Machine Intelligence 2 **Exercise Sheet 7**
SoSe 2020, Obermayer/Kashef due: 2020-06-11 23:59

Stochastic Optimization

Exercise 7.1: Simulated Annealing (5 points)

Simulated annealing can be used to optimize a cost function $E : \underline{\mathbf{s}} \to \mathbb{R}$ where the state $\underline{\mathbf{s}}$ is a set of discrete state variables $s_i \in \{-1, +1\},\ i = 1, \ldots, N$. For an all-to-all connected "network" with $N = 6$ binary nodes, this means that $\underline{\mathbf{s}} \in \{-1, +1\}^6$. Our cost ("energy") is defined as follows:

$$E(\underline{\mathbf{s}}) = -\frac{1}{2} \sum_{i,j=1}^{N} w_{ij} s_i s_j,$$

where $w_{ij} = w_{ji} \in \mathbb{R}$, and $w_{ii} = 0$.
The probability that the network is in a specific state $\underline{\mathbf{s}}$ with energy $E(\underline{\mathbf{s}})$ is given by

$$P(\underline{\mathbf{s}}) = \frac{1}{Z} \exp\big(-\beta E(\underline{\mathbf{s}})\big),$$

where the *partition function* Z guarantees $P(\underline{\mathbf{s}})$ to be a valid probability mass function and is given as the sum over all possible configurations, i.e. $Z = \sum_{\underline{\mathbf{s}}} \exp\big(-\beta E(\underline{\mathbf{s}})\big)$.

Write a program that finds the optimal configuration $\underline{\mathbf{s}}^*$ for a given set of weights $\underline{\mathbf{W}}$ as follows

A: Initialization:

1. Set both β_0,$\tau > 1$ (make sure β_0 is small enough)[1], and $t_{max} > 150$. Try both $M = 1$ and $M = 20$.
2. Set the initial state $\underline{\mathbf{s}}$ randomly;
3. Set $\underline{\mathbf{W}}$ arbitrarily, e.g. $w_{ij} \in [-1, 1]$ uniformly random, but force (also during optimization) $\underline{\mathbf{W}}$ to be symmetric and have a zero diagonal.

B: Optimization: for each iteration $t = 0, \ldots, t_{max}$

1. Repeat the following M times (state update loop):
   (i) Select node i randomly
   (ii) Determine the energies for the two options of s_i and compute their difference

$$E_{s_i} = -\frac{1}{2} \sum_{j \in \mathcal{N}_i} w_{ij} s_i s_j \quad \to \quad \Delta E = E_{-s_i} - E_{s_i} = -2E_{s_i}$$

   where $\mathcal{N}_i$ is the set of neighbors of node i. Since we have all-to-all connections the set of neighbors is effectively **all other** nodes.
   (iii) flip state s_i with probability $P(s_i \to -s_i) = \dfrac{1}{(1 + e^{\beta_t \Delta E})}$
2. Increase β using $\beta_{t+1} = \tau \beta_t$ before iterating over the state update loop another M times.

[1] By setting $\beta > 1$ we will have actually already skipped the exploration phase.

C: Visualization: For both $M=1$ and $M=20$ in the same figure.

1. Plot the temperature $T_t = \frac{1}{\beta_t}$ and the energy $E(\underline{\mathbf{s}})$ over the iterations $t = 0, \ldots, t_{max}$.
2. Show the energy $E(\underline{\mathbf{s}})$ for all possible 2^6 states using a bar plot. The order of the states is not relevant. Additionally, plot the probabilities $P(\underline{\mathbf{s}})$ for different β as a bar plot. No need to do this for all values of β, simply choose $\{\beta\}$ in a way, that the probability distributions differ sufficiently to illustrate the effect of β on $P(\underline{\mathbf{s}})$ (e.g. β changes by one order of magnitude).

Exercise 7.2: Mean-Field Annealing (5 points)

Mean-field annealing is a deterministic approximation of simulated annealing. During optimization each node s_i corresponds to the first moment of the original i-th binary variable and thus takes continuous values. The states s_i represent the mean with respect to the factorized distribution $Q(\underline{\mathbf{s}}) \approx P(\underline{\mathbf{s}})$.

Consider again an all-to-all connected network with $N = 6$ nodes. The cost (energy) function remains the same:

$$E(\underline{\mathbf{s}}) = -\frac{1}{2} \sum_{i,j=1}^{N} w_{ij} s_i s_j,$$

where the components $w_{ij} \in \mathbb{R}$ are symmetric, and $w_{ii} = 0$. The approximated probability of a state $\underline{\mathbf{s}}$ is now given by

$$Q(\underline{\mathbf{s}}) = \frac{1}{Z_Q} \exp\Big(-\beta \sum_j e_j s_j \Big).$$

Write a program that finds the optimal configuration $\underline{\mathbf{s}}^*$ of the network for given weights $\underline{\mathbf{W}}$. It should execute the following steps:

A: Initialization:

1. β_0 small enough, $\tau > 1$, set t_{max}, ε small enough
2. Set the initial state $\underline{\mathbf{s}}$ with non-zero values chosen randomly.
3. Choose the same $\underline{\mathbf{W}}$ used in the exercise above.

B: Optimization: for each iteration $t = 0, ..., t_{max}$

1. repeat the following until convergence, i.e., $|\boldsymbol{e}_{\text{new}} - \boldsymbol{e}_{\text{old}}| < \varepsilon$
   - for $i = 1, \ldots, N$:
     - compute mean-field
       $$e_i = -\sum_{j \in \mathcal{N}_i} w_{ij} s_j$$
       where $\mathcal{N}_i$ is the set of neighbors of node i. In the case of all-to-all connections it is effectively the set of **all other** nodes.
     - update the state using $s_i = \tanh(-\beta e_i)$
2. Increase β using $\beta_{t+1} = \tau \beta_t$.

C: Visualization:

1. Plot the temperature $T_t = \frac{1}{\beta_t}$ and the energy $E(\underline{\mathbf{s}})$ over the iterations $t = 0, \ldots, t_{max}$.

Total 10 points.