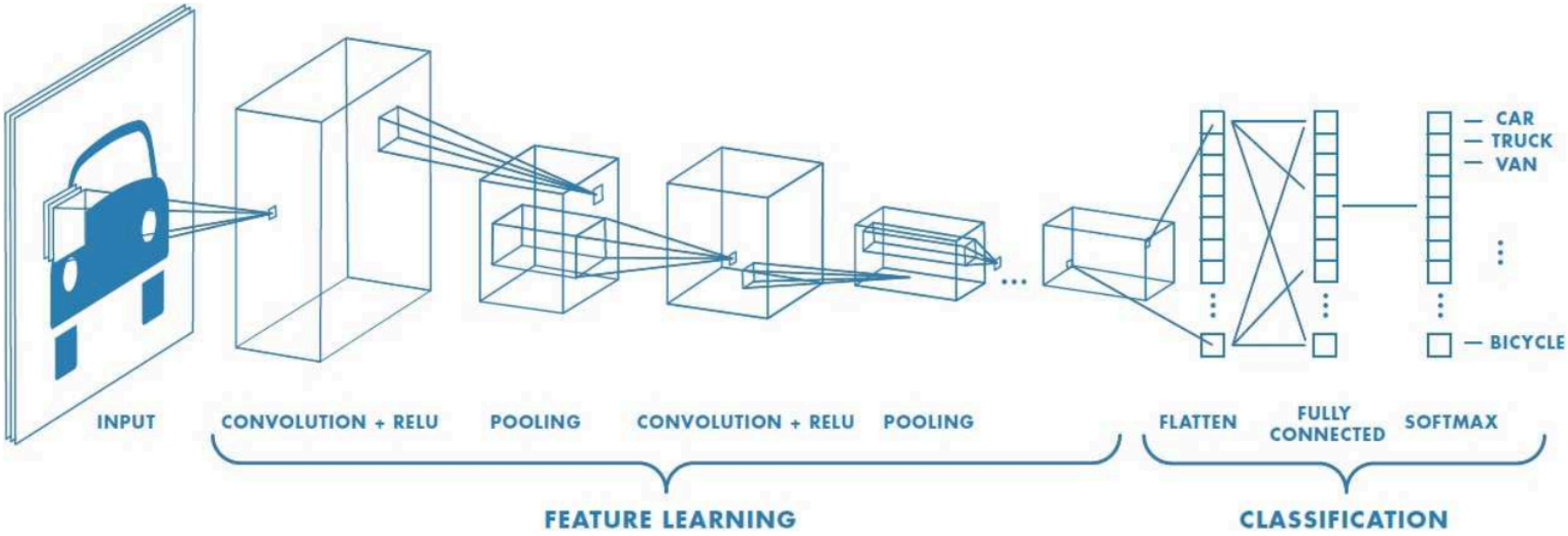
Prediction from our network

No idea!?!

Convolution is the solution

How Convolution Works




CAR
TRUCK
VAN
BICYCLE
INPUT
CONVOLUTION + RELU
POOLING
CONVOLUTION + RELU
POOLING
FLATTEN
FULLY CONNECTED
SOFTMAX
FEATURE LEARNING
CLASSIFICATION

Constructing the right CNN

- Number of layers
- Number of convolution filters
- Learning rate
- Number of epochs
- Batch size
- Activation function
- Weight initialization
- Dropout rate

Hyperparameter tuning

- Grid Search
  - $O(n^k)$
- Random Search (Bergstra & Bengio, 2012)
- Bayesian Optimization

Bayesian Optimization

Bayesian Optimization

- Tool for globally optimizing objective functions which are costly or slow to evaluate
- Sequential model-based optimization (SMBO)

$$x_M = \arg\min_{x \in \mathcal{X}} f(x)$$

1. Initialize a Gaussian Process on a small set of samples from our domain, $\mathcal{X}$ to compute a prior probability model
2. Sequentially select new locations within the domain by optimizing some *acquisition function*
3. At each iteration, we update our belief of what our objective function *f* looks like, having computed the loss for a new configuration of hyperparameters.

Gaussian Process

- A Gaussian Process is the generalization of a Gaussian distribution to a distribution over functions

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

- Induces a posterior distribution over the loss function that is analytically tractable

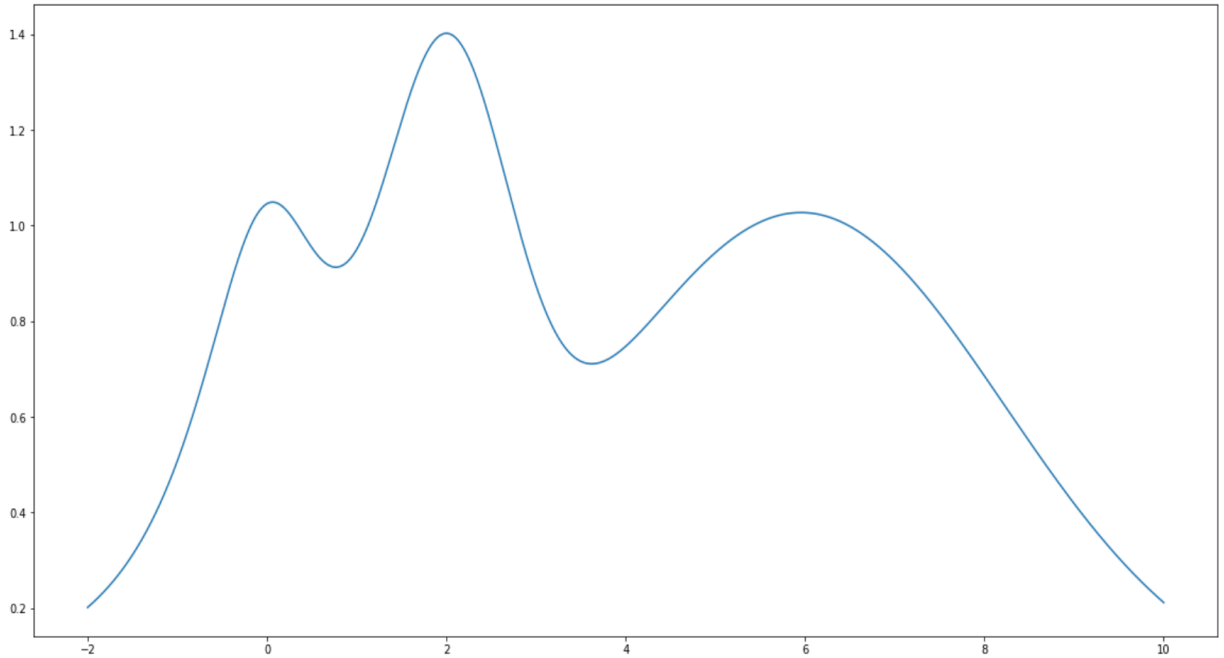
$$\text{Posterior probability} \propto \text{Likelihood} \times \text{Prior probability}$$

Acquisition Function

- Surrogate function that represents our beliefs of $f(x)$
  - Expected Improvement
  - Upper Confidence Bound (UCB)
  - Maximum probability of improvement (MPI)
- Exploration/Exploitation trade off

