Assignment 1

Deep Learning, WS24

Student		
Last name	First name	Matriculation Number
Nožić	Naida	12336462

Task 1 - Maximum Likelihood Estimation

For this task we are given a dataset $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, where $x_i \in \mathbb{R}^d$ and $y_i \in \{0, 1\}$, $i = 1, \ldots, n$. The x_i is a feature vector and y_i the corresponding class. In the following tasks we will be dealing with a classification problem. Additionally, each (x_i, y_i) tuple is assumed to be an i.i.d. sample from some true, unknown joint distribution $p^*(x, y)$

Task 1.1

For $X = (x_1, \ldots, x_n)^T$ and $y = (y_1, \ldots, y_n)^T$, the likelihood of the entire dataset under our model $p_\theta(y \mid X)$ is:

$$p_\theta(y \mid X) = \prod_{i=1}^{n} p_\theta(y_i \mid x_i) \tag{1}$$

The negative log-likelihood $\mathrm{NLL}(\theta) = -\log(p_\theta(y \mid X))$ in terms of single-sample likelihoods:

$$\mathrm{NLL}(\theta) = -\sum_{i=1}^{n} \log(p_\theta(y_i \mid x_i)) \tag{2}$$

Task 1.2

We are given the empirical distribution induced by $\mathcal{D}$: $p_\mathcal{D}(x, y) = p_\mathcal{D}(x)p_\mathcal{D}(y \mid x)$ with

$$p_\mathcal{D}(y \mid x_i) = \begin{cases} 1 & \text{if } y = y_i \\ 0 & \text{else} \end{cases} \qquad p_\mathcal{D}(x) = \frac{1}{n}\sum_{i=1}^{n} \delta(x - x_i)$$

where $\delta(\cdot)$ is the Dirac delta function centered around 0.

The Maximum-Likelihood estimator minimizes the expected *KL-Divergence* between the empirical distribution and the model distribution:

$$\arg\min_\theta \mathrm{NLL}(\theta) = \arg\min_\theta \mathbb{E}_{x \sim p_\mathcal{D}(x)} \left[D_{\mathrm{KL}} \left(p_\mathcal{D}(\cdot \mid x), p_\theta(\cdot \mid x) \right) \right]$$

Proof

$$D_{\mathrm{KL}} \left(p_\mathcal{D}(\cdot \mid x), p_\theta(\cdot \mid x) \right) = \mathbb{E}_{y \sim p_\mathcal{D}(\cdot \mid x)} \left[\log \frac{p_\mathcal{D}(y \mid x)}{p_\theta(y \mid x)} \right]$$

$$D_{\mathrm{KL}} \left(p_\mathcal{D}(\cdot \mid x), p_\theta(\cdot \mid x) \right) = \mathbb{E}_{y \sim p_\mathcal{D}(\cdot \mid x)} \left[\log p_\mathcal{D}(y \mid x) \right] - \mathbb{E}_{y \sim p_\mathcal{D}(\cdot \mid x)} \left[\log p_\theta(y \mid x) \right]$$

Since we are dealing with a minimization problem that depends on θ, then the left term $\mathbb{E}_{y \sim p_\mathcal{D}(\cdot \mid x)} \left[\log p_\mathcal{D}(y \mid x) \right]$ can be ignored because it will always be constant and does not depend on the θ.

For the right term we will use this general formula for expectation:

$$\mathbb{E}_{x \sim p(x)}[f(x)] = \sum_x p(x) \cdot f(x)$$

Therefore, the new equation is equal to:

$$D_{\mathrm{KL}} \left(p_\mathcal{D}(\cdot \mid x), p_\theta(\cdot \mid x) \right) = \mathbb{E}_{y \sim p_\mathcal{D}(\cdot \mid x)} \left[\log p_\mathcal{D}(y \mid x) \right] - \sum_y p_\mathcal{D}(y \mid x) \log p_\theta(y \mid x)$$

As mentioned, I will ignore the constant term and only include the relevant part of the optimization problem.

