(Prof. Permuter Haim, Mr. Eli Shmuel, Mr. Dor Tsur, 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) **True or False (24 Points)**:
   a) For a source X, the average length of a non-singular code must be **greater than or equal** its entropy, $H(X)$.
   False. Take $\mathcal{X} = \{0, 1, 00, 11\}$ and let X be uniformly distributed over $\mathcal{X}$. The average length this non-singular code is $\frac{1}{4}(1+1+2+2) = 1.5 < H(X) = \log_2(4) = 2$.
   b) All typical sequences in $\mathcal{A}_\epsilon^{(n)}$ have the same probability up to a negligible factor.
   True. All typical sequences in $\mathcal{A}_\epsilon^{(n)}$ have **approximately** the same probability which is $\approx 2^{-nH(X)}$, since by definition, the typical set $\mathcal{A}_\epsilon^{(n)}$ with respect to $p(x)$ is the set of sequences $x^n \in \mathcal{X}^n$ such that
   $$2^{-n(H(X)+\epsilon)} \leq p(x^n) \leq 2^{-n(H(X)-\epsilon)}. \tag{1}$$
   c) If $W \perp\!\!\!\perp X$, $X \perp\!\!\!\perp Y$ and $W \perp\!\!\!\perp Y$ then (W, X, Y) are mutually (jointly) independent.
   False. Take W and X, each distributed according to Bernoulli(0.5), and $Y = W \oplus X$.
   Then $1 = H(Y) \neq H(Y|W, X) = 0$ and Y is dependent of (W, X).
   d) Let $P(y|x)$ characterize a discrete memoryless channel (DMC) with input and output alphabets $\mathcal{X} = \{1, 2, \ldots, m\}, \mathcal{Y} = \{1, 2, \ldots, n\}$, respectively. Assume for all $y \in \mathcal{Y}$ that $P(y|X = 1) = P(y|X = 2) = \cdots = P(y|X = m)$. The capacity of this DMC is 0.
   True. For any chosen P_X, we have $P(y) = P(y|x)$ for all $x, y \in \mathcal{X} \times \mathcal{Y}$, i.e., $Y \perp\!\!\!\perp X$ and $H(Y|X) = H(Y)$. Hence $I(X;Y) = H(Y) - H(Y|X) = 0$.

2) **Channels with dependence between letters (32 points)**: Consider a channel over a binary alphabet that takes in two bit symbols and produces a two bit output, as determined by the following mapping: $00 \to 01$, $01 \to 10$, $10 \to 11$, and $11 \to 00$. Thus if the two bit sequence 01 is the input to the channel, the output is 10 with probability 1. Let X_1, X_2 denote the two input symbols and Y_1, Y_2 denote the corresponding output symbols.
   a) Find $I(X_1, X_2; Y_1, Y_2)$ as a function of the input distribution on the four possible pairs of inputs.
   Remark: For convenience, denote the input distribution on the four possible pairs by p_{00}, p_{01}, p_{10}, and p_{11}.
   Solution:
   If we look at pairs of inputs and pairs of outputs, this channel is a noiseless four input four output channel. Let the probabilities of the four input pairs be p_{00}, p_{01}, p_{10}, and p_{11}, respectively. Then the probability of the four pairs of output bits are p_{11}, p_{00}, p_{01}, and p_{10}, respectively. Accordingly,
   $$\begin{aligned} I(X_1, X_2; Y_1, Y_2) &= H(Y_1, Y_2) - H(Y_1, Y_2|X_1, X_2) \\ &= H(Y_1, Y_2) - 0 \\ &= H(p_{11}, p_{00}, p_{01}, p_{10}). \end{aligned}$$
   b) What is the capacity of a pair of transmissions on this channel?
   Solution:
   The capacity of the channel is achieved by a uniform distribution over the inputs, which produces a uniform distribution on the output pairs. That is
   $$C = \max_{P(x_1,x_2)} I(X_1, X_2; Y_1, Y_2) = 2,$$
   while the maximizing $P(x_1, x_2)$ puts probability 0.25 on each of the pairs 00, 01, 10, and 11.
   c) Calculate $I(X_1; Y_1)$ under the maximizing input distribution.
   Solution:
   To calculate $I(X_1; Y_1)$, we need to calculate the joint distribution of X_1 and Y_1. From the joint distribution of X_1X_2 and Y_1Y_2 under an uniform distribution (which is optimal), it is easy to calculate the joint distribution

of X_1 and Y_1. In particular we obtain:

$$P(x_1, y_1) = 0.25,$$

for any $x_1, y_1 \in \{0, 1\}$. Therefore, we can see that the marginal distributions of X_1 and Y_1 are both $(0.5, 0.5)$ and that the joint distribution is the product of the marginals, i.e., X_1 is independent of Y_1, and therefore $I(X_1; Y_1) = 0$.