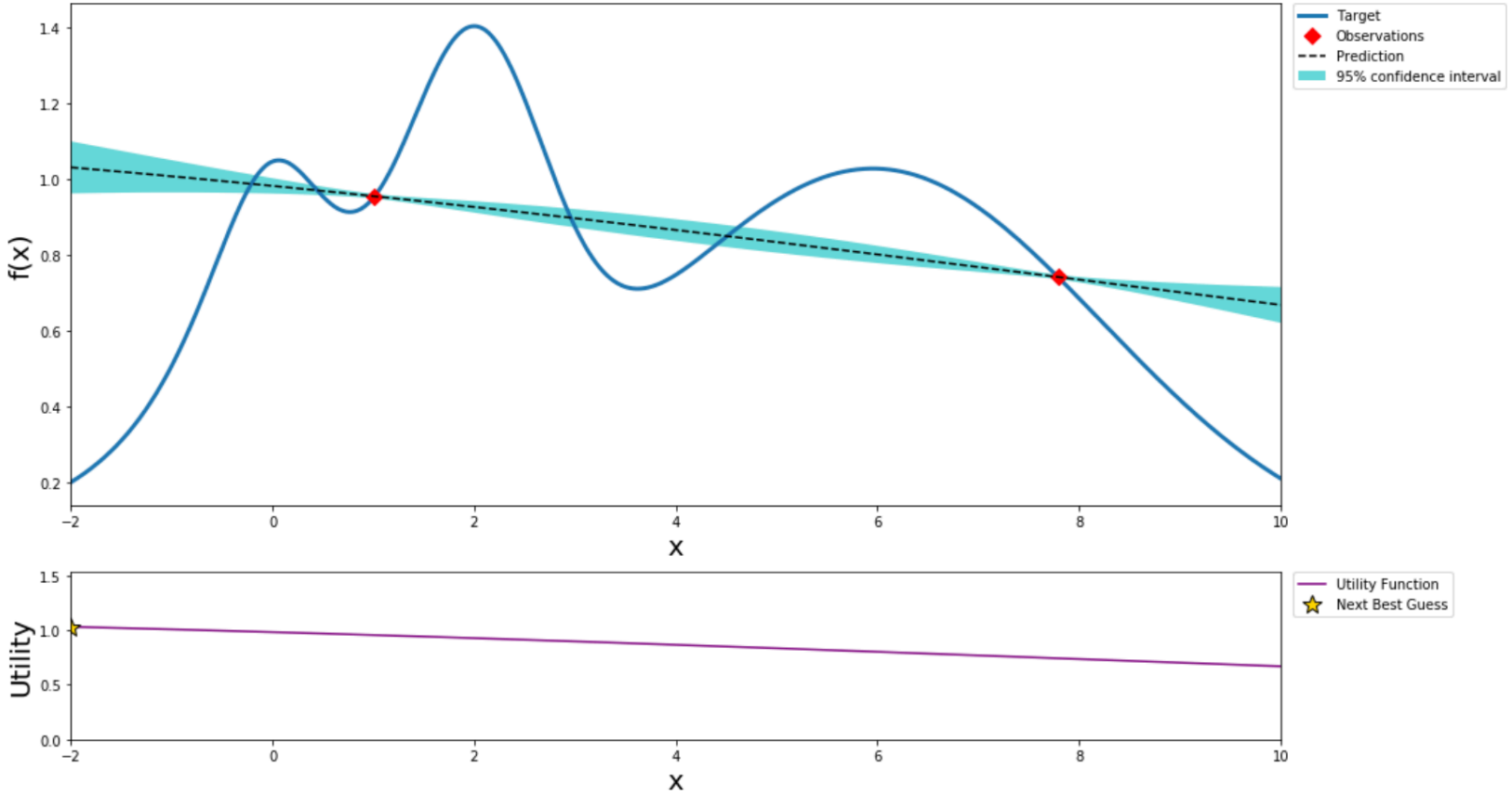
One Dimensional Bayesian Optimization Example

$$f(x) = e^{-(x-2)^2} + e^{-\frac{(x-6)^2}{10}} + \frac{1}{x^2+1}$$

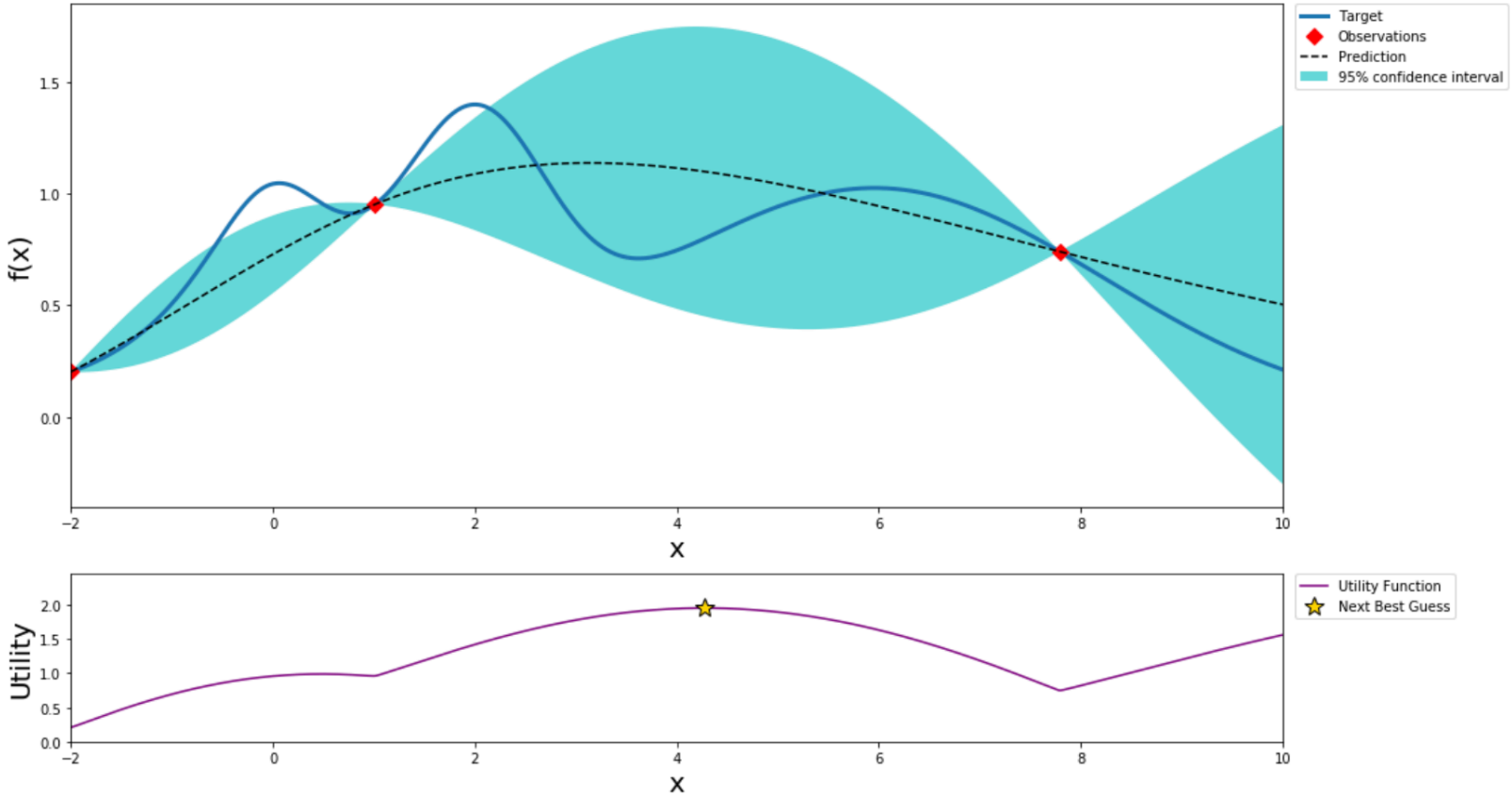


https://github.com/niallturb/pylondinium18/blob/master/basic_bayes_opt_demo.ipynb

