Exam of
Foundational Principles of Machine Learning (FPML)
December 15, 2023 – 3h

DON'T RETURN THIS SHEET BEFORE YOU ARE ALLOWED TO DO SO!
(you can write your name on the blank papers in the meantime)

General advice:

- **Authorized documents: 6 pages of personal notes.**
- Do not hesitate to do the exercises in any order you like: start with the ones you feel are quick to deal with.
- When you are allowed to start, before you start the first exercise, go through the subject quickly. In each exercise, the most difficult question is not necessarily the last, please feel free to skip some questions. Don't hesitate to go and scrap off points where they are easy to take. (vous pouvez "aller grapiller les points")
- The grading points (scale) is indicative, if the exam is too long a correction factor will be applied. So don't panic in front of the length, what you do, do it right! Also, you may notice that points sum up to 21 (5+8+5+3) instead of 20, so, I will do *something*.
- **French:** Vous êtes autorisés à composer en Français. (Y compris en insérant des mots techniques comme overfitting ou regularization en anglais quand vous ne savez pas la traduction).
- **French:** si certains bouts de l'énoncé ne sont pas clairs, je peux les traduire ! N'hésitez pas à demander si vous n'êtes pas sûrs.
- Calculators not allowed (and useless). No electronic device allowed (cell phone, etc).
- At the end, we will collect your papers. You can leave after you have returned your paper.

DON'T RETURN THIS SHEET BEFORE YOU ARE ALLOWED TO DO SO!
(you can write your name on the copies in the meantime)

1 Lecture related questions, all independent (5 points)

1. (1 pt) Explain the purpose of the validation set and of the test set.
2. (1.5 pt) Explain the typical (expected) relation between overfitting and large weights. Explain the purpose of Ridge regression, the idea behind it, and how/why it works (Ridge is the one with the term $+\lambda||w||_2^2$).
3. (0.5 pt) Let's assume we have finite amound of data and a given model in mind, that we can esily train on the data. What experiment can we do to estimate whether more data would be helpful?
4. (1 pt) Define the K-fold cross-validation procedure (the one seen in class). What are its benefits ? (give at least two)
5. (1 pt) When you have overfitting, what are the things you can do? (assuming we keep the same family of models). Cite as many possible solutions as you know, and each time, quickly explain your choice (1-2 lines per "solution" to overfitting).

2 Hinge loss - to be written starting from this one (8 points)

We have a dataset of input data $X = \{\vec{x}_1, \dots, \vec{x}_N\}$ and corresponding binary labels $Y = \{(y_1 \dots, y_N\}$. The data is D dimensional, $\vec{x} \in \mathbb{R}^D$. The labels are encoded like this: $y_n \in \{-1, 1\}$. We want to consider the hinge loss:

ERRATUM: I forgot to write $\hat{y}_n$ and wrote $\hat{y}$ instead... sorry.

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^{N} \ell_{hinge}(\hat{y}_n, y_n) \tag{1}$$

$$\ell_{hinge}(\hat{y}_n, y_n) = max(0, 1 - \hat{y} y_n) \tag{2}$$

Where $\hat{y}_n \in \mathbb{R}$ is the output of the model (for the input $\vec{x}_n$).

We denote $H(z)$ the Heaviside function (step function) : $H(z) = \begin{cases} 1 \text{ if } z \geq 0 \\ 0 \text{ if } z < 0 \end{cases}$.

You can introduce other notations if you find them useful.

If you feel more comfortable denoting the ground truth labels as y_n^{GT} or t_n, please do so (it will avoid forgetting the hat on $\hat{y}$ and then confusing the model's output and the ground truth labels).

1. (0.25 pt) Write the prediction function, $y_{predicted} = \text{Readout}(\hat{y}) =?$. **ERRATUM: to answer this question more easily, it's better to first read questions 6, i.e. know what is the model.**
2. (0.5 pt) Draw the function $\ell_{hinge}(\hat{y}, y_n)$ as a function of $\hat{y}$ when $y_n = 1$. In particular, pay attention to the values this function takes at the points $\hat{y} = 0, 1, 2$.
3. (0.25 pt) Draw the function $\ell_{hinge}(\hat{y}, y_n)$ as a function of $\hat{y}$ when $y_n = -1$. In particular, pay attention to the values this function takes at the points $\hat{y} = -2, -1, 0$.
4. (0.5 pt) What is $\frac{\partial \ell_{hinge}}{\partial \hat{y}}(\hat{y}, y_n)$ when $y_n = 1$? What is $\frac{\partial \ell_{hinge}}{\partial \hat{y}}(\hat{y}, y_n)$ when $y_n = -1$?
5. (0.5 pt) Taking advantage of the fact that $\forall n, y_n^2 = 1$, and using the Heaviside function H, summarize these two results in a single formula, i.e. write down as explicitly as possible $\frac{\partial}{\partial \hat{y}} \ell_{hinge}(\hat{y}, y_n)$.
6. (0.5 pt) We will use a linear model, $\hat{y}_n = \vec{w} \cdot \vec{x}_n$. Compute $\vec{\nabla}_{\vec{w}} \hat{y}_n$, detailing the steps (expand the dot product and compute for instance $\frac{\partial}{\partial w_1} \hat{y}$).
7. (1 pt) Using this model, deduce what is $\vec{\nabla}_{\vec{w}} \mathcal{L}$. *This question depends on most of the previous ones, but if you cannot complete it, you can still do a lot in the next questions.*
8. (0.5 pt) What is the Gradient Descent update for a full epoch, then? Re-write it using as many matrix operations as possible, in the spirit of vectorization (for leveraging numpy's or GPU's capabilites).
9. *(Independent question)* We have the following points: $X = \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix} \right)$ with labels $Y = (1, 1, 1, -1)$ and 2 possible planes $\mathcal{P}(\vec{w})$ that separate them: $\vec{w}_a = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \vec{w}_b = \frac{1}{\sqrt{5}} \begin{pmatrix} 2 \\ 1 \end{pmatrix}$

