Classification of ordinal data in deep learning: an experimental study

Victor-Manuel Vargas, Pedro-Antonio Gutiérrez and César Hervás

***Abstract*—The abstract goes here.**

***Index Terms*—IEEE, IEEEtran, journal, LATEX, paper, template.**

I. Introduction

DEEP LEARNING, introduced by Yann Lecun [1], combines multiple Machine Learning techniques and allows computational models that are composed of numerous processing layers to learn representations of data with various levels of abstraction. These methods have dramatically improved the state-of-the-art in many domains, such as image classification [2], [3], [4], speech recognition [5], control problems [6], object detection [7], [8], privacy protection [9], recovery of human pose [10], semantic segmentation [11] and image retrieval [12]. Deep learning discovers complex structures in large datasets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the features in the previous layer.

Convolutional Neural Networks (CNN) are designed to process data that comes in the form of multiple arrays. CNNs are suited for images, video, speech and audio processing, and they have been used extensively in the last years for automatic classification tasks [13], [14], [15]. On image classification tasks, each colour channel is represented by a 2D array. In this case, convolutional layers extract the main features from the pixels of the images and, after that, a fully connected layer classify every sample based on its extracted features. At the output of the convolutional net, a softmax function provides the probabilities of the set of classes predefined in the model for classification tasks. These outputs are compared against the correct values.

The backpropagation algorithm is a stochastic gradient descent algorithm that minimises a predefined loss function. It has been used in the last years in many works [16], [17], [4], [18] for training shallow and deep neural networks. It updates the layer's parameters after backpropagating the loss function gradients through the network. Learning rate hyper-parameter controls the strength of the changes that are applied to those parameters. Some works have checked the importance of finding the optimal value for this parameter [19] and have tried different approaches to try to improve the training process [20].

Batch normalization is another technique that is used for this kind of networks. It reduces the internal covariate shift by normalizing layer inputs. It was presented in 2015 [21] and gives a critical enhancement to the training phase. Batch size is an important hyper-parameter that should be adjusted as it affects the layer's parameters updates and also the normalization [22][23].

Ordinal classification problems are those where labels are ordered, and there are different inter-classes weights for each pair of classes. This kind of problem can be treated as a nominal classification problem, but when you do this, you are discarding ordinal information. A better approach is to use specific methods that take into account this kind of information to improve the performance of the classification model. One way to use the ordinal information is to evaluate the model using an ordinal metric. Multiple metrics exist in the literature of machine learning and statistics [24], [25]. Kappa index is a well-known statistic coefficient defined by Cohen [26] to measure inter-rater agreement on classifying elements into a set of categories. Later, Weighted Kappa (WK) is a modified version of the Kappa statistic calculated to allow assigning different weights to different levels of aggregation between two variables. Weighted kappa loss was described in previous work [18] and is a cost function that is based on the WK metric. These functions are indicated for problems where different inter-classes weights are assigned, and that work proved that using these functions improves the model performance and reduce the overfitting risk. This kind of problems is associated with ordinal data processing. Those weights are predefined and depend on the type of the chosen WK. In a linear penalization, the weight is proportional to the distance between the predicted and the real class. In the Quadratic Weighted Kappa (QWK) the penalization is proportional to the square of the distance. In this work, we will use the QWK metric and the QWK loss.

Softmax function is the most widely used for the output layer in all kind of neural networks for classification tasks. It converts the output of the network to a set of probabilities corresponding with each class. The Proportional Odds Model (POM) [27], [28] will be used in this work as it should give better performance in ordinal problems. Both output functions will be compared to demonstrate this fact. We will contrast the results obtained with statistical analysis to provide more robust results. An approximated ANOVA III [29] test followed by a Student's t-test [CITA] will be performed because of the limitations of the computational time required to run a higher number of executions.

The experiments will be run using two different ordinal datasets: Diabetic Retinopathy [18], that contains high-resolution fundus images related with diabetes disease, and Adience [31], that is formed of human faces images that are assigned an age range.

This paper is organized as follows: in Section II, we take a look at previous works related to this paper. Sections III and IV present a formal description of an ordinal problem and the Proportional Odds Model. In Section V we describe the model, the experiments and the datasets used, in Section VI we present the results obtained and the statistical analysis and finally in Section VII we expose the conclusions of this work.

