

Technische Universität Berlin

Faculty of Electrical Engineering and Computer Science

Dept. of Computer Engineering and Microelectronics

Remote Sensing Image Analysis Group



ANALYSIS OF DEEP LEARNING LOSS FUNCTIONS FOR MULTI-LABEL REMOTE SENSING IMAGE CLASSIFICATION

Master of Science in ICT Innovation

March, 2020

Hichame Yessou

Matriculation Number: 406319

Supervisor: Prof. Dr. Begüm Demir

Advisor: Gencer Sümbül

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe angefertigt habe. Sämtliche benutzten Informationsquellen sowie das Gedankengut Dritter wurden im Text als solche kenntlich gemacht und im Literaturverzeichnis angeführt. Die Arbeit wurde bisher nicht veröffentlicht und keiner Prüfungsbehörde vorgelegt.

Hereby I declare that I wrote this thesis myself with the help of no more than the mentioned literature and auxiliary means.

Berlin, Date

.....

Name Surname

Acknowledgements

I wish to express my deep appreciation to my supervisor Professor Dr Begüm Demir and advisor Gencer Sümbül, who have guided me in this year-long journey through findings and challenges. They inspired and conveyed in me the mindset of research and profound understanding for which I'm deeply grateful. They have vision and devotion to push the boundaries of the research in a meaningful field such as Remote Sensing, sparking and growing my interest in it. It was a great pleasure to work and study with them. Without their direction, this work would not have been possible. This work marks the end of EIT Digital Master School program which I am appreciative for all the experiences and wonderful people I met, such as Federico Schiepatti from Politecnico di Milano.

I would like to express my special regards to my family, for which no amount of words would describe my gratefulness and appreciation to them. My mother Mina, my father Ahmed and my brothers Rayan and Sami, with their love, prayers and sacrifices; have been a central part in developing the person I am today. Without their support, nothing would have been possible.

A significant thanks belong to my best friends, Federico and Riccardo which have given me guidance, support and lifted me during ups and downs. An important credit goes to my best friends Lucia and Irene which have been extremely caring, helpful in solving obstacles and exchanging opinions. A big thank is directed to Federico and Giada, with which I've shared the journey at Politecnico di Milano and Technische Universität of Berlin.

A deep thank you needs to be addressed to my cousin Zakaria, with his tenacity and conscious mind has inspired me to push throughout challenges. Lastly, but certainly not least; to my all friends at NeonWood, where we have shared so many moments, becoming a second family away from home.

Abstract

This thesis analyzes and compares different deep learning loss functions in the framework of the multi-label Remote Sensing (RS) image scene classification problems. We consider seven loss functions: 1) Cross-Entropy Loss; 2) Weighted Cross-Entropy loss; 3) Focal Loss; 4) Hamming Loss; 5) Huber Loss; 6) SparseMax Loss; and 7) Ranking Loss. Our analysis aims to reveal their performance-wise differences and with greater significance, which loss functions are most suitable in specific contexts. All the considered loss functions are analyzed for the first time in RS and theoretically compared in terms of their: 1) capability to address class imbalanced data (for which the number of samples associated to each class significantly varies); 2) capability to consider outliers; 3) convexity and differentiability; and 4) required time to reach a high performance (i.e., efficiency of learning). After the theoretical comparison, experimental analysis is carried out on the publicly available Sentinel-2 benchmark archive, BigEarthNet, to compare different loss functions by considering the constraints of the learning problem, the training methodologies and the expectations from deep learning models. Based on our analyses, some guidelines are derived for a proper selection of a loss function in the context of multi-label RS image classification.

Zusammenfassung

In dieser Arbeit wurden verschiedene Deep Learning Loss-Funktionen zu Multi-Label-Problemen in der Fernerkundung verglichen. Dies wurde unter Verwendung des frei verfügbaren Fernerkundungsarchivs “BigEarthNet” durchgeführt, das verschiedene reale Herausforderungen einführt und dabei hilft, die Eigenschaften jedes Verlusts hervorzuheben. Die Analyse ergab leistungsbezogene Unterschiede und mit größerer Bedeutung, welche Verlustfunktionen in bestimmten Kontexten am besten geeignet sind. Abgesehen von herkömmlichen Ansätzen gibt es in der Literatur eine Reihe neuartiger Verlustfunktionen, die im Bereich der Fernerkundung noch nie verglichen wurden. Unter diesen ist es erwähnenswert: der Fokusverlust, der Hamming-Verlust, der SparseMax-Verlust, der Huber-Verlust und der Ranking-Verlust. Der absolute analytische Vergleich wurde aufgrund des Gleichgewichts zwischen Präzision und Rückruf unter Verwendung des F1-Scores als Metrik durchgeführt. Der qualitative Vergleich berücksichtigt zwei Faktoren, indem er die Merkmale der Verluste entfaltet und die Variationen über mehrere Stichproben des positiven (oder negativen) Beitrags mit einem Rahmen zur Erklärung von DNN beobachtet. Diese Arbeit, beginnend mit den einzelnen Ansprüchen der Verlustfunktion, führt zu Verknüpfungen zwischen ihren theoretischen Merkmalen und pragmatischen Anwendungen in verschiedenen Umgebungen.

Contents

List of Acronyms	vii
List of Figures	viii
List of Tables	ix
1 Introduction	1
2 Related Work	4
2.1 Convolutional Neural Networks	4
2.2 Deep Learning Training	7
2.3 Multi-label Remote Sensing Image Classification	9
3 Deep Learning Loss Functions for Multi-Label Classification	11
3.1 Cross-Entropy Loss	11
3.2 Focal Loss	13
3.3 Weighted Cross-Entropy Loss	15
3.4 Hamming Loss	15
3.5 Huber Loss	16
3.6 Ranking Loss	17
3.7 SparseMax Loss	18
4 Comparative Analysis of Deep Learning Loss Functions	20
4.1 Capability of Loss Functions to Handle Class Imbalance	20
4.2 Capability of Loss Functions to Handle Outliers	21
4.3 Convexity and Differentiability of Loss Functions	22
4.4 Efficiency of the Considered Learning Mechanism	23
5 Data Set Description and Design of Experiments	25
5.1 Description of the Data Set	25
5.2 Considered Deep Learning Technologies	25
5.3 Design of Experiments	27

6	Experimental Results	32
6.1	Comparison of the Classification Performance of Loss Functions	32
6.2	Evaluation of Loss Functions in the Context of Class Imbalance Awareness	39
6.3	Evaluation of Loss Functions in the Context of Outlier Awareness	44
6.4	Evaluation of Loss Functions in the Context of Convexity and Differentiability	46
6.5	Evaluation of Loss Functions in the Context of Efficiency of the Learning Mechanism	49
7	Conclusion and Discussion	53
	Bibliography	55
	Appendix	59

List of Acronyms

CEL	Cross Entropy Loss
FL	Focal Loss
W-CEL	Weighted Cross-Entropy
HL	Huber Loss
HAL	Hamming Loss
SML	SparseMax Loss
RL	Ranking Loss
DNNs	Deep Neural Networks
NN	Neural Network
MLC	Multi-Label Classification
SGD	Stochastic Gradient Descent
API	Application Programming Interface
BR	Binary Relevance
ReLU	Rectified Linear Unit
LRP	Layerwise Relevance Propagation

List of Figures

2.1	Convolutional operation over an input. The output is restricted only to positions where the kernel is entirely included in the input, also called “valid” convolution. . . .	5
2.2	Max-Pooling operation over an input	6
5.1	LRP procedure showing the redistribution of the relevance scores from each neuron to the lower layer.	31
6.1	Class-based accuracies in F1-score on the test set of the BigEarthNet archive	40
6.2	Overall classification accuracies in F-1 Score of the validation set of the BigEarthNet archive.	49
6.3	A BigEarthNet patch used for the comparison of LRP heatmaps	50

List of Tables

4.1	Comparison of MLC losses under considered criteria	20
4.2	Cost-sensitivity comparison between Cross-Entropy and Focal Loss	21
5.1	An example of Sentinel-2 image patches and their multi-labels in the BigEarthNet archive	26
5.2	Number of samples in train, validation and test set for each class	27
6.1	Overall classification accuracies on the test set of the BigEarthNet archive	33
6.2	Class-based sample-averaged Precision on the test set of BigEarthNet	34
6.3	Class-based sample-averaged Recall on the test set of BigEarthNet	35
6.4	Class-based classification accuracies in F1-Score sample-averaged on the test set of BigEarthNet	36
6.5	An example of BigEarthNet patches, their multi-labels and LRP heatmaps for the considered loss functions.	37
6.6	An example of BigEarthNet patches, their multi-labels and LRP heatmaps for the considered loss functions.	38
6.7	LRP heatmaps comparing the detection of the minority class "Industrial or commercial units" and "Inland waters"	42
6.8	LRP heatmaps comparing the detection of the minority class "Industrial or commercial units"	43
6.9	An example of RS images, their multi-labels and LRP heatmaps for Cross-Entropy Loss and Huber Loss Functions.	44
6.10	Cross-Entropy Loss and Focal Loss LRP heatmaps for outliers	45
6.11	LRP heatmaps showing different degrees of accuracy on different classes	48
6.12	Change of LRP heatmaps during training of Cross-Entropy, SparseMax and Huber Loss Functions for the <i>Marine Waters</i> class	51
6.13	Change of LRP heatmaps during training of Cross-Entropy, SparseMax and Huber Loss Functions for the <i>Beaches, dunes, sands</i> class	52

1 Introduction

The wealth of data available and its higher quality are among the foundations for enabling better performances of Deep Learning approaches. Deep Learning methods have attracted great recognition in the Remote Sensing realm thanks to their achievements. There are several tasks in which Deep Learning methods are used and in which we see a growing application of novel techniques. Examples of that are Land Use and Land Cover (LULC) classification, Scene Classification, and Object Detection. [29] Scene Classification is a critical field in Remote Sensing with extensive operative applications, which can be defined as the categorization of scene images into a discrete set of LULC classes coherently to its semantic content. Past approaches were focusing on single-label applications which are summarizing the totality of the sample to one single class. This is an oversimplification of the underlying problem which cannot be omitted and that requires a different approach. Remote Sensing scenes and generally speaking, most of the images within the Computer Vision world, contains semantically different objects. A more resembling approach to real-world situations is the Multi-Label Classification (MLC), which attributes one or several classes to the instance. MLC is a favoured approach since it characterizes precisely the semantic content of the image with one or several labels. Scene Classification in Remote Sensing has a particular need for multiple labels to describe an image for its nature of the problem. Relatively to their spatial resolution, images usually describe large portions of Earth Observations, including several morphologically different land portions. This, however, introduces several challenges such as skewed distributions of classes, exponential combinations of labels and potential underlying correlations among them. These MLC characteristics can also be extended to other fields such as text classification, recommender systems and generally speaking, to any classification task. Several approaches have been presented in the RS literature. Traditional methods use a combination of Cross-Entropy loss and Sigmoid activations.[34] Despite the popularity of existing Cross-Entropy based approaches, they neglect several factors which are crucial for achieving meaningful results and more adherent to the real-world applications of the models. Among them, we find several failings such as not having a cost-sensitive behaviour. The Cross-Entropy Loss does not acknowledge the class distribution within the data. [8] The usage of Loss Functions or a technique that takes into consideration long-tailed class distributions is a need in many real-world applications which require considering minority classes. A frequently studied problem is how to design noise-robust learning techniques. Supervised DNN relies on large-scale archives to achieve greater performance. [23] Often datasets can be affected by errors in the labelling process and models with loss functions agnostic to noisy labels can affect their performances. In [48] is studied the robustness of a modified Cross-Entropy Loss with awareness to noisy data, achieving superior performances.

1 Introduction

Li, Xiaoxu, et al. [28] analyses the limitation of the Cross-Entropy Loss in terms of its inability to push the decision boundaries. Losses which aim to maximize the margin show better performances, more stability and easier optimizations.

Inappropriate Loss Function could lead to suboptimal performances or inaccurate correlations between predictions and the semantic content of instances. Loss Functions are a critical component of a Deep Learning approach since defines the evaluation of model improvements. This a highly-complex problem for standard classification tasks and even more challenging for multi-label classification due to the various circumstances in which predictions can be evaluated. Models using different Loss Function can have different performances and imply different conditions, which are characteristics that have to be evaluated relatively to the operational use of the model itself. Therefore is essential choosing a function that faithfully represents the multi-label objective and the context in which the model will operate. This work aims to foster the importance of carefully selecting Loss Functions considering the dataset, the objective of the model and operational context of the models. We analyze and compare several Deep Learning loss functions for MLC in the context of Remote Sensing. State-of-the-art loss functions from other fields are presented, that to the best of our knowledge has never been considered in Remote Sensing. With the need for a framework for studying the different loss functions relevant to common real-world challenges, a set of most relevant features has been defined.

Using a large-scale archive and with similar challenges to operational settings is a crucial factor which comes with drawbacks. One of them is the dataset size itself, forcing the change of training procedures from directly using raw images to protocol buffers. The wide range of characteristic of a model is a significant aspect in the analysis of different training procedures. There are numerous features that each model has by using a specific Loss Function and choosing a subset which is the most relevant to the Remote Sensing context is a challenging task. Being able to compare them in an objective way and meaningfully with respect to the Remote Sensing field is a non-trivial challenge. Moreover, having a comprehensive understanding of what the models are learning is another area with great importance. Using state-of-the-art methodologies helped to gain insights on this component of the analysis.

The thesis has been structured in 7 chapters.

Chapter 2 introduces the main concepts studied. Introducing the Convolutional Neural Networks and the basics of Loss Minimization to understand the importance of the Loss Function. Also, it's present an introduction to the MLC problem with a focus on Remote Sensing.

Chapter 3 describes the different Loss Functions used in the MLC context. The concepts behind every loss are introduced with their characteristics and providing an insight into the behaviour of the models.

In **Chapter 4** are defined the main properties needed to be evaluated when choosing a Loss Function. These might be dependent on the application of the model and the dataset itself.

Chapter 5 describes the “BigEarthNet” dataset and why has been selected for this comparison.

The technologies and parameters used for the experiments are explained. There is a breakdown of the type of experiments and how they have been carried out.

In **Chapter 6** The results obtained in the experiments are presented, showcasing the different performances and the behaviours of the models in various situations.

Chapter 7 summarizes the study and the findings. This last chapter provides also hints on how to choose the proper Loss Function and presents a future extension of this work.

2 Related Work

This work has been structured around the concepts of Deep Learning and Image Classification within the Remote Sensing field. Therefore is beneficial to describe the main concepts and provide foundations for the understanding of the following work.

2.1 Convolutional Neural Networks

Neural Networks are weighted graphs, which take inspirations from the simplification of the architecture of the human brain. The underlying concept is that each neuron creates outputs based on the received inputs from other neurons. While being an oversimplification of the real biological process, Neural Networks have proved to be highly successful in a vast set of applications. The Multi-Layer Perceptron is the traditional architecture associated with Neural Networks. Multiple nodes form an input layer, with one or more hidden layers and an output layer. The layers are fully connected among them, meaning that each node from the previous layer is connected to every node of the next layer. The inputs are processed with non-linear transformations allowing to compute non-trivial problems. In Image Classification, the Multi-Layer Perceptron has a significant drawback. Its total number of parameters can grow vastly while disregarding the spatial information of the inputs. There are many types of architectures and layers in the Neural Network domain that are great alternatives with imagery data. The Convolutional Neural Networks (CNN) are among the most successful architectures used in the image classification task. They draw inspiration as well from the biological realm, imitating the overlapping receptive fields of the neurons. The CNN architecture is composed of three different types of layers: the Convolutional layer, the Activation and Pooling layer and the Fully Connected layer.

The convolutional layer is the foundation of the architecture, employing the “convolution” mathematical linear operation. For simplicity, we will consider an input image in a two-dimensional form. The convolution operation covers the input image with a filter (or kernel), sliding over all the pixels sequentially. The filter itself is a matrix, usually 3×3 or 5×5 , which is learned and used to compute the element-wise product that is then summed.

This process is repeated at every location of the input, by sliding the position of the filter to the right by s steps. (or pixels) The final result of the convolution over all the image is called the “feature map”. The network will learn filters that will get activated when receiving a certain semantic feature in any location of the input. The resulting feature map is smaller than the initial input since it can be computed only from a minor subregion of the image. There are several hyperparameters to be

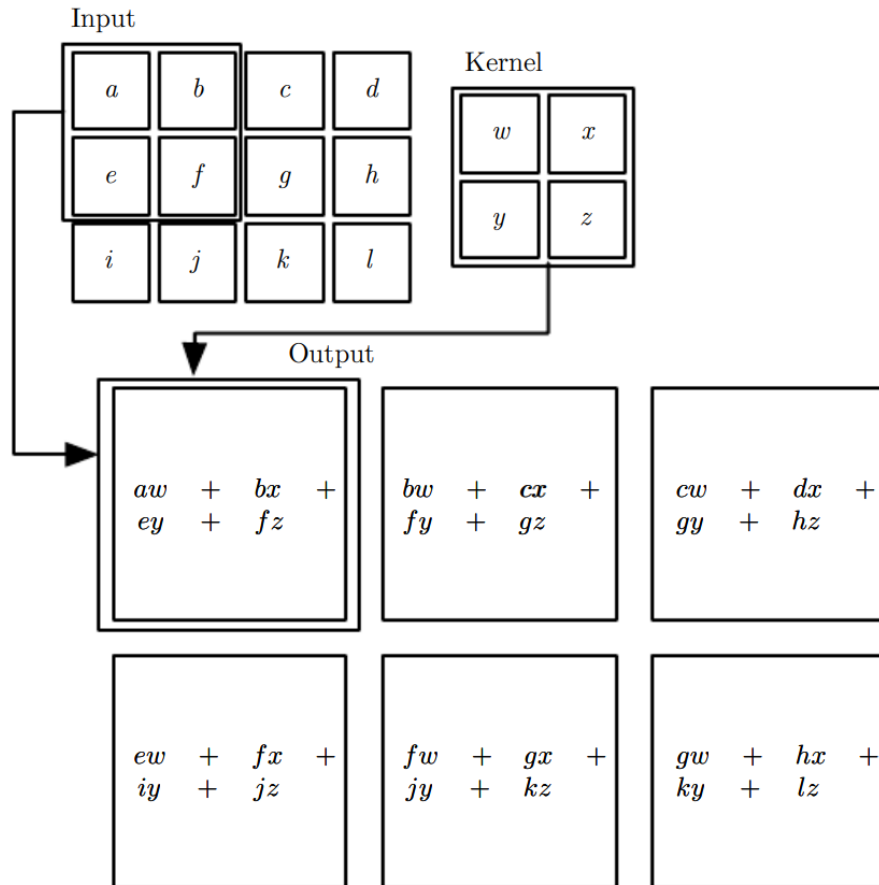


Figure 2.1: Convolutional operation over an input. The output is restricted only to positions where the kernel is entirely included in the input, also called “valid” convolution.