(a) (1 pt) Draw the 4 points (use different markers for the 2 classes) and the 2 planes in a system of coordinates. Note that the planes have both $b = 0$ implicitly, i.e. are defined only by their orientation (they are orthogonal to the $\vec{w}$ that defines them) and they go through O (the origin of your coordinate system).

(b) (1 pt) Compute the Losses $\mathcal{L}(X, Y, \vec{w}_a)$, $\mathcal{L}(X, Y, \vec{w}_b)$ for both planes $\vec{w}_a, \vec{w}_b$. Which one is larger ?

(c) (0.5 pt) With or without the previous computation, which plane corresponds to the smaller loss ? Do you think other planes may have a Loss as good as that ? Why ? You can use geometrical arguments.

(d) (0.5 pt) Using the Perceptron Loss $\mathcal{L}_{perceptron} = \frac{1}{N}\sum_{n=1}^{N} max(0, -\vec{w}\cdot\vec{x}_n\ y_n)$, quickly write the result for the perceptron Loss, i.e. give the value of $\mathcal{L}_{perceptron}(X, Y, \vec{w}_a)$ and of $\mathcal{L}_{perceptron}(X, Y, \vec{w}_b)$ Why does the hinge loss seems better than the perceptron one ? *(No computation needed).*

10. (0.25 pt) If you remember the gradient of the classic perceptron update *(the one where the loss is written: $\mathcal{L}_{perceptron} = \frac{1}{N}\sum_{n=1}^{N} max(0, -\vec{w}\cdot\vec{x}_n\ y_n)$)*, what are the similarities and the difference(s) between the gradients of $\mathcal{L}$ and of $\mathcal{L}_{perceptron}$? *(Answering the previous question helps in this one, but it's not required to answer this one).*

11. (0.25 pt) Do you see some intuitive connection between our hinge loss model and SVMs ?

12. (0.5 pt) *(Independent question)* We now have the idea that the data comes from the following process:

$$y_n = \vec{w}\cdot\vec{x}_n + \varepsilon_n \tag{3}$$

where ε_n are noise terms, each ε_n is an i.i.d. Gaussian variable: $\rho(\varepsilon) = \mathcal{N}(0, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\varepsilon^2}{2\sigma^2}}$.

We also have a Gaussian prior on each component w_d of the vector $\vec{w}$, $\rho(w_d) = \mathcal{N}(0, \tau) = \frac{1}{\tau\sqrt{2\pi}} e^{-\frac{w_d^2}{2\tau^2}}$. Can you guess the result of a MAP estimate ? *Note: you cannot and are not asked to provide the solution $\vec{w}_{\text{MAP}}$ explicitly, you should just write the problem that needs to be solved numerically to get $\vec{w}_{\text{MAP}}$, and notice to what it corresponds to.*

3 Maximum A Posteriori (MAP) (5 points)

We have access to a data set $\widetilde{X} = \{x_1, \ldots, x_N\}$ of empirical binary ($x_n \in \{0, 1\}$) observations that are assumed to be independent and identically distributed. We may refer to the empirical mean as $\overline{x} = \frac{1}{N}\sum_{n=1}^{N} x_n$.

We model the observations as random variables X_n following a Bernoulli law, $P(x) = p^x(1-p)^{1-x}$ (where x can take the values $\{0, 1\}$). We want to compute the MAP estimate of the parameter p assuming an exponential prior for p (which by definition, is positive), $\rho(p) = \lambda e^{-\lambda p}$, where $\lambda > 0$.
Reminder: The event "I observe the data is $\widetilde{X}$" can be written $X = \widetilde{X}$.
Reminder: $\mathbb{E}_{\rho_\lambda}[p] = \int_0^\infty p\ \lambda e^{-\lambda p}\text{dp} = 1/\lambda$.
Reminder: $\log(a^b) = b\log(a)$ and $\log(u)' = \frac{u'}{u}$ (as e.g. $\log(x)' = \frac{1}{x}$).