II. Related work

There are many works related to the application and development of convolutional neural networks models. Few works treat ordinal classification problems, though.

J. de la Torre et al. [18] proposed the use of QWK loss function for the optimization algorithm. They compared this cost function against the traditional log-loss function across three different databases, including the Diabetic Retinopathy database as the most complex one. They proved that their function could improve the results as it reduces the overfitting and the required training time. Also, they checked the importance of hyper-parameter tuning.

Christopher Beckham and Christopher Pal [31] proposed a straightforward technique to constrain discrete ordinal probability distributions to be unimodal via the use of the Poisson and binomial probability distributions. They evaluated this approach in the context of deep learning on two large ordinal image datasets, including the Adience dataset used in this paper, and they obtained promising results.

Adience dataset has been used in other works for human age estimation. E. Eidinger [32] presented an approach using support vector machines and neural networks. J.-C. Chen [33] proposed a coarse-to-fine strategy for deep convolutional networks. G. Levi [34] presented another convolutional network model for age estimation.

At present, there is a great discussion in terms of finding the best activation function that offers good performance and mitigates the effects of the gradients vanishing problem [35], [36]. Rectified Linear Unit (ReLU) [37] is widely used in most deep learning works, but recently, Clevert et al. proposed the Exponential Linear Unit (ELU) [38]. They proved that ELUs alleviate the vanishing gradient problem via the identity for positive values. In their experiments, ELUs lead not only to faster learning but also to significantly better generalization performance than ReLUs and Leaky ReLU functions (LReLU) on networks with more than five layers. That's why we decided to use the ELU function for the experiments described in this paper.

Sergey Ioffe and Christian Szegedy [21] described Batch Normalization and its benefits in previous work. It reduces the internal covariate shift by normalizing layer inputs. Their method draws its strength from making normalization a part of the model architecture and performing the normalization for each training batch. It allows them to use higher learning rates and be less careful about initialization. It also eliminates the need for using regularization techniques like Dropout.

III. Ordinal problem definition

The ordinal problem consists of predicting the label y of an input vector $\mathbf{x}$, where $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^K$ and $y \in \mathcal{Y} = \{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_Q\}$, i.e. $\mathbf{x}$ is in a K-dimensional input space, and y is in a label space of Q different labels. The objective of the ordinal problem is to find a function $r : \mathcal{X} \rightarrow \mathcal{Y}$ to predict the labels or categories of new patterns, given a training set of N points, $D = \{(x_i, y_i), i = 1, ..., N\}$. A natural label ordering exists for ordinal problems: $\mathcal{C}_1 \prec \mathcal{C}_2 \prec ... \prec \mathcal{C}_Q$. The assumption of order between class labels makes that two different elements of $\mathcal{Y}$ can always be compared by using the relation $\prec$, which is not possible under the nominal classification setting. In regression (where $y \in \mathbb{R}$), real values in $\mathbb{R}$ can be ordered by the standard $<$ operator, but labels in ordinal regression ($y \in \mathcal{Y}$) do not carry metric information, so the category serves as a qualitative indication of the pattern rather than a quantitative one.

IV. Proportional Odds Model

Arising from a statistical background, the Proportional Odds Model (POM) is one of the first models designed explicitly for ordinal regression [39], dated back to 1980. It is a member of a wider family of models recognised as Cumulative Link Models (CLMs) [28]. CLMs predict probabilities of groups of contiguous categories, taking the ordinal scale into account. In this way, cumulative probabilities $P(y \prec C_q|\mathbf{x})$ are estimated, which can be directly related to standard probabilities:

$$P(y \preceq \mathcal{C}_q|\mathbf{x}) = P(y = \mathcal{C}_1|\mathbf{x}) + ... + P(y = \mathcal{C}_q|\mathbf{x}), \quad (1)$$

$$P(y = \mathcal{C}_q|\mathbf{x}) = P(y \preceq \mathcal{C}_q|\mathbf{x}) - P(y \preceq \mathcal{C}_{q-1}|\mathbf{x}), \quad (2)$$

with $q = 2, ..., Q-1$, and considering that $P(y = \mathcal{C}_1|\mathbf{x}) = P(y \preceq \mathcal{C}_1|\mathbf{x})$ and $P(y \preceq \mathcal{C}_Q|\mathbf{x}) = 1$.