Image source [16]

considered when working with a convolutional layer.

- The filter size is usually regarded as a matrix, however, in practical applications, its depth is equal to the full depth of the input. The size covered by the filter is associated with the concept of the receptive field. The convolution uses the spatially local correlation of the input to enforce patterns sparsely connected in a local area. This results in neurons connected only to a small region of the input layer.
- The stride is the parameter that regulates the number of pixels or steps which the filter will slide to the right, for every step. Having a stride set to 1 will move the filter by one pixel at the time, providing overlapping receptive fields.
- In certain occasions, the corner pixels are “convoluted” only one time as opposed to multiple times in other regions. To overcome this, the padding adds additional data around the inputs, such that the true edges are considered as many times as the other regions.

2 Related Work

Generally, after the convolution layer is applied a non-linear activation function. The activation function of a neuron specifies the output of that node, provided the inputs. They represent the biological behaviour, whether the neuron is firing or not. Having an activation function which has only a positive firing rate, introduces a non-linearity which enhances the decision-making capabilities. The convolutional operator is simply a linear operator and using a non-linear activation will limit the problem of vanishing gradient. Using an activation function such as the ReLU will remove negative values from the activation map. Introducing non-linear properties in the architecture will increase the decision power of the network without affecting the receptive fields or generalization accuracy. [16] The resulting output is usually fed into the Pooling layer. The underlying concept of the pooling operator is to non-linearly down-sample, reducing the spatial size of the representations, parameters and computational load. The *max pooling* function is the most common, partitioning the input in non-overlapping regions based on the filter size and outputs the maximum value among that area. The specific location of a feature is negligible compared to its approximated position relative to other features. As shown in Fig. 2.2, pooling with filters in size of 2×2 with a stride of 2, downsamples the input by 2 along its dimensions, reducing the activations by 75%. An additional benefit from the pooling operation is that it introduces translation invariance, allowing the network to detect features in various positions in the inputs.

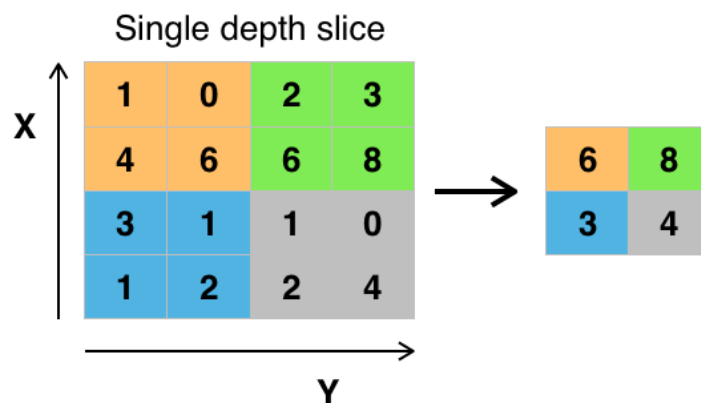


Figure 2.2: Max-Pooling operation over an input
Image source [36]

The final layer of a CNN architecture is usually a Fully Connected layer. The inputs are provided by the previous Activation or Pooling layer, resulting in N fully connected nodes equivalent to the number of classes. The Fully Connected layer decides which values in the resulting feature maps are the most relevant to a specific class. Generally, the resulting vector describes the hypothesized probability outputs over the classes.

2.2 Deep Learning Training

A *loss function* is a broad term that defines the mapping of an event into a real number, representing a cost (or loss) associated with the event. [5] In Deep Learning, often the Loss Function is a topic which is treated lightly even if needs greater attention and understanding. The underlying concept is that, due to the unknown true distribution of the events, we can minimize the (“empirical”) risk on a known set of (training) events which are drawn from the same distribution as the true one. The risk associated with a model is defined as the expectation of the loss function. We can define the empirical risk as:

$$\mathbb{E}_{\mathbf{x}, y \sim \hat{p}_{data}(\mathbf{x}, y)} [L(f(\mathbf{x}; \theta), y)] = \frac{1}{m} \sum_{i=1}^m L(f(\mathbf{x}^{(i)}; \theta), y^{(i)}) \quad (2.1)$$

where M is the number of training samples, $f(\mathbf{x}^{(i)}; \theta)$ the model predictions with the parameters θ on the i -th sample having a label $y^{(i)}$. [16] The significance behind the Loss Function is to summarize all the aspects of the problem to a scalar so that an improvement would translate into a better model. This is a non-trivial task, especially in the MLC case, considering that the Loss Function has to define a meaningful goal for the search within a considerably more complex parameter space. Therefore is essential that the function faithfully represents our design goals, avoiding unrelated error functions to our problem. A different understanding of the Loss Function can be formulated from a geometrical point of view. The Loss Function $L(w)$ can be seen as a surface sitting over the weight space w , where every point has a local gradient given by the vector ∇L . The trajectories of the optimizers are described over the Loss Function surface. The Loss Function defines the “goodness” of a model, however, due to the enormous amount of parameters in Neural Networks, there is the need for searching the optimal parameters (weights) that will minimize it. The optimizer is strictly linked to the loss function, defining the direction in the parameter space of the greatest improvement. The learning process is defined as the search of optimal parameters θ that minimize the loss function $L(f(x; \theta), y)$. While it’s beneficial having a good performance on training data, what we care the most is having good performances on the true distribution, suggesting having generalization capabilities. The direction of the greatest improvement is given by gradient methods, allowing the update of the parameters. The traditional formulation is:

$$\theta_{t+1} = \theta_t - \eta_t \nabla F(\theta_t) \quad (2.2)$$

where η_t is the step size of the update (also defined as “learning rate”) and $\nabla F(\theta_t)$ is the gradient for the t -th iteration. [16] Batch or deterministic gradient methods calculate the exact gradient using the entirety of the dataset. On the other spectrum of the approach, there are online methods, computing the gradients from a single example. Practical Deep Learning applications use approaches within these two, computing the gradients with a minibatch of samples, generally addressed as **stochastic** methods. The dimension of the batch size defines an index of accuracy of the gradient computation while offering regularization effects due to the noise provided. The Stochastic Gradient Descent

2 Related Work

(SGD) and its variants are among the most popular optimization algorithms in most of the Deep Learning applications. The method relies on an estimate of the gradient, based on a batch of m samples. The SGD iterative procedure presented in the Algorithm 1 updates the weight of the model according to the direction of the greatest improvement, using the estimated gradient shown in Eq. 2.2. This is performed until a stopping criterion has been met (Eq. 2.3), which usually is computed in terms of the delta between the minimum achieved loss and the current or last n -loss results.

Algorithm 1: Stochastic Gradient Descent (SGD)

Data: Learning rate η and initial parameters θ

while *stopping criterion not met* **do**

 Sample m instances $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ from the training set

 Compute gradient estimation for the current model θ_t

$\nabla F(\theta_t) = \frac{1}{m} \nabla_{\theta_t} \sum_i L(f(\mathbf{x}^{(i)}; \theta_t), \mathbf{y}^{(i)})$

 Apply update $\theta_{t+1} = \theta_t - \eta t \nabla F(\theta_t)$

The learning rate η impacts greatly this process and its choice is not a trivial task. The preliminary phases of the training try to address this problem by monitoring the learning curves of the Loss Function over the epochs. Having a learning rate that is too small, the training process will be slow with the risk to be stuck in a local minimum. If the learning rate is too large, the learning curve will be irregular with strong fluctuations and the risk of skipping the minimizer. The convergence criterion is defined using the excess error \mathbb{E} :

$$\mathbb{E} = J(\theta) - \min_{\theta} J(\theta) \quad (2.3)$$

where $J(\theta)$ is the current loss by the current model θ , while $\min_{\theta} J(\theta)$ is the loss achieved by the best performing model. The delta by which the current model improves with regards to the best model obtained during the training, correspond to the general trend of model improvement. When the model does not achieve substantial improvements we can stop the training process due to convergence to the minimizer. Gradient-based methods used on convex problems have an excess error of $O(\frac{1}{\sqrt{k}})$ after k iterations, while strongly convex problems have an excess error of $O(\frac{1}{k})$ after k iterations, where the lower the error the better the training. These bounds are of greater benefit, however, they cannot be further developed without implying additional conditions.

2.3 Multi-label Remote Sensing Image Classification

The growing number of satellites present in Earth orbits have provided a greater amount of data available for Remote Sensing applications. Scene Classification is a critical field in Remote Sensing with extensive applications which have received greater attention. Traditional approaches and datasets describe the entirety of the sample with a single class associated with the most relevant part of the instance. Remote Sensing scenes can represent a multitude of land-covers which can significantly vary among the same image. MLC is a favoured approach since it characterizes precisely the semantic content of the image with one or several labels. This, however, introduces several challenges such as skewed distributions of classes, exponential combinations of labels and potential underlying correlations among them. Moreover, identifying precisely similar categories (e.g. “Broad-leaved forest” and “Mixed forest”) is not a trivial task, especially when an instance contains analogous ones. Several approaches try to solve the MLC problem, with the same purpose, addressing the complex correlation between the semantic content of the input and the broad output space. In [44] a multi-label active learning framework is proposed, relying on a multi-label support vector machine. (SVM) A conditional random field (CRF) framework is proposed by Zeggada et al. [46] exploiting simultaneously spatial contextual information and cross-correlation between labels, Karalas et al. [22] analyses a set of ensemble-based multi-label learning architectures, based on binary relevance classifiers and label powerset classifiers, achieving good performances. Yet, the semantic complexity of Remote Sensing instances influences heavily their capacity to generalize on spatially distant samples. Deep Learning methodologies have attracted great attention in RS for performance-wise advances and generalization capabilities. In [45] is presented one of the first efforts using Deep Learning approaches on Multi-Label Classification tasks for UAV imagery. This approach uses a radial basis function neural network with a multi-labelling layer made with specific thresholding operations. Given the limited size of the dataset, they have applied a transfer learning approach using a pre-trained model based on the ILSVRC2014 dataset. The problem with this approach is that the learned features of these models are significantly different from the required features of Remote Sensing applications. Y. Wei et al. [43] proposed a flexible CNN framework where proposed object hypotheses are taken as the inputs of a shared CNN that is connected with each hypothesis. The results are aggregated with max-pooling to produce the ultimate multi-label predictions from the initial hypothesis. In [37] a novel multi-attention driven system that cooperatively utilizes a Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) to classify multi-label remote sensing (RS) images. The system uses a “K-Branch CNN” which extracts local descriptors using different CNNs for specific spatial resolutions, modelling their spatial relationship with a bidirectional RNN. The resulting local descriptors are used to produce multiple attention scores accounting for the correlation among the classes and provide the attention-based local descriptors. The final descriptors are used for the MLC task. Hua et al. [19] propose a 3-elemental module network, distinguished by its capabilities to produce discriminative label-wise features and reasoning about label relations in a meaningful way. Most of the works in MLC in Deep Learning and Remote Sensing settings use a traditional

2 Related Work

combination of Cross-Entropy loss and Sigmoid activations. Using Sigmoid activations each output of the model will consider the classification task with a one-vs-all approach, correctly assuming class independence. The cross-entropy has been extensively used in several fields and its effectiveness is backed by its information theory origin. While this has been effective in several studies it's not always the most suitable choice since this approach lacks the intrinsic properties of the MLC problem, such as sparse output distributions, distinguished importance towards specific classes and correlation among the labels. W. Edwards Deming and Nassim Nicholas Taleb support that loss functions need much greater attention in its choice should be carefully considered. [10]

Images can be annotated with multiple labels and modelling the rich semantic information in a precise way is crucial for image understanding. The labels might describe objects, scenes, actions and attributes and the Computer Vision field has addressed the MLC problem proposing several approaches. Wang et. al [42] has proposed a CNN and RNN framework employing an end-to-end model that exploits the semantic redundancy and co-occurrence dependency, both inline with the multi-label objective. The RNN component can model high-order dependencies and uses an attention mechanism to improve the prediction of small objects. A similar approach that uses attention maps is proposed in [49], where the network is able to exploit semantic and spatial relations between labels. The result translates in a network not only achieving greater performances but also providing more accurate activations with regards to the considered classes. Kang et. al [12] has presented a novel framework that explicitly focuses on the high-order correlation between labels. The improvement compared to traditional systems rely on the simultaneous propagation of multiple labels during the training. Alfassy et. al [3] address the problem of multi-label few-shot classification. The novel technique couples pairs of examples in the feature space, producing an integrated feature vector with labels obtained through set operations on the corresponding input pairs. Although these novel approaches produce great improvements, they generally address the training procedures blindly with the traditional Cross-Entropy Loss. Different Loss Functions, carefully chosen, might enhance the improvements obtained by these techniques.

3 Deep Learning Loss Functions for Multi-Label Classification

Traditional classification approaches were built around the assumption that each instance can be categorized simply by a single class. Recent advances in Machine Learning and real-world applications showed that there is a need for multiple labels in order to have a more accurate semantic description of the inputs. Existing work on MLC within Remote Sensing focuses mostly on the conventional combination of Sigmoid and Cross-Entropy Loss. The reason toward the usage of the Sigmoid Activation Function is the resulting independent Bernoulli distributions as opposed to multinomial distributions with a Softmax Function. From the Sigmoid function it's understandable that the predictions of one class are unrelated to the predictions of another class, producing independent probabilities for each one of the labels. The Sigmoid function is defined as:

$$P(\hat{y}_i|x_i) = \frac{1}{1 + e^{-z_i}} \quad (3.1)$$

where z_i is the class score for the i -th label.

We can formalize the problem by defining an archive $X = \{x_1, \dots, x_M\}$ that consists of M images. The sample x_i expresses the i^{th} instance. The archive contains instances that are associated with one or more classes from a label set $L = \{l_1, \dots, l_C\}$ with $|L| = C$. The association of the image x_i with the label information is determined by a binary label vector $y_i \in \{0, 1\}^C$, where every element of y_i indicates the presence or absence of a label $l_c \in L$. A MLC task can be formalized as the minimization of a general cost function $J(\theta) = L(f(x; \theta), y)$. The loss function is defined as $L(\cdot)$, the predicted model scores for the input x are defined as $f(\cdot)$ or \hat{y} , while y_i is the ground truth for the $i - th$ instance. [37]

3.1 Cross-Entropy Loss

The Cross-Entropy Loss (CEL) has strong foundations from Information Theory. Its principle is a general method of inference about an unknown probability density, given a prior estimate and new information as constraints on expected values. From a probabilistic standpoint, the Cross-Entropy between two probability distributions, drawn from the same underlying set of events, measures the average number of bits needed to distinguish an event over the set if the coding scheme for the set is optimized for the predicted probability distribution, rather than the true distribution. [32] The

3 Deep Learning Loss Functions for Multi-Label Classification

definition of the Cross-Entropy of a distribution \mathbf{q} relative to a distribution \mathbf{p} is formulated as:

$$H(p, q) = -E_p[\log q] \quad (3.2)$$

It is defined applying the *Kullback-Leibler divergence* $D_{KL}(p||q)$ which is the relative entropy of \mathbf{q} with respect to \mathbf{p} .

$$H(p, q) = H(p) + D_{KL}(p||q) \quad (3.3)$$

where $H(p)$ is the *entropy* of \mathbf{p} . Concerning the classification context, having discrete probability distributions \mathbf{p} and \mathbf{q} with the same support χ translates in:

$$H(p, q) = - \sum_{x \in \chi} p(x) \log q(x) \quad (3.4)$$

The principle behind it is the *Kraft-McMillan theorem* where any directly decodable coding scheme for coding a message to identify a value x_i out of a set of possible values $\{x_1, \dots, x_n\}$ can be defined as representing an implicit probability distribution $q(x_i) = (\frac{1}{2})^{l_i}$ over $\{x_1, \dots, x_n\}$, where l_i is the length of the code for the values x_i in bits.[5] Consequently, the Cross-Entropy can be explained as the expected message-length per datum when an hypothesized distribution q is assumed while the data is originated from a distribution p , motivating the expectation over the probability distribution p . The expectation of the message-length under the true distribution p is:

$$E_p[l] = -E_p \left[\frac{\ln q(x)}{\ln(2)} \right] = -E_p[\log_2 q(x)] = - \sum_{x_i} p(x_i) \log_2 q(x_i) = - \sum_x p(x) \log_2 q(x) = H(p, q) \quad (3.5)$$

Used in classification problems, the Cross-Entropy is correlated with the likelihood. The likelihood of the training set to be maximized can be defined as:

$$L(p, q) = \prod_i q_i^{N p_i} \quad (3.6)$$

where the number of samples in the training set is N , the predicted probability for a class i is \mathbf{q}_i and the empirical probability for the class i is \mathbf{p}_i . The resulting log-likelihood is obtained by dividing with the number of samples N , showing that its maximization is equivalent to the minimization of the Cross-Entropy.

$$\frac{1}{N} \log \prod_i q_i^{N p_i} = -H(p, q) \quad (3.7)$$

A possible approach to understand the Cross-Entropy is to define it as the log-likelihood for the ground truth y under a model \hat{y} . Evaluating the log-likelihood of a dataset under a model can be explained as the number of bits expected to use for encoding this data given that the encoding scheme

is based on the model hypothesis.

