F-Measure as the error function to train Neural Networks

Joan Pastor-Pellicer[1], Francisco Zamora-Martínez[2], Salvador España-Boquera[1], and María José Castro-Bleda[1]

[1] DSIC, Universitat Politècnica de València, Valencia, Spain
[2] ESET, Universidad CEU Cadenal Herrera, Valencia, Spain

Abstract. Imbalance datasets impose serious problems in machine learning. For many tasks characterized by imbalanced data, the F-Measure seems more appropiate than the Mean Square Error or other error measures. This paper studies the use of F-Measure as the training criterion for Neural Networks by integrating it in the Backpropagation algorithm. This novel training criterion has been validated empirically on a real task for which F-Measure is typically applied to evaluate the quality. The task consists in cleaning and enhancing ancient document images which is performed, in this work, by means of neural filters.

Keywords: Neural Networks, Error-backpropagation algorithm, F-Measure, Imbalance datasets

1 Introduction

It is not uncommon in many real tasks that the number of patterns of one class is significantly lower than other classes. Examples of tasks with very imbalanced data are information retrieval (a lot of information and very few useful data) or medical diagnosis (less ill than healthy patients). Imbalance datasets impose serious problems in machine learning and, particularly, in Artificial Neural Networks (ANN) training. Some authors have addressed this problem by resampling the data in order to balance the occurrences, others have modified the training algorithm [2, 15].

We have followed the second approach, designing a new training algorithm which uses the F-Measure [13] as an objective error function for the Backpropagation (BP) algorithm. The F-Measure (FM) may be more suitable than the Mean Square Error (MSE) or other error measures, for problems with imbalanced data, because it is a quality measure computed as a combination between Precision and Recall. Though there are different approaches for the optimization of the F-Measure using supervised techniques like SVMs [9], logistic regression [8] and other approaches [1], no such algorithm exists for ANNs to the best of our knowledge.

In order to illustrate the interest of this proposal in a real task, we have studied a problem for which the F-Measure has been typically used to assess the quality: image cleaning and enhancement of ancient document images. This

task consists in estimating the probability of ink in each pixel of the cleaned image given the noisy counterpart, and it can be considered to be an imbalanced problem since only a few percentage of pixels in an image corresponds to ink. This task is required not only to improve the readability of these documents by humans but is also the first stage in most preprocessing pipelines applied in text recognition systems. This stage is quite critical because any mistake could be propagated to the following ones. The relevance of this stage depends on the quality of the documents and it is particularly important in historical documents which suffer many types of degradation.

The rest of the paper is structured as follows. First, Section 2defines the F-Measure for continuous values and explains how this measure can be used as objective error function in the Backpropagation algorithm. In order to illustrate the performance of the new training criteria, a task of image cleaning and enhancement of ancient printed and handwritten documents is proposed (see Section 3). The proposed method is successfully applied to different competition datasets and experimental results are presented in Section 4. The conclusions are finally drawn at the end.

2 Error-backpropagation with F-Measure

The Backpropagation (BP) algorithm updates the ANN weights following the derivative of a given error function. The MSE function is widely used. For a given ANN output $o_1, o_2, \ldots, o_n$ and a its corresponding target output $t_1, t_2, \ldots, t_n$, the MSE for one pattern is computed as $MSE = 1/2 \cdot \sum_{i=1}^{n}(o_i - t_i)^2$, and its derivative is $\partial MSE/\partial o_i = (t_i - o_i)$.

This equation for training ANNs is well known, and has been successfully applied in several pattern recognition tasks (classification, regression, forecasting, ...). Different weight updating modes exists [4]:

- the *batch* training, which computes and sums the derivatives of all training patterns and updates weights once every epoch;
- the *on-line* training, which computes the derivative of one training pattern and updates weights once for each pattern every epoch; and
- finally, the *mini-batch* or *bunch* mode [3], which computes and sums the derivatives of a few training patterns, updating weights once for the mini-batch size, but several times for one epoch.

Mini-batch and on-line modes have some advantages compared with batch mode: convergence is faster and the result is equal or even more accurate.