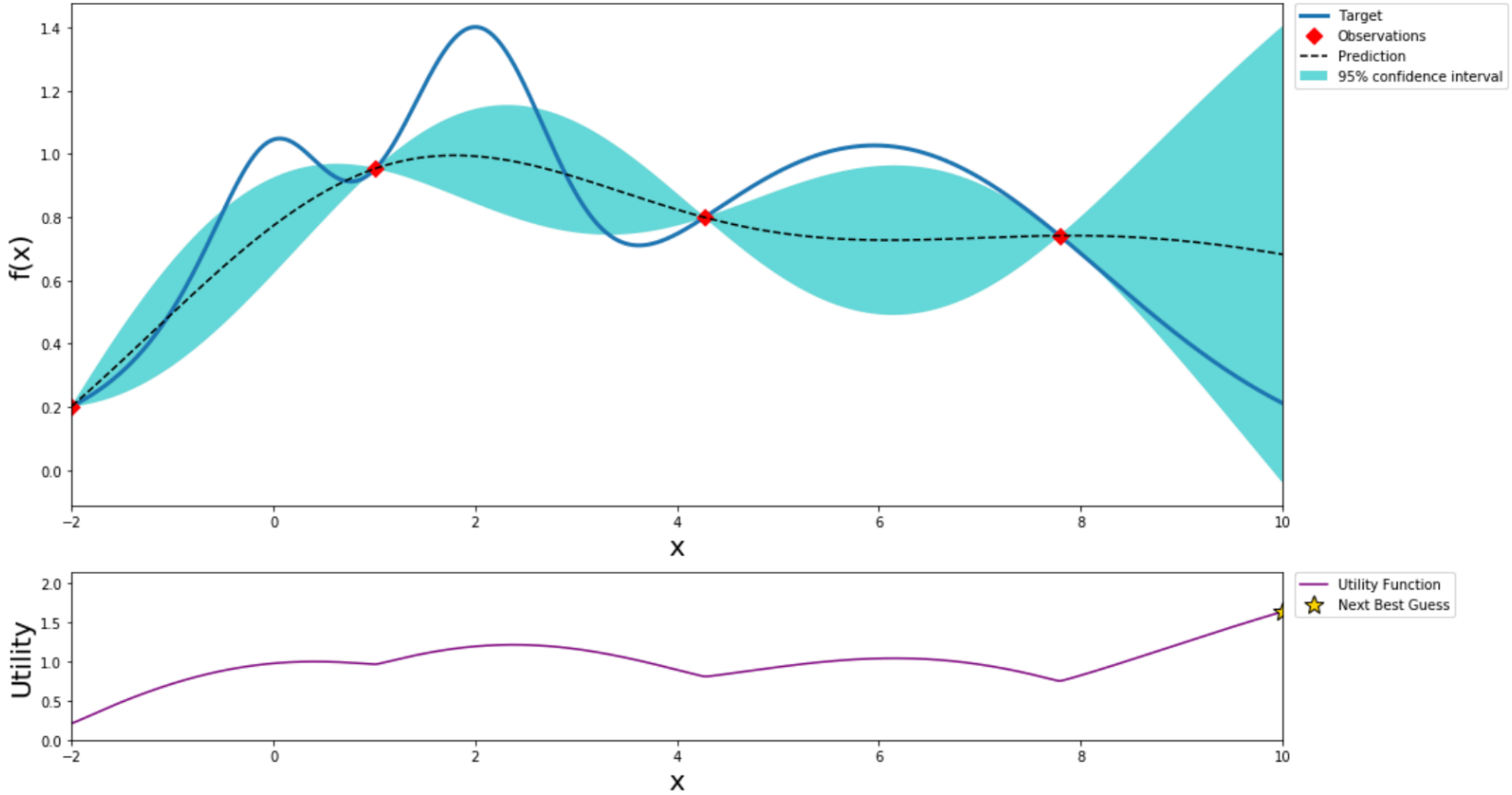
Initialize with 2 random points



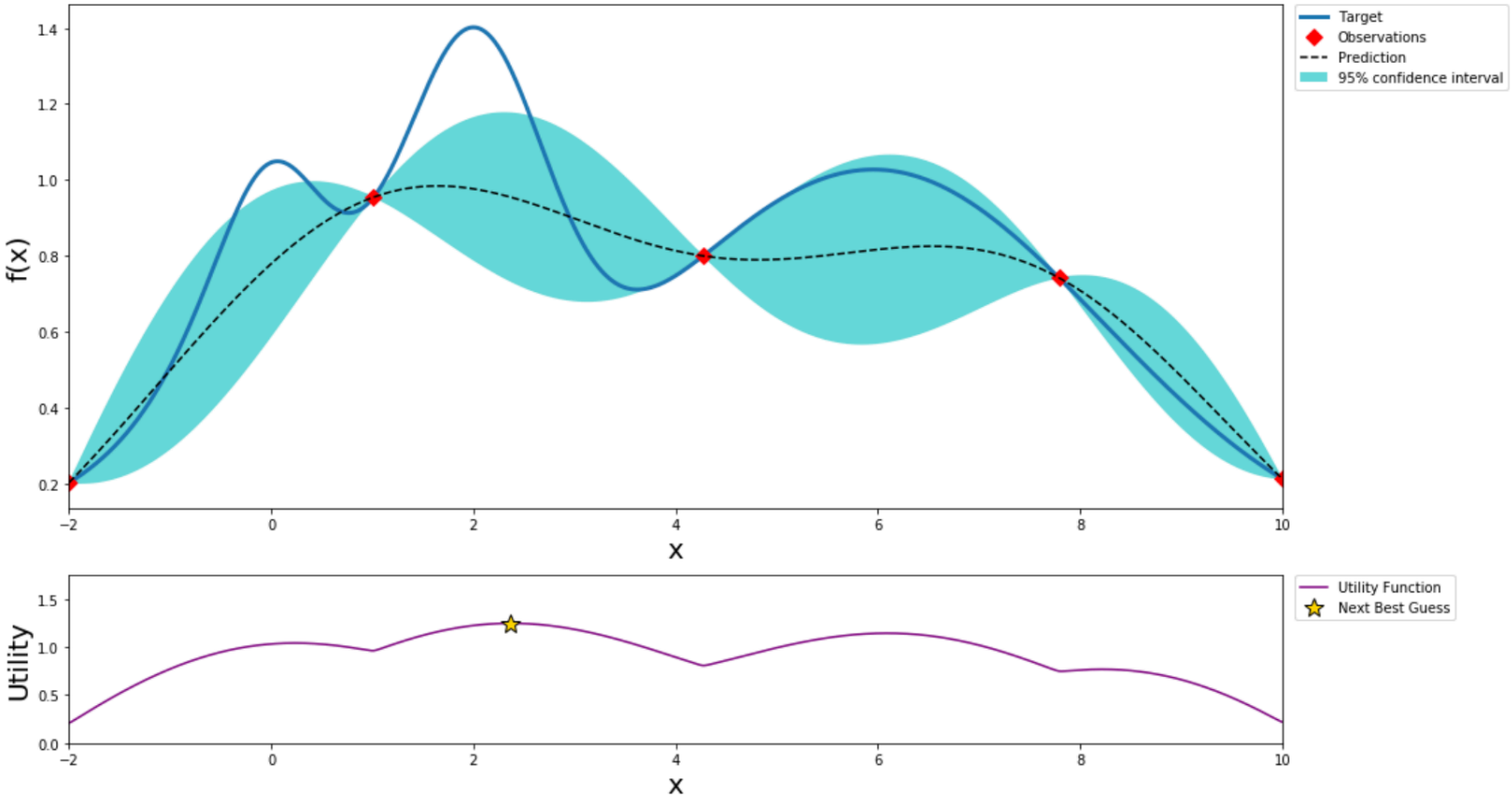
After 1 step of GP (and two random points)