A more understandable and informal description can be provided as the “measure of surprise”. Having a model that predicts exactly zero probability to a specific class means that it has a strong hypothesis regarding it. However, if the class itself is present, it translates in $-\log(0) = \infty$ infinite *surprise*. Meaning that the model had a strong hypothesis and yet was extremely surprised of something did not account for, requiring infinite bits to encode that “impossible” event. When used as Loss Function the true distribution is given by the true classes, while the estimated distribution corresponds to the model predictions. Its minimal value (0) is found when the two distributions are equal, and typically it is computed by taking the average of Cross-Entropies in the dataset. Its effectiveness has been widely proven in several fields, such as Computer Vision and Remote Sensing. For notational simplicity, we rewrite the loss as:

$$L_{CE}(y, \hat{y}) = \begin{cases} -\log(\hat{y}) & \text{if } y = 1 \\ -\log(1 - \hat{y}) & \text{otherwise} \end{cases} \quad (3.8)$$

In the classification case, the derivative of the Cross-Entropy Loss with a Sigmoid Activation for a particular output unit has the same form of the regression case, formally:

$$\frac{\partial CE}{\partial x} = y(p_t - 1) \quad (3.9)$$

where $y \in \{0, 1\}$ is the binary target variable and p_t is the network output in the form, p if $y = 1$ and $1-p$ otherwise.

Loss Function choice and the activation function are strictly correlated components. From the previous example and the formulation is trivial to understand that the cross-entropy loss will provide infinite loss for predicting zero probability of non-zero ground truth. This forces to use a conservative hypothesis model which leaves small probability for any event. From a practical standpoint, the Cross-Entropy and Sigmoid activations are a great fit, however, this does not leave room for sparse output probabilities which, especially with a high number of classes, is highly desirable in MLC settings. This methodology is not suitable in some circumstances such as with unbalanced datasets, presence of outliers, time constraints on the training phase or when there is a demand to conform to label correlations. These conditions are the norm for operational settings and the choice of the Loss Function is a crucial component in solving them. The Cross-Entropy Loss and Sigmoid activations are the most popular choices in the MLC task, however, other options better address the challenges described beforehand.

3.2 Focal Loss

Lin et al. [27] propose a novel approach to solve the problem of extreme imbalance between foreground and background classes for one-stage object detector. It has been designed to increase the

accuracy of the one-stage detectors, to match the performances of computationally heavier and more complex two-stage detectors. The main challenge that prevents one-stage detectors from achieving superior performances is the class imbalance. This class of detectors has a large pool of candidate object locations covering different spatial positions, scales and aspect ratios.

One problem of the standard Cross-Entropy criterion is that “treats equally” hard and easily classified samples. The Focal Loss introduces several properties and handles naturally the class imbalance without having to consider the class distribution.

The Cross-Entropy Loss is characterized by the fact that instances classified with high confidence obtain a considerable loss. When framed over the training set, a large number of easily classified instances can overpower the most difficult classes. From a numerical standpoint, easily classified samples compose the majority of the loss and control the gradient. It’s worth noting that this is particularly relevant because heavily directs the model updates and therefore degenerate the learning process. The Focal Loss reshapes the Cross-Entropy Loss, accounting for the sample “hardness”, for calculating the penalization factor. The following discussion is formulated around the assumption that the computation of the probability p is done with sigmoid activation, delivering improved accuracy and greater numerical stability. We define p_t :

$$p_t = \begin{cases} \hat{y} & \text{if } y = 1 \\ 1 - \hat{y} & \text{otherwise} \end{cases} \quad (3.10)$$

The Focal Loss adds a modulating factor to the Cross-Entropy, resulting in:

$$L_{FL}(p_t) = -(1 - p_t)^\gamma \log(p_t) \quad (3.11)$$

where the labels are defined as $y \in \{\pm 1\}$, the model estimated probability for positive classes as p_t and γ is the *focusing parameter*. The gradient w.r.t. x is defined as:

$$\frac{\partial FL}{\partial x} = y(1 - p_t)^\gamma (\gamma p_t \log(p_t) + p_t - 1) \quad (3.12)$$

where p_t is defined in Eq.3.10 and y is the ground truth.

The Focal Loss introduces two likeable properties:

- For instances misclassified with low confidence, the modulating factor is close to 1 and the loss for that sample is almost unaffected. As the confidence rises, the modulating factor approaches 0, down-weighting significantly the loss for well-classified instances.
- The rate at which instances are down-weighted is regulated by the focusing parameter γ .

With this mechanism the Focal Loss is able to address at training time, the imbalance between foreground and background classes depending on the focusing parameter. (e.g. 1:1000 for $\gamma = 2$) Compared to larger backbone networks or more complex two-stage detectors, one-stage detectors trained

with the Focal Loss have higher accuracy delivered with faster inference speeds on a simpler architecture. Recently, there have been proposed in literature several applications using the Focal Loss. An example of that is the detection of lung cancer using a CNN with the Focal Loss on Computed Tomography scans, achieving significant results. [41] Another case is the application of the Focal Loss on an imbalanced dataset of the drug to drug interactions, using a hybrid Recurrent-Convolutional Neural Network. [40]

3.3 Weighted Cross-Entropy Loss

When dealing with imbalanced dataset many solutions attempt to address the problem. Resampling techniques approach this challenge by downsampling a subset of the majority classes or artificially upsampling the minority classes. However, studies in this direction have shown that this approach does not lead to significant improvements even with optimized parameter settings. [7] A typical approach to overcome the class imbalance problems is with cost-sensitive re-weighting. Trivially, this results in treating the cost of misclassifying a minority class as many times like the loss incurred from another class. Generally, re-weighting is performed by multiplying the Loss Function by a weighting vector inversely proportional to the class distribution. Using prior probabilities into the Loss Function improves the performance on imbalanced datasets. This technique is agnostic to the loss function used. However, the goal of our work is to understand the behaviour of the loss function, also when it is artificially modified. In fact, by multiplying the loss function by a scalar, the magnitude of the gradient is affected and therefore optimizers that rely on it might have a different behaviour. To have a fair comparison, we will apply weighting to the Cross-Entropy Loss, since its popularity. We formalize the weighting approach for the Cross-Entropy loss as:

$$L_{W-CE}(y, \hat{y}) = -wL_{CE}(y, \hat{y}) \quad (3.13)$$

where $w \in R^k$ with elements $w_k > 0$ defined over the dimension of the label set $L = \{l_1, \dots, l_C\}$ with $|L| = C$. The weighting vector can also be treated as a hyperparameter set using cross-validation. It is worth noting that using a class-balanced term is complementary to a differentiated behaviour based on sample difficulty, such as with the Focal Loss.

3.4 Hamming Loss

In Multi-Label Classification tasks, the ultimate objective is to predict the set of label(s) of a sample \mathbf{x} from an output space L . Generally, the output space of multi-label problems increases exponentially with is the cardinality. A solution to this problem is to shift the initial problem with $2^{|L|}$ parameters, into a set of simpler problems. The Binary Relevance (BR) transformation technique solves the obstacle with independent binary problems for each class, driving down the estimation to $|L|$ parameters. Another problem of the MLC task is that there might be incomplete observations for certain

instances or where there is a high number of classes with a substantial set of irrelevant ones. This domain of labels is generally called *weak labels*. [11] The Hamming measure is mainly considered as a metric for reporting performances in MLC problems. The Hamming Loss is defined as the normalized count of incorrectly classified labels. Minimizing this measure is adherent to the multi-label task and its usage in this realm is having greater attention. This can be seen as the relaxed objective of multi-label problems, more forgiving than the 0/1 Loss which is considering a prediction correct only if there is the exact match. It is defined as:

$$L_{HA}(y, \hat{y}) = \frac{1}{|L|} xor(y, \hat{y}) \quad (3.14)$$

where XOR is the Boolean operator and \hat{y} the label prediction of the model.

However, it is a non-convex and discontinuous function therefore difficult or even impossible to optimize it for certain problems.[14] Another problem is that the Hamming Loss treats all the labels equally, neglecting the possibility of having different degrees of importance for the labels or pushing for the concept of label sparsity. Although the Hamming Loss addressed concisely the MLC problem, it might be unsuitable for situations where the output space is vast. Prabhu et. al [21] show discuss that the Hamming Loss is inadequate for solving Extreme Multi-Label problems with a number of labels orders of magnitude higher than the traditional MLC. For this context the Hamming Loss would penalize models that predict missing labels which could have been relevant to the instance. The Hamming Loss does not focus on a set of relevant labels, treating the uniformly the various classes and resulting in Extreme Multi-Label models that perform inadequately. Many works have been proposed to improve the simple formulation of the Hamming Loss. Among them it is worth noting Dembczyński et.[9] al analyzing the label dependence using the Hamming Loss, identifying mainly two scenarios conditional and marginal dependence. In [11] is described an approach for the estimation of approximated partial predictions. The Hamming Loss is used addressing the problem by efficiently focusing on label-wise information on convex sets of probabilities. Results are shown in terms of improvements regarding the prediction with missing or incomplete data.

3.5 Huber Loss

The Huber Loss is regarded as an important tool in robust statistics. It's a loss function mainly used in regression problems, however, its likeable properties are helpful also for classification settings. The Huber Loss is less sensitive to outliers when compared to a squared error loss. This Loss Function has a mixed behaviour; based on two functions, the squared loss and the absolute loss:

$$L_{HU}(y, \hat{y}) = \begin{cases} \max(0, 1 - y\hat{y})^2, & \text{for } y\hat{y} \geq -1 \\ -4y\hat{y}, & \text{otherwise} \end{cases} \quad (3.15)$$

The idea is to combine them and exploit the advantages of each one of them. The drawback of the

squared loss is that it's heavily influenced by outliers, shifting the focus of the training away from inliers. The drawback of the absolute loss instead is that, when close to the neighbourhood of its minimum, does not reward enough changes toward the goal. The Huber Loss combines the advantages from the previous two losses providing robustness on data including outliers. An interesting property of the Huber Loss is that it's strongly convex in a uniform neighbourhood of its target. Strong convexity implies several conditions, among them the Polyak-Lojasiewicz (PL) inequality which can be formalized as:

$$\frac{1}{2} \|\nabla f(x)\|^2 \geq \mu(f(x) - f^*), \forall x \quad (3.16)$$

The linear convergence guaranteed by the PL inequality can be applied to functions which could be not convex. Therefore gradient method can have linear convergence to global minimizers without the convexity constraint. This property enables faster and linear convergence rate compared to other approaches. More specifically the Huber Loss yields an excess error of $O(\frac{1}{k})$ after k iterations. This great convergence rate hits the boundaries of the achievable decrease of generalization error. [6] discusses that would not be beneficial searching for optimization algorithms with convergence rate faster than $O(\frac{1}{k})$ because would lead to increased overfitting. Different than the function convexity, another characteristic which supports the optimization of loss functions is its actual differentiability. The Huber Loss is once-differentiable since has an MSE behaviour in the proximity of the target, as opposed to the MAE. It's worth noting that the differentiability is not a sufficient condition for guaranteeing convergence to a global minimum, however, it's a required condition for providing a non-zero gradient back to the model. (i.e. backpropagation)

3.6 Ranking Loss

Ranking frameworks have recently got great attention, achieving excellent results in several fields. Label dependency is a significant component of the MLC problem that ranking approaches are leveraging. Several variations have been proposed in literature, however, we will focus on the standard Ranking approach to represent this class of techniques. The concept behind the Ranking Loss is that, while it is certainly relevant to classify positive labels, it is also important for the model to perform "sensible" mistakes by assigning higher ranks to the positive labels than to most of the negative labels. An equivalent perspective on the approach is discussed in [22], where the Ranking Loss is defined as the evaluation of the average fraction of label pairs that are ordered incorrectly. This introduces likable properties in terms of continuous improvements of the predictions for irrelevant labels, efficiently using the sample information. The Ranking Loss induce the model to produce a vector of prediction with higher values for the positive labels y (ground truths) than the negative labels.

$$f_u(x) > f_v(x), \quad \forall u \in y, v \notin y \quad (3.17)$$

The Ranking Loss can be defined as:

$$L_R(y, \hat{y}) = \sum_{v \notin y} \sum_{u \in y} \max(0, \alpha + f_v(x_i) - f_u(x_i)) \quad (3.18)$$

where u is the label set associated with the relevant ground truths y , v is its complement of irrelevant labels and α is an hyperparameter that introduces a margin, usually set to 1. One drawback is that the ranking objective, minimizing the occurrences where a positive label has a lower rank than a negative label; is different from the actual multi-label objective. In other words, the Ranking Loss optimizes the area under the ROC curve (AUC) but does not directly optimize the top- k annotation accuracy. When Ranking Losses are applied to multi-label problems, they lack a decision-maker component defining what labels are included in the predictions. Several approaches can be included in the training process to overcome this problem such as top- k and thresholding. Li et. al [26] is an example of a smooth version of the Ranking Loss with a label decision module that provides estimations with the optimal confidence thresholds for each class. A different approach is the one by Gong et. al [15] where there is an approximate top- k ranking objectives, accurately selecting the first k labels to be included in the final prediction. However, they do not take into consideration the semantic content of the image since an instance might contain a single visual concept and be forced to include k classes. A drawback of the Ranking Loss is that, generally, this approach can not model higher-order correlations.

3.7 SparseMax Loss

A radically different approach is the one introduced by the SparseMax Loss, which coupled with the SparseMax activation function can output sparse support, assigning exactly a probability of zero to some outputs. Having sparse posterior distributions is appealing when there is the need for filtering large output spaces, to identify which group of variables are potentially relevant and to predict multiple labels. Filtering large output spaces and predicting multiple labels makes the SparseMax loss and SparseMax transformation very appealing for the MLC problem since addresses specifically the multi-label objective. The SparseMax transformation introduces these properties while yielding most of the likeable properties of the softmax activations. An example of that is the two-class case, where the softmax activation becomes the sigmoid function, which employs the same for the SparseMax transformation. It uses the softmax function as a starting point, by acknowledging the limitation of the resulting probability distribution, which always has full support. Defining the $(K-1)$ -dimensional simplex $\delta^{K-1} := \mathbf{p} \in \mathbb{R}^K | 1^T \mathbf{p} = 1, \mathbf{p} \geq 0$, the SparseMax transformation maps a vector of real weights \mathbb{R}^K (e.g. label scores) to δ^{K-1} probability distributions. The SparseMax transformation is defined as:

$$sparsemax(\mathbf{z}) := \underset{p \in \nabla^{K-1}}{\|\mathbf{p} - \mathbf{z}\|^2} \quad (3.19)$$

Which results in the Euclidean projection of the input vector \mathbf{z} onto the probability simplex. This

projection is likely to be on the boundaries of the simplex resulting in sparse support with regard to the initial input vector. The construction of the SparseMax Loss starts with defining a *gradient* that resembles the one from logistic loss with softmax activations. The gradient for the SparseMax Loss for the multi-label classification case is:

$$\nabla_{\mathbf{z}} L_{\text{sparsemax}}(\mathbf{z}; \mathbf{q}) = -\mathbf{q} + \text{sparsemax}(\mathbf{z}) \quad (3.20)$$

where δ_k is the delta distribution on sample label, providing $[\delta_k]_j = 1$ if $j = y_i$ and 0 otherwise. Following with the SparseMax Loss formulation

$$L_{SM}(q, z) = -q^T z + \frac{1}{2} \sum_{j \in S(z)} (z_j^2 - \tau^2(z)) + \frac{1}{2} \|q\|^2 \quad (3.21)$$

where τ is the thresholding function, $S(z)$ is the support of $\text{sparsemax}(z)$ and q the target distributions. The SparseMax transformation computes a threshold $\tau(z)$ for which coordinates above it will be shifted by the support of the input, while the others will be floored to zero. The resulting Loss Function is differentiable everywhere, with a gradient defined in Eq. 3.20 and it's convex. Another appealing property in the context of classification is that holds a separation margin like the Hinge Loss. However, retaining important properties for smooth optimization methods, such as being differentiable everywhere and convex. Furthermore, the SparseMax Loss in the binary case reduces to the Huber classification loss. The Eq. 3.21 for $|S(z)| = 1$ is:

$$L_{\text{sparsemax}}(\mathbf{z}; k) = -z_k + z_{(1)} \quad (3.22)$$

While for $|S(z)| = 2$ becomes:

$$L_{\text{sparsemax}}(\mathbf{z}; k) = -z_k + \frac{1 + (z_{(1)} - z_{(2)})^2}{4} + \frac{z_{(1)} - z_{(2)}}{2} \quad (3.23)$$

where $z_{(1)} \geq z_{(2)} \geq \dots$ are the sorted components of \mathbf{z} and $t = z_1 - z_2$.

$$L_{\text{sparsemax}}(t) = \begin{cases} 0 & \text{if } t \geq 1 \\ -t & \text{if } t \leq -1 \\ \frac{(t-1)^2}{4} & \text{if } -1 < t < 1 \end{cases} \quad (3.24)$$

4 Comparative Analysis of Deep Learning Loss Functions

There are several properties that can be used for studying and comparing the above-mentioned loss functions. We have identified the most relevant ones in terms of applicability to real problems that a model might incur, which are: 1) capabilities to handle imbalanced data sets; 2) robustness to outliers; 3) function convexity and differentiability; 4) learning efficiency. Loss functions that do not have these features lack in terms of performance and, to a certain degree, their applicability is limited. We use generally the term “feature” as a broad term to define properties or applicability on operative contexts. In Table 4.1 we categorize the features of the various Loss Functions in terms of *None* (-), *Low*, *Medium* or *High* adherence to the concept.