As it was previously stated, for tasks with imbalanced data, the F-Measure (FM) function is more appropriate than MSE or other error measures. F-Measure is a quality measure computed as a combination between Precision (PR) and Recall (RC). It is possible to compute a version of the F-Measure interpreting the output of the model as a binary value (for 2-class problems: 1 for relevant and 0 for non-relevant), being $o^{(i)}$ the output of the model for pattern i and $t^{(i)}$ the real-class value (0 or 1) for pattern i. The computation of FM is a harmonic

mean of PR and RC, and leads to the final formulation of FM in terms of *true positives* (TPs), *false positives* (FPs) and *false negatives* (FNs).[3] TPs, FPs, and FNs are computed over a dataset of m patterns: $TP = \sum_{i=1}^{m} o^{(i)} \cdot t^{(i)}$, $FP = \sum_{i=1}^{m} o^{(i)} \cdot (1 - t^{(i)})$, and $FN = \sum_{i=1}^{m} (1 - o^{(i)}) \cdot t^{(i)}$. FM is formalized as usual, although the formula can be simplified by substituting TPs, FPs and FNs with previous definitions:

$$FM_\beta = \frac{(1+\beta^2) \cdot PR \cdot RC}{\beta^2 \cdot PR + RC} = \frac{(1+\beta^2) \cdot TP}{(1+\beta^2) \cdot TP + \beta^2 \cdot FN + FP} = \tag{1}$$

$$= \frac{(1+\beta^2) \cdot \sum_{i=1}^{m} o^{(i)} \cdot t^{(i)}}{\sum_{i=1}^{m} \left(o^{(i)} + \beta^2 \cdot t^{(i)} \right)} \tag{2}$$

In order to use the F-Measure as the objective error function in BP algorithm it is required to derive it by $o^{(i)}$:

$$\frac{\partial FM_\beta}{\partial o^{(i)}} = \frac{(1+\beta^2) t^{(i)}}{\sum_{j=1}^{m} \left(o^{(j)} + \beta^2 \cdot t^{(j)} \right)} - \frac{(1+\beta^2) \cdot \sum_{j=1}^{m} o^{(j)} \cdot t^{(j)}}{\left[\sum_{j=1}^{m} \left(o^{(j)} + \beta^2 \cdot t^{(j)} \right) \right]^2} \tag{3}$$

Since BP is defined for minimization, the sign of the F-Measure function has to be inverted. Note that F-Measure derivative of pattern i depends on the others $m - 1$ patterns, so it is not separable as the MSE. Therefore, the exact computation of this derivative forces to use batch training mode. However, batch training is slow and inaccurate when the number of patterns m is large (in the reported experiments, millions of patterns). Because of these issues, we decided to use a mini-batch training mode, which leads to an approximation highly correlated with the true F-Measure computed on the entire dataset. Also, for large training partitions, it is better to train each epoch with a shorter random replacement sampled from training data. In this way, the error of one epoch will be the mean of each mini-batch FM. Let b be the size of the mini-batch, and R the replacement sample size, weights are updated $\lceil R/b \rceil$ times every epoch.

The derivative of F-Measure has some issues that need to be discussed: the use of mini-batch mode combined with random replacement makes it possible to sample a bunch of patterns where every target is *false* (class 0). In this case, the FM and its derivative are both zero, meaning that these mini-batch presentations

[3] TPs are those positive samples correctly classified (no error); FPs are those negative samples incorrectly classified as positive samples (error); and FNs are positive samples incorrectly classified as negative samples (error).

does not update the weights. This problem becomes more likely the lower the mini-batch size and the more imbalanced the data is. Since each sample selection is independent of others, the probability of occurrence of this situation can be easily computed from the mini-batch size b and the proportion of 0's in the entire training dataset (of size m) as $(F/m)^b$, where $F = \sum_{j=1}^{m}(1 - t^{(j)})$. This problem reduces convergence speed because mini-batches suffering from this problem does not update weights even if the output of the model is not correct.