The model is inspired by the notion of latent variable, considering a linear transformation $f(x) = \mathbf{w}^\mathbf{T}\mathbf{x}$ of the input variables as the one-dimensional mapping. The decision rule $r : \mathcal{X} \rightarrow \mathcal{Y}$ is not fitted directly, but stochastic ordering of space $\mathcal{X}$ is satisfied by the following general model form [40]:

$$g^{-1}(P(y \preceq \mathcal{C}_q|\mathbf{x})) = b_q - \mathbf{w}^\mathbf{T}\mathbf{x}, \quad q = 1, ..., Q-1, \quad (3)$$

where $g^{-1} : [0, 1] \rightarrow (-\infty, +\infty)$ is a monotonic function often termed as the inverse link function and b_q is the threshold defined for class $\mathcal{C}_q$. Consider the latent variable $y^* = f(x)^* = \mathbf{w}^\mathbf{T}\mathbf{x} + \epsilon$, where ϵ is the random variable of the error. The most common choice for the probability distribution of ϵ is the logistic function (which is the default function for POM). Label $\mathcal{C}_q$ is proposed if and only if $f(x) \in [b_{q-1}, b_q]$, where the function f and $\mathbf{b} = (b_0, b_1, ..., b_{Q-1}, b_Q)$ are to be determined from the data. it is assumed that $b_0 = -\infty$ and $b_Q = +\infty$, so the real line defined by $f(x), x \in \mathcal{X}$, is divided into Q consecutive intervals. Each interval corresponds to a category. The constraints $b_1 \leq b_2 \leq ... \leq b_{Q-1}$ ensures that $P(y \preceq \mathcal{C}_q|\mathbf{x})$ increases with q [39]. In this work, we use this ordinal model with different link functions for the probability distribution of ϵ, including logit, probit and complementary log-log. These functions will be detailed in Section V-B.

V. Experiments

A. Data

We make use of two ordinal datasets appropriate for deep neural networks:

- *Diabetic Retinopathy (DR)*[1]. DR is a dataset consisting of extremely high-resolution fundus image data. The training set consists of 17563 pairs of images (where a pair consists of a left and right eye image corresponding to a patient). In this dataset, we try to predict from five levels of diabetic retinopathy: no DR (25810 images), mild DR (2443 images), moderate DR (5292 images), severe DR (873 images), or proliferative DR (708 images). The images are taken in variable conditions: by different cameras, varying conditions of illumination and different resolutions. These images come from the EyePACS dataset that was used in a Diabetic Retinopathy Detection competition that was hosted on the Kaggle platform. Also, this dataset was used in later works [18], [41] and ordinal techniques (such as an ordinal cost function) were applied in order to achieve better performance. A validation set is set aside, consisting of 10% of the patients in the training set. The images are resized to 128 by 128 pixels and rescaled to $[0, 1]$ range. Data augmentation techniques, described in Section V-C, are applied to achieve a higher number of samples.
- *Adience*[2]. This dataset consists of 26580 faces belonging to 2284 subjects. We use the form of the dataset where faces have been pre-cropped and aligned. The dataset was preprocessed, using the methods described in a previous work [31], so that the images are 256 pixels in width and height, and pixels values follow a $(0; 1)$ normal distribution. The original dataset is split into five cross-validation folds. The training set consists of merging the first four folds which comprise a total of 15554 images. From this, 10% of the images are held out as part of a validation set. The last fold is used as test set.

B. The model

Convolutional Neural Networks (CNN) has been used for both datasets. The different architectures of CNN used in these experiments are presented in tables I and II. The architecture for DR is the same that was used in [18] and the network for Adience is a small Residual Network

[1]https://www.kaggle.com/c/diabetic-retinopathy-detection/data
[2]http://www.openu.ac.il/home/hassner/Adience/data.html

TABLE I
Description of the architecture used in the DR experiments.

Layer	Output shape
2 x Conv3x3@32s1	252x252x32
MaxPool2x2s2	126x126x32
2 x Conv3x3@64s1	122x122x64
MaxPool2x2s2	61x61x64
2 x Conv3x3@128s1	57x57x128
MaxPool2x2s2	28x28x128
2 x Conv3x3@128s1	24x24x128
MaxPool2x2s2	12x12x128
Conv4x4@128s1	9x9x128

TABLE II
Description of the architecture used in the Adience experiments.

