Assignment #1: Simulated Annealing and Sampling - Evgeny Sorokin [BS16-DS2]

[S20] Practical Machine Learning & Deep Learning

https://hackmd.io/@nomemm/Sy9f3N7-8

Task 1

Model architecture:

```
class IrisNet(nn.Module):
    def __init__(self):
        super(IrisNet, self).__init__()
        self.fc1 = nn.Linear(4, 50)
        self.fc2 = nn.Linear(50, 50)
        self.fc3 = nn.Linear(50, 3)
        self.softmax = nn.Softmax(dim=1)
        self.apply(self.init_weights)

    def forward(self, X):
        X = F.relu(self.fc1(X))
        X = self.fc2(X)
        X = self.fc3(X)
        X = self.softmax(X)

        return X
```

Analysis

0. Algorithm adds a random noise (either Gaussian or Uniform) to weights of the network, thus updates the energy function.
1. `GaussianSampler` provides better results than `UniformSampler` (see `sampler`)
2. More aggressive weights' update leads to better loss & precision rates (not only $\text{uniform}(-1,1)$, but $N(0,3)$)
3. Linear decay $(T-a)$ of temperature works stable & converges other than geometric decay $(T*a)$ (see `T_decay` lambda function).
4. Adding alpha decay improved results in case of long runs (see `alpha_decay`)
5. One of the best results is achieved by

```
T0=4000.0,
alpha=1.5,
alpha_decay=lambda a, t: a * 0.95 if t % 500 == 0 else a,
T_decay=lambda T, a: T-a,
sampler=GaussianSampler(0, 3)
```

Results:

```
SA took 3216 steps, final temperature = -0.6068306399342867
Test loss : 0.8409
prediction accuracy: 0.7105263157894737
macro precision: 0.8166666666666668
micro precision: 0.7105263157894737
macro recall: 0.7380952380952381
micro recall: 0.7105263157894737
```

6. Average accuracy is 40%, when `UniformSampler(-1, 1)` is used.

```
T0=20000.0,
alpha=4.5,
alpha_decay=lambda a, t: a * 0.8 if t % 300 == 0 else a,
T_decay=lambda T, a: T-a,
sampler=UniformSampler(-1, 1)
```

Results:

```
SA took 6150 steps, final temperature = -0.6121586356881892
Test loss : 1.1304
prediction accuracy: 0.42105263157894735
macro precision: 0.3680555555555555
micro precision: 0.42105263157894735
macro recall: 0.40740740740740744
micro recall: 0.42105263157894735
SA with uniform sampling took 2.812 seconds
```

7. Time:

- Average time `SA + Uniform` **= 2.75sec**
- Average time `SA + Gaussian` **= 1.48 sec**.
- `SGD` time on 1000 epochs = **0.7 sec**
  Thus Gaussian converges faster than Uniform and has better results.

8. SGD:

- 1000 epochs
- lr=0.01
- no complex decay policies

```
Test loss : 0.6538
prediction accuracy: 0.9210526315789473
macro precision: 0.9076923076923077
micro precision: 0.9210526315789473
macro recall: 0.9153439153439153
micro recall: 0.9210526315789473
Adam took 0.774 seconds
```