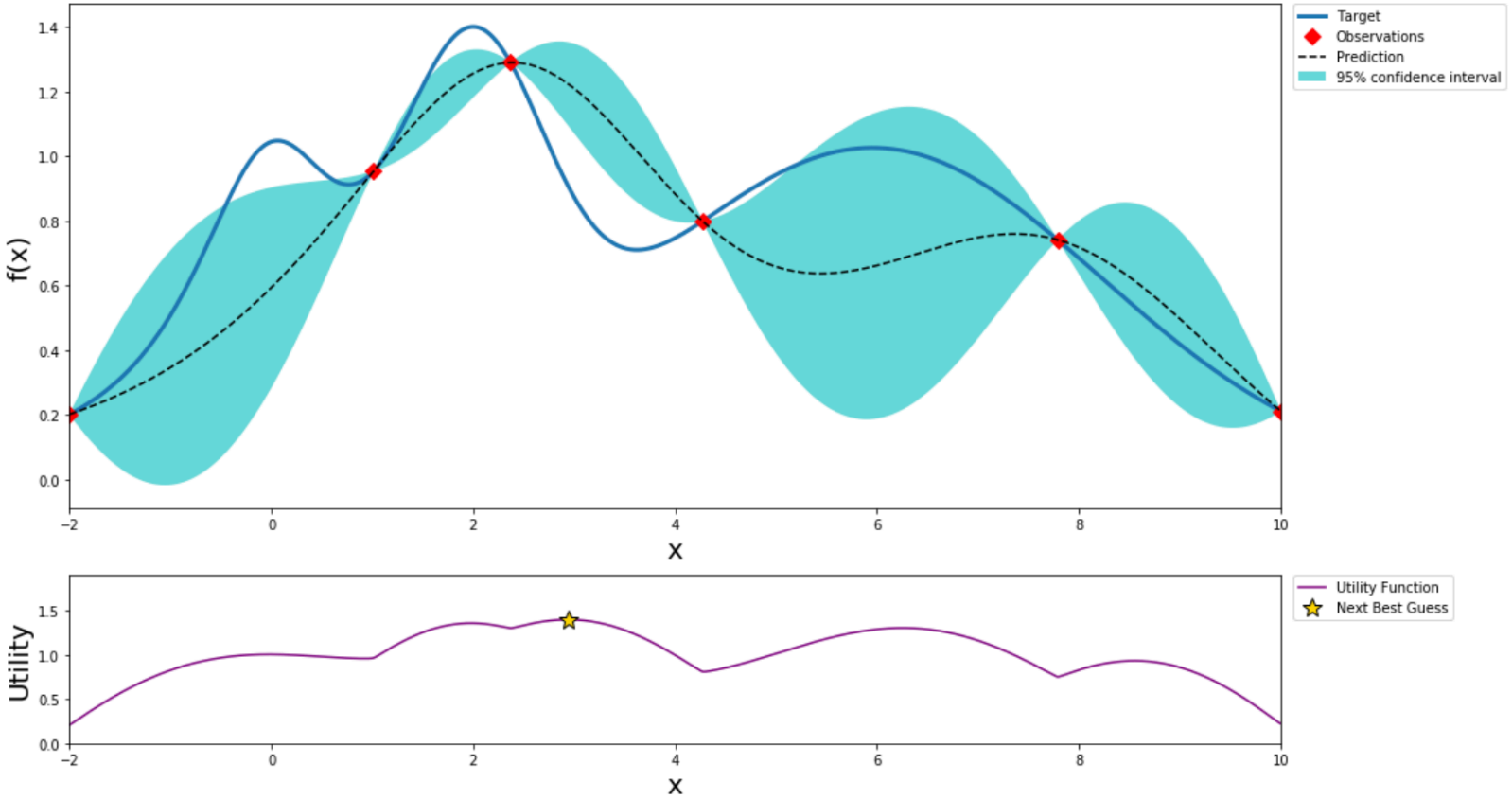
After 2 steps of GP (and two random points)



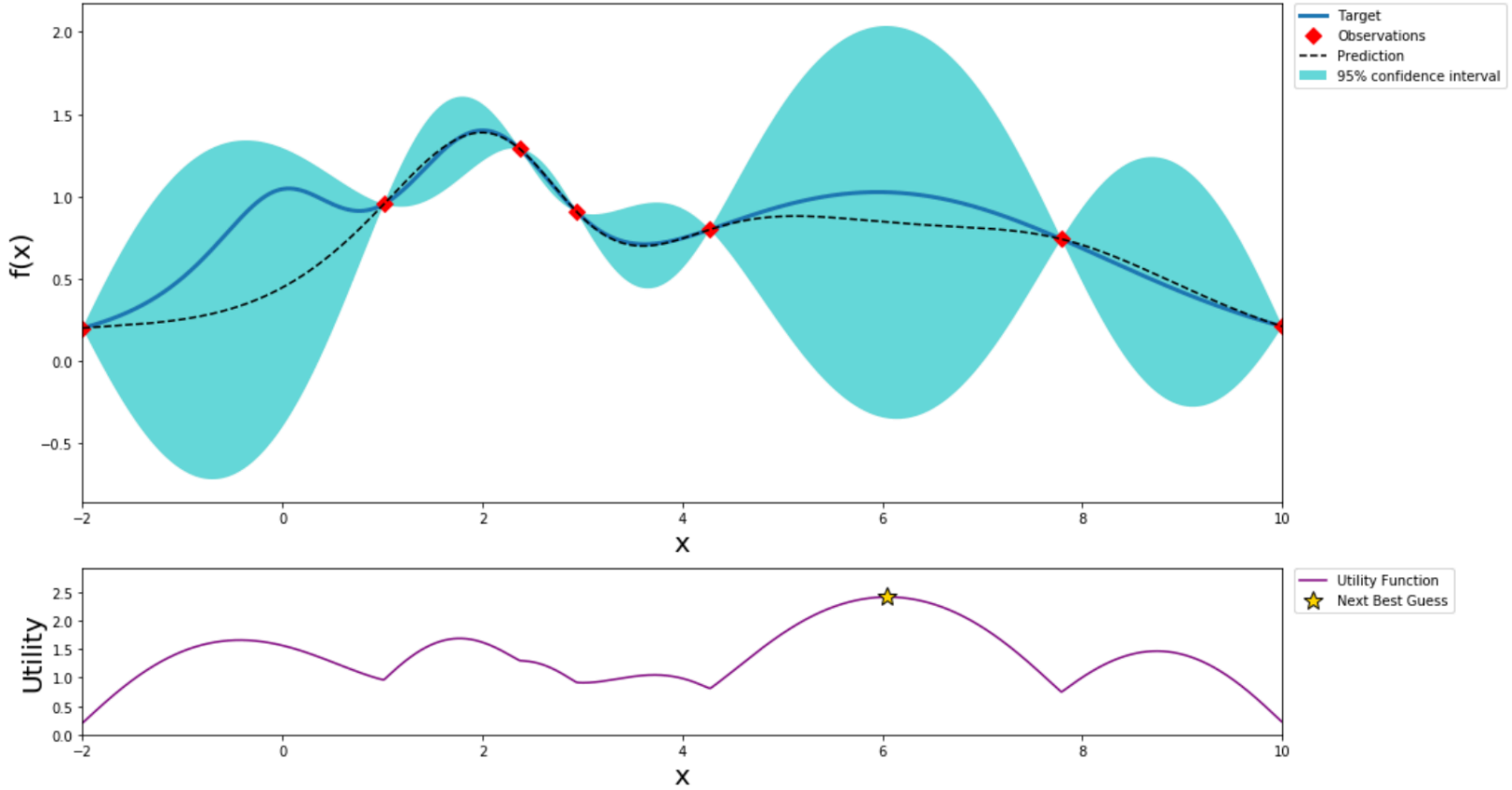
After 3 steps of GP (and two random points)