Table 4.1: Comparison of MLC losses under considered criteria

Losses	Class Imbalance Awareness	Outlier Awareness	Convex and Differentiable	Learning Efficiency
CEL [18]	L	L	M	H
W-CEL [18]	H	L	M	L
FL [27]	H	L	M	M
HAL [13]	L	L	–	M
HL [20]	L	H	H	M
SML [30]	L	L	M	H
RL [26]	L	L	–	H

4.1 Capability of Loss Functions to Handle Class Imbalance

The majority of real-world applications include some degree of class imbalance within the dataset. This can be summarized as when one or multiple classes are not equally represented in the data. Traditional Deep Learning classifiers such as with the Cross-Entropy Loss with low awareness toward specific labels will incur in bias towards the majority class, and possibly neglecting the minority class. This obstacle is indeed enhanced in MLC because one class might not be represented in most of the samples, due to the intrinsic nature of the problem. Within the same domain of learning procedures, class-weighting techniques are a traditional approach dealing with skewed class distributions that introduce differentiated consideration toward specific classes. The intent is to adjust the contribution to

the total loss roughly on the same scale, from the different classes. This approach is de-coupled from the actual loss-choice and can be applied independently to any function. As discussed in Section 3.3, artificially modifying the magnitude of the gradient significantly changes the training procedure and therefore the weighting vector has to be chosen very precisely. Similarly, a distinct approach is based on the sample “hardness”, which generally speaking, is more likely to belong to poorly represented classes. The Focal Loss exploits this effect, however, there is no direct correlation between sample hardness and class numerosity, which might lead the training process to focus on hard samples but associated with vastly represented classes. To have an understanding of the difference with regards to the standard Cross-Entropy Loss is useful to compare them from a numerical standpoint.

Table 4.2: Cost-sensitivity comparison between Cross-Entropy and Focal Loss

	CEL	FL	Down-weighting power
P(0.1)	1	0.81	≈ 1.23
P(0.5)	0.3	0.0752	≈ 3.92
P(0.9)	0.04575	0.0004575	100
P(0.968)	0.01412	0.00001446	≈ 976

With a default focusing parameter $\gamma = 2$, we see that for hard samples the down-weighting effect is limited at most $\approx 4x$. In Table 1 we show that for easily classified samples, the Focal Loss has a much lower loss ($\approx 1000x$) compared to the standard Cross-Entropy. The results translate in a loss that focuses on instances with low support, directing the training on a sparse set of hard examples. Depending on the γ parameter, this behaviour has a much stronger down-weighting effect compared to the Weighted-Cross Entropy Loss since has an exponential nature. However, it is worth noting the two techniques are not exclusive and could be used together to further improve the prediction towards difficult and minority classes. The Hamming, Huber and Ranking Loss do not specifically address a differentiated behaviour for particular classes like the SparseMax Loss, however, the latter has shown greater performances on minority classes.

4.2 Capability of Loss Functions to Handle Outliers

Large scale datasets are key elements that allow DNNs to achieve great performances. They rely on high-quality datasets which are usually expensive and often limited in size. Crowdsourcing platforms (such as Amazon Mechanical Turk) for classifying datasets, allow annotating a great number of instances in a relatively cheap and fast way. However, the quality of the annotation process is correlated with price and time availabilities; often resulting in noisy datasets. It follows that many datasets contain noise and many applications require robustness to outliers; more specifically that a model is not *heavily* influenced or biased by outliers. Outlier and noise are mostly concepts that are well covered in the regression field, however, they affect as much as in the classification settings. Training procedures that can learn effectively from noisy datasets would allow a broader applica-

tion. It's worth noting that in the regression context is much easier to define an outlier, yet, for the classification realm, is necessary to abstract the concept to a broader set of possible situations. Relatively to the setting, there can be many types of outliers (e.g. wrongly labelled data, missing labels, instances with Gaussian noise, areas with extremely high spectral value,.. [1]), which can greatly affect models that do not consider these sets of instances. These approaches would result in models overly sensitive to outliers, unreliable predictions and having an inconsistent behaviour in the training over the samples. The Cross-Entropy Loss (and therefore the Weighted version) does not show robustness to outliers since its equivalent to a Maximum Likelihood Estimation, which is known for not exhibiting robustness. The Hamming Loss does not have a differentiated behaviour for addressing the problem of outliers. The Huber Loss, initially conceived for regression problems, has a differentiated form depending on the situation. It combines the advantages of the linear loss for large misclassifications, with the quadratic loss for margin values close to the target providing high awareness to possible outliers. This behaviour shifts the attention to observations that are correctly classified and therefore linearly penalize the samples which are incorrectly classified. As introduced in Section 3.2, one drawback of the Focal Loss is that could focus extensively on a small set of outliers, biasing the training procedure for precisely annotated samples. We can relate the two losses in terms of the nature of samples on which the focus is addressed. Intuitively, the Huber Loss reduces the contribution of *outliers* by down-weighting the loss of samples with large errors (hard-samples). Differently, the Focal Loss instead of addressing outliers, down-weights *inliers* (easy samples) focusing on a sparse set of hard examples.[27] From here the feature that, if the training information is highly affected by outliers, the Focal Loss might not be the best choice and that the Huber Loss should be considered instead. For their nature, the Ranking and the SparseMax Loss do not have special attention to outliers.

4.3 Convexity and Differentiability of Loss Functions

Good performances of Deep Neural Networks are based on the ability to properly minimize the loss; obtaining a model, usually in a neighbourhood of a minimizer, using gradient methods. The optimization tasks of Deep Neural Networks are generally non-convex problems, exhibiting convex properties in the trajectory of gradient minimizers. [17] It's worth specifying that the term "convex" and "convexity" are related to the function itself, which differs from the non-convexity nature of Deep Neural Networks with regards to the model parameters. However, has been shown that using convex Loss Functions yields likeable properties, such as the better trainability due to a smoother profile or pushing for large-margin solutions resulting in better generalization capabilities. [35] The concept of optimizing a loss function is an NP-complete problem however gradients methods provide a way to determine minimizers. Some convex Loss Functions provide a flat "minima" region where any of the points could be a good model. Opposed, with non-convex functions, we can encounter a high number of local minima and saddle points. We can state that the actual convexity or non-

convexity structure of a Loss Function is correlated to its actual trainability. The characteristics of the landscape of the neighbourhood within the minimizer heavily impacts the generalization capabilities of a model and its reliability in terms of predictions. These features are defined by the geometry of the surface of the loss and are influenced by several choices such as the network depth, the optimizer, the network initialization and many other factors. Because of its nature, (requiring training over all the dataset) studying and evaluating Loss Functions is extremely time-consuming, therefore the research in this field is primarily theoretical. Entropy-based Losses are convex, the Hamming Loss and the Ranking Loss do not satisfy the required conditions, while the SparseMax Loss has been designed with a convex and differentiable nature. Some functions exhibit additional properties to just convexity such as the Huber Loss with strong convexity. This property enables faster and linear convergence rate compared to normal convexities which will be later covered. The byproduct of a strongly convex function is the well-behaved surface loss, providing a much smoother baseline that could be better optimized even with deeper architectures. As shown in Section 3.5 the Huber Loss is quadratic in the proximity of the target. Following certain conditions, when the Loss Function is quadratic, every local minimum of the empirical loss are global minimizers. Yet, having a global minimum does not translate in no misclassification error. Different than the function convexity, another characteristic which supports the optimization of loss functions is its actual differentiability. It's worth noting that the differentiability is not a sufficient condition for guaranteeing convergence to a global minimum, however, it's a required condition for providing a non-zero gradient back to the model. (i.e. backpropagation) [4] Several techniques allow the training of loss functions which have points of non-differentiability. However, in doing so, they will undesirably change the scope of the cost function and introduce an additional layer of complexity computationally-wise. Generally speaking, we can also link the differentiability of a loss function, simplifying the topic, as the introduction of a certain degree of local smoothness.

4.4 Efficiency of the Considered Learning Mechanism

Applications may require fast training procedures. This can be interpreted as reaching a specific performance in fewer iterations or achieving superior performances with equal training time. Producing a model with greater performances in fewer iterations can be defined as “learning efficiently”. This can be visualized as a dimensionless performance concept that relates results and the data used. The name “efficiency of learning mechanism” has been chosen to define how much more improvement of the parameter space is provided by equal amounts of information. Having all the models performing the training procedure on the same dataset is helpful to highlight the Loss Functions that have better optimization trajectories. From a mathematical perspective, the different Loss Functions yields different gradients during the backpropagation. Deep Neural Networks problems are highly non-convex problems, therefore an analysis of their performance using convex optimization methods is a non-trivial task. A different interpretation can be obtained by reframing the problem relative to

4 Comparative Analysis of Deep Learning Loss Functions

the starting point of the training procedure and the final minimizer within the loss landscape. Models with stronger gradients have better usage of the data, providing a smaller distance between the initial training state and the neighbourhood of the minimizer. As shown in Section 2.2, the properties of the Loss Function affect the bounds of the rate at which a problem can converge. Using Stochastic Gradient Descent, after k iterations, convex functions will provide a **excess error** of $O(\frac{1}{\sqrt{k}})$ while strongly convex functions will have an **excess error** of $O(\frac{1}{k})$. This is a relevant property when evaluating the convergence of a procedure, however as shown in our results, the other aspects of the models might considerably affect its convergence. The number of local minima and saddle points plays an important role in the trajectory of the optimizers, reflecting on the actual performance of the model. There are several aspects from the architectural perspective that affect this, such as the network depth, the usage of skip connections or the model width. Neglecting the analysis of the influence optimizers with adaptive learning rates, we can investigate and compare the behaviour of the different loss functions in the first epochs of the training. Improvements in this direction are very beneficial, not only for fixed-time training procedure but also to have an optimization process with better trajectories within the landscape loss.

5 Data Set Description and Design of Experiments

5.1 Description of the Data Set


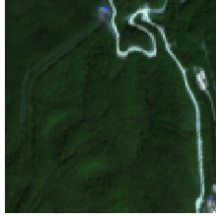
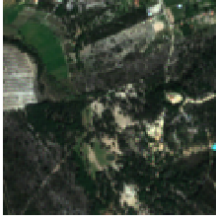

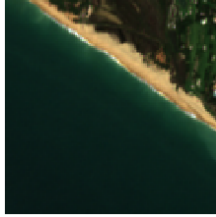
Comparing different models trained with different Loss Functions needs a meaningful framework that would challenge them under several aspects. The experiments have been carried out on the large-scale archive “BigEarthNet”, formed over 125 Sentinel-2 tiles on 10 European countries and acquired between June 2017 and May 2018. [38] The tiles have been divided into 590,326 non-overlapping patches, initially annotated with 43 land-cover classes (i.e. multi-labels) provided from the CORINE Land Cover database of the year 2018. Consequently, the archive has been updated with an improved labelling structure including 19 land-cover classes, providing less ambiguity while achieving an accurate sample description. [39] The size and diversity of the archive are an important factor for providing a realistic benchmark to the various Loss Functions.

The images are composed of 12 spectral bands, with different spatial resolutions. (120x120 pixels for the 10m bands; 60x60 pixels for the 20m bands and 20x20 pixels for the 60m bands) The samples have a number of labels that extends from 1 to 12, with 5% of the images described by more than five classes. In Tab. 5.1 are shown examples of BigEarthNet-19 scenes with their multi-labels. The dataset identifies 70,987 patches that are fully covered by seasonal snow, cloud and cloud shadow, which we promptly removed. The remaining images are shuffled and split in 52% for the training set, 24% for the validation set and 24% for the test set, respectively with 269695, 123723 and 125866 instances. In order to identify the most and least represented classes is useful to know the class distribution over the dataset. Table 5.2 shows the number of samples for each class in the various splits of the dataset.

5.2 Considered Deep Learning Technologies

Comparing different Loss Functions requires extensive iterations between the training of a model, evaluating its performances and assessing its meaningfulness. In order to speed up this process and have an agile approach that allows fast changes, it has been employed a cloud-based Python notebook allowing the transformation of data, training on GPUs and visualization of the results. More specifically Google Colab has been chosen as the platform for assessing the validity of the data, starting the training of the models and then analysing the results. However, for the complete training of the

Table 5.1: An example of Sentinel-2 image patches and their multi-labels in the BigEarthNet archive

 <p><i>Arable land, Mixed forest, Transitional woodland/shrub and Urban fabric</i></p>	 <p><i>Arable land</i></p>	 <p><i>Broad-leaved forest</i></p>
 <p><i>Complex cultivation patterns, Coniferous forest and Mixed Forest</i></p>	 <p><i>Beaches, dunes, sands, Marine waters and Urban fabric</i></p>	 <p><i>Beaches, dunes, sands, Marine waters and Permanent crops</i></p>

models, we used the High-Performance Computing (HPC) cluster kindly provided by Faculty IV at TU Berlin. This has been necessary since the training was carried out on 80 epochs, exceeding the availability time of the Google Colab sessions. Within the HPC, the greater computational power for the training was provided by the GPUs, the Tesla P100, enabling faster training times. The framework used for training the models is Tensorflow, providing a comprehensive, flexible ecosystem, libraries and a large community for developing Deep Learning models. An example of a tool made available by Tensorflow which has been used to plot some results is TensorBoard, which seamlessly integrates within the framework allowing the visualization of the training losses and metrics. Another relevant set of component used to compare the different losses has been the explainability element. In doing so, the state-of-the-art techniques have been employed for evaluating the predictions. More specifically, the Layer-wise Relevance Propagation (LRP)[2] has been used which decompose the prediction in terms of the contributions of individual input features in a simple way, allowing the visualization of the explanations in the same form as the input data. LRP perform a conservative relevance redistribution procedure with a backward pass on the Neural Network.

Table 5.2: Number of samples in train, validation and test set for each class

Label	Num. Images		
	Train	Val	Test
Agro-forestry areas	15790	7598	7261
Arable land	100394	46604	47150
Beaches, dunes, sands	1197	118	221
Broad-leaved forest	73411	33759	34130
Complex cultivation patterns	53534	25031	25638
Coniferous forest	86569	38674	39532
Costal wetlands	1037	219	310
Industrial or commercial units	6182	2875	2808
Inland waters	35349	15751	16177
Inland wetlands	11620	5131	5349
Land principally occupied by agriculture, ..	67260	31325	32052
Marine waters	39114	17740	18023
Mixed forest	91930	41996	42641
Moors, heathland and sclerophyllous vegetation	8438	3970	3859
Natural grassland and sparsely vegetated areas	6663	2560	2799
Pastures	50981	23846	24170
Permanent crops	15862	6676	6812
Transitional woodland/shrub	77593	35146	36211
Urban fabric	38783	18180	17928

5.3 Design of Experiments

Optimization of Deep Neural Network is generally a difficult task for different aspects introduced by the complex nature of this approach. Therefore the evaluation of different training procedures requires a comprehensive analysis under different points of view. The experiments have been designed to avoid bias, providing a fair platform for analysing and compare the different losses. A fine choice of training parameters (batch size, learning rate, optimizer) produces superior performance and minimizers that generalize better. The causes for these changes are not well understood [25], however, the goal of our study is not to obtain the greatest performances but to understand the changes in behaviour among the models trained with different losses and therefore the training parameters have been chosen in an agnostic way. We have identified 4 factors, that generally affects the different training procedure the *network depth*, the *batch size*, the *learning rate* and the *activation function*.

Most of pragmatic Deep Learning training procedures have the assumption that we can compute the exact gradient for our model. In reality, we have access only to an approximated version which could introduce error or bias. Algorithms rely on mini-batches to sample the dataset and compute an approximation of the true gradient. One way of mitigating this factor is by choosing a relatively large mini-batch which would have a slightly more reliable gradient compared to a small one. This is especially helpful in determining which Loss Function focus on less represented classes. Having a small

mini-batches could avoid entirely the evaluation of such classes because they would not be included in the sample itself and therefore not highlight the differentiated behaviour of specific losses. Architecture choices impact heavily the landscape of the Loss Function with noticeable differences in term of the sharpness of minimizers. One crucial choice is the actual structure of the architecture, which can be defined in terms of type of layers, usage of skip connections, filters and more importantly the depth of the network. In [25] is visualized the correlation between the network depth and the resulting loss surface, visibly becoming more complex and sharp, as the number of layers increases. The degradation of the loss surface can be characterize in terms of non-convex sections, such as large regions with chaotic gradient directions and increased steepness of the surface in most of the directions. For example purposes, we can define a deep model as several iterations of the multiplication of a weight matrix W . Supposing that the matrix W can be eigen-decomposed in $W = Vdiag(\lambda)V^{-1}$. Having T layers, can be associated to as many multiplications, which can be formalized as:

$$\underbrace{W \times W \times W \dots W}_{|T| \text{ layers}} = W^T = (Vdiag(\lambda)V^{-1})^t = Vdiag(\lambda)^t V^{-1} \quad (5.1)$$

