From machine learning basics to Feed Forward Neural Networks

Typical machine learning problems can be decomposed into two different classes.

Classification The problem is to learn whether an individual from a given state space $\mathbb{R}^p$ belongs to some class. The focus is usually set on learning with a known number M of classes so that an individual is associated with a label in $\{1, \dots, M\}$. The statistical model is then given by $(X, Y) \in \mathbb{R}^p \times \{1, \dots, M\}$ and the objective is to define a function $f : \mathbb{R}^p \to \{1, \dots, M\}$, called classifier, such that $f(X)$ is the best prediction of Y in a given context.

Regression The observation associated with X is assumed to be given by

$$Y = f(X) + \varepsilon \,,$$

where ε is a centered noise independent of X. The statistical model is then given by $(X, Y) \in \mathbb{R}^p \times \mathbb{R}^m$ and the objective is to define the best estimator of f in a given context. An element of $\mathbb{R}^p$ contains all the features the label prediction or the regression estimate has to be based on.

Loss and risk functions

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. Assume that (X, Y) is a couple of random variables defined on $(\Omega, \mathcal{F}, \mathbb{P})$ and taking values in $\mathbb{R}^p \times \{1, \dots, M\}$ or $\mathbb{R}^p \times \mathbb{R}^m$ where $\mathbb{R}^p$ is a given state space. In the case of nonparametric models, it is not assumed that the joint law of (X, Y) belongs to any parametric or semiparametric family of models. The best prediction is defined as

$$h_* \in \operatorname*{argmin}_{h \in \mathcal{H}} \mathcal{R}(h) \quad \text{where} \quad \mathcal{R}(h) = \mathbb{E}[\ell(Y, h(X))] \,,$$

and ℓ is a loss function measuring the goodness of the prediction of Y by $h(X)$ and $\mathcal{H}$ is a chosen set of possible candidates. Some widespread choices of loss function are:

$$(\textbf{Classification}) \quad \ell(Y, h(X)) = |Y - h(X)|^2 \quad \text{and} \quad (\textbf{Regression}) \quad \ell(Y, h(X)) = 1_{Y \neq h(X)} \,.$$

In most cases, the risk $\mathcal{R}$ cannot be computed nor minimized, it is instead estimated by the empirical classification risk defined as

$$\mathcal{R}_n(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, h(X_i)) \,,$$

where $(X_i, Y_i)_{1 \leqslant i \leqslant n}$ are independent observations with the same distribution as (X, Y). The classification problem then boilds down to solving

$$\widehat{h}_n \in \operatorname*{argmin}_{h \in \mathcal{H}} \mathcal{R}_n(h) \,.$$

In this context, several practical and theoretical challenges arise from the minimization of the empirical classification risk.

Gentle start - classification with a mixture of two Gaussian distributions

In this first example, consider a parametric model, that is, the joint distribution of (X, Y) is assumed to belong to a family of distributions parametrized by a vector θ with real components. For $k \in \{-1, 1\}$, write $\pi_k = \mathbb{P}(Y = k)$. Assume that conditionally on the event $\{Y = k\}$, X has a Gaussian distribution with mean $\mu_k \in \mathbb{R}^p$ and covariance matrix $\Sigma \in \mathbb{R}^{p\times p}$. The probability density function of X given that $\{Y = k\}$ is

$$g_k : x \mapsto \sqrt{\det(2\pi\Sigma)}\exp\left\{-\frac{1}{2}(x-\mu_k)' \Sigma^{-1}(x-\mu_k)\right\} .$$

In this case, the parameter $\theta = (\pi_1, \mu_1, \mu_{-1}, \Sigma)$ belongs to the set $\Theta = [0, 1] \times \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^{d\times d}$. The parameter π_{-1} is not part of the components of θ since $\pi_{-1} = 1 - \pi_1$. The explicit computation of $\mathbb{P}(Y = 1|X)$ writes

$$\mathbb{P}(Y = 1|X) = \frac{\pi_1 g_1(X)}{\pi_1 g_1(X) + \pi_{-1} g_{-1}(X)} = \frac{1}{1 + \frac{\pi_{-1} g_{-1}(X)}{\pi_1 g_1(X)}} = \sigma\left(\log(\pi_1/\pi_{-1}) + \log(g_1(X)/g_{-1}(X))\right) ,$$

where $\sigma : x \mapsto (1 + \mathrm{e}^{-x})^{-1}$ is the sigmoid function. Then,

$$\mathbb{P}(Y = 1|X) = \sigma(x'\omega + b) ,$$

where

$$\omega = \Sigma^{-1}(\mu_1 - \mu_{-1}) , b = \log(\pi_1/\pi_{-1}) + \frac{1}{2}(\mu_1 + \mu_{-1})' \Sigma^{-1}(\mu_{-1} - \mu_1) .$$