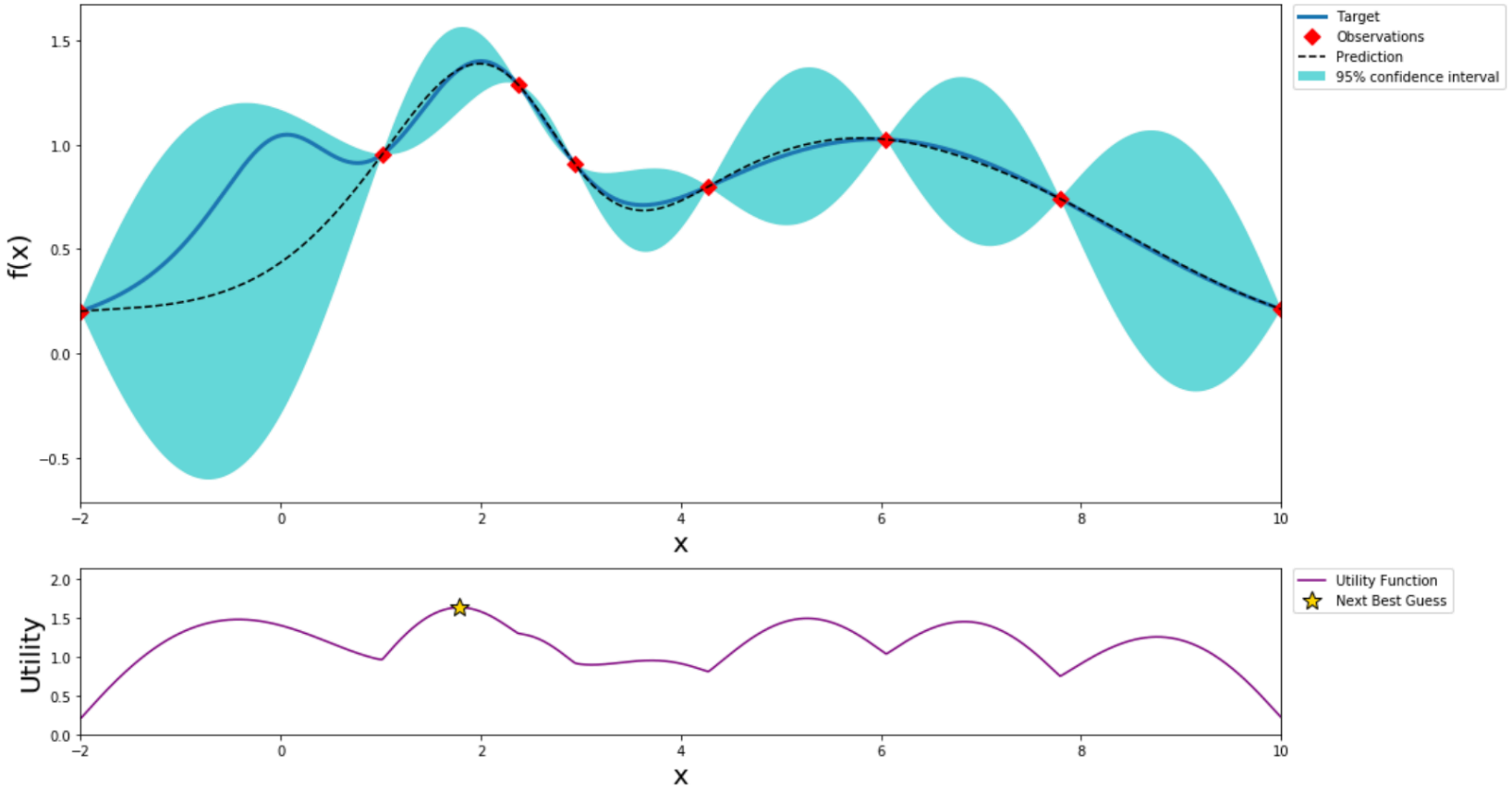
After 4 steps of GP (and two random points)



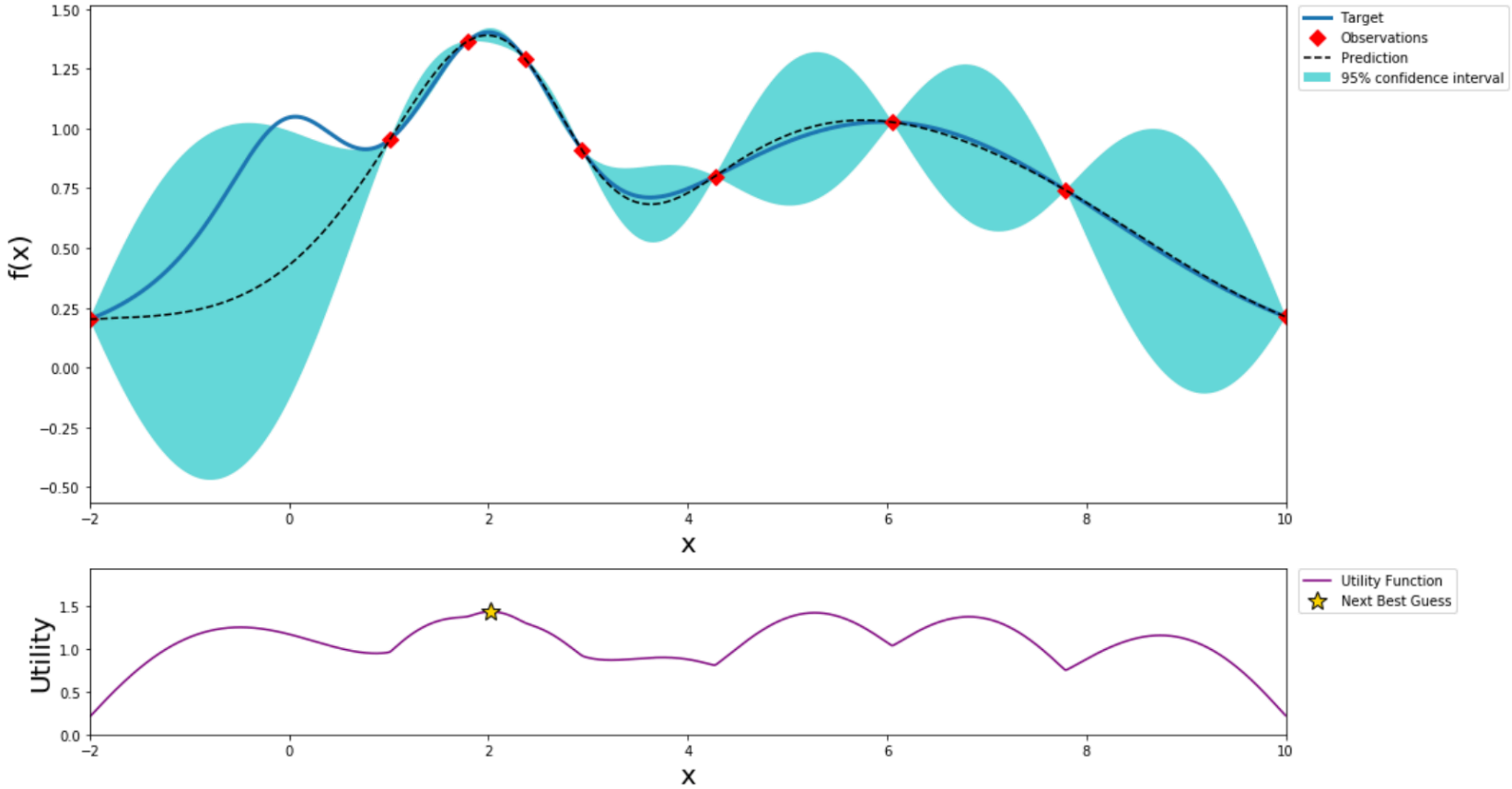
After 5 steps of GP (and two random points)