3 Cleaning and enhancement as a probability pixel estimation problem

Image cleaning and enhancement, specially for ancient documents, are common and crucial steps for any document recognition system. Traditionally, the output of image cleaning is a binarized image where black pixels mean the presence of ink in this region. Nevertheless, since many preprocessing techniques can also deal with gray level images, it is possible to consider the gray level of cleaned image pixels as the probability of ink. Thus, cleaned images are not considered as arbitrary gray level images but, rather, as a soft estimation of a black and white image which tries to represent, in a limited resolution, the set of ideal strokes. Gray values are a way to represent the probability of picking a black sub-pixel in this pixel, so intermediate gray values are expected to be found in the borders of strokes. This idea has a correspondence with the desirable anti-aliasing property of geometrical transformations applied in most common preprocessing stages such as the correction of the skew of the page, the slope of the words in the lines and the slant of the strokes.

This problem can be considered as the joint estimation of the probability of finding ink in pixel areas, as the classification of pixels into two classes or as the retrieval of ink areas in the whole document.

It is not easy to classify image cleaning and binarization techniques, many are based on geometrical heuristics, but we propose the use of supervised machine learning techniques. Although other machine learning techniques exists (e.g. based on Markov Random Fields [14]) we have used neural network filters [7] which estimate the probability of ink for each pixel given a window of the original image centered at the pixel to be cleaned (see Figure 1). This has two main limitations: each pixel is estimated independently, and only local information is taken into account. The first limitation is alleviated by the high correlation of window inputs of neighboring pixels.

There exists many assessment measures to evaluate the quality of image cleaning and binarization which can be classified into three categories [10]: by means of human supervision, indirectly by evaluating the overall performance of a recognition system and, finally, by comparing the cleaned image with a reference or *ground truth*. Several measures have been proposed in the literature for the last option which is the only considered in this work. Note that, since this task is imbalanced (the percentage of ink pixels ranges between 5 and 15 percent), the F-Measure seems quite suitable in this case. As a matter of fact,

many prestigious image binarization contests [6] employ the F-Measure to rank contenders.

MSE is a common error metric in BP training algorithms which has been applied in probability estimation tasks and which can also be used to measure the quality of image cleaning, but the use of F-Measure seems more appropriate. Indeed, a cleaned image with lower MSE may appear subjectively of lower quality than another one which may assign little mistakes on white pixels which are the majority. That is why we have opted for the use of this measure as the training criterion in BP. Since the output of the neural network is represented as a real-value, it is straightforward to compute a "soft" F-Measure and error derivatives interpreting the output of the model as a binary probability where the value for a pattern i is set as $o^{(i)} = P(\text{relevant}|\text{sample})$ and $1 - o^{(i)} = P(\text{non-relevant}|\text{sample})$. The soft F-Measure is computed as stated in Section 2.

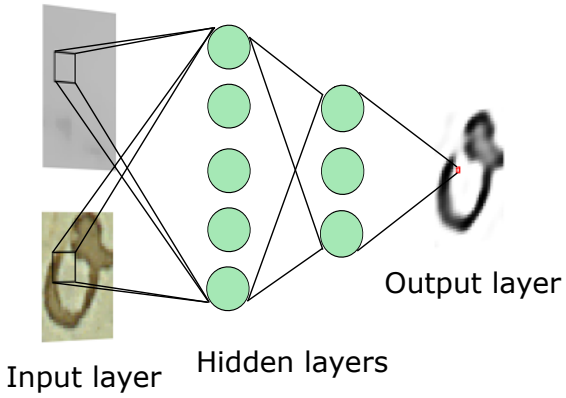


Fig. 1. Scheme of a neural network filter: a feedforward neural network estimates the probability of ink of a pixel on the cleaned image from a window centered at the same pixel in the original image. Another optional window receives an estimation of background computed by means of a median filter.

4 Experimental Setup and Results

The ANNs for enhancement and cleaning have been applied to the four Document Image Binarization COntest (DIBCO) datasets (DIBCO 2009 [5], DIBCO 2010 [12], DIBCO 2011 [6], and DIBCO 2012 [11]), partitioned as follows:

- Training set: includes DIBCO 2009 and DIBCO 2010 datasets. A total of 24 images (19.8 Mpx, 6.6% classified as ink).
- Validation set: includes DIBCO 2011 dataset. A total of 12 images (10.0 Mpx patterns, 9.0% classified as ink).

– Test: includes DIBCO 2012 dataset. A total of 14 images. (19.2 Mpx, 6.7% classified as ink).