When Σ and μ_1 and μ_{-1} are unknown, this classifier cannot be computed explicitely. We will approximate it using the observations. Assume that $(X_i, Y_i)_{1\leqslant i\leqslant n}$ are independent observations with the same distribution as (X, Y). The loglikelihood of these observations is given by

$$\log \mathbf{p}_\theta(X_{1:n}, Y_{1:n}) = \sum_{i=1}^n \log \mathbf{p}_\theta(X_i, Y_i) ,$$

which yields

$$\log \mathbf{p}_\theta(X_{1:n}, Y_{1:n}) = -\frac{nd}{2}\log(2\pi) - \frac{n}{2}\log\det\Sigma + \left(\sum_{i=1}^n 1_{Y_i=1}\right)\log\pi_1 + \left(\sum_{i=1}^n 1_{Y_i=-1}\right)\log(1-\pi_1)$$
$$- \frac{1}{2}\sum_{i=1}^n 1_{Y_i=1}(X_i-\mu_1)'\Sigma^{-1}(X_i-\mu_1) - \frac{1}{2}\sum_{i=1}^n 1_{Y_i=-1}(X_i-\mu_{-1})'\Sigma^{-1}(X_i-\mu_{-1}) .$$

Maximizing the log likelihood function to estimate θ is equivalent to minimizing the empirical cross-entropy risk function:

$$\theta \mapsto -\frac{1}{n}\sum_{i=1}^n \sum_{k\in\{-1,1\}} 1_{Y_i=k}\left(\log\pi_k - \frac{\log\det\Sigma}{2} - \frac{1}{2}(X_i-\mu_k)'\Sigma^{-1}(X_i-\mu_k)\right) .$$

The gradient of $\log \mathbf{p}_\theta(X_{1:n}, Y_{1:n})$ with respect to θ is therefore given by

$$\frac{\partial \log \mathbf{p}_\theta(X_{1:n}, Y_{1:n})}{\partial \pi_1} = \left(\sum_{i=1}^n 1_{Y_i=1}\right)\frac{1}{\pi_1} - \left(\sum_{i=1}^n 1_{Y_i=-1}\right)\frac{1}{1-\pi_1} ,$$

$$\frac{\partial \log \mathbf{p}_\theta(X_{1:n}, Y_{1:n})}{\partial \mu_1} = \sum_{i=1}^n 1_{Y_i=1}\left(2\Sigma^{-1}X_i - 2\Sigma^{-1}\mu_1\right) ,$$

$$\frac{\partial \log \mathbf{p}_\theta(X_{1:n}, Y_{1:n})}{\partial \mu_{-1}} = \sum_{i=1}^n 1_{Y_i=-1}\left(2\Sigma^{-1}X_i - 2\Sigma^{-1}\mu_{-1}\right) ,$$

$$\frac{\partial \log \mathbf{p}_\theta\left(X_{1:n}, Y_{1:n}\right)}{\partial \Sigma^{-1}} = \frac{n}{2}\Sigma - \frac{1}{2}\sum_{i=1}^{n} 1_{Y_i=1}\left(X_i-\mu_1\right)\left(X_i-\mu_1\right)' - \frac{1}{2}\sum_{i=1}^{n} 1_{Y_i=-1}\left(X_i-\mu_{-1}\right)\left(X_i-\mu_{-1}\right)' .$$

The maximum likelihood estimator (i.e. the cross-entropy based estimator) is defined as the only parameter such that all these equations are set to 0. For $k \in \{-1, 1\}$, it is given by

$$\widehat{\pi}_k^n = \frac{1}{n}\sum_{i=1}^{n} 1_{Y_i=k} ,$$

$$\widehat{\mu}_k^n = \frac{1}{\sum_{i=1}^{n} 1_{Y_i=k}} \sum_{i=1}^{n} 1_{Y_i=k}\, X_i ,$$

$$\widehat{\Sigma}^n = \frac{1}{n}\sum_{i=1}^{n} \left(X_i - \widehat{\mu}_{Y_i}^n\right)\left(X_i - \widehat{\mu}_{Y_i}^n\right)' .$$