d) Does the maximizing input distribution that you found necessarily maximize the mutual information between the individual symbols, and their corresponding outputs? Explain your answer.
Solution:
From the previous section we can conclude that the distribution of the input sequences that achieves capacity does not necessarily maximize the mutual information between individual symbols and their corresponding outputs. In particular, we obtained that $I(X_1; Y_1) = 0$ which, for an arbitrary binary channel, is clearly not necessarily the optimal value when maximizing over $P(x_1)$.

3) **Classifier (18 points)** A classifier can be written as $H(x) = sign(F(x))$, where $H(x) : \mathbb{R}^d \to \{-1, 1\}$ and $F(x) : \mathbb{R}^d \to \mathbb{R}$. Alternatively, for a given x we want to label it 1 or -1 (instead of 1 or 0 as we saw in class). The labeling is done by the sign of a given scalar function $F(x)$, which does not matter for us.
To obtain the parameters in $F(x)$, we need to **minimize** the loss function averaged over the training set $\{x_i, y_i\}_{i=1}^N$, $Loss = \frac{1}{N} \sum_{i=1}^N \mathcal{L}(y_i, F(x_i))$, where $x_i \in \mathbb{R}^d$ is the features vector, $y_i \in \mathbb{R}$ is the label, and $\mathcal{L}$ is the loss function. Define the loss function as follows:

$$\mathcal{L}(y_i, F(x_i)) = l\left(y_i \cdot F(x_i)\right), \quad l(\cdot) : \mathbb{R} \to \mathbb{R}$$

For **example**, for linear classifiers, $F(x_i) = w_0 + \sum_{j=1}^d w_j x_{i,j}$, and $y_i \cdot F(x_i) = y \cdot (w_0 + \sum_{j=1}^d w_j x_{i,j})$.
Notation - x_j^i is the j element in the features vector x^i.

a) Which loss functions from Figure 1 are appropriate to use in classification? For the ones that are not appropriate, explain why. In general, what conditions does l have to satisfy in order to be an appropriate loss function? The x axis is $y_i F(x_i)$, and y axis is $l(y_i F(x_i))$.
Answer: We want the term $y_i F(x_i)$ to be always positive, because we want y_i and $F(x_i)$ will have the same sign (positive value of $y_i F(x_i)$ is correct classification and negative is the incorrect). Thus, (a) and (b) are appropriate to use in classification - negative values of the multiplication are penalized, while positive aren't. In (c), there is very little penalty for extremely misclassified examples, which correspond to very negative $yF(x)$. In (d) and (e), correctly classified examples are penalized, whereas misclassified examples

