Assignment 1: Neural Networks with Few Labels

Emanuele Sansone*

November 2, 2016

1 From binary to multiclass classification

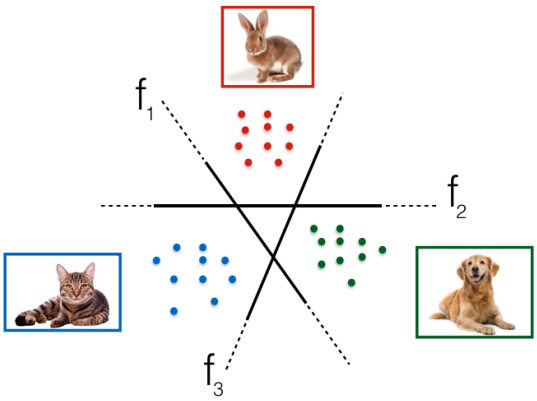

Figure 1: Example of multiclass classification problem. Each point corresponds to an image. Each color represents a class (the blue color is associated with class "cat" and the green color is associated with class "dog" and the red color is associated with class "rabbit"). The goal is to learn the boundaries (highlighted in black) between the three classes given the available images.

Recall that in binary classification, a classifier can be learnt by solving the following optimization problem

$$\mathcal{R}^{\lambda}_{emp}(f) = \frac{\pi}{|D_b^+|}\sum_{\boldsymbol{x}_i \in D_b^+} \ell(f(\boldsymbol{x}_i), 1) + \frac{(1-\pi)}{|D_b^-|}\sum_{\boldsymbol{x}_i \in D_b^-} \ell(f(\boldsymbol{x}_i), -1) + \lambda\Omega(f) \tag{1}$$

where D_b^+ and D_b^- are the portions of training dataset $D = \{(\boldsymbol{x}_i, y_i) : \boldsymbol{x}_i \in \mathbb{R}^d, y_i \in \{1, -1\}\}_{i=1}^m$ corresponding to the positive and negative classes, respectively; π is the positive class prior; $\ell(\cdot, \cdot)$ is the loss function; $\Omega(\cdot)$ is the regularizer; λ is a real-positive hyperparameter weighting the relative importance of $\Omega(\cdot)$ with respect to the first two terms in (1); f is the function that we want to learn.

When dealing with multiple classes, the goal is to learn multiple binary classifiers. This is easily seen from Figure 1, where each binary classifier discriminates samples belonging to one class against samples belonging to all other classes. This strategy is called **one-versus-all classification** [Aly05]. In particular, if there are K classes, then the multiclass classification problem can be decomposed into K binary classification subproblems (see Figure 2 for an example of decomposition). Therefore, each subproblem j has its own dataset $D_b^j = D_b^{j+} \cup D_b^{j-}$ (where class j is the positive class and all other classes are regarded as the negative class), its own positive class prior π_j and its own function f_j to be learnt. Based on these considerations, problem (1) can be reformulated in the following way:

$$\mathcal{R}^{\lambda}_{emp}(\boldsymbol{f}) = \sum_{j=1}^{K}\left\{\frac{\pi_j}{|D_b^{j+}|}\sum_{\boldsymbol{x}_i \in D_b^{j+}} \ell(f_j(\boldsymbol{x}_i), 1) + \frac{(1-\pi_j)}{|D_b^{j-}|}\sum_{\boldsymbol{x}_i \in D_b^{j-}} \ell(f_j(\boldsymbol{x}_i), -1)\right\} + \lambda\Omega(\boldsymbol{f}) \tag{2}$$