Relaxing the Gaussian assumption - Logistic regression

In some situations, it may be too restrictive to assume that the joint distribution of (X, Y) belongs to a parametric family. One of the most widespread model is the logistic regression which is defined by

$$\mathbb{P}\left(Y=1|X\right) = \sigma(X'\omega + b) ,$$

where $b \in \mathbb{R}$, $\omega \in \mathbb{R}^d$ and for all $x \in \mathbb{R}^d$. The parameter θ is thus $\theta = (b, \omega) \in \mathbb{R} \times \mathbb{R}^p$. When b and ω are unknown, this quantity cannot be computed explicitely and is approximated using the observations. Assume that $(X_i, Y_i)_{1 \leqslant i \leqslant n}$ are independent observations with the same distribution as (X, Y). The conditional likelihood of the observations $Y_{1:n}$ given $X_{1:n}$ is:

$$\mathbf{p}_\theta\left(Y_{1:n}|X_{1:n}\right) = \prod_{i=1}^{n} \mathbf{p}_\theta\left(Y_i|X_i\right) = \prod_{i=1}^{n} \left(\sigma(X_i)\right)^{(1+Y_i)/2} \left(1-\sigma(X_i)\right)^{(1-Y_i)/2} ,$$

which yields

$$\mathbf{p}_\theta\left(Y_{1:n}|X_{1:n}\right) = \prod_{i=1}^{n} \left(\frac{\mathrm{e}^{b+\langle\omega;X_i\rangle}}{1+\mathrm{e}^{b+\langle\omega;X_i\rangle}}\right)^{(1+Y_i)/2} \left(\frac{1}{1+\mathrm{e}^{b+\langle\omega;X_i\rangle}}\right)^{(1-Y_i)/2} .$$

The associated conditional loglikelihood is therefore

$$\log \mathbf{p}_\theta\left(Y_{1:n}|X_{1:n}\right) = \sum_{i=1}^{n} \left\{ \frac{1+Y_i}{2} \log\left(\frac{\mathrm{e}^{b+\langle\omega;X_i\rangle}}{1+\mathrm{e}^{b+\langle\omega;X_i\rangle}}\right) + \frac{1-Y_i}{2} \log\left(\frac{1}{1+\mathrm{e}^{b+\langle\omega;X_i\rangle}}\right) \right\} ,$$

i.e.

$$\log \mathbf{p}_\theta\left(Y_{1:n}|X_{1:n}\right) = \sum_{i=1}^{n} \left\{ \frac{1+Y_i}{2}\left(b+\langle\omega;X_i\rangle\right) - \log\left(1+\mathrm{e}^{b+\langle\omega;X_i\rangle}\right) \right\} .$$

Note again that maximizing this loglikelihood is equivalent to minimizing the empirical cross-entropy. It cannot be done explictly yet numerous numerical optimization methods are available to maximize $(\omega, b) \mapsto \log \mathbf{p}_\theta\left(Y_{1:n}|X_{1:n}\right)$ (see next session on a up-to-date overview of gradient based algorithms).

The multilayer perceptron - Feed Forward Neural Networks

The first mathematical model for a neuron was the Threshold Logic Unit (McCulloch and Pitts, 1943), with Boolean inputs and outputs. The response associated with an input $x \in \{0,1\}^d$ is defined as $f : x \mapsto 1_{\omega \sum_{j=1}^{d} x_j + b \geqslant 0}$. This construction allows to build any boolean function from elementary units. This elementary model can be extended to the **Perceptron** with real valued

inputs (Rosenblatt, 1957) by writing $f : x \mapsto 1_{\sum_{j=1}^{d} \omega_j x_j + b \geqslant 0}$. In this case, the nonlinear activation function is $\sigma : x \mapsto 1_{x \geqslant 0}$ and the ouput defined as:

$$f : x \mapsto \sigma(\omega' x + b)\,.$$

Linear discriminant analysis and logistic regression are other instances with the sigmoid activation function. The perceptron weakens the modeling assumptions of LDA or logistic regression and composed in parallel q of these perceptron units to produce the output. Then, $x = z_0 \in \mathbb{R}^p$, $b_1 \in \mathbb{R}^q$, $\omega_1 \in \mathbb{R}^{p \times q}$ and

$$z_1 = \sigma(\omega_1' x + b)\,,$$

with σ the elementwise activation function. The **multi-layer perceptron**, also known as the fully connected feedforward network, connects these units in series. For a given number L of layers,

$$z_1 = \sigma(\omega_1' x + b_1)\,, \quad z_2 = \sigma(\omega_2' z_1 + b_2)\,, \quad \ldots\,, \quad z_L = \sigma(\omega_L' z_{L-1} + b_L)\,.$$

As there is no modelling assumptions anymore, virtually any activation function may be used. The relu activation $x \mapsto \max(0, x)$ and its extensions are the default recommendation in modern implementations (Jarrettet al., 2009), (Nair and Hinton, 2010), (Glorot et al., 2011), etc. One of the major motivation arises from the gradient based parameter optimization which is numerically more stable with relu, (see next session on a up-to-date overview of gradient based algorithms). Assume that the network contains L layers, then the output layer is of the form:

$$z_L = \sigma(\omega_L' z_{L-1} + b_L)\,.$$

The choice of this last activation function greatly relies on the task the network is assumed to perform.