In order to compare the proposed technique, different configurations have been tried with which differ in the error criteria:

– Logistic output unit ANN trained using the MSE error criteria.
– Logistic output unit ANN trained using the FM error criteria.

using training and validation data to 1) find a common topology which works fine with both error criteria, and 2) adjust parameters afterwards. Finally, the trained networks have been used to compute the performance on the test set. The quality of ANN filters has been measured by comparing the cleaned images with the ground-truth.

Each type of error criteria has been tested on a network which share the same input, hidden and output layers with the other network. The input layer is composed by 90 input neurons: 81 pixels corresponding to a window of size 9×9 centered at the pixel to be cleaned and 9 additional context pixels associated to a 3×3 window with an estimation of background using a median filter. Relating the hidden layers, the best configuration was two hidden layers of sizes 64 and 16, respectively. 9 different random initialized networks have been trained in order to reduce the effect of local minima. Table 1 shows the average of the FM and MSE measures, along with the standard deviation on validation and test sets for a training with a mini-batch of 32 samples. Also, an example of a test set image cleaned with both ANNs is depicted in Figure 2. In general, both training techniques perform quite well when measured either on MSE or on FM, since a well cleaned image gives good results on both metrics. Comparing these results with [11], which is measured on the same test, the results are not competitive compared with the best models although they are better than method 1 which is also based on neural networks (they obtain a FM 0.82, and we obtain a FM 0.836 for a soft measure which is a lower bound on the binarized measure. We can also observe, from Table 1, that ANNs trained with the FM error function obtains better F-Measure than ANNs trained with the MSE error function. Conversely, the second model outperforms the first one in MSE in both validation and test. As expected, each training criteria prioritizes a different goal.

Table 1. Average and Standard Deviation of the MSE and FM.

	Validation Data		Test Data	
	$\mu \pm \sigma$ MSE	$\mu \pm \sigma$ FM	$\mu \pm \sigma$ MSE	$\mu \pm \sigma$ FM
MSE training	0.0254 ± 0.0010	0.708 ± 0.013	0.0165 ± 0.0004	0.754 ± 0.007
FM training	0.0376 ± 0.0036	0.774 ± 0.012	0.0181 ± 0.0006	0.836 ± 0.009

In order to study the influence of the size of the mini-batch, different trainings have been carried out varying this parameter and the reported F-Measure is

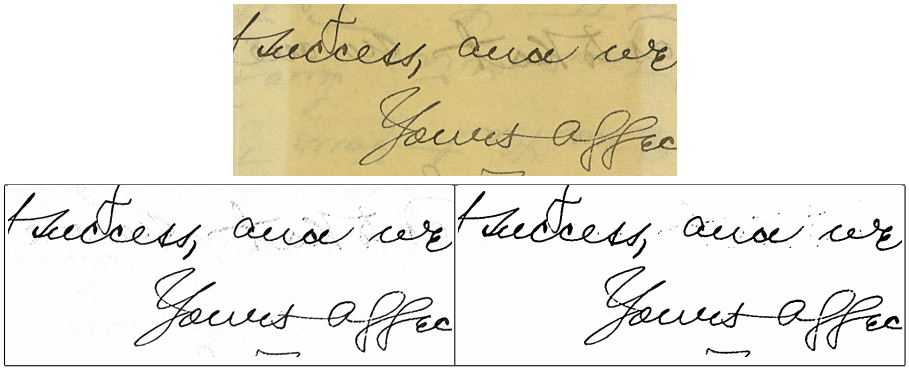
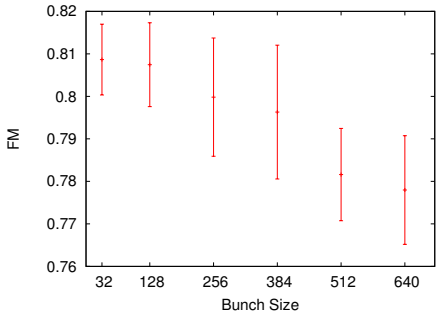
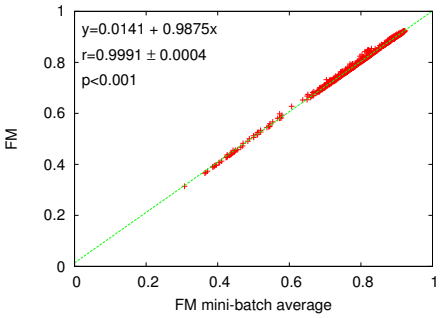


