Assignment 4
CS289: Algorithmic Machine Learning, Fall 2016
Due: November 30, 10PM

Guidelines for submitting the solutions:

- The assignments need to be submitted on Gradescope. Make sure you follow all the instructions - they are simple enough that exceptions will not be accepted.
- To save my eyes (and perhaps, more importantly, a few of your points) please use a good scanner and/or digitize the scan as per the instructions on the class web page.
- Start each problem or sub-problem on a separate page even if it means having a lot of white-space and write/type in large font.
- The solutions need to be submitted by 10 PM on the due date. No late submissions will be accepted.
- Please adhere to the code of conduct outlined on the class page.

1. Consider a neural network with a) one input layer with two inputs (x_1, x_2), b) two hidden layers with two nodes each, c) one output node. (Thus, there are four layers in total including the two hidden layers).

   Let $f(x)$ denote the output of the network on input x with the activation function being the Sigmoid. Compute the partial derivative of f with respect to the following weights. Your answer should be in terms of the X_i^ℓ, Y_i^ℓ variables that we defined in class.

   - w_{11}^3.
   - w_{11}^2.
   - w_{11}^1.

   You do not have to type-in all your calculations. Just the most important ones and the final answers. [2 points]

   (Recall: w_{ij}^ℓ denotes the weight of the edge from node i in layer $\ell - 1$ to node j in layer ℓ; X_i^ℓ denotes the output of the i'th node of the ℓ'th layer; Y_i^ℓ denotes the 'input' of the i'th node of the ℓ'th layer. The layers are numbered $0, 1, 2, 3$ with 0 being the input layer).

2. For $i \leq m$, let $e_i \in \mathbb{R}^m$ be the vector with the i'th coordinate being 1 and the rest being 0. Let $C = ConvexHull(e_1, e_2, \ldots, e_m)$. Give an algorithm that given a vector $v \in \mathbb{R}^m$ checks if $v \in C$ without using linear programming. Your algorithm should run in time $O(m)$. [2 points]

3. The goal of this exercise is to partially show one of the properties of Gram matrices we stated in class. Let $A \in \mathbb{R}^{m \times k}_{\geq 0}$ be a 'topics' matrix for a topic model (i.e., the columns sum up to 1). Let $\mathcal{W}$ be a distribution on weight vectors $\mathbb{R}^k_{\geq 0}$, i.e., $\mathcal{W}$ is a distribution on non-negative vectors in $\mathbb{R}^k$ whose sum of entries is 1. Consider the following process for generating a document according to the corresponding topic model: (a) Sample a weight vector w according to $\mathcal{W}$ and then sample s words independently according to the distribution Aw on words. For two words i, j, let $G_{i,j}$ be the probability that if we generate a document d using the above model, then the first two words of d are word i, word j respectively. Let $G \in \mathbb{R}^{m \times m}$ matrix of these probabilities. Show that $G = ARA^T$ for some non-negative matrix R. [3 points]

   [Hint: Let Y_1, Y_2 be the random variables denoting the first and second words of the document. Then $G_{i,j} = \mathsf{Pr}[Y_1 = i, Y_2 = j]$. Compute this by conditioning on the topics t_1, t_2 that lead to generating Y_1, Y_2.]

4. *(This problem is perhaps a bit trickier than others we've seen, but the hints will help.)

   Define $X \in \mathbb{R}^{m \times m}$ by $X_{ij} = (i - j)^2$. Show that for $m = 2^k$, the non-negative rank (nnr) of X is at most $2k$. [3 points]

   [Hint: Let X_k be the matrix for $m = 2^k$. Split X_k into four quadrants X^1, X^2, X^3, X^4 corresponding to entries $\{i, j : 1 \leq i, j \leq 2^{k-1}\}$, $\{i, j : 1 \leq i \leq 2^{k-1}, 2^{k-1} + 1 \leq j \leq 2^k\}$, $\{i, j : 2^{k-1} + 1 \leq i \leq 2^k, 1 \leq j \leq 2^{k-1}\}$, $\{i, j : 2^{k-1} + 1 \leq i, j \leq 2^k\}$.

   What can you say about the pieces X^1, X^4 in relation to X_{k-1}? Next, for indices i, j corresponding to X^2 show the following: if $j' = 2^k - j + 1$, then, $X^2_{i,j} = (j - i)^2 = (2^k - j' - i + 1)^2 = (j' - i)^2 + (2^k - 2i + 1)(2^k - 2j' + 1)$.

   Use the above facts to show that $nnr(X_k) \leq nnr(X_{k-1}) + 2$ and then use induction.]