Layer	Output shape
Conv7x7@32s2	112x112x32
MaxPool3x3s2	55x55x32
2 x ResBlock3x3@64s1	55x55x32
1 x ResBlock3x3@128s2	28x28x64
2 x ResBlock3x3@128s1	28x28x64
1 x ResBlock3x3@256s2	14x14x128
2 x ResBlock3x3@256s1	14x14x128
1 x ResBlock3x3@512s2	7x7x256
2 x ResBlock3x3@512s1	7x7x256
AveragePool7x7s2	1x1x256

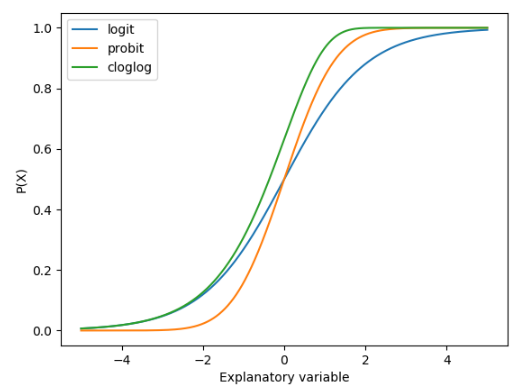

Fig. 1. Representation of link functions.

(ResNet) [3] that was used in [31]. For convolutional layers, ConvWxH@FsS = F filters of size WxH and stride S. For pooling layers, PoolWxHsS = pool size of WxH and stride S.

Every convolutional layer is followed by an ELU activation layer [38] and a batch normalization [21]. At the output, a Proportional Odds Model is used with different link functions [28]. The logit link function is commonly used within POM. In this paper, we are comparing other link functions like probit or complementary log-log with the logit link. These three types of links are explained below and represented in Figure 1. They all follow the same form $P(y \prec \mathcal{C}_q) = \Phi(b_q - \mathbf{w}^\mathbf{T}\mathbf{x})$ for a continuous cdf Φ.

- *Logit.* Logit link function is the most widely used function for Proportional Odds Models. The logit link is shown in 4 and 5.

$$\begin{aligned} logit[P(y \preceq \mathcal{C}_q)] = log\frac{P(y \preceq \mathcal{C}_q)}{1 - P(y \preceq \mathcal{C}_q)} = \\ = b_q - \mathbf{w^T x}, \quad q = 1, ..., Q-1 \end{aligned} \tag{4}$$

or the equivalent expression

$$P(y \preceq \mathcal{C}_q) = \frac{1}{1 + e^{-(b - \mathbf{w^T x})}} \tag{5}$$

- *Probit*. Probit link function is the inverse of the standard normal cummulative distribution function (cdf) Φ. Its expression is shown in 6 and 7.

$$\begin{aligned} \Phi^{-1}[P(y \preceq \mathcal{C}_q)] = b_q - \mathbf{w^T x}, \quad q = 1, ..., Q-1 \\ P(y \preceq \mathcal{C}_q) = \Phi(b_q - \mathbf{w^T x}), \qquad q = 1, ..., Q-1 \end{aligned} \tag{6}$$

or

$$P(y \preceq \mathcal{C}_q) = \int_{-\infty}^{b_q - \mathbf{w^T x}} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} dx \tag{7}$$

- *Complementary log-log*. Like the logit and the probit transformation, the complementary log-log transformation takes a response that is restricted to the (0,1) interval and converts it into something in $(-\infty, +\infty)$ interval. Complementary log-log expression is shown in 8 and 9.

$$\begin{aligned} log[-log[1 - P(y \preceq \mathcal{C}_q)]] = \\ = b_q - \mathbf{w^T x}, \quad q = 1, ..., Q-1 \end{aligned} \tag{8}$$

or

$$P(y \preceq \mathcal{C}_q) = 1 - e^{-e^{b_q - \mathbf{w^T x}}}, \quad q = 1, ..., Q-1 \tag{9}$$

Logit and probit links are symmetric, that is

$$link[P(y \preceq \mathcal{C}_q)] = -link[1 - P(y \preceq \mathcal{C}_q)] \tag{10}$$