*https://emsansone.github.io/

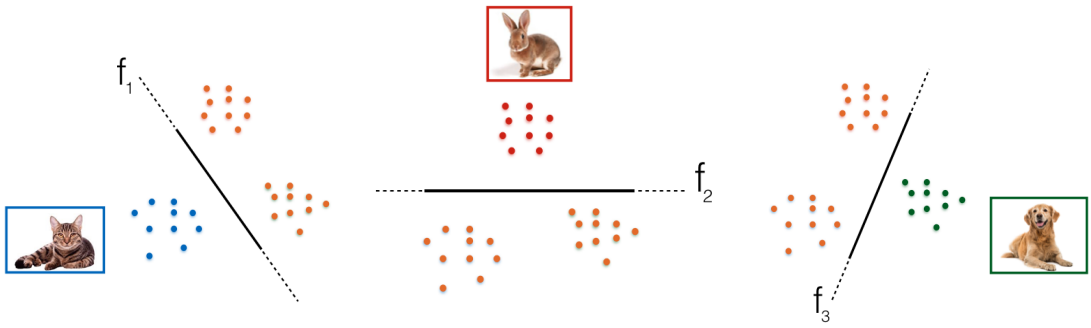

Figure 2: Decomposition of the multiclass classification problem.

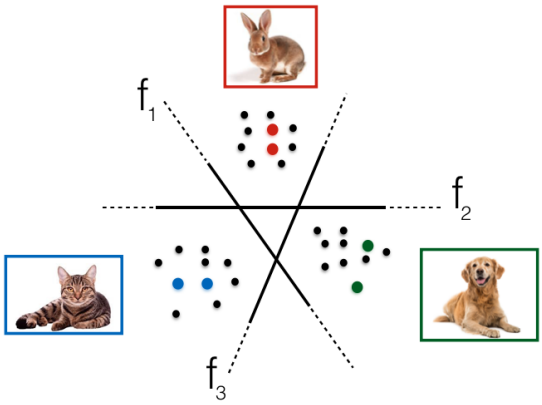

Figure 3: Multiclass classification problem in the presence of few labeled data (coloured points) and unlabeled data (black points)

Note that instead of training K binary classifiers independently using K different optimization problems, in (2) we minimize only a single objective function. This is particularly useful for cases where the models have vectorial outputs. This concept will be clarified in the following sections.

2 Learning with few labels

We are interested in learning multiclass classifiers in scenarios where only few labeled data are available. In general, minimizing (2) in such context does not produce satisfactory and reliable results. This is due to the fact that using only a limited amount of training data is in general not enough to estimate correctly the true multiclass risk functional (obtained by replacing summations with integrals in (2)).

Nevertheless, if unlabeled data are available, then we can still have the possibility to achieve good results, namely learn good classifiers. Figure 3 shows an example of multiclass classification with few labeled data and plenty of unlabeled samples. The problem can be decomposed into many positive unlabeled (PU) learning subproblems (see Figure 4 for an example). The main difference with respect to the case of traditional multiclass classification is that the goal of each subproblem is to learn a classifier based on labeled samples belonging to only one class (regarded as positive) and unlabeled data belonging to both the positive and negative (all other) classes. In the next section, we review the formulation of PU learning.

3 Positive unlabeled (PU) learning

In PU learning, the training set is split into two parts, namely a set of samples $D_p = \{\mathbf{x}_i \in X\}_{i=1}^{p}$ drawn from the positive class and a set of "not labeled" samples $D_n = \{\mathbf{x}_i \in X\}_{i=1}^{n}$ drawn from both the positive and the negative classes. The goal is the same of the binary classification problem, but this time the supervised information is available only for one class. The learning problem can

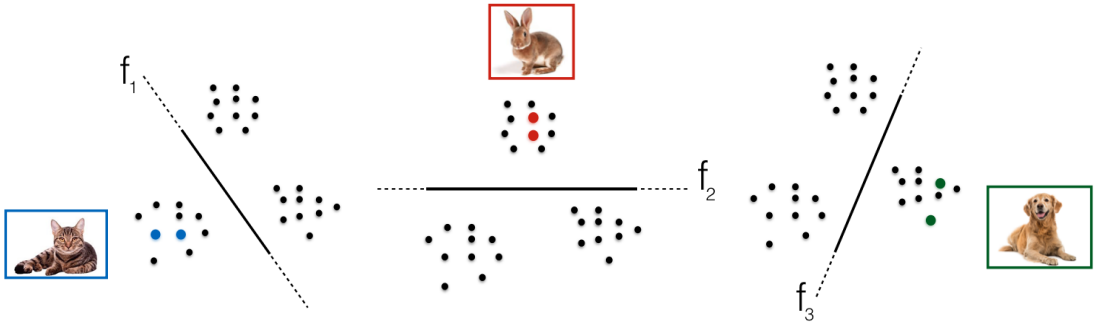

Figure 4: Decomposition of the multiclass classification problem with few labeled data into PU learning subproblems.