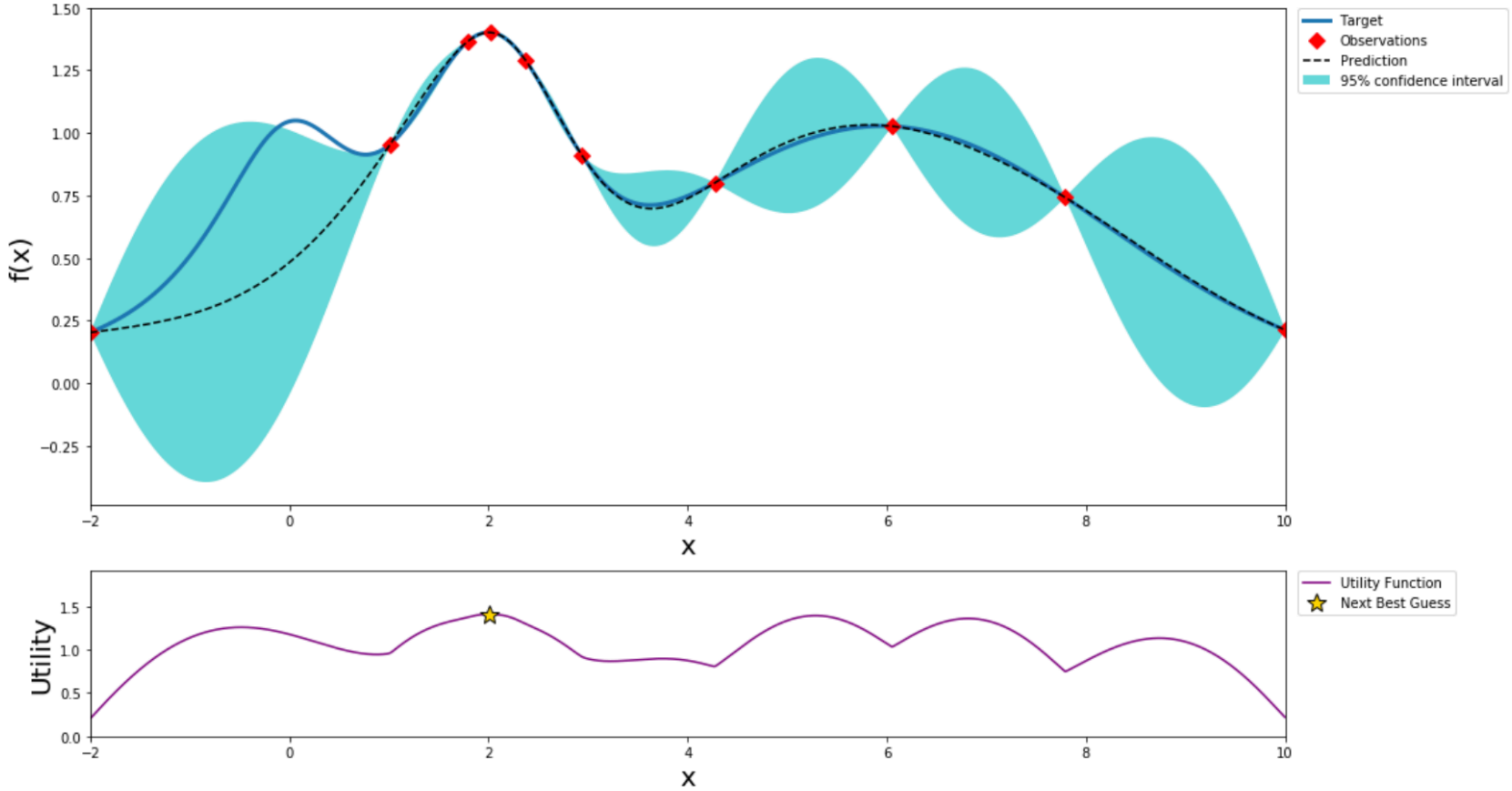
After 6 steps of GP (and two random points)



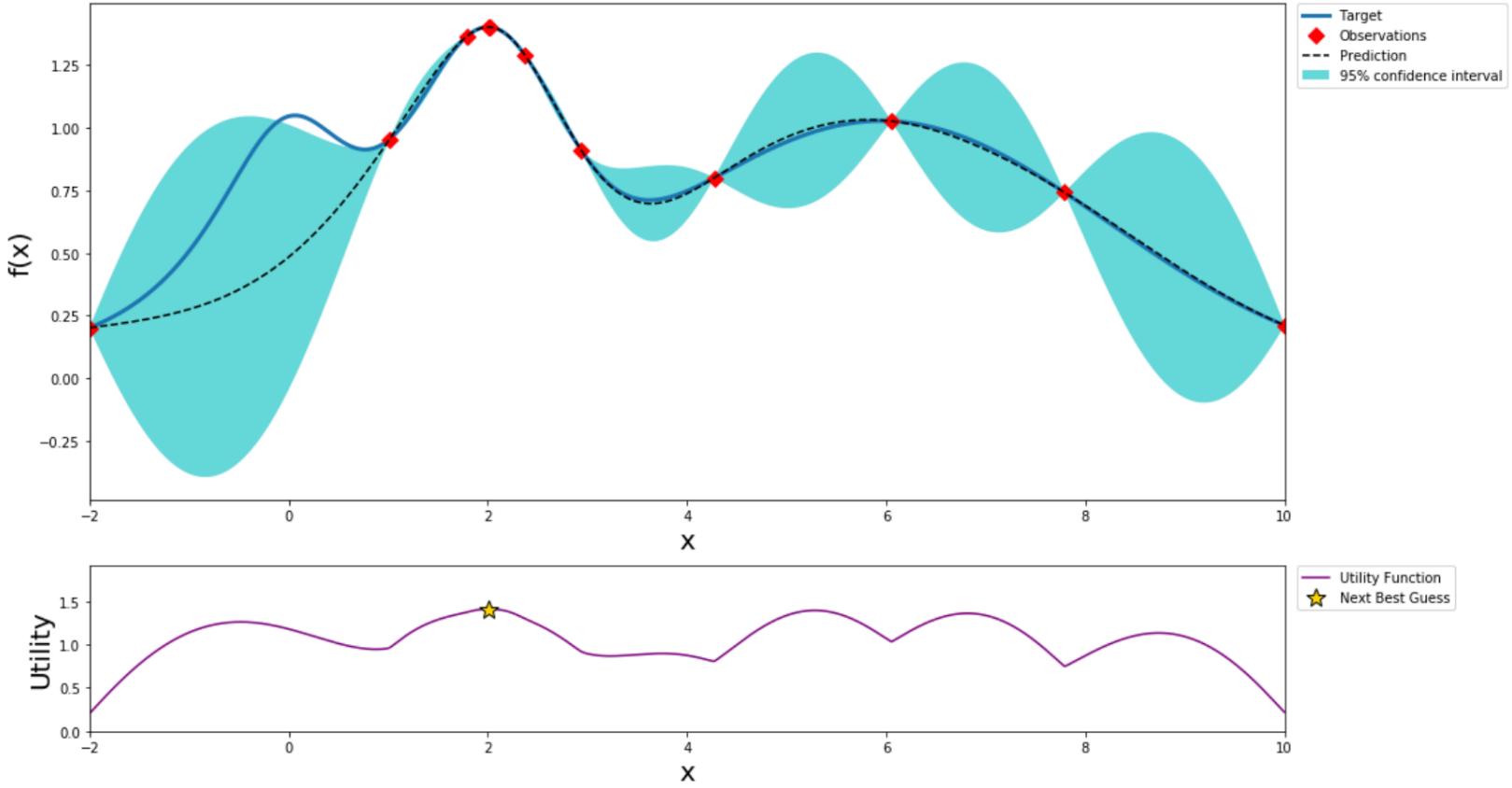
After 7 steps of GP (and two random points)



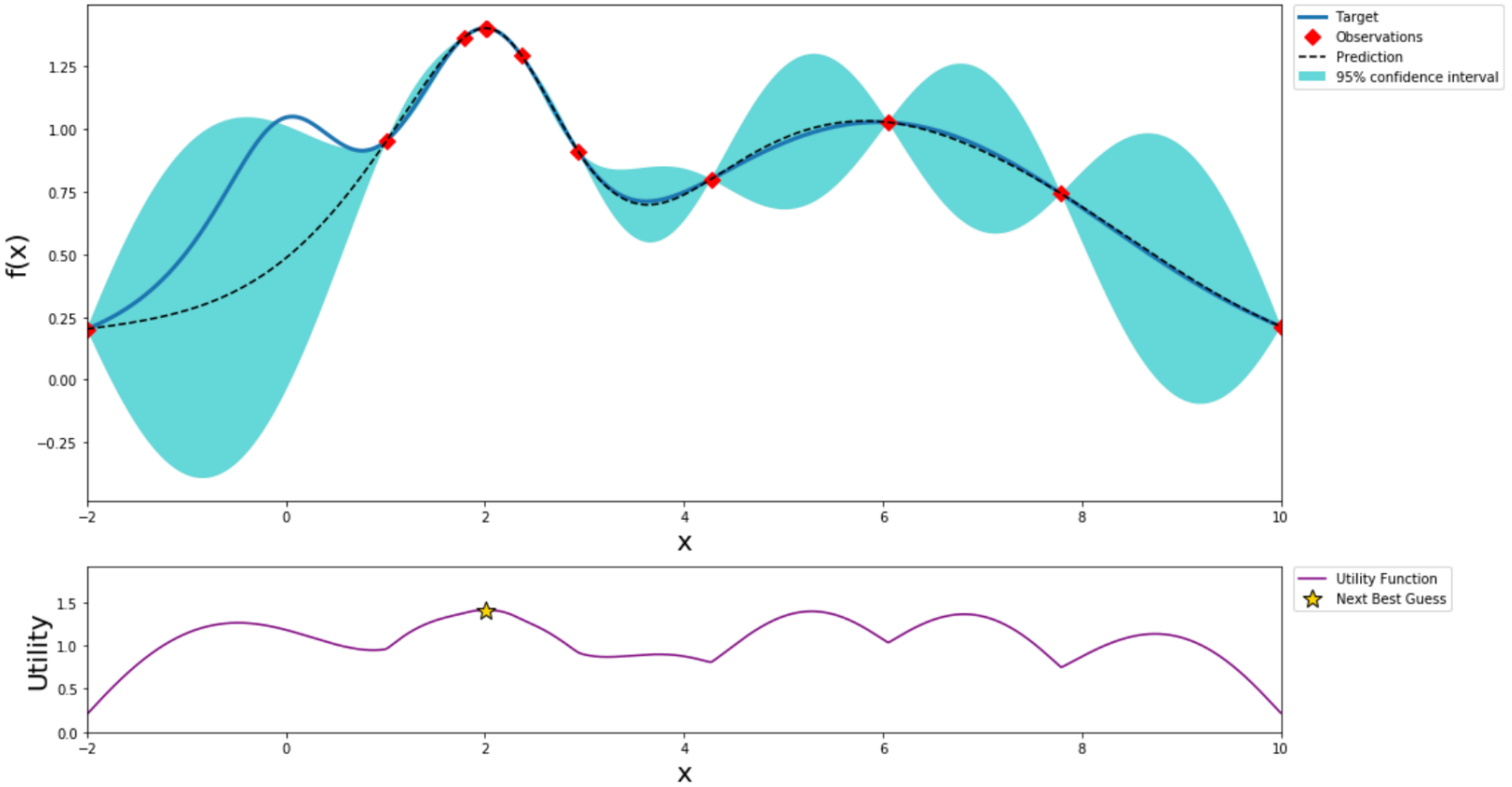
After 8 steps of GP (and two random points)