Resulting in a gradient balanced with respect to $diag(\lambda)^t$. Eigenvalues that are less than 1 in absolute value, will vanish, significantly increasing the challenge to detect the course in which the Loss Function should improve. Eigenvalues that are greater than 1 in absolute values, will explode resulting in very steep cliffs within the loss surface. The goal of this work is to analyse the differences within the Loss Functions in an objective way and not to achieve state-of-the-art numerical performances, so using a low number of layers is a meaningful choice in addressing it. Having a shallow network architecture will less likely form irregular and scattered regions in the loss landscape. Another feature linked to the network architecture is the choice of activation functions. Architectures using Rectifiers (ReLU activation function) are less prone to vanish the gradient since they saturate only in one direction. Neural Networks that uses ReLU activations are trainable in faster times and do not penalize the generalization accuracy. [24] Other than being among the most used activation functions and therefore more adherent to operational settings, using ReLU activations, would provide a framework to understand the features of generally less trainable Loss Functions. Goldstein et al. [25] support that generally, the majority of the training process is spent tracing a wide arc around a mountain-shaped minimizer. Several works in the literature plan to address this problem by finding good initial points for the training procedure. Other features that factor for the goodness of the trajectory and the training process, are the *batch size* and *learning rate*. It's generally agreed that having training procedures with small-batches are more likely to result in "flat" minima while large-batches will result in "sharp" minima. The shape of the minima is strongly related to the generalization capabilities of a model, so having a large batch-sizes will lead to poor generalization capabilities, while small batch-sizes can lead to good generalization capabilities. This is explained by the "noise" introduced by the approximation of the small-batch, which reduce the certainty on the training set and avoiding overfitting. However, having a batch size that is too small, will most likely lead to poor performance,

so it's helpful searching for a good batch size between the two extremes. Poor conditioning and discontinuous gradients of the Loss Function might steepen (or tightening) the area with a minimizer, yet the initial experiments showed that a batch size of 1024 is a good spot between generalization capabilities and trainability. The second parameter that affects the training to good minimizers is the learning rate. It affects both the trainability and the generalization capabilities of the model. Lower learning rates provide better training procedures and have poor generalization ability, while larger learning rates provide more irregular training and better generalization abilities. A logical approach would be to sensibly reduce the step size of the local descent however this would result in higher computational costs to reach the minimizer. In other scenarios, such as with an extremely small learning rate, the local descent could result in landing on an area with saddle point or wide flat regions, impeding continuing the search towards valid solutions. [25] For this work, the learning rate has been chosen to 10^{-4} with a set of preliminary experiments for all the losses, to provide a good trainability and good generalizations capabilities. Later we present how Loss Function with poor conditioning and discontinuous gradients does not have a well-behaved profile, incurring in more frequent saddle points and irregular contribution heatmap. The comparison of models trained with different Loss Functions can be performed under different aspects. The main two perspectives are the analytical performances and empirical results in terms of heatmaps. A trivial approach to compare Loss Functions is by examining their prediction performances. Due to the various nature of the losses a numerical comparison of the changes in terms of absolute value is meaningful only when the Loss Function has the same underlying concept. This can be applied to the Cross-Entropy Loss, Focal Loss and Weighted Cross-Entropy Loss, since are all constructed around the concept of entropy. Comparing the raw loss values would be helpful in the cases above, for showcasing on which cases the Loss Function would draw more attention, such as on less represented classes. In traditional multi-class classification methods, the evaluation of the performance of a model is a much more simplified task due to its unambiguous definition and therefore evaluation of correct (or wrong) prediction. In multi-label classification problems, this role is much more complex since predictions could be neither completely wrong nor completely right. It is easy to understand why the absolute accuracy is not an appropriate metric since would not measure how models are performing when predictions are not completely correct. Other metrics such as Precision and Recall offer more flexibility on this front, allowing different measurements in terms of labels or instances. [47] We can define Precision as the average portion of predicted correct labels to the total number of actual labels:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (5.2)$$

The Recall can be defined as the average portion of predicted correct labels to the total number of predicted labels:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (5.3)$$

Precision and Recall measure disjoint aspects of predictions, so it is more meaningful to take into

5 Data Set Description and Design of Experiments

consideration a metric that takes into consideration both aspects. The F1-Measure is the harmonic mean of precision and recall, allowing to shrink to a single number the performance of a model in a reliable way.

$$F1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (5.4)$$

For these metrics, we can have a distinction in terms of how we evaluate the labels. One approach called *macro*-averaging is to evaluate the individual class labels first and then average it over the classes. (e.g. calculating precision for each label and finding their unweighted mean) The opposite approach called *micro*-averaging is to evaluate globally over all the instances and all the class labels. (e.g. calculating the precision by finding the total true positives and false positives)

$$\text{precision}_{\text{micro}} = \frac{TP_1 + \dots + TP_n}{TP_1 + \dots + TP_n + FP_1 + \dots + FP_n} \quad (5.5)$$

where $TP_1 + \dots + TP_n$ is the sum of the true positives for each of n classes and $FP_1 + \dots + FP_n$ is the sum of false positives for each of n classes. Another approach is the *weighted* one, where the macro-averaging is calculated with an average weighted by the class support. This can change the F1-Score and shift its value outside the precision-recall range. The last approach is the *sample* one, where the metric is calculated for each instance and then averaged over them providing a balanced metric for the MLC problem. The results and the discussion have been articulated in terms of *sample*-average metrics. One way of comparing the various models, with one of the previous approaches, is on the validation set over the training. In this way, it is easy to identify the model with the best (or worst) overall performance. However, this information is relatively important since some applications have interests in understanding the performances in different classes and possibly choose a model (or Loss Function) based on that. Having a relatively large number of classes, it is challenging to visualize the performances for each class of each model. A different case in which the F1-Score helps to understand the behaviour of a model is by plotting the performance of the model on the validation set, during the training process. This allows understanding how fast a model can learn and therefore achieve a specific performance. Some applications have strict requirements in terms of duration of the training phases and having a Loss Function that allows a model to achieve much higher performance (F1-Score in our case) with the same number of epochs, is of great advantage.

Deep Neural Networks generally have high predictive accuracy but the results are usually not easily understandable by a human. Having the ability to interpret the results of a model helps to gain trust and define some traits from the heatmap contributions. [31] Explanations of Deep Learning models are relative and they make a significant difference when explaining a (correct or incorrect) prediction, visualizing what a model “sees” about a true class or relatively to another class choice. Methods like the Layer-wise Relevance Propagation address these topics. Models trained with different Loss Functions can express different properties and showcase different behaviour within the contribution heatmaps. LRP works by propagating the prediction $f(x)$ backwards in the neural network, subject to a conservation property intrinsic of the Neural Network architecture, where the input of a specific

neuron is redistributed to the lower layer in equal amounts. [33] Fig. 5.1 shows the overall LRP procedure, propagating the relevance score backwards in the Neural Network to the inputs. Defining two neurons j and k of consecutive layers, the propagation of relevance scores $((R_k)_k)$ is defined by the rule:

$$R_j = \sum_k \frac{z_{jk}}{\sum_j z_{jk}} R_k \quad (5.6)$$

where the quantity z_{jk} defines the contribution from neuron j towards neuron k . The relevance score R_k is expressed as a function of the lower-level activations on which are performed first-order Taylor expansions for specific reference points in the space of activations. [33]

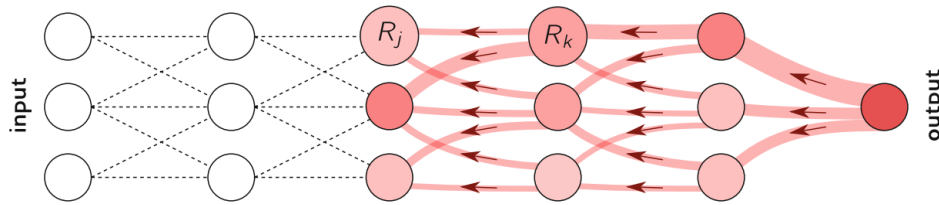


Figure 5.1: LRP procedure showing the redistribution of the relevance scores from each neuron to the lower layer.

Image source [33]

The LRP heatmapping technique has been used to compare the following features:

- Behaviour with imbalanced data. Areas of heatmaps, of models addressing this problem, with less represented classes have different degrees of contributions than models using traditional losses
- Outlier awareness. Loss Functions that deals with outliers should have lower (or higher) contributions on instance sections that are semantically different from the class descriptors and possibly have better predictions.
- Convexity and differentiability. Loss Functions that are convex and differentiable are generally smoother with a well-behaved surface. The contribution profile of these models should be more coherent with the semantic content of the instance and smoother with regard to the number of areas with positive and negative contributions.
- Learning efficiency. Showing the evolution of the LRP heatmaps over the training showcase how quickly, different Loss Functions, can correlate the semantic content of the input to a class.

From an empirical perspective, having the LRP contributions of a model helps to define what a model is actually learning. In the context of loss choice, having an overview of which losses leads to models with more meaningful predictions is of a great contribution.

6 Experimental Results

Several experiments have been carried out in order to assess the performances and behaviour of the different Loss Functions in various settings. The qualities analyzed are the ones identified for the different losses; class imbalance awareness, outlier awareness, convexity and differentiability, and the learning efficiency. The experiments assessing the class imbalance awareness aims to showcase the ability to recognise samples of minority classes in imbalanced datasets. The identification of these classes can be evaluated in terms of actual performances (F1-Score) over the various classes but also visually, with LRP heatmaps with respect to specific classes. Similarly, the experiments testing a model's ability to deal with outliers can be visualized in terms of LRP heatmaps. Analyzing the performances in an analytical way was not possible due to the dataset arrangement and therefore was neglected. The analysis of Loss Functions which are convex and differentiable is mainly a theoretical topic. However, it is possible to showcase the better behaviour of the losses with these properties in terms of a better accuracy between the semantic content of the image and the pixel-wise LRP contributions. The learning efficiency of a Loss Function has been evaluated in two aspects. In analytical terms by presenting the F1-Score over the validation set at training time and in empirical terms, presenting the evolution of the LRP contributions over the training phase.

6.1 Comparison of the Classification Performance of Loss Functions

The overall multi-label classification performance in terms of F1-Score over the test set is summarized in Table 6.1, using the samples averaging approach. In Tab. 6.2, Tab. 6.3 and Tab. 6.4 are shown the analytical performances on different metrics of all the Loss Functions over all the BigEarthNet classes with different averaging techniques. It is worth noting that, for a limited set of applications, the objective relies on the maximization of either the Precision or the Recall. However, these metrics happen to be in tension, improving one will lead to lower results in the other one. In order to comprehensively evaluate the performance of a model is important to consider the F1-Score since summarizes both metrics. From a general perspective, the models trained with the SparseMax Loss achieves significantly better performances, both from an analytical and empirical point of view. A significant improvement is also provided by the weighting approach of the Cross-Entropy Loss, sensibly increasing the recall and therefore the proportion of actual positives correctly identified. Tab. 6.2 and Tab. 6.3 shows that Cross-Entropy-based Losses, the Hamming Loss and the Huber Loss

6.1 Comparison of the Classification Performance of Loss Functions

have a more skewed distribution towards the precision metric. Instead, the Ranking Loss shows a strongly skewed distribution towards the Recall despite failing to identify several classes, such as “Agro-forestry areas”, “Beaches, dunes, sands”, “Costal wetlands” and so on. A likeable feature of the SparseMax Loss is that has a more balanced behaviour between precision and recall. It’s worth noting that the overall best performing loss from an analytical perspective is the SparseMax Loss, due to the marked gap between the other Loss Functions in terms of F1-Score under every averaging method. There are very few “class” cases where the Huber Loss and Weighted Cross-Entropy Loss match the SparseMax Loss F1-score performances. More specifically it’s worth noting the Weighted Cross-Entropy Loss achieving an F1-Score of 0.66 on “Broad-leaved forest”, 0.93 on “Marine waters” and 0.34 on “Natural grassland” like the SparseMax Loss. Likewise, the Huber Loss meets the same SparseMax Loss F1-Score of 0.79 on “Coniferous forest”. It is worth noting the good performance overall of the Hamming Loss, which is close to the performances achieved by the SparseMax Loss but lacks in minority classes such as “Industrial or commercial units” with 0.13 or “Permanent crops” with 0.02. This suggests that, possibly with a good weighting technique, this Loss Function might yield very good performances on the majority of the classes. Another relevant aspect is that most of the Loss Functions have similar performances on the most represented and easier classes, while the significant gaps are among difficult classes. An example of that is for the “Permanent crops”, which is barely recognized by the Cross-Entropy, Focal and Hamming Losses (0.03, 0.02, 0.01 F1-Scores), while the SparseMax Loss achieves a considerably better performance. (0.39)

Table 6.1: Overall classification accuracies on the test set of the BigEarthNet archive

	CEL	FL	HAL	HL	SML	RL	W-CEL
P_{micro}	0.75	0.72	0.74	0.76	0.66	0.52	0.74
P_{macro}	0.69	0.70	0.70	0.70	0.61	0.35	0.70
P_{sample}	0.75	0.72	0.76	0.77	0.71	0.58	0.76
R_{micro}	0.51	0.51	0.54	0.53	0.70	0.73	0.57
R_{macro}	0.35	0.34	0.38	0.38	0.53	0.48	0.40
R_{sample}	0.58	0.58	0.60	0.60	0.74	0.77	0.64
$F1_{micro}$	0.61	0.60	0.62	0.63	0.68	0.61	0.65
$F1_{macro}$	0.42	0.40	0.44	0.45	0.54	0.40	0.46
$F1_{sample}$	0.62	0.61	0.64	0.64	0.70	0.63	0.66

In Tab. 6.5 and Tab. 6.6 are shown the different contribution heatmaps for every Loss Function with their predictions. We can start noting strongly distinct behaviours, with models being sensitive to different areas of the input image. They have different intensities and patterns, perceiving different parts of the input image as important. The Cross-Entropy and Hamming Loss have a strong perception of relevant areas, which could saturate other classes that might be more accurate. It’s also notable, the reduced and less marked LRP contribution in the Focal and Weighted Cross-Entropy Loss which effectively are less prone to be biased to the majority or specific classes. The Huber Loss

6 Experimental Results

shows a smooth profile, coherent with the content of the image, while the Ranking Loss has a more irregular shape and the SparseMax Loss shows an adequate amounts of intensity from the input.

Table 6.2: Class-based sample-averaged Precision on the test set of BigEarthNet

Label	Precision						
	CEL	FL	HAL	HL	SML	RL	W-CEL
Agro-forestry areas	0,94	0,97	0,82	0,89	0,78	0,00	0,76
Arable land	0,85	0,88	0,86	0,90	0,76	0,67	0,79
Beaches, dunes, sands	0,66	0,82	0,69	0,69	0,63	0,00	0,73
Broad-leaved forest	0,68	0,60	0,67	0,67	0,55	0,48	0,63
Complex cultivation patterns	0,67	0,59	0,74	0,68	0,57	0,47	0,71
Coniferous forest	0,77	0,81	0,75	0,80	0,79	0,59	0,85
Costal wetlands	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Industrial or commercial units	0,63	0,68	0,70	0,75	0,52	0,00	0,71
Inland waters	0,90	0,91	0,91	0,94	0,85	0,66	0,92
Inland wetlands	0,56	0,70	0,55	0,62	0,54	0,25	0,68
Land principally occupied by agriculture..	0,65	0,57	0,71	0,68	0,56	0,48	0,71
Marine waters	0,89	0,91	0,90	0,90	0,87	0,62	0,94
Mixed forest	0,75	0,77	0,73	0,79	0,72	0,59	0,76
Moors, heathland and sclerophyllous..	0,58	0,80	0,77	0,58	0,53	0,00	0,80
Natural grassland and sparsely vegetated..	0,57	0,37	0,40	0,51	0,30	0,00	0,37
Pastures	0,87	0,91	0,82	0,84	0,79	0,58	0,83
Permanent crops	0,72	0,67	0,84	0,70	0,56	0,36	0,63
Transitional woodland/shrub	0,58	0,59	0,56	0,59	0,52	0,39	0,58
Urban fabric	0,85	0,79	0,89	0,81	0,70	0,46	0,85

6.1 Comparison of the Classification Performance of Loss Functions

Table 6.3: Class-based sample-averaged Recall on the test set of BigEarthNet

Label	Recall						
	CEL	FL	HAL	HL	SML	RL	W-CEL
Agro-forestry areas	0,14	0,02	0,34	0,32	0,53	0,00	0,34
Arable land	0,64	0,56	0,62	0,55	0,80	0,80	0,76
Beaches, dunes, sands	0,28	0,19	0,28	0,23	0,30	0,00	0,19
Broad-leaved forest	0,51	0,64	0,57	0,62	0,80	0,80	0,69
Complex cultivation patterns	0,29	0,44	0,19	0,37	0,66	0,61	0,34
Coniferous forest	0,78	0,71	0,83	0,79	0,79	0,91	0,70
Costal wetlands	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Industrial or commercial units	0,04	0,04	0,07	0,03	0,22	0,00	0,07
Inland waters	0,44	0,36	0,45	0,41	0,51	0,55	0,41
Inland wetlands	0,15	0,12	0,19	0,18	0,25	0,36	0,11
Land principally occupied by agriculture..	0,34	0,47	0,28	0,33	0,67	0,60	0,37
Marine waters	0,93	0,91	0,93	0,93	0,95	0,96	0,91
Mixed forest	0,63	0,61	0,75	0,61	0,77	0,90	0,71
Moors, heathland and sclerophyllous..	0,01	0,00	0,01	0,03	0,17	0,00	0,01
Natural grassland and sparsely vegetated..	0,11	0,20	0,24	0,17	0,39	0,00	0,31
Pastures	0,40	0,34	0,46	0,45	0,51	0,57	0,46
Permanent crops	0,01	0,01	0,01	0,06	0,30	0,33	0,07
Transitional woodland/shrub	0,56	0,49	0,61	0,60	0,78	0,94	0,65
Urban fabric	0,39	0,42	0,36	0,47	0,62	0,76	0,46


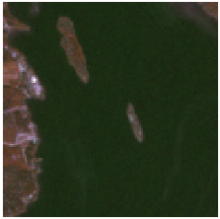
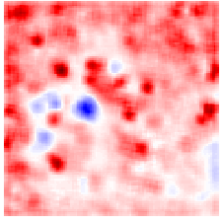
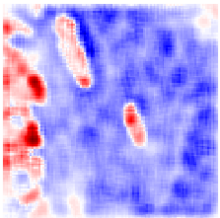
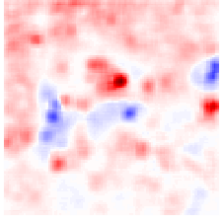
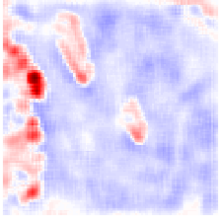
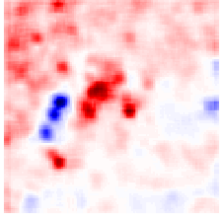
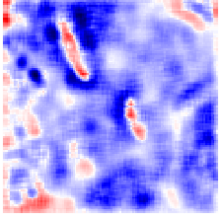
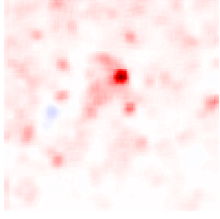
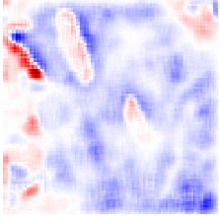
6 Experimental Results