$$\arg\min_\theta -\mathbb{E}_{x \sim p_\mathcal{D}(x)} \left[\sum_y p_\mathcal{D}(y \mid x) \log p_\theta(y \mid x) \right]$$

$$\arg\min_\theta -\sum_x p_\mathcal{D}(x) * \sum_y p_\mathcal{D}(y \mid x) \log p_\theta(y \mid x)$$

$$\arg\min_\theta -\frac{1}{n} \sum_x \sum_{i=1}^{n} \delta(x - x_i) \sum_y p_\mathcal{D}(y \mid x) \log p_\theta(y \mid x)$$

The general definition od the Dirac function is given as:

$$\delta(x) = \begin{cases} +\infty & \text{if } x = 0, \\ 0 & \text{if } x \neq 0 \end{cases} \quad \text{and} \quad \int_{-\infty}^{\infty} \delta(x)\, dx = 1$$

We are not interested in values for which the Dirac function is equal to zero. Because of this, we will only focus on cases where $x = x_i$. This will eliminate the sum over the x values, since we are only observing the x_i. Our proof continues as follows:

$$\arg\min_\theta -\frac{1}{n} \sum_{i=1}^{n} \sum_y p_\mathcal{D}(y \mid x_i) \log p_\theta(y \mid x_i)$$

At the beginning, we have stated this formula as well:

$$p_\mathcal{D}(y \mid x_i) = \begin{cases} 1 & \text{if } y = y_i \\ 0 & \text{else} \end{cases}$$

Once again, we are not interested in cases where $p_\mathcal{D}(y \mid x_i) = 0$. Therefore, we will only focus on cases where $y = y_i$, with $p_\mathcal{D}(y \mid x_i) = 1$. This removes the sum over the y values in our equation and leaves:

$$\arg\min_\theta -\frac{1}{n} \sum_{i=1}^{n} \log p_\theta(y_i \mid x_i)$$

This is the end of our proof. The obtained equation is equal to the negative log-likelihood calculated under Task 1.1.

$$\arg\min_\theta \text{NLL}(\theta) = \arg\min_\theta -\sum_{i=1}^{n} \log(p_\theta(y_i \mid x_i))$$

Task 1.3

The Maximum-Likelihood estimator also minimizes the expected cross-entropy between the empirical distribution $p_\mathcal{D}(y \mid x)$ and the model distribution $p_\theta(y \mid x)$, i.e.,

$$\arg\min_\theta \text{NLL}(\theta) = \arg\min_\theta \mathbb{E}_{x \sim p_\mathcal{D}(x)} \left[\mathcal{H}_{\text{ce}}(p_\mathcal{D}(\cdot \mid x), p_\theta(\cdot \mid x))\right]$$

Proof

$$\mathbb{H}_{\text{ce}}(p_\mathcal{D}(\cdot \mid x), p_\theta(\cdot \mid x)) = \mathbb{E}_{y \sim p_\mathcal{D}(\cdot \mid x)} \left[-\log(p_\theta(y \mid x))\right]$$

$$\arg\min_\theta \text{NLL}(\theta) = \arg\min_\theta \mathbb{E}_{x \sim p_\mathcal{D}(x)} \left[\mathbb{E}_{y \sim p_\mathcal{D}(\cdot \mid x)} \left[-\log(p_\theta(y \mid x))\right]\right]$$

$$\arg\min_\theta \text{NLL}(\theta) = \arg\min_\theta \mathbb{E}_{x \sim p_\mathcal{D}(x)} \left[\sum_y p_\mathcal{D}(y \mid x) * \left[-\log(p_\theta(y \mid x))\right]\right]$$

$$\arg\min_\theta \text{NLL}(\theta) = \arg\min_\theta -\sum_x p_\mathcal{D}(x) * \sum_y p_\mathcal{D}(y \mid x) * \log(p_\theta(y \mid x))$$

$$\arg\min_{\theta} \mathrm{NLL}(\theta) = \arg\min_{\theta} -\sum_{x} \frac{1}{n} \sum_{i=1}^{n} \delta(x - x_i) \sum_{y} p_{\mathcal{D}}(y \mid x) * \log(p_\theta(y \mid x))$$

