Simulated Annealing and Sampling

Practical Machine Learning & Deep Learning course assignment 1

Denis Rangulov
Innopolis University
d.rangulov@innopolis.ru

ABSTRACT

This document contains description of my implementation an optimization algorithm called Simulated Annealing (SA) in two tasks.

The first task is fitting the neural network using SA as an optimization method. Then comparing the results with traditional fitting using back-propagation.

Due to the nature of this optimization approach, it does not require the function to be differentiable. This allows to apply this method to optimize combinatorial problems. So the second task is finding the optimal traveling salesman path using SA for 30 Russian cities.

The document follows this structure for two tasks:

- Specific of implementation
- Evaluating the results

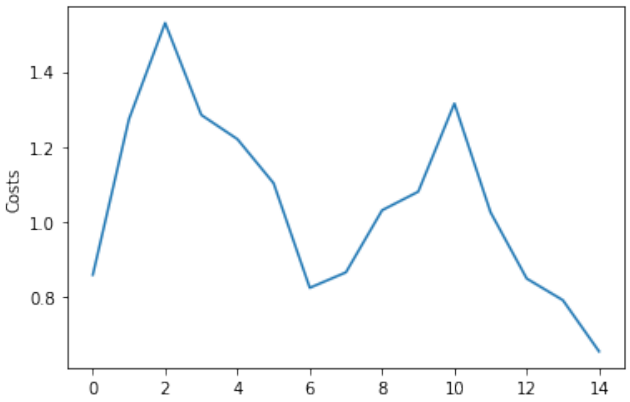

Figure 1: Loss function of SA optimization with first annealing rates

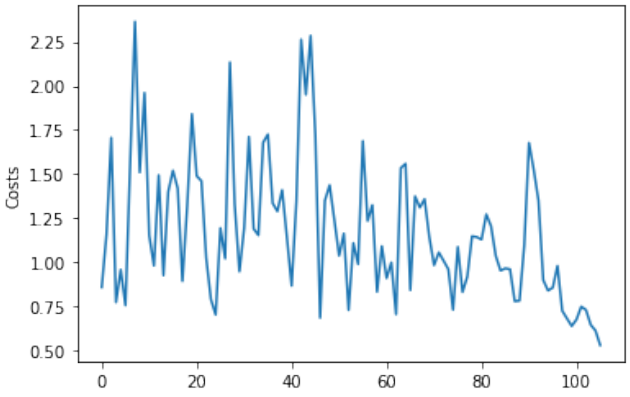

Figure 2: Loss function of SA optimization with second annealing rates

1 USER MANUAL: HOW TO REPRODUCE CODE

Both task are in jupiter notebook files. To reproduce code you should run it in Google Colab (solution files contain a button to do it quickly).

2 TASK 1

The model of neural network is implemented using Keras framework. It contains two Dense layers with five neurons. It does not use bias due to simpleness of iterating through model layer weights. So model has 60 weight parameters in total to be optimized. They use default initialization. By default, Keras uses Glorot initialization with a uniform distribution.

After default initialization model has 35.7% train accuracy and 26.3% test accuracy. Two annealing rates was chosen for evaluating algorithm. Duration of training with last annealing rate is 3.356 seconds.

Also, model are trained using vanilla gradient decent with learning rate 0.01 for 917 epochs. Train accuracy is 66.7% and test accuracy is 65.8%. Duration is 5.338 seconds.

Annealing rates	Number of steps	Train accuracy	Train accuracy
0.9	88	66.7%	65.8%
0.99	917	82.1%	84.2%

Table 1: Results of experiments

3 TASK 2

Data was loaded and preprocessed using pandas library. Standart SA are implemented. Improvement is difference between distance of final path and initial. Two annealing rates are analysed.

Annealing rates	Number of steps	Improvements
0.99	1375	2.49%
0.9995	27625	27.6%

Table 2: Results of experiments

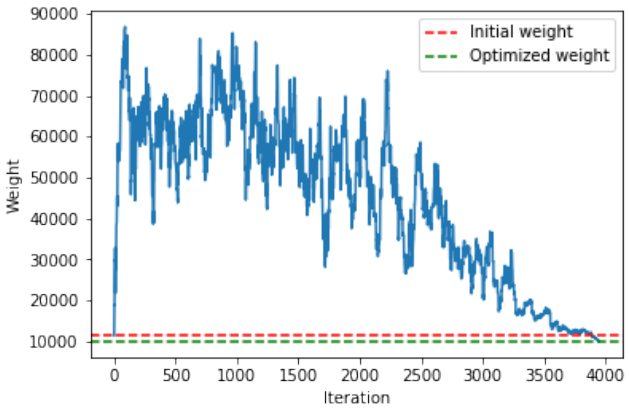

Figure 3: Optimization

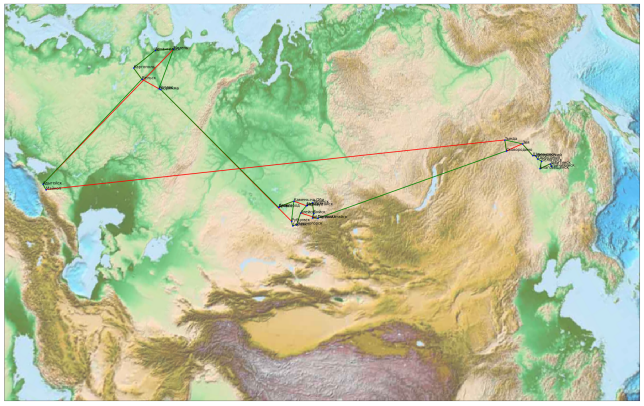

Figure 4: Visualization

4 CONCLUSION

As many other randomized algorithms, SA appears to be surprisingly simple, yet efficient at achieving different tasks comparing with traditional approaches.

Link to github repository: https://github.com/DenisRang/a1-simulated-annealing-and-sampling

REFERENCES