This means that the response curve for $P(y \preceq \mathcal{C}_q)$ has a symmetric appearance about the point $P(y \preceq \mathcal{C}_q) = 0.5$, and so $P(y \preceq \mathcal{C}_q)$ has the same rate for approaching 0 than for approaching 1. This symmetric property can be demonstrated as follows:

1) If we define $P(y \preceq \mathcal{C}_q) \equiv p$, we have

$$\begin{aligned} link(p) = logit(p) = log\left(\frac{p}{1-p}\right) = \\ = log(p) - log(1-p) \end{aligned} \tag{11}$$

while

$$\begin{aligned} -link(1-p) = -logit(1-p) = \\ = -log\left(\frac{1-p}{p}\right) = -log(1-p) + log(p) \end{aligned} \tag{12}$$

The symmetry has been proven for logit link.

2) If we define

$$\begin{aligned} p \equiv P(y \preceq \mathcal{C}_q) = \Phi(b_q - \mathbf{w^T x}) = \\ = \int_{-\infty}^{b_q - \mathbf{w^T x}} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} dx \end{aligned} \tag{13}$$

then

$$probit(p) = \Phi^{-1}(p) = b_q - \mathbf{w^T x} \tag{14}$$

and

$$-\,probit(1-p) = \Phi^{-1}(1-p) = -b_q + \mathbf{w^T x} \tag{15}$$

where

$$1 - p = \int_{-\infty}^{-b_q + \mathbf{w^T x}} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} dx \tag{16}$$

and

$$p = 1 - \int_{-\infty}^{-b_q + \mathbf{w^T x}} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2} dx \tag{17}$$

the symmetric property for the probit function has been proven.

Unlike logit and probit, the complementary log-log model is asymmetrical. It is frequently used when the probability of an event is very small or very large. When the given data is not symmetric in the $[0, 1]$ interval and increase slowly at small to moderate value but increases sharply near 1, the logit and probit models are inappropriate. However, in this situation, the complementary log-log model might give a satisfying answer.

C. Procedure

The model is optimized using a batch based first-order optimization algorithm called Adam [42]. We study different initial learning rates in order to find the optimal one for each problem. We apply an exponential decay across training epochs to the initial learning rate.

Both datasets have been artificially equalized using data augmentation techniques [43], [4]. However, different transformations were applied to each one. Diabetic Retinopathy dataset augmentation was based on image cropping and zooming, horizontal and vertical flipping, brightness adjustment and random rotations. Horizontal flipping was the only transformation applied to Adience dataset.

In the case of Diabetic Retinopathy Detection, the epoch size has been fixed to 100000 images per epoch. For the Adience dataset, epoch size is the number of images in the training set.

The model is evaluated with Quadratic Weighted Kappa metric (QWK) [44]. This evaluation measure gives a higher weight to the errors that are further from the correct class. Quadratic weighted kappa loss is considered as loss function for the optimizer as it gives better performance for ordinal classification problems [18].

Also, other evaluation metrics have been used to ease the comparison with other works: Minimum Sensitivity (MS) [24], Mean Absolute Error (MAE) [24], Mean Squared Error (MSE) [45], Correct Classification Rate (CCR), Top-2 CCR [31], Top-3 CCR [31] and 1-off accuracy [33], [34], [32].

Experiments were run with the standard cross-entropy loss and the softmax function too in order to prove the performance improvement of the QWK loss and the Proportional Odds Model. The results of these experiments are analysed in Section VI-D.

D. Factors

Three different factors have been considered: learning rate, batch size and link function for the final output layer.

- *Learning rate.* Learning rate is one of the most critical hyper-parameters to tune for training deep neural networks. Optimal learning rate can vary depending on the dataset and the CNN architecture. Previous works have presented some techniques that adjust this parameter in order to achieve better performance [20], [19]. Within this work, we have considered three different values for this parameter: 10^{-2}, 10^{-3} and 10^{-4}.
- *Batch size.* Batch size is also an important parameter as it controls the number of weight updates that are made on every epoch. It can affect the training time and the model performance. In this paper, we have tried three separate batch sizes for each dataset. For DR dataset, we have used 5, 10 and 15 while, for the Adience dataset, 64, 128 and 256 was used. We took the batch sizes that were used in [18] and [31] as a reference, and we expanded the range on both sides.
- *Link function.* Different link functions have been used for the POM at the last layer output: logit, probit and complementary log-log.