Similarly as in task 1.2, we will only observe the cases for which $x = x_i$.

$$\arg\min_{\theta} \mathrm{NLL}(\theta) = \arg\min_{\theta} -\frac{1}{n} \sum_{i=1}^{n} \sum_{y} p_{\mathcal{D}}(y \mid x_i) * \log(p_\theta(y \mid x_i))$$

Also, similarly to task 1.2, we will only take into consideration values $y = y_i$, for which $p_{\mathcal{D}}(y \mid x_i) = 1$

$$\arg\min_{\theta} \mathrm{NLL}(\theta) = \arg\min_{\theta} -\frac{1}{n} \sum_{i=1}^{n} \log(p_\theta(y \mid x_i))$$

Since we got to the standard formula for the negative log likelihood, we can state that the Maximum-Likelihood estimator also minimizes the expected cross-entropy between the empirical distribution and the model distribution.

Task 1.4

The following proof shows the connection between $D_{\mathrm{KL}}(p, q)$, $\mathbb{H}_{\mathrm{ce}}(p, q)$ and $\mathbb{H}(p)$, where $\mathbb{H}(p) = \mathbb{E}_{z \sim p(z)}[-\log(p(z))]$ is the entropy of p.

Proof
The entropy formula, which is the expected amount of information/surprise over p:

$$\mathbb{H}(p) = \mathbb{E}_{Z \sim p(z)}[-\log(p(z))]$$

We can express the Kullback-Leibler(KL)Divergence as:

$$\mathbb{D}_{\mathrm{KL}}(p, q) = \mathbb{E}_{z \sim p(z)} \left[\log \frac{p(z)}{q(z)} \right]$$

$$\mathbb{D}_{\mathrm{KL}}(p, q) = \mathbb{E}_{z \sim p(z)} \left[\log p(z) \right] - \mathbb{E}_{z \sim p(z)} \left[\log q(z) \right]$$

The cross-entropy between two distributions p(x), q(x) is given by:

$$\mathbb{H}_{\mathrm{CE}}(p, q) = \mathbb{E}_{z \sim p(z)} \left[-\log q(z) \right]$$

If we insert $\mathbb{H}_{\mathrm{CE}}(p, q)$ into the previous equation, we get:

$$\mathbb{D}_{\mathrm{KL}}(p, q) = \mathbb{E}_{z \sim p(z)} \left[\log p(z) \right] + \mathbb{H}_{\mathrm{CE}}(p, q)$$

The final calculated connection between $D_{\mathrm{KL}}(p, q)$, $\mathbb{H}_{\mathrm{ce}}(p, q)$ and $\mathbb{H}(p)$ is:

$$\mathbb{D}_{\mathrm{KL}}(p, q) = \mathbb{H}_{\mathrm{CE}}(p, q) - \mathbb{H}(p)$$

Task 2 - Decision Theory

Task 2.1

We are assuming that we have access to the true posterior $p^*(y \mid \mathbf{x})$, $\mathbf{x} \in \mathbb{R}^d$. For the following task, the defined loss function is the zero-one loss:

$$\mathcal{L}(y, \hat{y}) = \begin{cases} 1 & \text{if } y \neq \hat{y} \\ 0 & \text{else} \end{cases}$$

where y is the *true, observed* label, and $\hat{y}$ is the model's prediction.