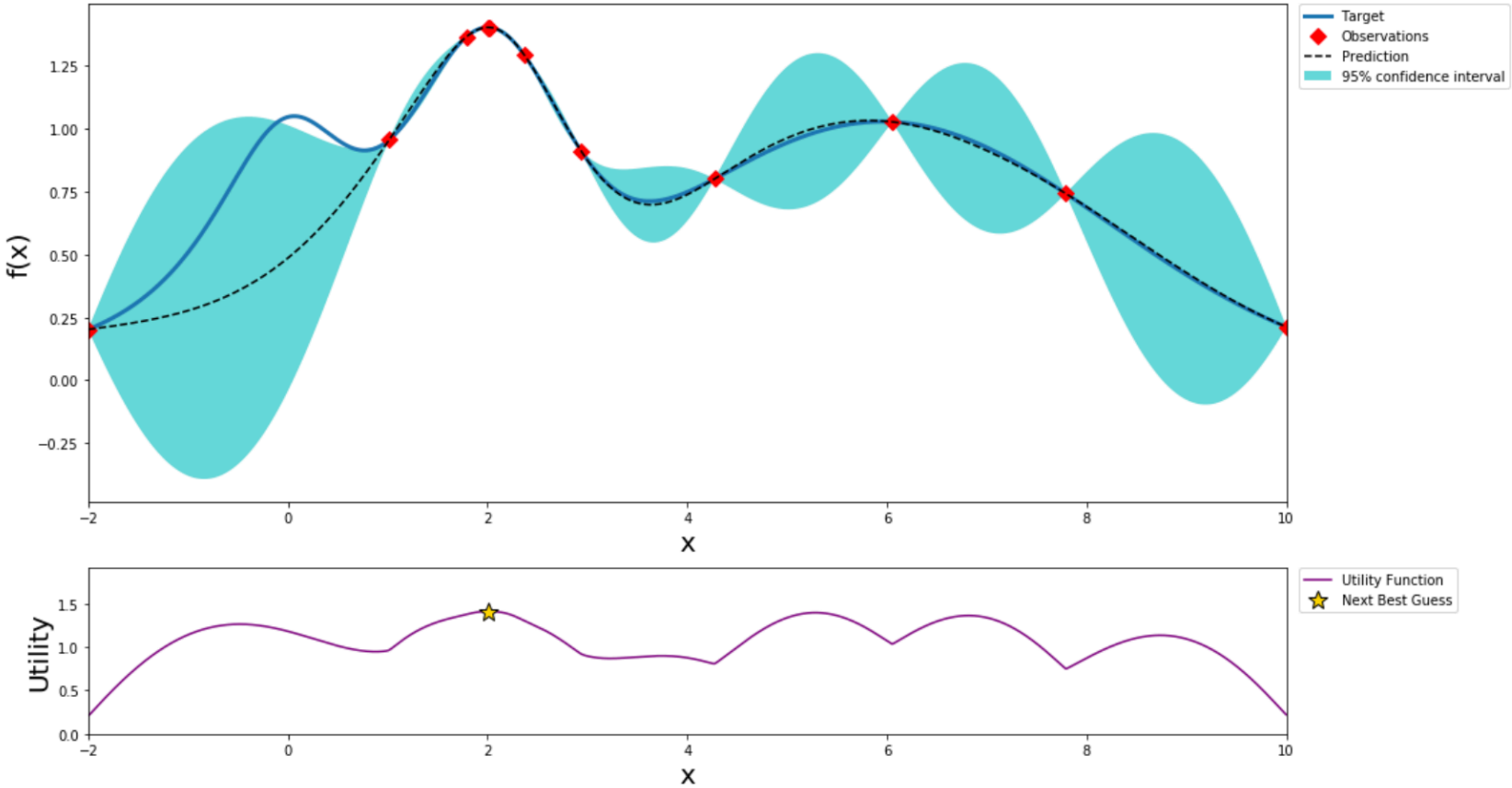
After 9 steps of GP (and two random points)



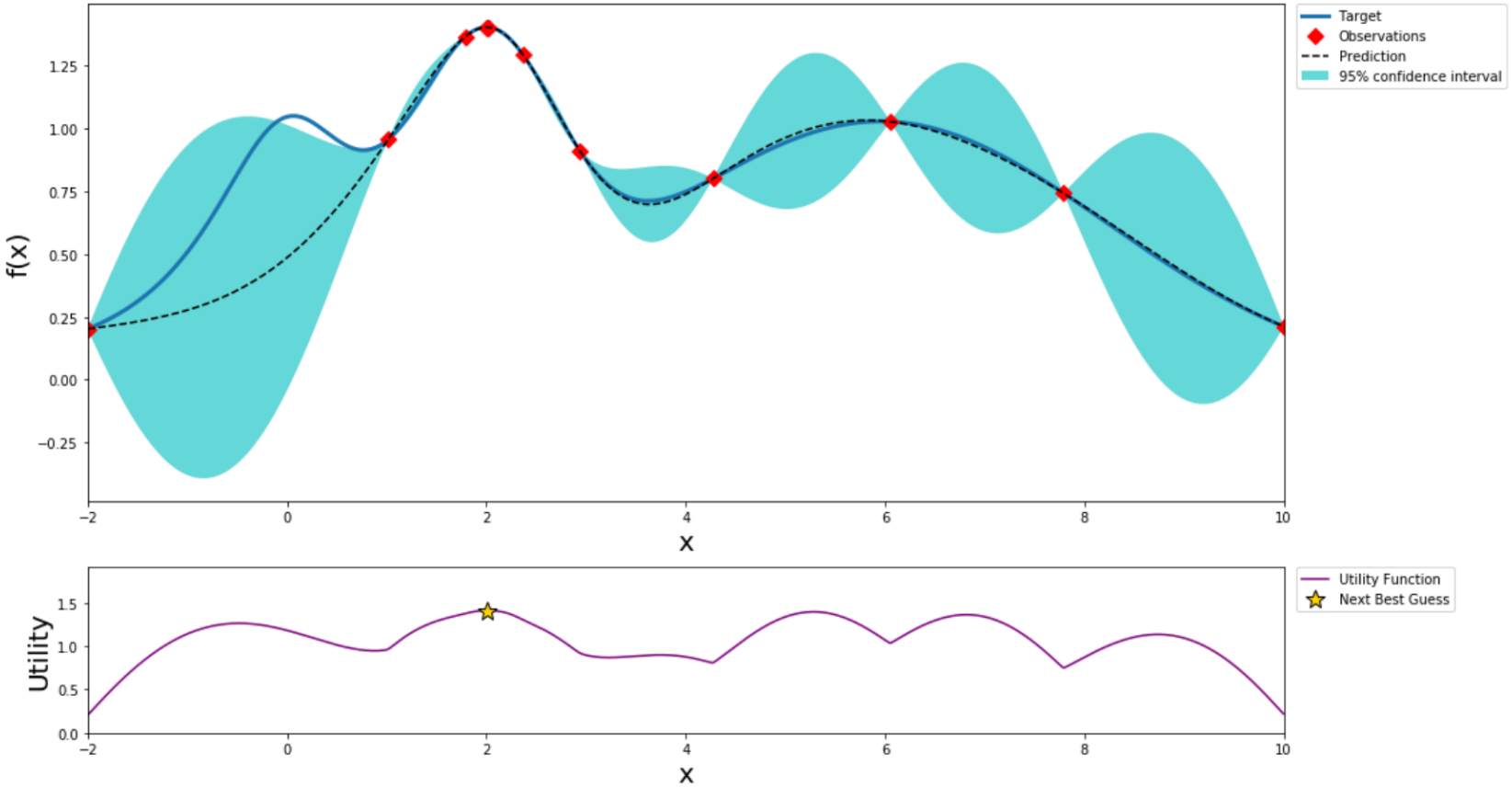
After 10 steps of GP (and two random points)