1. (0.25 pt) What is the range of possible values for the empirical average $\overline{x} = \frac{1}{N}\sum_n x_n$?

2. We want to compute the MAP estimate $\hat{p}_{\text{MAP}}$ from the start, i.e. from the definition, recalling the fundamental steps (that are general to all cases) as well as the computations that apply to this precise case. This can be broken into steps of increasing difficulty:

   (a) (1.5 pt) As a first step, you should do the general reasoning, and when you arrive at concrete computations, you may simplify by injecting $\lambda = 0$ into the formula to recover the MLE result: $\hat{p}_{MLE} = \overline{x}$. If you are lost with MAP things, you can also just do the MLE reasoning from the start to compute $\hat{p}_{MLE}$, and still score some points.

   (b) (1 pt) Now, under the general case $\lambda \neq 0$ you should obtain $\hat{p}_{MAP}$. *(Reminder: The solutions (solving for x, assuming $a, b, c \in \mathbb{R}$) of the equation $ax^2 + bx + c = 0$ are $x_+ = \frac{-b+\sqrt{\Delta}}{2a}, x_- = \frac{-b-\sqrt{\Delta}}{2a}$, where $\Delta = b^2 - 4ac$. This can be summarized as $x_\pm = \frac{-b\pm\sqrt{\Delta}}{2a}$.)* Write down the solution $\hat{p}_{MAP}$ exaclty. It does not simplify much, I'm sorry about that.

   (c) (0.5 pt) Formally, you should have found two solutions. How should we deal with that ? (Hint: what are the values allowed for $\hat{p}$?).

   (d) (1 pt) From the general formula found in (b), simplify in the limits $N \sim \infty$ and $\lambda \sim 0$. In particular, you may assume that $\frac{\lambda}{N}\overline{x} \ll \left(1 + \frac{\lambda}{N}\right)^2$, and use the development: $\sqrt{1-\varepsilon} \sim 1 - \frac{1}{2}\varepsilon$ (under the assumption $\varepsilon \sim 0$). You should obtain a simple formula where $\hat{p}_{MAP}$ depends on $\overline{x}, \lambda, N$.

   (e) (0.25 pt) Without any computation, guess the zero-th order limit when $\lambda \to \infty$.

(f) (bonus points) From the general formula, simplify in the limit $\lambda \sim \infty$. This is somewhat similar to the previous case. You should obtain a simple formula where $\hat{p}_{MAP}$ depends on $\overline{x}, \lambda, N$.

3. (0.5) Comment on the meaning of the limits $N \sim \infty$, $\lambda \sim 0$ and $\lambda \sim \infty$.

4 Weighted PCA (3 pts)

We recall the PCA recipe.
We denote $\overline{\vec{x}}$ the empirical average of the data, feature by feature: $\overline{\vec{x}} = \frac{1}{N}\sum_n \vec{x}_n$
We denote C the empirical covariance matrix of the data: $C = \frac{1}{N-1}\sum_n \vec{x}_n \cdot \vec{x}_n^T$, which is thus a $D \times D$ matrix.
Diagonalizing C, we obtain the matrix of its eigenvectors U, that we can truncate (taking only D' columns in it) to obtain P.
The PCA projection formula is then $\Phi_{PCA}(\vec{x}) = (\vec{x} - \overline{\vec{x}})P$, where P is assumed to be a matrix of shape $D \times D'$, where $D' < D$ is the dimension after PCA, and D is the original dimension of each data point.

We want to define a balanced or weighted PCA where each sample is attributed an over-sampling coefficient α_n. For instance, if we have a minority class representing 10% of samples only, we may use $\alpha_n = 1/0.1$ for these samples, and $\alpha_n = 1/(0.9)$ for the other class (that has frequency of 90%). More generally, we may want to boost the importance of some samples in our PCA.

We want to find the exact recipe for weighted PCA that is equivalent to over-sampling by a factor α_n, but without the need to actually over-sample on the computer.

Let's denote $(.)^{(b)}$ the balanced version of something.

1. (1 pt) Write down the over-sampled version of the data average $\overline{\vec{x}}^{(b)}$. Try to write it in terms of the $\vec{x}_n$ and α_n.
2. (1 pt) How should we re-define P in the balanced case ? Define $P^{(b)}$. You may need to introduce $C^{(b)}$, the covariance matrix of the over-sampled data (in which case you need to define it, as you did for $\overline{\vec{x}}^{(b)}$.
3. (0.5 pt) Can this be re-written as a feature map on the x_n ? Why ? Does it make sense to you ?
4. (0.5 pt) Discuss in which sense this balanced or weighted PCA can be though of as an unsupervised method (as PCA) or as a supervised method.