be still formulated as a risk minimization. In fact, since $\mathcal{P}(\mathbf{x}) = \pi\mathcal{P}(\mathbf{x}|y=1)+(1-\pi)\mathcal{P}(\mathbf{x}|y=-1)$, the risk functional can be rewritten in the following way:

$$\begin{aligned}\mathcal{R}(f) =& \pi \int \ell(f(\mathbf{x}),1)\mathcal{P}(\mathbf{x}|y=1)d\mathbf{x} \\ &+ (1-\pi)\int \ell(f(\mathbf{x}),-1)\frac{\mathcal{P}(\mathbf{x})-\pi\mathcal{P}(\mathbf{x}|y=1)}{1-\pi}d\mathbf{x} \\ =& \pi\int \tilde{\ell}(f(\mathbf{x}),1)\mathcal{P}(\mathbf{x}|y=1)d\mathbf{x} + \int \ell(f(\mathbf{x}),-1)\mathcal{P}(\mathbf{x})d\mathbf{x} \end{aligned} \tag{3}$$

where $\tilde{\ell}(f(\mathbf{x}),1) = \ell(f(\mathbf{x}),1) - \ell(f(\mathbf{x}),-1)$ is called the **composite loss** [DPNS15].

The risk functional in (3) can not be minimized since the distributions are unknown. In practice, one considers the empirical risk functional in place of (3), where expectation integrals are replaced with the empirical mean estimates computed over the available training data, namely

$$\mathcal{R}(f) \approx \mathcal{R}_{emp}(f) = \frac{\pi}{|D_p|}\sum_{\mathbf{x}_i\in D_p}\tilde{\ell}(f(\mathbf{x}_i),1) + \frac{1}{|D_n|}\sum_{\mathbf{x}_i\in D_n}\ell(f(\mathbf{x}_i),-1) \tag{4}$$

The minimization of $\mathcal{R}_{emp}$ is in general an ill-posed problem. A regularization term is usually added to $\mathcal{R}_{emp}$ in order to restrict the solution space and to penalize complex solutions, leading to the definition of the following regularized empirical risk functional:

$$\mathcal{R}^{\lambda}_{emp}(f) = \frac{\pi}{|D_p|}\sum_{\mathbf{x}_i\in D_p}\tilde{\ell}(f(\mathbf{x}_i),1) + \frac{1}{|D_n|}\sum_{\mathbf{x}_i\in D_n}\ell(f(\mathbf{x}_i),-1) + \lambda\Omega(f) \tag{5}$$

As it is shown in [DPNS15, SDNZ16], the best loss function, that makes (5) a convex functional, is the double Hinge loss, namely:[1]

$$\ell(f(\boldsymbol{x}),y) = \max\left\{-yf(\boldsymbol{x}), \max\left\{0, \frac{1}{2} - \frac{yf(\boldsymbol{x})}{2}\right\}\right\} \tag{6}$$

Let us now rewrite $\mathcal{R}^{\lambda}_{emp}$ using the double Hinge loss function and denote the new regularized empirical risk as $\mathcal{R}^{\lambda,\mathcal{H}}_{emp}$, which allows to formulate the following optimization problem:

$$\begin{aligned} f^* =& \arg\min_{f\in\mathcal{F}} \mathcal{R}^{\lambda,\mathcal{H}}_{emp}(\boldsymbol{f}) \\ & \arg\min_{f\in\mathcal{F}}\left\{-\frac{\pi}{|D_p|}\sum_{\boldsymbol{x}_i\in D_p} f(\boldsymbol{x}_i) + \frac{(1-\pi)}{|D_n|}\sum_{\boldsymbol{x}_i\in D_n}\max\left\{f(\boldsymbol{x}_i), \max\left\{0,\frac{1}{2}+\frac{f(\boldsymbol{x}_i)}{2}\right\}\right\}\right\} + \lambda\Omega(\boldsymbol{f}) \end{aligned} \tag{7}$$

This is our final formulation of the PU learning problem.