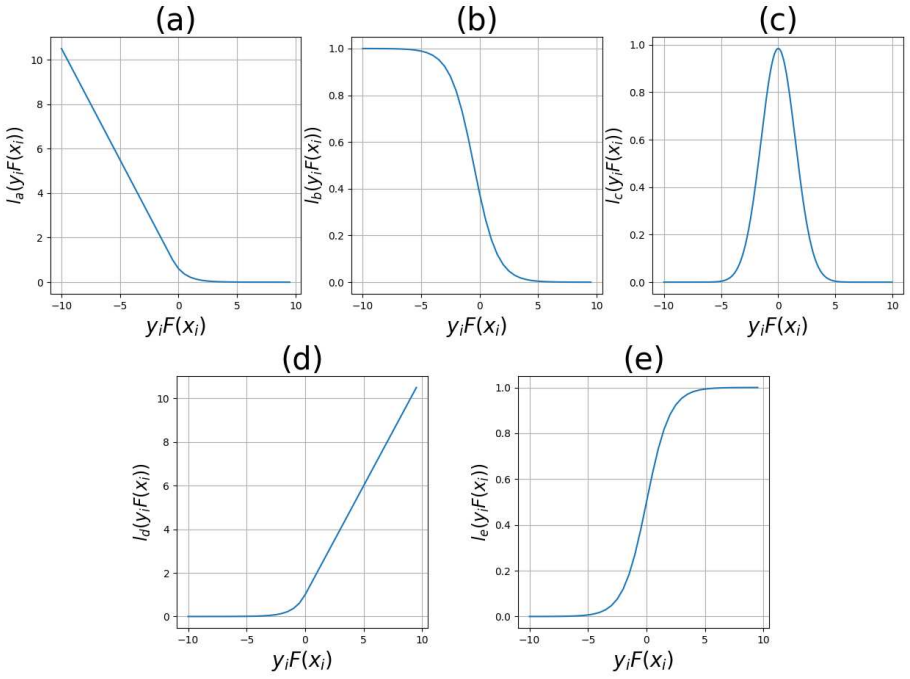

Fig. 1: Loss functions: The x axis is $y_i F(x_i)$, and the y axis is $l(y_i F(x_i))$.

are not. In general, l should approximate the 0-1 loss, and it should be a non-increasing function of $yF(x)$.[3]

b) Of the loss functions appropriate to use in classification (your previous section's answer), which one is the most immune to outliers? In other words, which loss function doesn't give large penalty for greater errors? Provide an explanation.
Remark: In the given model, x is outlier if $yF(x)$ is too negative or too positive.
Answer: (b) is more robust to outliers. For outliers, $yF(x)$ is often very negative. In (a), outliers are heavily penalized. So the resulting classifier is largely affected by the outliers. On the other hand, in (b), the loss of outliers is bounded. So the resulting classifier is less affected by the outliers, and thus more robust.

4) **Auto-Encoders (36 points)** Let $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}_x, \Sigma_x)$. We would like to create a generative model of $\mathbf{X}$.

a) (6 points) Assume we use a simple autoencoder (not variational) and X is some single-dimensional random variable (i.e., the encoder $F : \mathbb{R} \mapsto \mathbb{R}$ and so is the decoder $G : \mathbb{R} \mapsto \mathbb{R}$). What would be the optimal choice of F and G for MSE-minimizing?
Answer: Any bijection F will be a suitable choice for our encoder. Appropriately, we will choose $G = F^{-1}$. We will results with $\mathsf{MSE}(X, G(F)(X))) = \mathsf{MSE}(X, X) = 0$.

b) (8 points) In this section we consider the linear case for a variational autoencoder and we will apply the reparametrization trick. We assume that $\boldsymbol{X}$ is m-dimensional. Let:

$$\boldsymbol{\mu_z} = \begin{pmatrix} w_1^\mathsf{T}\mathbf{X} \\ \vdots \\ w_d^\mathsf{T}\mathbf{X} \end{pmatrix} + b, \qquad \Sigma_Z = A \cdot \mathit{diag}(\mathbf{X}), \qquad \mathit{diag}(\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix}) = \begin{pmatrix} x_1 & 0 & \dots & \dots & 0 \\ 0 & x_2 & 0 & \dots & 0 \\ \vdots & 0 & x_3 & \dots & 0 \\ 0 & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & \dots & x_m \end{pmatrix}$$

with $w_i \in \mathbb{R}^m$ for $i = 1 \dots d$, $A \in \mathbb{R}^{d \times m}$ and $b \in \mathbb{R}^d$.
What is the distribution of $\boldsymbol{\mu_z}$? What is the distribution of Z given a realization $X = x$?
Reminder: if $\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ and $Y = AX + b$ then $\mathbf{Y} \sim \mathcal{N}(A\boldsymbol{\mu} + b, A\Sigma A^\mathsf{T})$.
Answer: We can rewrite $\boldsymbol{\mu_z}$ as:

