HomeWork 5 : Inference and Machine learning

Most of this homework has to be done on a computer, with the language of your choice. Present your results with graphs, plots, and data.

1 Statistical inference and MSE

We shall consider the problem discussed in our lecture, where a system emits particle with a half-length decay λ, which we detect if $1 < \lambda < 20$. The probability to observe such a decay is thus

$$\begin{aligned} P_\lambda(x_i) &= \frac{e^{-x/\lambda}}{Z(\lambda)} \text{ if } 1 < x < 20 \\ P_\lambda(x_i) &= 0 \text{ otherwise} \end{aligned} \tag{1}$$

1. Compute $Z(\lambda)$ such that the probability is normalized. What the probability $P_\lambda(\{x\})$ to observe a set of n events $(\{x\})$? Write a program that output n such observation sampled from the probability distribution (1).
2. We now assume that we are given a set of n observations, without being told the true value of λ. We will try different estimators $\hat{\lambda}(\{x\})$ and for each of them, we shall define the squared error as $\text{SE} = (\hat{\lambda}(\{x\}) - \lambda)^2$.
   First we consider the maximum likelihood estimator
   $$\hat{\lambda}_{\text{ML}}(\{x\}) = \text{argmax}_\lambda P_\lambda(\{x\}) \tag{2}$$
   Create some data set with $n = 10, 100, 1000$ for different values of λ and see how the ML estimator performs. Note that finding the maximizer $\hat{\lambda}_{ML}$ can be done numerically.
3. If we average of many realizations of this process we can obtain the mean square error $MSE(\lambda, \hat{\lambda}, n)$, which is thus a function of n, λ and of the estimator $\hat{\lambda}$. Compute, for $n = 10000$, the curve $MSE(\lambda, \hat{\lambda}_{\text{ML}}, n = 10000)$.
   How does this curve compare with the Cramers-Rao bound $\text{MSE}(\hat{\lambda}) \geq \frac{1}{I(\lambda)}$, where $I(\lambda)$ is the total Fisher information.
   Is the ML estimator unbiased ?
4. We shall now adopt a Bayesian point of view. Following Jeffreys, we shall use the following prior $P(\lambda) \propto \sqrt{I(\lambda)}$. The posterior probability, for a single event, thus now reads
   $$\begin{aligned} P(\lambda|x_i) &\propto \frac{e^{-x/\lambda + \frac{1}{2}\log I(\lambda)}}{Z(\lambda)} \text{ if } 1 < x < 20 \\ P(\lambda|x_i) &= 0 \text{ otherwise} \end{aligned} \tag{3}$$
   Consider the MAP estimator with this prior
   $$\hat{\lambda}_{\text{MAP}}(\{x\}) = \text{argmax}_\lambda P(\lambda|(\{x\})) \tag{4}$$
   Is it an unbiased estimator ?
   Compute numerically, by averaging many instance, the curve $MSE(\lambda, \hat{\lambda}_{\text{MAP}}, n = 10000)$. How does it compare with the two other curves ?

2 The MNIST dataset

The MNIST database (Mixed National Institute of Standards and Technology database) is database of handwritten digits from 0 to 9 that is commonly used for training various image processing systems. It is widely used for training and testing in machine learning. It can downloaded automatically in many languages, but is also available here : `http://yann.lecun.com/exdb/mnist/` There are many simpler ways to download these data, that you can find on the internet.

The dataset is divided into a training set of 60000 digits, together with their correct assignment, that one shall use for training the algorithm. A second set, of 10000 digits, will be use to check the accuracy of our classification.

Our goal here is to train a machine learning algorithm. We will use a simple two layer neural net.

Consider for instance a classifier that should decide if an image is a 0 or not. We will consider the following model :

$$y = \theta \mathbf{z} = \theta f(F\mathbf{x}) \tag{5}$$

where $\mathbf{x}$ is a vector of dimension n that contains the values of the pixel for a given image, F is a matrix $F_{m\times n}$, $f(.)$ a function, θ a m-dimensional vector, and y should be the prediction label for the image. Ideally, we would like $y = 1$ is the image is a 0 and $y = -1$ over-wise. In practice, we will, of course, use $sign(y)$ since y is continuous.

In order to simplify our task, we will learn only the values of the vector θ, and will use, for F, a random Gaussian matrix. We will also use, for $f()$ the sigmoid function $1/1 + exp(-x)$.

— Assume that for all the images $i = 1 \to 60000$, we have the label $y_i = \pm 1$ arranged in a vector $\mathbf{Y}$. Show that in that case the best value of θ is given (after regularization) by

$$\hat{\theta} = (\mathbf{F^T F} + \Gamma \mathbf{I})^{-1} \mathbf{F^T Y} \tag{6}$$

— Why can the number M can be interpreted as a number of "hidden neurons" ? For moderate values of M, it is actually an easy task to solve this problem. Create 10 different such network ; each of them trained on recognizing $0, 1, 2, \ldots, 10$. For Γ use a small value, but make sure that the matrix is invertible.

— Once the training is done, we need to use these 10 network to make predictions on a new image.
For a given such new image, we will get 10 values of y using these 10 classifiers. To make our prediction, we shall assign the label corresponding to the largest value of y between these 10.
Compute, for moderate values of M (start by $M = 100$) the performance of our machine learning algorithm.

— For moderate values of M, you might have a memory problem (creating a $M \times 60000$ random matrix takes a lot of place !).
You could try to work, for instance, with a single precision matrix. You could also try to make the training with much less examples to make sure you are avoiding memory problems. There are other possibilities which I will let you think about...