Biclass classification The output z_L is the estimate of the probability that the class is 1 given the input x. The common choice in this case is the sigmoid function, one of the main reason is due to the gradient descent algorithm used to optimize the parameters, cf. next sessions.

Multiclass classification The output z_L is the estimate of the probability that the class is k for all $1 \leqslant k \leqslant M$, given the input x. The common choice in this case is the softmax function: for all $1 \leqslant i \leqslant r$,

$$\sigma(z)_i = \frac{\mathrm{e}^{z_i}}{\sum_{j=1}^{r} \mathrm{e}^{z_j}}\,.$$

Universal approximation properties

The universal approximation theorem sets a theoretical guarentee that feedforward networks with hidden layers provide a universal approximation framework. The first result of (Horniket al., 1989) and (Cybenko, 1989) states that a feedforward network with a linear output layer and at least one hidden layer can approximate any Borel measurable function from one finite-dimensional space to another, provided that the network is given enough hidden units. For a given activation function $\sigma : \mathbb{R} \to \mathbb{R}$, the set of neural networks with one hidden layer and a linear output associated with σ is given by:

$$\mathcal{N}_\sigma = \Big\{ f : \mathbb{R}^p \to \mathbb{R}\,;\, \exists\, q \geqslant 1\,, (c_1, \ldots, c_q, b_1, \ldots, b_q) \in \mathbb{R}^{2q}\,, (\omega_1, \ldots, \omega_q) \in \mathbb{R}^{pq}\,, \\ f : x \mapsto \sum_{j=1}^{q} c_j \sigma(\langle \omega_j ; x \rangle + b_j) \Big\}\,.$$

(Horniket al., 1989), (Cybenko, 1989) Let $\sigma : \mathbb{R} \to \mathbb{R}$ be a continuous activation function such that $\lim_{x\to\infty} \sigma(x) = 1$ and $\lim_{x\to-\infty} \sigma(x) = 0$. Then, $\mathcal{N}_\sigma$ is dense in $\mathcal{C}([0,1]^p, \mathbb{R})$ for the topology of the supremum norm.

While the original theorems were first stated in terms of units with activation functions that saturate for both very negative and very positive arguments, universal approximation theorems have also been proved for a wider class of activation functions, which includes the now commonly used rectified linear unit (Leshno et al., 1993).

(Leshno et al., 1993) Assume that $\sigma : \mathbb{R} \to \mathbb{R}$ is continuous and is not a polynomial activation function. Then, $\mathcal{N}_\sigma$ is dense in $\mathcal{C}(\mathbb{R}^p, \mathbb{R})$ for the topology of the supremum norm on compact subsets.

According to the universal approximation theorem, there exists a network large enough to achieve any degree of accuracy we desire, but the theorem does not say how large this network will be. In (Barron, 1993), the authors provides some bounds on the size of a single-layer network needed to approximate a broad class of functions. Unfortunately, in the worst case, an exponential number of hidden units (possibly with one hidden unit corresponding to each input configuration that needs to be distinguished) may be required.

(Barron, 1993) Let $r > 0$ and μ be a probability distribution on the closed ball centered at 0 and with radius r denoted by B_r. Let $\sigma : \mathbb{R} \to \mathbb{R}$ be a bounded and measurable activation function such that $\lim_{x\to\infty} \sigma(x) = 1$ and $\lim_{x\to-\infty} \sigma(x) = 0$. Assume that $f : \mathbb{R}^p \to \mathbb{R}$ is such that its Fourier transform $\widetilde{f}$ satisties $\int_{\mathbb{R}^p} |\omega||\widetilde{f}(w)|\mathrm{d}\omega < \infty$. Then, for all $q \geqslant 1$, there exist a Feedforward neural network model with one layer of q sigmoidal units:

$$f_q : x \mapsto \sum_{j=1}^{q} c_j \sigma(\langle \omega_j; x \rangle + b_j) + c_0$$

and $C > 0$ such that

$$\int_{\mathsf{B}_r} \left(f(x) - f_q(x)\right)^2 \mu(\mathrm{d}x) \leqslant \frac{C}{q}\,.$$