Table 6.4: Class-based classification accuracies in F1-Score sample-averaged on the test set of BigEarthNet

Label	F1-Score						
	CEL	FL	HAL	HL	SML	RL	W-CEL
Agro-forestry areas	0,24	0,05	0,48	0,47	0,63	0,00	0,47
Arable land	0,73	0,69	0,72	0,68	0,78	0,73	0,77
Beaches, dunes, sands	0,39	0,31	0,39	0,35	0,41	0,00	0,30
Broad-leaved forest	0,58	0,62	0,62	0,64	0,65	0,60	0,66
Complex cultivation patterns	0,40	0,51	0,30	0,48	0,61	0,53	0,46
Coniferous forest	0,77	0,76	0,78	0,79	0,79	0,72	0,77
Costal wetlands	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Industrial or commercial units	0,07	0,07	0,13	0,06	0,31	0,00	0,13
Inland waters	0,59	0,52	0,60	0,57	0,64	0,60	0,57
Inland wetlands	0,24	0,20	0,28	0,27	0,34	0,30	0,20
Land principally occupied by agriculture..	0,44	0,52	0,40	0,45	0,61	0,53	0,49
Marine waters	0,91	0,91	0,92	0,92	0,91	0,75	0,93
Mixed forest	0,69	0,68	0,74	0,69	0,75	0,71	0,73
Moors, heathland and sclerophyllous..	0,01	0,00	0,01	0,06	0,25	0,00	0,01
Natural grassland and sparsely vegetated..	0,19	0,26	0,30	0,26	0,34	0,00	0,34
Pastures	0,55	0,49	0,59	0,59	0,62	0,58	0,59
Permanent crops	0,03	0,02	0,01	0,12	0,39	0,35	0,13
Transitional woodland/shrub	0,57	0,54	0,58	0,60	0,63	0,55	0,61
Urban fabric	0,54	0,55	0,51	0,60	0,66	0,58	0,59

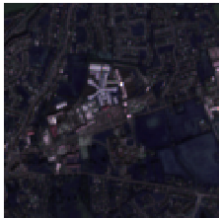
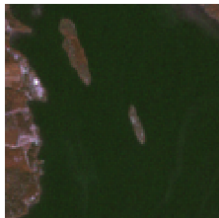
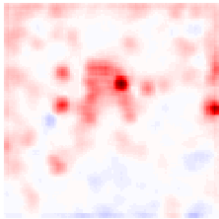
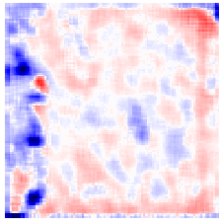
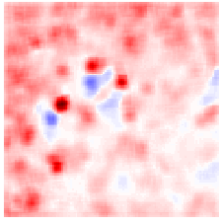
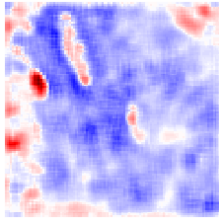
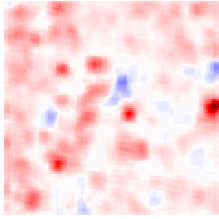
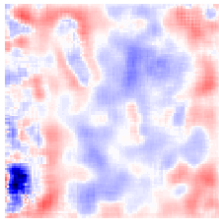
6.1 Comparison of the Classification Performance of Loss Functions

Table 6.5: An example of BigEarthNet patches, their multi-labels and LRP heatmaps for the considered loss functions.

Loss	a)	b)
Input	 <p><i>Industrial or commercial units, Pastures and Urban fabric</i></p>	 <p><i>Coniferous forest, Inland waters and Mixed forest</i></p>
CE	 <p><i>Urban fabric</i></p>	 <p><i>Coniferous forest, Inland waters and Mixed forest</i></p>
FL	 <p><i>Industrial or commercial units, Urban Fabric</i></p>	 <p><i>Coniferous forest, Inland waters and Mixed forest</i></p>
HAL	 <p><i>Pastures and Urban Fabric</i></p>	 <p><i>Coniferous forest, Inland waters and Mixed forest</i></p>
HL	 <p><i>Industrial or commercial units, Pastures and Urban Fabric</i></p>	 <p><i>Coniferous forest, Inland waters and Mixed forest</i></p>

6 Experimental Results

Table 6.6: An example of BigEarthNet patches, their multi-labels and LRP heatmaps for the considered loss functions.

Loss	a)	b)
Input	 <p><i>Industrial or commercial units, Pastures and Urban fabric</i></p>	 <p><i>Coniferous forest, Inland waters and Mixed forest</i></p>
RL	 <p><i>Transitional woodland-shrub and Urban fabric</i></p>	 <p><i>Coniferous forest, Inland waters, Marine waters and Mixed forest</i></p>
SML	 <p><i>Industrial or commercial units and Urban fabric</i></p>	 <p><i>Coniferous forest, Inland waters and Mixed forest</i></p>
W-CE	 <p><i>Industrial or commercial units, Pastures and Urban fabric</i></p>	 <p><i>Coniferous forest and Mixed forest</i></p>

6.2 Evaluation of Loss Functions in the Context of Class Imbalance Awareness

In this subsection, has been analyzed how different loss functions perform with minority classes. In Tab. 6.4 are shown the F1-Score performances in numerical terms, while in Fig. 6.1 the performances achieved by every loss function over all the classes are plotted to perceive better the overall trend. As discussed previously, in Tab. 6.5 and Tab. 6.6 are shown the respective performances on precision and recall. We can observe the superior performances of the SparseMax Loss over all the classes, both in the most and in the less represented ones, outperforming Loss Functions that specifically address the *class imbalance* problem. Examples of that are, the “Moors, heathland and sclerophyllous” and “Industrial or commercial units” classes, where the SparseMax Loss has achieved respectively 0.25 and 0.31 of F1-Score, while the Focal Loss has achieved 0.01 and 0.07 of F1-Score. Cost-sensitive Loss Functions, such as the Focal Loss and the Weighted Cross-Entropy Loss, generally show greater performances. As an example, in 6.4, we can observe that in the class “Natural grassland and sparsely vegetated area” the Focal Loss with 0.26 F1-Score and Weighted Cross-Entropy Loss with 0.34 F1-Score achieve better F1-Scores compared to the traditional Cross-Entropy Loss that produce a 0.19 F1-Score. We find a similar outcome with the “Industrial or commercial units” class, with the Weighted Cross-Entropy Loss achieving higher F1-Score, from 0.07 to 0.13. However, Focal Loss shows no improvements in this class, resulting in a 0.07 F1-Score as the Cross-Entropy Loss. Another significant difference is in the “Agro-forestry areas”, where the Cross-Entropy Loss scores the best precision (0.94) of any model, however, due to the very low recall (0.14), the resulting F1-Score is 0.24. Within this class, the SparseMax Loss has a slightly lower precision 0.78, however, due to the higher recall (0.53) results in a much higher F1-Score. (0.63) None of the losses has been able to detect “Coastal wetlands” on the test set due to its very low support, approximately ≈ 1500 samples in the whole dataset. The fine-tuning of the loss parameters has been avoided, which would have supported to achieve a more complete prediction profile and better performances. However, this was not the objective of this study, and doing so, would have biased the outcome and possibly also the loss surface landscape. Highlighting the behaviour of the loss function using an objective framework helps to have a more detailed understanding of the differences among them.

The Loss Functions that specifically address the class imbalance perform consistently better than the ones that do not. This is not only quantifiable in terms of increased F1-Scores as observed in Tab. 6.4, but better observed in the LRP heatmaps. As an example, Tab. 6.7 shows the differences in contributions among the different approaches for the predicted classes. The Cross-Entropy model has very strong contributions to the “Urban Fabric” class. The class “Industrial or commercial units” is missed in the prediction and has negative contributions to the objects in the instance that are associate to the class. In particular, the bottom left section includes a bright white industrial roof, and it’s negatively correlated with the “Industrial or commercial units” class. The section of the image which includes the river (“Inland waters”) is misclassified by the Cross-Entropy Loss, providing for

6 Experimental Results

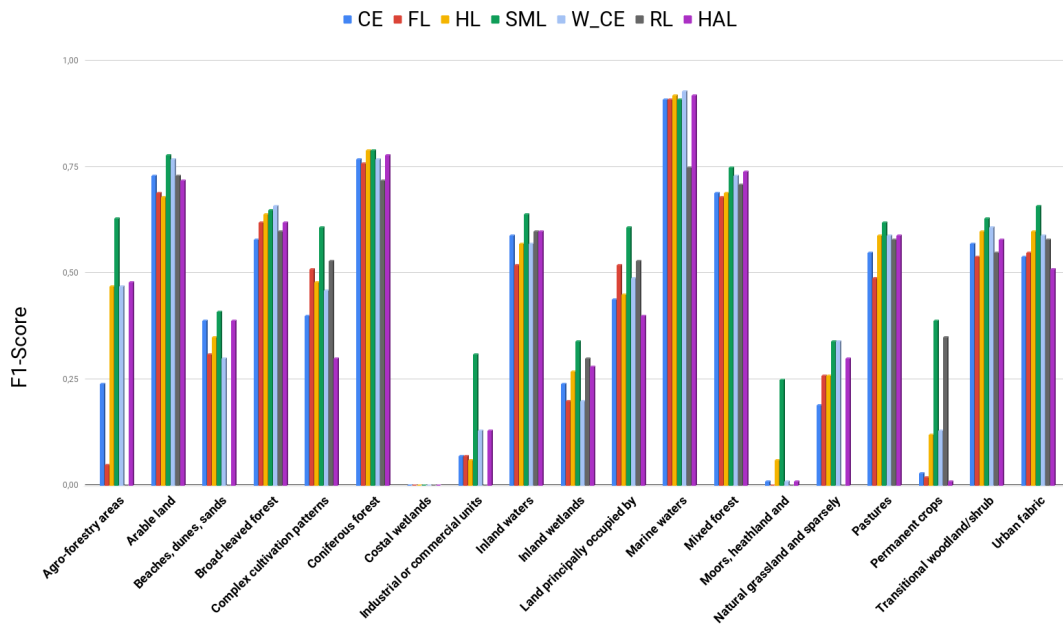


Figure 6.1: Class-based accuracies in F1-score on the test set of the BigEarthNet archive

that region a negative contribution to the related class and positively relating it to the "Urban Fabric" class. However, the model based on the Focal Loss has a greater behaviour, identifying the "Urban Fabric" class with the left section and the minority class ("Industrial or commercial units") with the right area, concurrently with the brighter roofs of the buildings which we can associate to the class. Another aspect worth noting is the balanced and homogeneous contribution of these sections as opposed to the Cross-Entropy Loss. The bottom-right section includes a river that the Focal Loss identifies successfully as the "Inland waters" class. The Weighted Cross-Entropy, has comparable contributions to the standard Cross-Entropy, with differences limited to the magnitude of the contribution and not to the correlation with the semantic content. The objects and regions of the instance are perceived in a very similar way to the standard Cross-Entropy Loss, while the Focal Loss can detect and have positive or negative contributions of semantically different areas. Another sample which highlights the difference in the evaluation of the various classes is shown in Tab. 6.8. The Cross-Entropy Loss correctly associates the "Urban Fabric" label to the left section of the instance, however, misses the prediction for "Industrial or commercial units" class that is wrongly associated with the top edges of the instance. The strong positive contributions of the top-corners of the instance are linked to semantically different regions, showing poor correlation of the prediction with the actual content. The Focal Loss has a more balanced contribution for the "Urban fabric" and correctly identifies the "Industrial or commercial units", relating it to the area with white roofs in the bottom and upper part. The class "Arable Land" is not predicted by the Cross-Entropy Loss and neither by the Focal Loss, but in the latter, we can see it associated with sections of land with patterns similar to the class. It's worth noting that unlike the first examples, the Weighted-Cross Entropy has a

6.2 Evaluation of Loss Functions in the Context of Class Imbalance Awareness

similar contribution heatmap to the Focal Loss in terms of adherence to the content of the input and intensities of the contributions. However, it predicts wrongly the "Arable Land" class in a section relatable to the "Industrial or commercial units" class. The detection of minority classes is consistent throughout the dataset and even if it's not significant in terms of F1-Score, the contrast is greatly marked in terms of LRP contributions. The Focal Loss and Weighted Cross-Entropy Loss have a more precise localization of the less represented classes compared to the other losses. It's also expected an improved performance with the usage of weighting techniques or with a careful choice of the parameters γ on the Focal Loss.

6 Experimental Results

Table 6.7: LRP heatmaps comparing the detection of the minority class "Industrial or commercial units" and "Inland waters"

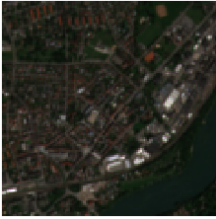
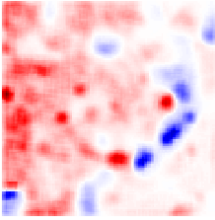
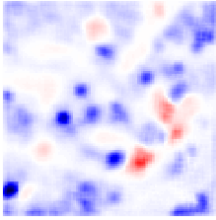
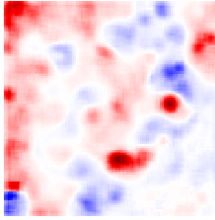
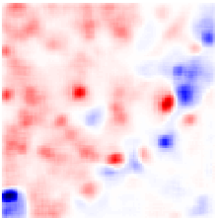
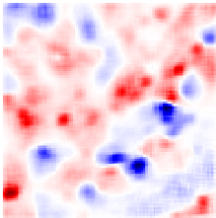
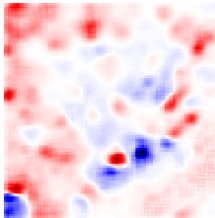
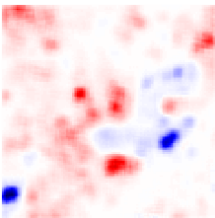
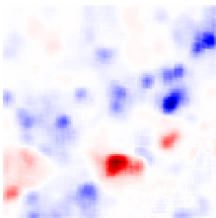
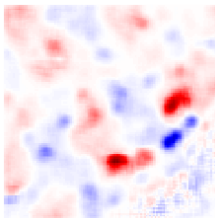
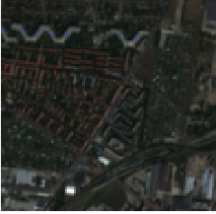
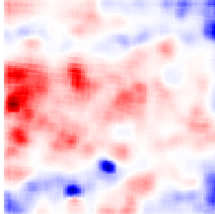
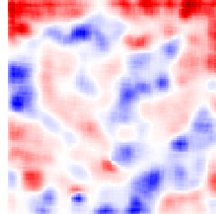
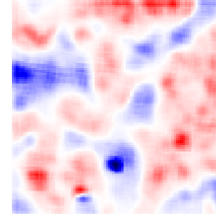
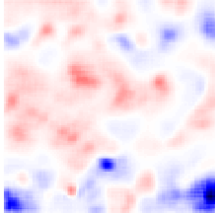
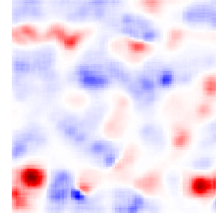
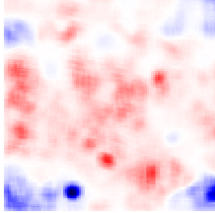
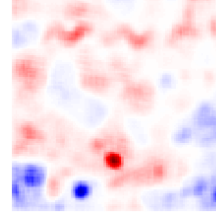
Input	 Industrial or commercial units, Inland waters, Urban fabric		
Loss and prediction	Urban Fabric	Industrial or commercial units	Inland waters
CE <i>Urban Fabric</i>			
FL <i>Urban Fabric, Industrial or commercial units and Inland waters</i>			
W-CE <i>Urban fabric</i>			

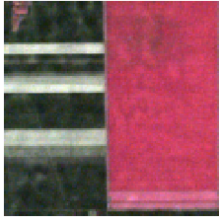
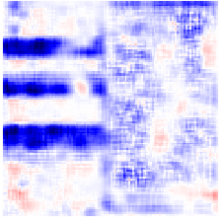
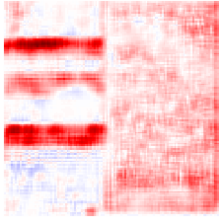

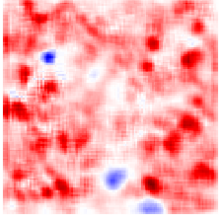
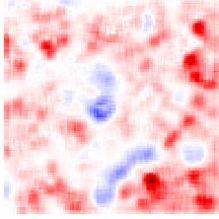
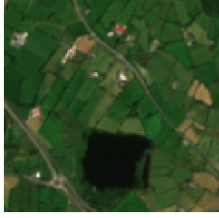
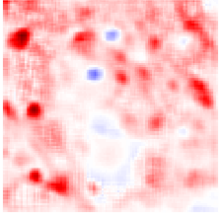
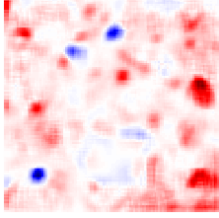
Table 6.8: LRP heatmaps comparing the detection of the minority class "Industrial or commercial units"

Input	 <p>Industrial or commercial units, Urban fabric</p>		
Loss and prediction	Urban Fabric	Industrial or commercial units	Arable Land
<p>CE <i>Urban Fabric</i></p>			
<p>FL <i>Urban Fabric, Industrial or commercial units</i></p>			
<p>W-CE <i>Urban fabric, Arable Land, Industrial or commercial units</i></p>			

6.3 Evaluation of Loss Functions in the Context of Outlier Awareness

Given the multivariate origin of outliers, it is a complex task having a comprehensive analysis of the behaviour of the models with different types of outliers. This subsection expands on the topic by selecting a set of outliers identified in the "BigEarthNet" dataset, showing the LRP heatmap with respect to a specific class and the predictions of these models. Other losses such as the SparseMax Loss have good performances on those samples, however, for visualization purposes, only the Cross-Entropy Loss, Focal Loss and the Huber Loss are shown. Having two semantically different regions in the same sample, considered as the same class, could be a difficult task to address with traditional approaches.