Below, we will showcase the logic behind obtaining the decision function $f : \mathbb{R}^d \to \{0, 1\}$ that minimizes the expected loss over the data generating distribution. We start off with the formula for the expected loss, while only taking into consideration $y \neq f(x)$, since otherwise the loss is zero and there will not be anything to minimize.:

$$\mathbb{E}_{(\mathbf{x},y)\sim p^*(\mathbf{x},y)} \left[\mathcal{L}(y, f(\mathbf{x}))\right] = \sum_{x \in X} \sum_{y \in Y : y \neq f(x)} p(x, y) * \mathcal{L}(y, f(x))$$

$$\mathbb{E}_{(\mathbf{x},y)\sim p^*(\mathbf{x},y)} \left[\mathcal{L}(y, f(\mathbf{x}))\right] = \sum_{x \in X} \sum_{y \in Y : y \neq f(x)} p(x) * p(y|x) * \mathcal{L}(y, f(x))$$

The loss is going to be equal to one due to $y \neq f(x)$.

$$\mathbb{E}_{(\mathbf{x},y)\sim p^*(\mathbf{x},y)} \left[\mathcal{L}(y, f(\mathbf{x}))\right] = \sum_{x \in X} p(x) \sum_{y \in Y : y \neq f(x)} p(y|x)$$

$$\mathbb{E}_{(\mathbf{x},y)\sim p^*(\mathbf{x},y)} \left[\mathcal{L}(y, f(\mathbf{x}))\right] = \sum_{x \in X} p(x) \left[1 - p(f(x)|x)\right]$$

The goal is to choose f, that minimizes the sum. We can ignore the $p(x)$ values since they will not change. The only values that we have an effect on are the conditional probabilities. In order for the sum to be minimized, the term $[1 - p(f(x)|x)]$ has to be minimal, meaning the $p(f(x)|x)$ has to be as large as possible. Therefore, if $f(x)$ is the new y, then the decision function can be stated as follows:

$$f(x) = \arg\max_y p(y|x)$$

Based on the task description, we are dealing with a binary classification problem. The above equation can then be rewritten as:

$$f(x) = \begin{cases} 1 & p(y = 1|x) \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

This was derived from the fact that the decision function always chooses the y that has the highest probability. So if we want to have $f(x) = 1$, then $p(y = 1|x) \geq p(y = 0|x)$ has to hold. The decision for $p(y = 1|x) = p(y = 0|x)$ will not matter too much, since it does not effect the loss. We can rewrite the mentioned inequality as:

$$p(y = 1|x) \geq 1 - p(y = 1|x)$$
$$2p(y = 1|x) \geq 1$$
$$p(y = 1|x) \geq 0.5$$

So the final decision function is given as follows and is also knows as the Bayes optimum classifier.

$$f(x) = \begin{cases} 1 & p(y = 1|x) \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

Task 2.2

Question

Does there exist a different decision function f, that in expectation over $p^*(x, y)$, makes fewer misclassifications? Explain why/why not.

Answer

No there does not. The decision function, we have introduced, is knows as the Bayes classifier and is proven to be optimal, in the sense of achieving the lowest loss.

Let's say that the function $\theta(x)$ is the probability that the prediction of $f(x)$ is wrong. Then we could in general define it as follows:

$$\theta(x) = \begin{cases} p(y \neq 0|x) & f(x) = 0, \\ p(y \neq 1|x) & f(x) = 1 \end{cases}$$

We can rewrite it also as:

$$\theta(x) = \begin{cases} p(y = 1|x) & f(x) = 0, \\ 1 - p(y = 1|x) & f(x) = 1 \end{cases}$$

Therefore, if $p(y = 1|x) < 1 - p(y = 1|x)$, then we choose $f(x) = 0$. If $p(y = 1|x) > 1 - p(y = 1|x)$, we should choose $f(x) = 1$. If these probabilities are equal, then the choice does not matter. To summarize, the $p(y = 1|x) > 1 - p(y = 1|x)$ is the same as $p(y = 1|x) > 0.5$, which pretty much brings us to the decision formula we have stated in task 2.1.

The Bayes classifier is optimal since it always chooses the y value that has the highest conditional probability for it, and respectively, the lowest misclassification probability. There cannot be a different function that will give different predictions for specific x values but with a lower loss, since the Bayes classifier already has picked the y with the lowest loss.

Task 2.3

Question

If f had access to the marginal $p^*(x)$, could we construct a decision function that achieves a lower expected loss? Explain why/why not.

