Introduction to Information Theory and Machine Learning July 15^{th}, 2021.[1]
(Prof. Permuter Haim, Mr. Bashar Huleihel and Mr. Omer Luxembourg)

Final Exam - Moed B
Total time for the exam: 3 hours!

Please copy the following sentence and sign it: " I am respecting the rules of the exam: Signature:_____ "

Important: For **True / False** questions, copy the statement to your notebook and write clearly true or false. You should prove the statement if true, or disprove it, e.g. by providing a counter-example, otherwise.

1) **Imbalanced Data and Backpropagation (32 Points):** You're trying to classify RGB images if giraffe present (1) and giraffe absent (0) using a deep neural network. Unfortunately, your data set is imbalanced, and consist of:
   - 2000 images with a giraffe
   - 200 images with no giraffe

   a) (4 points) To address the imbalance problem, we would like to oversample our data by using augmentations, while avoiding having the same example twice in our dataset. Suggest two data augmentation techniques you could use to help address the class imbalance problem.
   b) (4 points) Instead of data augmentation, you want to experiment with other techniques. Here's the architecture of your network:

$$\begin{aligned} z_1 &= W_1 x^{(i)} + b_1, \\ a_1 &= ReLU(z_1), \\ z_2 &= W_2 a_1 + b_2, \\ \hat{y}^{(i)} &= \sigma(z_2), \\ L^{(i)} &= \alpha \cdot y^{(i)} \cdot \log(\hat{y}^{(i)}) + \beta \cdot (1 - y^{(i)}) \cdot \log(1 - \hat{y}^{(i)}), \\ J &= -\frac{1}{m} \sum_{i=1}^{m} L^{(i)}, \end{aligned}$$

   where $\hat{y}^{(i)} \in \mathbb{R}$, $y^{(i)} \in \mathbb{R}$, $x^{(i)} \in \mathbb{R}^{D_x \times 1}$, $W_1 \in \mathbb{R}^{D_{a_1} \times D_x}$, $W_2 \in \mathbb{R}^{1 \times D_{a_1}}$. Note that m is the size of the dataset and that the RGB images are flattened into vectors of length D_x before being fed into the network. What are the dimensions of b_1 and b_2?
   c) (4 points) Explain why α and β are useful for the imbalance data problem?
   d) (6 points) What are a reasonable values for the pair (α, β)? Provide specific values for these weightings. **Hint:** if $\alpha = 1, \beta = 1$ we get the original Binary Cross-Entropy loss function. Think why this function isn't right for the question's scenario.
   e) (4 points) You decide to add $L2$ regularization to this model. Write your new cost function.
   f) (4 points) Using this new cost function, write down the update rule for W_1 as a function of $\frac{\partial J}{\partial W_1}$ and W_1.
   Hint: assume you are using gradient descent. Use η as your learning rate.
   g) (6 points) Suppose you use $L1$ regularization instead. How would you expect the weights learned using $L1$ regularization to differ from these learned using $L2$ regularization?

2) **Probability mass function estimation (34 Points):** In this question, we will develop an algorithm for probability mass function (PMF) estimation based on a given sample set. Let $X \sim P_X$, $Y \sim P_Y$ and denote the joint PMF of (X, Y) by P_{XY}. Further, let U_X be the PMF of the uniform discrete probability measure over the alphabet of X, i.e. $U_X(x) = \frac{1}{\mathcal{X}}$ for any $x \in \mathcal{X}$.
   a) (5 points) Prove the following equality:

$$H(X, Y) = H(P_{XY}, U_{XY}) - D_{KL}(P_{XY} || U_{XY}),$$

where $H(P_{XY}, U_{XY})$ denote the cross-entropy between P_{XY} and U_{XY}.

b) (5 points) Express $H(P_{XY}, U_{XY})$ as function of the alphabets $\mathcal{X}$ and $\mathcal{Y}$.

c) (8 points) Propose a neural network based algorithm to estimate $H(X, Y)$ from a sample set $\{(x_i, y_i)\}_{i=1}^{n} \sim P_{XY}$. Denote the estimator by $\hat{H}_n(X, Y)$, and provide a block diagram of your proposed algorithm.

d) (8 points) For sufficient large n, is your proposed algorithm provides a lower / upper bound on $H(X, Y)$? Theoretically, when will the algorithm achieve equality?

e) (8 points) Assume that, for a sufficient large n, your suggested algorithm is converged. Suggest how to estimate P_{XY} based on the previous items.

3) **Huffman codes (34 Points):** Consider a random variable X which takes 6 values $\{A, B, C, D, E, F\}$ with probabilities $(0.5, 0.25, 0.1, 0.05, 0.05, 0.05)$ respectively.

a) (5 points) Construct a binary Huffman code for this random variable. What is the average length of the code?

b) (5 points) Construct a quaternary Huffman code for this random variable, i.e., a code over the alphabet of four symbols (call them a, b, c, and d). What is the average length of this code?

c) (4 points) One way to construct a binary code for a random variable is to start with a quaternary code, and convert the symbols into binary using the mapping $a \to 00$, $b \to 01$, $c \to 10$, and $d \to 11$. What is the average length of the binary code for the above random variable constructed by this process?

For any variable X, let L_H be the average length of the binary Huffman code for the random variable, and let L_{QB} be the average length code constructed by first building a quaternary Huffman code and converting it to binary.

d) (6 points) **True/False:** The inequality $L_H \leq L_{QB}$ always holds.

e) (7 points) Show that $L_{QB} < L_H + 2$.
Hint: Consider to use the fact that the average length of a quaternary Huffman code satisfies $L_Q < \frac{H_2(X)}{2} + 1$.

f) (7 points) Give an example where the code constructed by converting an optimal quaternary code is also the optimal binary code, i.e., example for which $L_H = L_{QB}$.

Good Luck!