Table 6.9: An example of RS images, their multi-labels and LRP heatmaps for Cross-Entropy Loss and Huber Loss Functions.


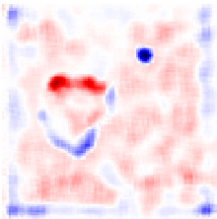
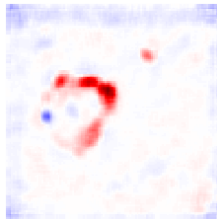
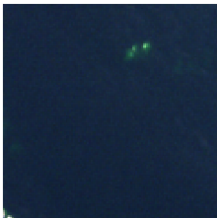
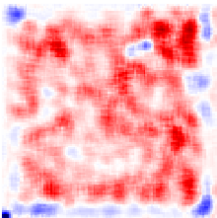
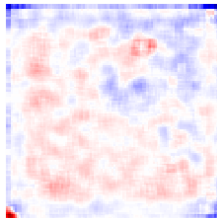
	Raw	CE	HL
a)			
	<i>Arable land</i>	–	<i>Arable land</i>
b)			
	<i>Pastures</i>	<i>Pastures</i>	<i>Pastures</i>
c)			
	<i>Pastures</i>	<i>Pastures</i>	<i>Pastures</i>

In Tab. 6.9.a is shown a sample of the class "Arable land" which is misclassified by the Cross-Entropy Loss. The sample has two contrasting regions belonging to the same class, which the tra-

6.3 Evaluation of Loss Functions in the Context of Outlier Awareness

ditional approach consider both as negative contributions with regards to the "Arable land" class. The Huber Loss instead positively correlate both the different sections of the sample and successfully predicts the class. In Tab. 6.9.b is shown a sample labelled as "Pastures", however, there is a noticeable portion of the image which (probably) is a mining site. The Cross-Entropy successfully predicts the label with a strong contribution in most of the input image, however, fails to recognize as a negative area the zone of the mining site. Contrastly, the Huber Loss has positive contributions for the class "Pastures" and precisely delimit the mining site as a negative contribution to the class. Another example from the same sample, that shows the accuracy of the Huber Loss in addressing the contribution profile to the input content, is the bottom part of the Tab. 6.9.b which includes a negatively correlated area. In the input image, this area represents a field with a noticeable elevation gain, which is less likely to be correlated with the "Pastures" class. Similarly, in Tab. 6.9.c, the sample is labelled as "Pastures" however, there is a large water body within the lower section. Both approaches correctly identify the class with similar objects positively correlated to the label. Yet, the Cross-Entropy partially recognizes the water body as a positive contribution, with extremely low areas marked with negative contributions with regards to this class. The Huber Loss has a larger negative contribution of the water body concerning the "Pastures" class.

Table 6.10: Cross-Entropy Loss and Focal Loss LRP heatmaps for outliers

	Raw	CE	FL
a)			
	<i>Marine Waters</i>	<i>Marine Waters</i>	<i>Marine Waters</i>
b)			
	<i>Marine Waters</i>	<i>Marine Waters</i>	<i>Marine Waters</i>

Although BigEarthNet is not heavily affected by outliers, the results confirm the properties yield by the Huber Loss in terms of efficacy on outliers. In contrast, we show how the behaviour of the Focal Loss compared to the traditional Cross-Entropy Loss. From a theoretical perspective is understandable a poor performance, since the Focal Loss directs the focus on a set of hard samples.

In Tab. 6.10.a is shown that the Cross-Entropy Loss has positive contributions around and concerning the island, with slight negative contributions in sparse sections of the sample. Oppositely, the Focal Loss draws most of its relevance from the island visual features and has a negative contribution in wide sections of the water bodies. In Tab. 6.10.b a similar example shows the drastic change of focus in the LRP contribution. The Cross-Entropy Loss correctly classifies the sample as *Marine Waters*, with some areas with negative contributions but overall a homogeneous profile. The Focal Loss instead has most of its attention toward a very small section of the sample in the bottom-left area while having negative contributions sparse across the area addressable as *Marine Waters*. This behaviour validates the underlying concept of the Focal Loss which focuses on a small set of hard examples. Having datasets with many outliers and/or very noisy samples should direct the Loss Function choice away from the Focal Loss and more towards to the Huber Loss.


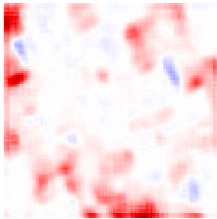
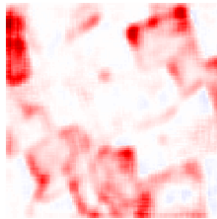

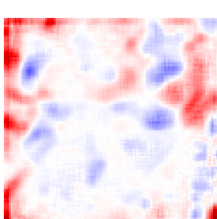
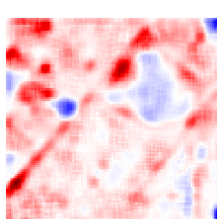
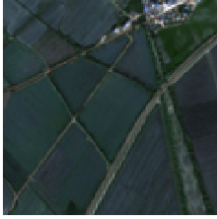
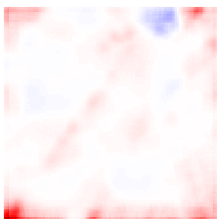
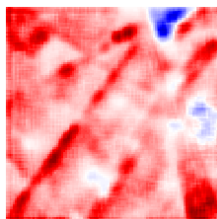
6.4 Evaluation of Loss Functions in the Context of Convexity and Differentiability

Although the analysis of the convexity and differentiability of the Loss Function is a theoretical aspect, it is also interesting to showcase the differences among them with this discriminant. The analysis is carried evaluating two aspects, the coherence of the LRP contribution with input regions of the same area and the consistency of the LRP contributions over the training epochs. Both aspects increase the reliability of the model, showing a better understanding (or learning) of the classes and better coherence through the training. Regarding the latter, having a model that consistently correctly classifies it's a crucial feature that introduces reliability in the predictions by design, regardless of the method for selecting a model during training. The generalization capabilities of a model are related to the geometry of the neighbourhood minimizers. [25] Having a surrounding landscape with a flat region of similar minimizers translates in a model that, regardless of the specific minima, will perform reliably and coherently in the proximity of the global minima. This also translates to a model that reaches a minimizer in a flat region with incremental gains. In Tab. 6.11 are shown examples that compare the coherence in the prediction of equivalent areas of the inputs. The Ranking Loss shows incongruities in several samples, while the SparseMax Loss (and other losses, such as the Huber Loss) show a more coherent LRP contribution. Tab 6.11 shows the LRP heatmaps concerning a specific class, allowing the understanding of which sections of the image compose the model prediction for a specific class. In detail, within Tab 6.11.a we can identify 3 objects (top-left, top right and bottom-centre part) that we can associate with the "Arable Land" class. The Ranking Loss has within the top-left and top-right areas, negative contributions to sections of these objects. The SparseMax better delimits those areas with strong positive contributions. In Tab. 6.11.b there is a large area (specifically, with brown/yellow land cover) of land associated with the "Arable Land" class, while the rest of the sample associates the green fields to the "Pastures" class. Logically, the contribution profile for the "Pastures" class should delimit and negatively correlate this area. In those

regards, the Ranking Loss has a scattered negative contribution towards the “Pastures” class. Instead, the SparseMax Loss delimits negatively the sections in the centre and mid-left in a precise manner, coherently among the input area. In Tab. 6.11.c is shown a sample where the Ranking Loss has most of its contribution from a small bottom-left area, while the actual majority of the instance is an homogeneous “Arable Land” area. In contrast, the SparseMax Loss has strong uniform contributions towards these areas. In Tab. 6.12 and Tab. 6.13 we can show the consistency of the LRP contribution over the various training epochs. The main aspect to note on different losses is the change of LRP contribution, from negative to positive or vice-versa. Generally, is acceptable a change of profile in the contribution, especially in the first epochs. However, starts to become a negative feature when this behaviour is observed in the last epochs. This because, the neighbourhood of a minimizer is where there is a higher chance of model selection from one epoch to the other, due to the lowest loss or best F1-Score. The Huber Loss in Tab. 6.12 has a marked change of the contribution for the Marine Waters class, however, in the 60th epoch reaches a good contribution for the class. The Cross-Entropy Loss has initially a positive correlation with the left section of the instance, associated with “Marine Waters”. However, later in the training, this changes towards a negative contribution. The SparseMax Loss instead, has a stable and reliable contribution from the epoch 20. In Tab. 6.13 the Cross-Entropy Loss has a similar behaviour for the class “Beaches, dunes, sands”, alternating with positive and negative contributions between the epoch 20, 40 and 60. The Huber Loss show a coherent evolution of the contributions, that are mostly linked to the white foam of crashed waves. The SparseMax Loss has a strong positive contribution to the objects relatable with “Beaches, dunes, sands”. However, start showing a change in contributions (from positive to negative) in the left section of the instance associated with a water body, in the epoch 60.

6 Experimental Results

Table 6.11: LRP heatmaps showing different degrees of accuracy on different classes

	Raw	RL	SML
a)			
	Heatmaps for <i>Arable Land</i>		
b)			
	Heatmaps for <i>Pastures</i>		
c)			
	Heatmaps for <i>Arable Land</i>		

6.5 Evaluation of Loss Functions in the Context of Efficiency of the Learning Mechanism

This subchapter provides insight on which losses have a more *efficient* learning procedure. The analysis of this property has been done via a performance-wise comparison and an empirical comparison of the learned regions in LRP heatmaps. Examining which models have a better and faster learning process allow the investigation of the behaviour of the losses in terms of reaching a global minimizer. A characteristics worth noting is that, different training runs might have minor differences in terms of intermediate F1-Scores and also LRP heatmaps. Having different starting points and random shuffling drives the training in different directions, however, those results have been observed through multiple experiment runs. This result in models that will likely reach global minimizers using different optimization trajectories. In Fig. 6.2 are shown the F1-Score performances of the models trained over different loss functions over the training. Models that achieve better performances at earlier stages are highly desirable. It is noticeable the superior performance in the initial epochs of the SparseMax and Ranking Loss. Similar performances are shown by other models trained with other losses only after 20 to 30 epochs. In numerical terms, an F1-Score of 0.55% is achieved by the SparseMax Loss after 12 epochs, by the Ranking Loss after 17 epochs, while the Huber loss achieves it only after 35 epochs. This is opposite with respect to the theoretical properties of the Hu-

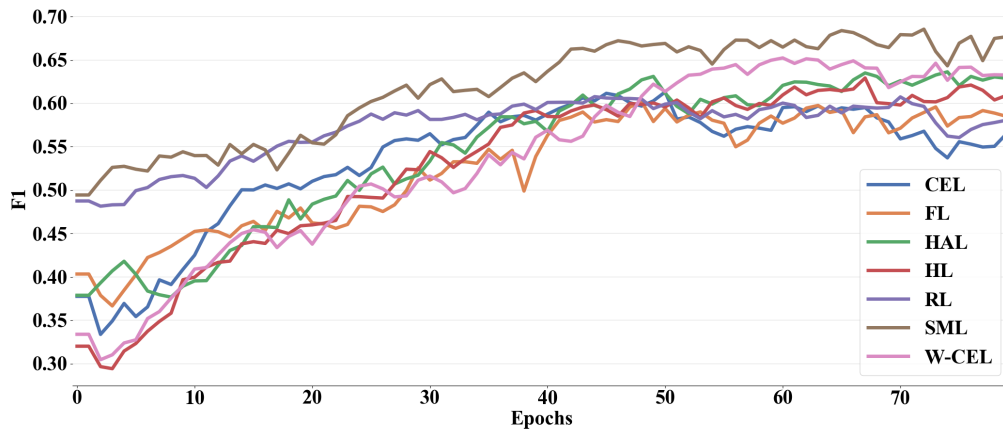


Figure 6.2: Overall classification accuracies in F-1 Score of the validation set of the BigEarthNet archive.

ber Loss, which thanks to the PL inequality, has a linear convergence. This behaviour can be possibly explained by the shallow structure of the CNN architecture. Deeper models, which could be more difficult to be trained, could exalt the performance-over-epoch gains of the Huber Loss compared to Loss Functions that do not guarantee linear convergence. By analyzing Fig. 6.2, it is also relevant that applying a weighting approach to the Cross-Entropy Loss, slows down the performance curve compared to the un-weighted Cross-Entropy Loss which is surpassed at a later stage. The Focal Loss

6 Experimental Results

and the Weighted Cross-Entropy Loss reach an F1-Score of 0.55% around the 40 epoch mark, a major difference compared to the other Loss Functions. This is understandable since these Loss Functions are harsher in their evaluation since the majority of the attention of the training is focused on hard samples and/or minority classes that might only affect lightly the F1-Score performance. Summarizing, the SparseMax Loss, Ranking Loss and Cross-Entropy Loss can deliver superior performances with an equal number of epochs.

A diversified approach to understand the learning efficiency of the different Loss Functions is to plot the LRP heatmaps during training. *Efficient* Loss Functions will produce models which have better-shaped heatmaps and semantically more adherent contributions, at earlier stages compared to “slower” Loss Functions. Displaying the evolution of the learned classes at different stages of the training phase supports the findings based on F1-Score performances over the training process. Using Fig. 6.3 as an input instance, we can understand how different the Loss Functions perform.



Figure 6.3: A BigEarthNet patch used for the comparison of LRP heatmaps

In Tab. 6.12 is shown the evolution of the contribution for the class Marine Waters during training. Due to visualization reasons, we have shown the evolution of the heatmaps every 20 epochs, yet a more detailed approach would be showcasing the development with a finer scale. (e.g. every 2-5 epochs) The Cross-Entropy Loss and the SparseMax Loss display a positive contribution of the left section, associated with the actual water body of the input image, from the 20th epoch, while the Huber Loss achieves it at a later stage. It's worth noting the consistent shape and intensity of the LRP contributions of the SparseMax Loss, as opposed to the marked changes of the LRP contributions of the Cross-Entropy Loss. There a strong positive contribution also of the border with beaches and sand areas, mostly because it's usual to find the class "Marine Waters" presence of these objects in the dataset. Similar behaviour is presented in Tab. 6.13 with the class "Beaches, dunes, sands". In this case, the SparseMax Loss shows the localization of the zone associated (central part to the central-bottom area) with the label from the 20th epoch. The positive contribution is marked and continuous, precisely describing the input image. Meanwhile, the Cross-Entropy Loss and the Huber Loss partially fail to identify these regions with scattered contributions in wrong areas, moreover showing negative contributions in the position that we would associate with the label. The Huber Loss, without considering the first epoch, has a smooth and consisted evolution. Instead, the Cross-Entropy Loss has an unreliable profile providing an alternating behaviour, switching from strongly positive to strongly negative LRP contributions. An example of that is the top-right corner, with an object relatable to the white foam of the crashing waves. The Cross-Entropy Loss has a strongly

6.5 Evaluation of Loss Functions in the Context of Efficiency of the Learning Mechanism

negative contribution in the epoch 20, subsequently, in the epoch 40, it is considered as a strongly correlated area. At a later stage, there is a further scattered contribution. This alternating behaviour shows an unreliable component of the model, where the quality of the prediction is highly dependent on the model at a specific epoch. More reliable models such as the SparseMax Loss and Huber Loss, are preferred since the learned classes are consistent regardless of the method of model selection.