Answer

No, having access to the marginal distribution $p^*(x)$, would not allow us to construct a decision function that achieves a lower expected loss than the Bayes classifier. The Bayes classifier $f(x) = \arg\max_y p(y|x)$ is already optimal because it minimizes the probability of misclassifications, by choosing the y with the highest posterior probability for the given x. The $p^*(y|x)$ is the most relevant piece of information and the optimization problem does not depend on $p^*(x)$. It does not provide any additional information and its values, for each x, do not change, while the conditional probability values $p(y|x)$ are something that changes depending on the definition of our decision function.

Task 2.4

The new loss function, with $\mathcal{L}(y, \hat{y}) = L_{y,\hat{y}}$, is defined as:

$$L = \begin{bmatrix} 0 & 1 \\ 10 & 0 \end{bmatrix}$$

We will showcase the definition of a new decision function that minimizes the expected loss:

$$\mathbb{E}_{(x,y)\sim p^*(x,y)}\left[\mathcal{L}(y, g(x))\right]$$

By using the given loss function, we have as follows:

- for $g(x) = 0$

$$\mathcal{L}(y, 0) = \begin{cases} 0 & y = 0, \\ 10 & y = 1 \end{cases}$$

- for $g(x) = 1$

$$\mathcal{L}(y, 1) = \begin{cases} 1 & y = 0, \\ 0 & y = 1 \end{cases}$$

By using the formula below, we could calculate the total expected loss.

$$\mathbb{E}_{(\mathbf{x},y)\sim p^*(\mathbf{x},y)}\left[\mathcal{L}(y, g(\mathbf{x}))\right] = \sum_{x\in X} \sum_{y\in Y: y\neq g(x)} p(x) * p(y|x) * \mathcal{L}(y, g(x))$$

We will ignore the $p(x)$, since that will not help us in minimizing the loss. We will only expand the following sum:

$$\sum_{y\in Y: y\neq g(x)} p(y|x) * \mathcal{L}(y, g(x)) = p(y = 0|x) * 1 + p(y = 1|x) * 10$$

In summary, to minimize the loss we have to choose:

$$g(x) = \begin{cases} 0 & p(y = 1|x) < 1/11, \\ 1 & otherwise \end{cases}$$

Task 2.5

Question

For a particular x, we assume $p^*(y = 0|x) = 0.9$ and $p^*(y = 1|x) = 0.1$. What is the output of f(x) and g(x)?

Answer

Our original $f(x)$ will always choose the y that has the higher conditional probability. Therefore, since $p^*(y = 0|x) > p^*(y = 1|x)$, then the $f(x) = 0$.

By using the formula for $g(x)$, we can notice that the inequality $p(y = 1|x) < 1/11$ will not hold because of $0.1 > 1/11$. Because of this, we will choose $g(x) = 1$. The output of these two decisions is different, due to the different decision function definition.

Task 2.6

Question

Assume that y encodes if a patient (with feature vector x) has a disease ($y = 1$), or is healthy ($y = 0$). In words, briefly describe what the matrix L encodes in this case.

Answer

We are using the same L matrix as in the task 2.4:

$$L = \begin{bmatrix} 0 & 1 \\ 10 & 0 \end{bmatrix}$$

The $L_{0,1} = 1$ would represent the cost of a false positive. This means that our decision function predicted that a healthy person has the disease. The cost is low and logically could make sense, since it might lead to the person only undergoing unnecessary medical tests to rule out any disease. This does not have serious consequences for the person.

The $L_{1,0} = 10$ represents a higher cost compared to the above case. This is a false negative, that predicts that a patient with a disease is healthy. The cost is higher, because it could cause worse consequences to the person, who might fail to prevent his disease from developing and taking necessary medication, due to the incorrect prediction of our function.

The $L_{0,0} = 0$ represents a correct prediction, therefore there is no cost.

The $L_{1,1} = 0$ represents a correct prediction, therefore there is no cost.