$$\boldsymbol{\mu_z} = W^\mathsf{T} X + b, \qquad W := (w_1, \dots, w_d).$$

Therefore, its distribution is $\boldsymbol{\mu_z} \sim \mathcal{N}(W^\mathsf{T}\boldsymbol{\mu_X} + b, W^\mathsf{T}\Sigma_X W)$. Given a realization $X = x$ we know by the reparametrization trick that $Z = \boldsymbol{\mu}_Z + \Sigma_Z \epsilon$, and because we have a specific realization of $\mathbf{X} = X$ then the distribution parameters of Z are also deterministic. Therefore,

$$Z = W^\mathsf{T} X + b + A \cdot \mathit{diag}(X)\epsilon.$$

The randomness of Z follows from $\epsilon \sim \mathcal{N}(\mathbf{0}, I)$. Denote:

$$M = A \cdot \mathit{diag(x)}, \quad \boldsymbol{n} = W^\mathsf{T} x + b,$$

and all that's left is to substitute them into the reminder equation.

c) (5 points) We now focus on the one-dimensional case. In class we had defined the following loss for the variational-autoencoder:

$$\mathcal{L} := \underbrace{\mathbb{E}_{q(z)}\left[\frac{(x - f(z))^2}{2c}\right]}_{:=\mathsf{MSE}} + \underbrace{D_{KL}(\mathcal{N}(g(x), h^2(x)), \mathcal{N}(0, 1))}_{:=\mathsf{D}} \tag{2}$$

where our goal is to apply $\min_{f,g,h} \mathcal{L}$. We define the functions as follows:

$$f(z) = z, \quad g(x) = \sum_{i=1}^{m} w_i x^i, \quad h(x) = \sum_{j=1}^{m} \exp(-u_j x)$$

Calculate $\frac{\partial \mathsf{MSE}}{\partial w_i}$ for some general i.
Answer: By the cahin rule we know that:

$$\frac{\partial \mathsf{MSE}}{\partial w_i} = \frac{\partial \mathsf{MSE}}{\partial g}\frac{\partial g}{\partial w_i}.$$

For the calculation of the first term, we note that:

$$\begin{aligned}\mathsf{MSE} &= \mathbb{E}_{q(z)}\left[\frac{(x-f(z))^2}{2c}\right] \\ &= \frac{1}{2c}\mathbb{E}_{q(z)}\left[x^2 - 2xz + z^2\right] \\ &= \frac{1}{2c}\mathbb{E}_{q(z)}\left[x^2\right] - \frac{2x}{2c}\mathbb{E}_{q(z)}\left[z\right] + \frac{1}{2c}\mathbb{E}_{q(z)}\left[z^2\right] \\ &= \frac{1}{2c}x^2 - \frac{x}{c}g(x) + \frac{1}{2c}(g^2(x) + h^2(x)).\end{aligned}$$

The rest of the solution follows from simple derivation based on the mentioned steps.

d) (8 points) The KL-divergence between two Gaussians $G_1 = \mathcal{N}(\mu_1, \sigma_1^2)$ and $G_2 = \mathcal{N}(\mu_2, \sigma_2^2)$ is given by:

$$D_{KL}(G_1, G_2) = \log\frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{2\sigma_2^2} - \frac{1}{2}.$$

Calculate $\frac{\partial \mathsf{D}}{\partial w_i}$ for some general i.
Answer: The solution follows from simply plugging the given distributions into the KL divergence formula and calculating the derivative, again, with the mentioned chain rule.

e) (3 points) Given some fixed learning rate μ, write down the SGD update rule for the models weights $\{w_i\}_{i=1}^m$.
Answer: Based on the calculated derivatives we have:

$$w_{i,t+1} = w_{i,t} + \mu\frac{\partial \mathcal{L}}{\partial w_{i,t}}.$$

f) (6 points) This section is independent of the previous ones. Assume we want to constraint our latent vector Z to have similar statistical characteristics as some other random vector Y. Propose a modification for the loss in (2) to obtain this request. Suggest a method to control how strongly we want to impose this constraint.
Answer: When we want to impose a statistical similarity constraint we can do this through regularization of some related KL-divergence. Therefore, if we want similarities between the statistical properties of Z and Y, we would like to minimize the KL divergence $D_{KL}(Q_Z||P_Y)$. If we want to control over the impact of this regularization factor, we will multiply it by some hyper-parameter β. The bigger β is, the stronger influence this factor has.

Good Luck!