Table 6.12: Change of LRP heatmaps during training of Cross-Entropy, SparseMax and Huber Loss Functions for the *Marine Waters* class

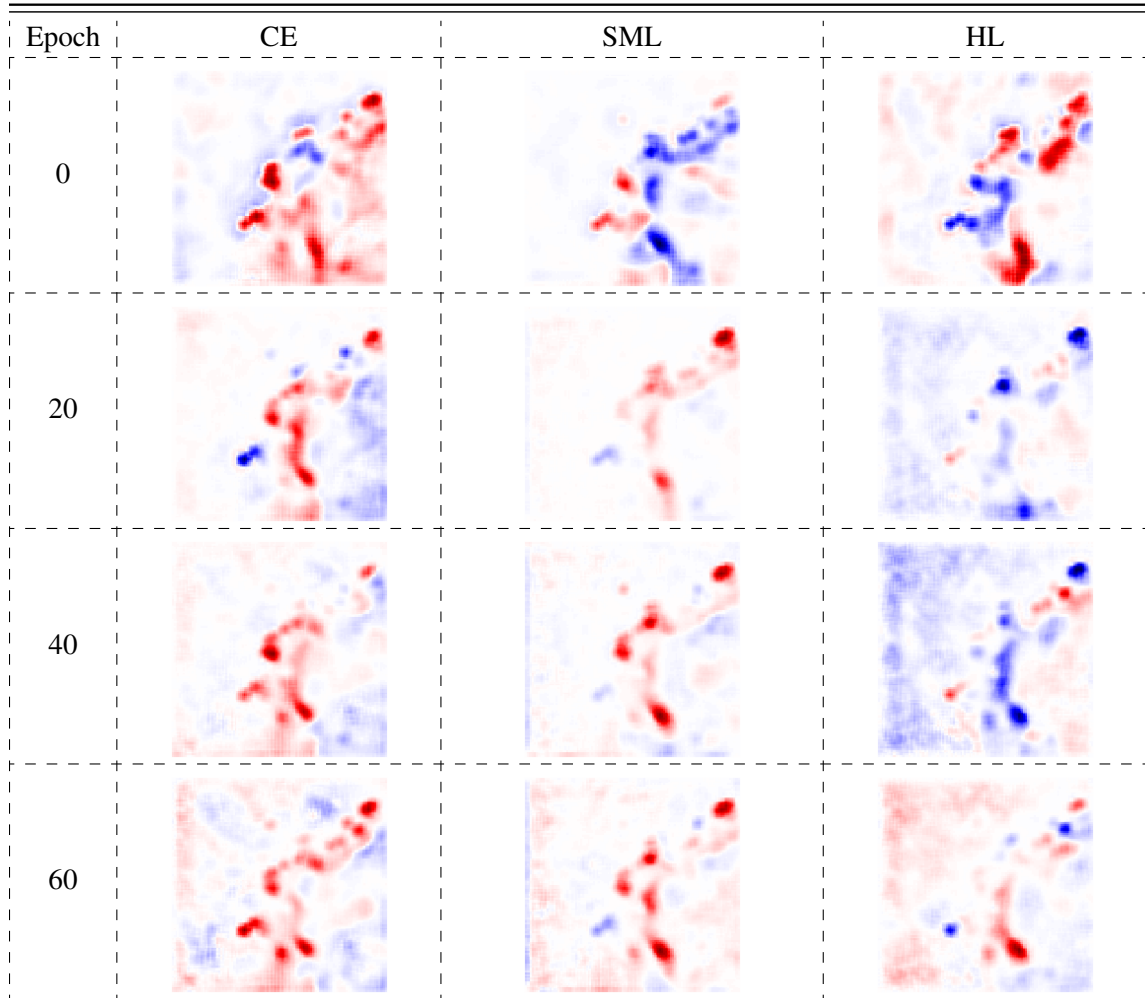
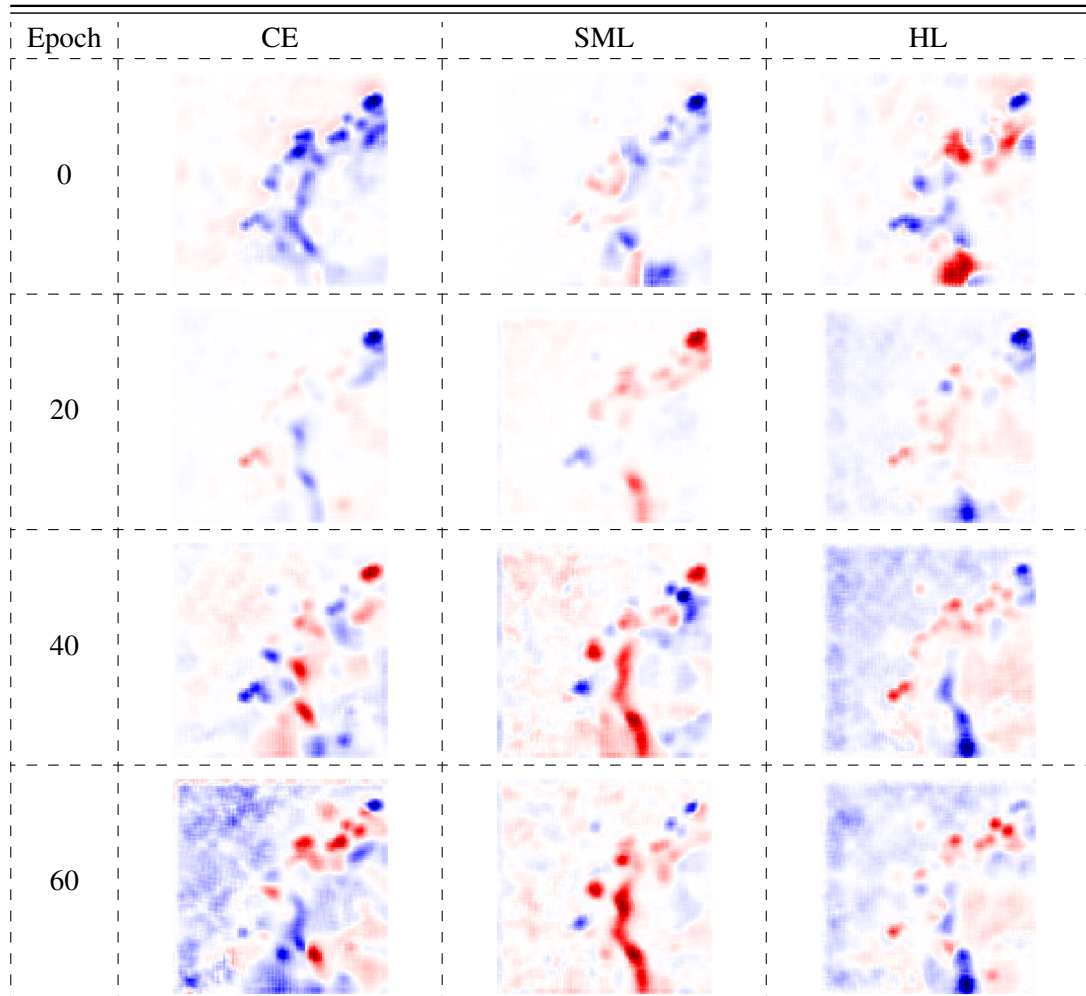


Table 6.13: Change of LRP heatmaps during training of Cross-Entropy, SparseMax and Huber Loss Functions for the *Beaches, dunes, sands* class



7 Conclusion and Discussion

The choice of Loss Function is a crucial factor in the design of Deep Learning models, especially guiding the training in a considerably more complex output space such as in the MLC. We have tested different Loss Functions on the BigEarthNet dataset to have a framework that challenges the models with real-world problems. The best performing Loss Function is the SparseMax Loss, both from a performance-wise perspective (in terms of F1-Score) and from an empirical perspective. (in terms of LRP heatmaps) The advantages of this Loss Function and its activation function are multiple, among them: the sparse output distribution which is closely related with the multi-label objective, holding of a separation margin and its convex and differentiable properties. These set of properties are highly desirable, both from a theoretical perspective and an operational outlook. (e.g. extreme multi-label classification) The high performances compared to the Cross-Entropy and the more meaningful heatmaps confirm these findings, which has never been explored in the Remote Sensing realm. The SparseMax Loss does not have a differentiated behaviour for minority classes, however, has outstanding performances also on this set of samples. The Focal Loss which focuses on hard samples has an improved performance on less represented classes, showing a more meaningful heatmap profile as opposed to the Weighted Cross-Entropy. Directing the training on instances with low support can include samples from minority classes. However, there is not enough correlation between the "hardness" of a sample and its class distribution in the dataset. The negative consequence of this approach would be diverging the focus of the training on a set of difficult samples. (e.g. outliers, wrongly labelled data) The Weighted Cross-Entropy Loss avoids this problem by explicitly setting class weights. The drawback of this approach is that scales the range of the loss from a numerical perspective. Optimizers that rely on the range of the gradient might (e.g. Stochastic Gradient Descent) not be usable, posing a strong constraint on the implementation. Performance-wise the Weighted Cross-Entropy Loss might seem a better choice for handling minority classes, however, the heatmap analysis has shown unreliable results and a more precise outcome is obtained with the Focal Loss. It's worth noting that the two approaches can be combined and possibly achieving remarkable performances, also with a finer choice of class weights and loss parameters.

Models that require the training on datasets with a large number of outliers are suggested to choose the Huber Loss. The LRP heatmap has shown that models trained with traditional Cross-Entropy Loss do not have a clear detection of sections of the instance which might be unrelated to the label associated. Also, the Focal Loss directs the focus on a set of *hard* samples, therefore is specifically not suitable on datasets with a significant presence of outliers. On the contrary, the Huber Loss has excellent behaviour, precisely detecting the classes associated with the semantic content of the image

7 Conclusion and Discussion

and providing smoother contribution heatmaps. However, due to the various nature of the potential outliers, is not possible to generalize this feature to similar situations without further analysis. This emphasizes the importance of evaluating models not only in terms of performance-wise metrics but also with a more pragmatic approach, on what the models are learning. Loss functions that are convex and differentiable have shown better LRP heatmaps, resulting in more coherent and accurate output contributions to the classes. Applications that require fast training procedures should use the SparseMax Loss. Besides superior performances, the experiments show that SparseMax Loss can deliver specific performances with a fewer number of epochs compared to the traditional Cross-Entropy. Contrarily to its theoretical properties, the Huber Loss has shown a “slow” behaviour compared to the other losses. This behaviour can be explained as its linear convergence rate could be better appreciated in more complex and deeper architectures. It’s possible to improve the learning efficiency of the training procedures with a careful choice of optimizers and parameters, however, having a Loss Function that provides a higher baseline for faster training is a desirable choice. Concluding, this work has shown that the choice of Loss Functions is not a trivial task, touching several components of the problem. The performance metrics are not the only aspect to evaluate and equally important is the understanding of the behaviour of the model. Addressing the right training procedure and understanding what Neural Networks are learning is one step forward in building reliable and trustable AI systems.

In terms of future work, there are several directions in which this could evolve. Personally, the most interesting topic is the visualization of the different Loss Function landscapes. Performing these set of experiments would give an understanding of the Loss Function, not only for regions in proximity of the optimizer path but also in the surroundings. Another branch of studies would be related to the behaviour of these losses with more complex architectures, such as VGGs and ResNets. Analyzing the behaviour of these models, their performances, it’s trainability and contribution heatmaps; is of great interest. Lastly, another relevant future study would explore the improvements of having an ensemble approach, with multiple losses which ideally would be joint for complementary features.

Bibliography

- [1] Salem Al-amri, Namdeo Kalyankar, and Khamitkar Santosh. “A Comparative Study of Removal Noise from Remote Sensing Image”. In: *International Journal of Computer Science Issues* 7 (2010).
- [2] Maximilian Alber et al. *iNNvestigate neural networks!* 2018. arXiv: 1808.04260.
- [3] Amit Alfassy et al. *LaSO: Label-Set Operations networks for multi-label few-shot learning*. 2019. arXiv: 1902.09811.
- [4] Francis Bach, Benjamin Goehry, and Antoine Havet-Morel. “Statistical Machine Learning and Convex Optimization Lecture 1 - February 18th”. In: 2016.
- [5] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 2007.
- [6] Léon Bottou and Olivier Bousquet. “The Tradeoffs of Large Scale Learning”. In: *Advances in Neural Information Processing Systems*. 2008, pp. 161–168.
- [7] Cristiano L Castro and Antônio P Braga. “Novel Cost-Sensitive Approach to Improve the Multilayer Perceptron Performance on Imbalanced Data”. In: *IEEE Transactions on Neural Networks and Learning Systems* 24.6 (2013), pp. 888–899.
- [8] Yin Cui et al. “Class-balanced loss based on effective number of samples”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9268–9277.
- [9] Krzysztof Dembczyński et al. “On label dependence and loss minimization in multi-label classification”. In: *Machine Learning* (2012), pp. 5–45.
- [10] W. Edwards Deming. “Out of the Crisis”. In: The MIT Press, 2000.
- [11] Sebastien Destercke. “Multilabel Prediction with Probability Sets: The Hamming Loss Case”. In: *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*. Springer. 2014, pp. 496–505.
- [12] Feng Kang, Rong Jin, and R. Sukthankar. “Correlated Label Propagation with Application to Multi-label Learning”. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2006, pp. 1719–1726. DOI: 10.1109/CVPR.2006.90.
- [13] Eibe Frank and Mark Hall. “A Simple Approach to Ordinal Classification”. In: *Proceedings of the 12th European Conference on Machine Learning*. Berlin, Heidelberg: Springer-Verlag, 2001, pp. 145–156.

Bibliography

- [14] Wei Gao and Zhi-Hua Zhou. “On the Consistency of Multi-Label Learning”. In: *Artificial Intelligence* 199-200 (2013), pp. 22–44. DOI: 10.1016/j.artint.2013.03.001.
- [15] Yunchao Gong et al. *Deep Convolutional Ranking for Multilabel Image Annotation*. 2013.
- [16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, 2016. ISBN: 0262035618.
- [17] Ian Goodfellow, Oriol Vinyals, and Andrew M. Saxe. *Qualitatively characterizing neural network optimization problems*. 2014. arXiv: 1412.6544.
- [18] Geoffrey E Hinton et al. “The ”wake-sleep” algorithm for unsupervised neural networks”. In: (1995), pp. 1158–1161.
- [19] Yuansheng Hua, Lichao Mou, and Xiao Xiang Zhu. *Relation Network for Multi-label Aerial Image Classification*. 2019. arXiv: 1907.07274.
- [20] Peter J. Huber. “Robust Estimation of a Location Parameter”. In: *The Annals of Mathematical Statistics* (1964), pp. 73–101.
- [21] Himanshu Jain, Yashoteja Prabhu, and Manik Varma. “Extreme Multi-Label Loss Functions for Recommendation, Tagging, Ranking & Other Missing Label Applications”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 2016, pp. 935–944. DOI: 10.1145/2939672.2939756.
- [22] K. Karalas et al. “Land Classification Using Remotely Sensed Data: Going Multilabel”. In: *IEEE Transactions on Geoscience and Remote Sensing* 54.6 (2016), pp. 3548–3563. ISSN: 1558-0644. DOI: 10.1109/TGRS.2016.2520203.
- [23] Ashish Khetan, Zachary C. Lipton, and Anima Anandkumar. *Learning From Noisy Singly-labeled Data*. 2017. arXiv: 1712.04577.
- [24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Neural Information Processing Systems* (2012). DOI: 10.1145/3065386.
- [25] Hao Li et al. *Visualizing the Loss Landscape of Neural Nets*. 2017. arXiv: 1712.09913.
- [26] Yuncheng Li, Yale Song, and Jiebo Luo. *Improving Pairwise Ranking for Multi-label Image Classification*. 2017. arXiv: 1704.03135.
- [27] Tsung-Yi Lin et al. *Focal Loss for Dense Object Detection*. 2017. arXiv: 1708.02002.
- [28] Weiyang Liu et al. *Large-Margin Softmax Loss for Convolutional Neural Networks*. 2016. arXiv: 1612.02295.
- [29] Lei Ma et al. “Deep Learning in Remote Sensing applications: A meta-analysis and review”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* (2019), pp. 166–177.

- [30] Andre F. T. Martins and Ramon Fernandez Astudillo. *From Softmax to Sparsemax: A Sparse Model of Attention and Multi-Label Classification*. 2016. arXiv: 1602.02068.
- [31] Gregoire Montavon, Wojciech Samek, and Klaus-Robert Müller. “Methods for interpreting and understanding Deep Neural Networks”. In: *Digital Signal Processing* (2018), pp. 1–15. DOI: <https://doi.org/10.1016/j.dsp.2017.10.011>.
- [32] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.
- [33] Wojciech Samek et al. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. 2019. DOI: 10.1007/978-3-030-28954-6.
- [34] I. Shendryk et al. “Deep Learning - A new approach for Multi-Label Scene Classification in Planetscope and Sentinel-2 Imagery”. In: *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*. 2018, pp. 1116–1119. DOI: 10.1109/IGARSS.2018.8517499.
- [35] Daniel Soudry et al. *The Implicit Bias of Gradient Descent on Separable Data*. 2017. arXiv: 1710.10345.
- [36] Sargur Srihari. *Introduction to Machine Learning Course*. URL: <https://cedar.buffalo.edu/~srihari/CSE574>.
- [37] Gencer Sumbul and Begum Demir. “A Novel Multi-Attention Driven System for Multi-Label Remote Sensing Image Classification”. In: *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium* (2019). DOI: 10.1109/igarss.2019.8898188.
- [38] Gencer Sumbul et al. “BigEarthNet: A Large-Scale Benchmark Archive for Remote Sensing Image Understanding”. In: *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium* (2019), pp. 5901–5904.
- [39] Gencer Sumbul et al. *BigEarthNet Dataset with A New Class-Nomenclature for Remote Sensing Image Understanding*. 2020. arXiv: 2001.06372 [cs.CV].
- [40] Xia Sun et al. “Drug-Drug Interaction Extraction via Recurrent Hybrid Convolutional Neural Networks with an Improved Focal Loss”. In: *Molecular Diversity Preservation International - Entropy* (2019), p. 37. DOI: 10.3390/e21010037.
- [41] Giang Son Tran et al. “Improving Accuracy of Lung Nodule Classification Using Deep Learning with Focal Loss”. In: *Journal of Healthcare Engineering* (2019), pp. 1–9. DOI: 10.1155/2019/5156416.
- [42] Jiang Wang et al. *CNN-RNN: A Unified Framework for Multi-label Image Classification*. 2016. arXiv: 1604.04573.
- [43] Y. Wei et al. “HCP: A Flexible CNN Framework for Multi-Label Image Classification”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2016), pp. 1901–1907. DOI: 10.1109/TPAMI.2015.2491929.

Bibliography

- [44] Bayable Teshome Zegeye and Begüm Demir. “A Novel Active Learning Technique for Multilabel Remote Sensing Image Scene Classification”. In: *Image and Signal Processing for Remote Sensing XXIV*. 2018, pp. 100–107. DOI: 10.1117/12.2500191.
- [45] A. Zeggada, F. Melgani, and Y. Bazi. “A Deep Learning Approach to UAV Image Multilabeling”. In: *IEEE Geoscience and Remote Sensing Letters* (2017), pp. 694–698. DOI: 10.1109/LGRS.2017.2671922.
- [46] A. Zeggada et al. “Multilabel Conditional Random Field Classification for UAV Images”. In: *IEEE Geoscience and Remote Sensing Letters* (2018), pp. 399–403. DOI: 10.1109/LGRS.2018.2790426.
- [47] Min-Ling Zhang and Zhi-Hua Zhou. “A Review On Multi-Label Learning Algorithms”. In: *IEEE Transactions on Knowledge and Data Engineering* (2014), pp. 1819–1837. DOI: 10.1109/TKDE.2013.39.
- [48] Zhilu Zhang and Mert R. Sabuncu. *Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels*. 2018. arXiv: 1805.07836 [cs.LG].
- [49] Feng Zhu et al. “Learning spatial regularization with image-level supervisions for multi-label image classification”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 5513–5522.

Appendix