Fig. 2. (Top) Example of a noisy test image. (Bottom left) The same image cleaned with the ANN trained with the MSE error criteria. (Bottom right) The same image cleaned with the F-Measure error criteria.

illustrated in Figure 3(a). In this case, two different factors may influence the results in opposite ways: on the one side, the bigger the mini-batch size, the more accurate the approximation to the true F-Measure is. On the other side, a lower mini-batch value corresponds to a training scheme closer to the online version of BP which may have faster convergence. As can be observed, as the mini-batch size is increased, the F-Measure performs worse, which means that even smaller mini-batches may be highly correlated with the global F-Measure. In order to study this correlation, a statistic experiment has been carried out (see Figure 3(b)) obtaining a Pearson product-moment correlation coefficient $r = 0.9991 \pm 0.0004$ with a confidence interval 99.9% ($p < 0.001$).



(a) Influence of mini-batch size.



(b) Correlation between the average FM of 100 000 mini-batches of size 32 taken randomly and the FM value computed on the concatenation of all validation images.

5 Conclusions

In this work, a novel objective error function for the Backpropagation algorithm is proposed based on the F-Measure and it has been explained how it can be adapted to mini-batch training mode of BP. In order to empirically validate this training mode a real task using an imbalanced dataset of several millions of patterns has been carried out. The task has consisted in the estimation of a cleaned image from a noisy one by means of neural network filters. Experimental results show that, although ANNs trained with MSE or with FM performs quite well, each training mode prioritizes its corresponding assessment measure. This error criteria can be used in tasks where F-Measure makes sense, as is the case of information retrieval or document classification. As a future work, we plan to extend this work to other symmetrical measures such as the Matthews correlation coefficient.

References

1. An exact algorithm for f-measure maximization.
2. L. Al-Haddad, C.W. Morris, and L. Boddy. Training radial basis function neural networks: effects of training set size and imbalanced training sets. *J. of microbiological methods*, 43(1):33–44, 2000.
3. J. Bilmes et al. Using PHiPAC to speed error back-propagation learning. In *Proc. of ICASSP*, volume 5, pages 4153–4156, 1997.
4. Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley, 2. edition, 2001.
5. B. Gatos, K. Ntirogiannis, and I. Pratikakis. ICDAR 2009 document image binarization contest (DIBCO 2009). In *Proc. of ICDAR*, pages 1375–1382, 2009.
6. B. Gatos, K. Ntirogiannis, and I. Pratikakis. DIBCO 2009: document image binarization contest. *Int. J. on document analysis and recognition*, 14(1):35–44, 2011.
7. J.L. Hidalgo et al. Enhancement and cleaning of handwritten data by using neural networks. In *Patt. Rec. & Image Anal., vol. 3522*, pages 376–383. Springer, 2005.
8. M. Jansche. Maximum expected f-measure training of logistic regression models. In *Proc. of HLT&EMNLP*, pages 692–699, 2005.
9. D.R. Musicant et al. Optimizing f-measure with support vector machines. In *Proc. of Int. Florida AI Research Society Conference*, pages 356–360, 2003.
10. K. Ntirogiannis, B. Gatos, and I. Pratikakis. A Performance Evaluation Methodology for Historical Document Image Binarization. 2012.
11. I. Pratikakis, B. Gatos, and K. Ntirogiannis. ICFHR 2012 Competition on Handwritten Document Image Binarization (H-DIBCO 2012).
12. I. Pratikakis, B. Gatos, and K. Ntirogiannis. H-DIBCO 2010-handwritten document image binarization competition. In *Proc. of ICFHR*, pages 727–732, 2010.
13. C.J. van Rijsbergen. A theoretical basis for the use of co-occurrence data in information retrieval. *J. of Documentation*, 33(2):106–119, 1977.
14. C. Wolf. Document Ink Bleed-Through Removal with Two Hidden Markov Random Fields and a Single Observation Field. *IEEE PAMI*, 32(3):431–447, 2010.
15. Z.H. Zhou and X.Y. Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Trans. on Knowledge and Data Engineering*, 18(1):63–77, 2006.