Timur Khazhiev BS4-DS2

Assignment 1: Simulated Annealing and Sampling

[github link]

Task 1

In this task we needed to produce neural network which will be optimized by simulated annealing approach. To do so we need to specify the simple model (Keras used):

```
Neural Network Model Summary:
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
relu1 (Dense)                (None, 10)                40
_________________________________________________________________
relu2 (Dense)                (None, 10)                100
_________________________________________________________________
sotfmax (Dense)              (None, 3)                 30
=================================================================
Total params: 170
Trainable params: 170
Non-trainable params: 0
_________________________________________________________________
None
```

note: to show activations name field used, for loss categorical cross entropy used

For the annealing these parameters were used:

```
T = 100.0
T_min = 1e-8 Temperature to stop
anneal_rate = 0.1
max_steps = 100 Steps between annealings
```

Results were pretty surprising:

Simulated annealing	Adam (lr=0.001) on same model	SGD (lr=0.01) on same model

`Time taken: 14.9 sec`
`Test set loss: 0.128090`
`Test set accuracy: 0.966667`
	`Time taken: 15.0 sec`
	`Test set loss: 0.146576`
	`Test set accuracy: 0.966667`
		`Time taken: 13.2 sec`
		`Test set loss: 0.146884`
		`Test set accuracy: 0.966667`

For conclusion simulated annealing approach (at least on this problem) can compete with optimizer giants with about similar results, thus making it useful for some tasks.

References used:

Task itself

SA implementation in PyTorch

Simple model for Iris

Keras docs

Paper on SA and DL

Task 2

In this task we were required to solve classical Travelling salesman problem using simulated annealing on top30 cities in Russia.

For the annealing these parameters were used:

```
T = 1000.0
T_min = 1e-8 Temperature to stop
max_steps = 100 Steps between annealings
```

Anneal rate 0.1 (slow)	Anneal rate 0.001 (Fast)	Anneal rate 0.01 (Mid)

`Initial distance: 66906.0`
`Time taken: 53.6 sec`
`Best path distance: 25467.1`
	`Initial distance: 55232.6`
	`Time taken: 17.6 sec`
	`Best path distance: 23606.6`
		`Initial distance: 62472.3`
		`Time taken: 27.54 sec`
		`Best path distance: 27245.8`

For the conclusion I found that anneal rate affects the time but not always result. Most of code runs converged to 20000 - 25000 being the best path. Observations showed that first several temperature reports (every anneal) showed huge jump between distances, slowing down at the last anneals (as, obviously, expected).

References used:

Task itself

Java implementation