[1] Recall that in the binary classification problem, the best convex loss function was the Hinge loss. In the PU learning scenario, the Hinge loss produces a non-convex risk functional (due to the presence of the composite loss).

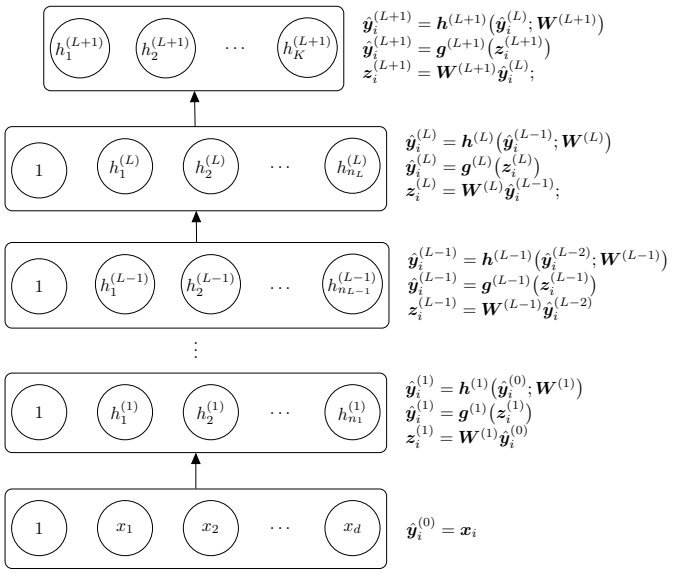

Figure 5: Feedforward neural network.

4 Multiclass PU learning

The extension of (7) to the multiclass scenario can be performed by adopting a similar strategy used in Section 1. In particular, each subproblem j has its own dataset $D^j = D_p^j \cup D_n^j$ (where class j is the positive class), its own positive class prior π_j and its own function f_j. Therefore, the multiclass classification problem in the presence of few labeled data, called **multiclass PU learning** problem, is formulated in the following way:

$$\mathcal{R}_{emp}^{\lambda,\mathcal{H}}(\boldsymbol{f}) = \sum_{j=1}^{K} \left\{ -\frac{\pi_j}{|D_p^j|} \sum_{\boldsymbol{x}_i \in D_p^j} f_j(\boldsymbol{x}_i) + \frac{(1-\pi_j)}{|D_n^j|} \sum_{\boldsymbol{x}_i \in D_n^j} \max\left\{ f_j(\boldsymbol{x}_i), \max\left\{ 0, \frac{1}{2} + \frac{f_j(\boldsymbol{x}_i)}{2} \right\} \right\} \right\} + \lambda\Omega(\boldsymbol{f}) \tag{8}$$

5 Problem assignment

Derive the backpropagation algorithm (in a similar way to what has been done during the lectures), considering the following requirements:

- The objective function is defined according to (8).
- The parametric model consists of a feedforward neural network with K linear neurons in output (i.e. the j-th output neuron computes function f_j, where $j = 1, \ldots, K$), and with L hidden layers characterized by rectifier linear neurons. See Figure 5.
- The regularizer $\Omega(\cdot)$ corresponds to the L_1 norm computed over the network parameters.

The steps required to complete the assignment are:

1. Define $\boldsymbol{\theta}$ (vector containing network parameters).
2. Define $\boldsymbol{\xi}(\cdot)$.
3. Define $J(\cdot)$.
4. Derive the backpropagation algorithm.

Submit the report in PDF format using LaTeX.

References

[Aly05] Mohamed Aly. Survey on Multiclass Classification Methods. 2005.

[DPNS15] Marthinus Du Plessis, Gang Niu, and Masashi Sugiyama. Convex Formulation for Learning from Positive and Unlabeled Data. In *ICML*, pages 1386–1394, 2015.

[SDNZ16] Emanuele Sansone, Francesco GB De Natale, and Zhi-Hua Zhou. Efficient Training for Positive Unlabeled Learning. *arXiv:1608.06807*, 2016.