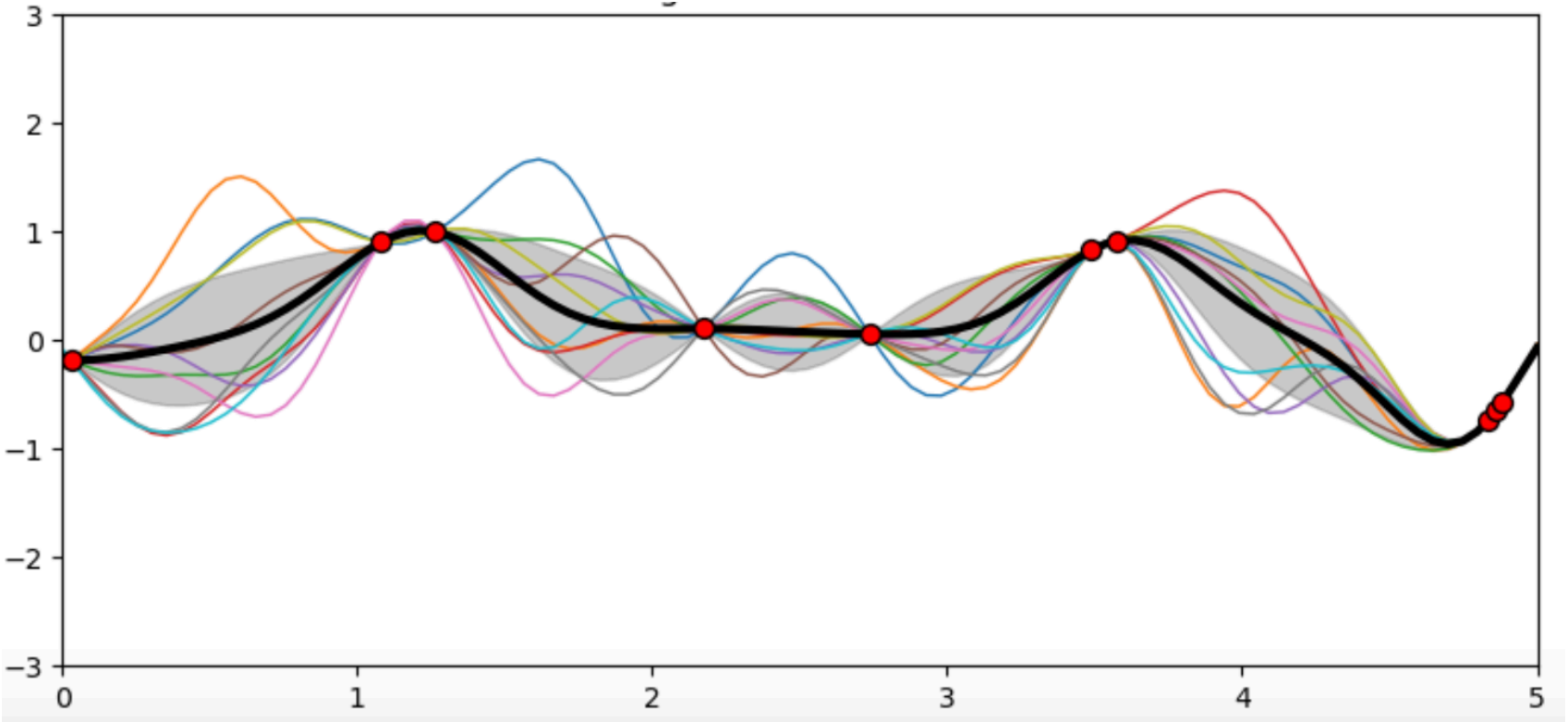
After 11 steps of GP (and two random points)



After 12 steps of GP (and two random points)



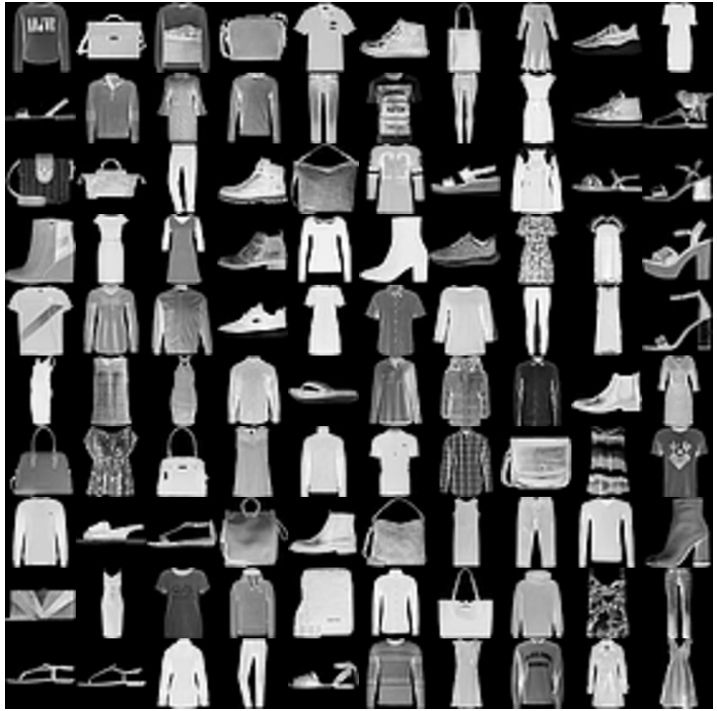
Functional Perspective



Fashion MNIST Demo

Fashion MNIST Dataset

- 60,000 training examples
- 10,000 test examples
- Each example is a 28x28 grayscale image
- 10 label classes
- Intended replacement for the original MNIST dataset



Label	Description
0	T-shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot

https://github.com/zalandoresearch/fashion-mnist

Demo

https://nbviewer.jupyter.org/github/niallturb/pylondinium18/blob/master/fashion_mnist_bayes_opt.ipynb

References

Snoek et al., 2012 – Practical Bayesian Optimization of Machine Learning Algorithms
http://papers.nips.cc/paper/4522-practical-bayesian-optimization-of-machine-learning-algorithms.pdf

Bergstra & Bengio, 2012 – Random Search for Hyperparamter Optimization
http://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf

Rasmussen & Williams, 2006 – Gaussian Processes for Machine Learning
http://www.gaussianprocess.org/gpml/

François Chollet, 2017 – Deep Learning with Python
https://www.manning.com/books/deep-learning-with-python

Thank you

Questions?