VI. Results

In this section, we present the results of the experiments. For each dataset, we show a table with the detailed experiments done training the model with each combination of parameters. Every parameter combination has been run five times. These tables show the mean value and the standard deviation (SD) of each metric across these five executions for the test set. These values are exposed using the format $\text{MEAN}_{(SD)}$.

A. Diabetic Retinopathy

Detailed results for the Diabetic Retinopathy dataset are presented in Table III. The best result for each metric is marked in bold and the second best is in italic font.

The best mean QWK value was obtained with the complementary log-log link function using a batch size of 10 and a learning rate of 10^{-3}. However, the best CCR value was obtained with a batch size of 15, the logit link and a learning rate of 10^{-4}. The optimal configuration depends on the metric we are analysing. In this case, as we are working with an ordinal problem, the most reliable metric is the QWK.

B. Adience

Results for the experiments made with the Adience dataset are shown in Table IV. The best result for each metric is marked in bold and the second best is in italic font.

The best mean QWK value was obtained with the logit link function using a batch size of 64 and a learning rate of 10^{-4}. Also, this configuration obtained the best score for Top-2, Top-3 and 1-off accuracy, and the second best for MS, MAE and CCR. In this case, this configuration can be selected as the optimal for this problem.

C. Statistical analysis

In this subsection, a statistical analysis will be performed in order to obtain conclusions from the results presented in this section.

The significance and relative importance of the parameters concerning the results obtained, as well as suitable values for each, were obtained using the ANalysis Of the VAriance (ANOVA) test.

The ANOVA test [29] is one of the most widely used statistical techniques. ANOVA is essentially a method of analysing the variance to which a response is subject into its various components, corresponding to the sources of variation which can be identified.

ANOVA, in this case, examines the effects of three quantitative variables (termed factors) on one quantitative response. Considered factors are the link function, which can be logit, probit and complementary log-log, the learning rate for the Adam optimization algorithm (10^{-2}, 10^{-3} and 10^{-4}) and the batch size (5, 10 and 15 for DR and 64, 128 and 256 for Adience).

Following the setup of the previous study, we performed an ANOVA III analysis and multiple comparison tests. We assume that five executions are enough to do the statistical tests because of the computational time limitations.

We denote by $QWK_{i,j,k,l}(i = 1, ..., 3; j = 1, ..., 3; k = 1, ..., 3)$ the value observed when the first factor is at the i-th level, the second at the j-th level and the third at k-th level. We assume that the three factors do not act independently and therefore there exists an interaction between each pair of them and between the three factors. In this case, the observations fit 18.

$$\begin{aligned} QWK_{i,j,k,l} =& \mu + L_i + P_j + B_k + LP_{i,j} + LB_{i,k} \\ &+ PB_{j,k} + LPB_{i,j,k} + \epsilon_{i,j,k,l} \end{aligned} \tag{18}$$

where μ is the fixed effect that is common to all the populations; L_i is the effect associated with the *i-th* level of the link factor (logit, probit, complementary log-log); P_j is the effect associated with the *j-th* level of the parameter factor and B_k is the effect associated with the *k-th* level of the batch size factor. The term $LP_{i,j}$ denotes the joint

TABLE III
DIABETIC RETINOPATHY RESULTS. BS STANDS FOR BATCH SIZE, LF FOR LINK FUNCTION AND LR FOR LEARNING RATE.

BS	LF	LR	$\overline{\text{QWK}}_{(SD)}$	$\overline{\text{MS}}_{(SD)}$	$\overline{\text{MAE}}_{(SD)}$	$\overline{\text{MSE}}_{(SD)}$	$\overline{\text{CCR}}_{(SD)}$	$\overline{\text{Top-2}}_{(SD)}$	$\overline{\text{Top-3}}_{(SD)}$	$\overline{\text{1-off}}_{(SD)}$
5	c log-log	10^{-2}	$0.414_{(0.057)}$	$0.075_{(0.042)}$	$0.177_{(0.023)}$	$0.165_{(0.020)}$	$0.556_{(0.057)}$	$0.833_{(0.042)}$	$0.968_{(0.011)}$	$0.816_{(0.021)}$
5	c log-log	10^{-3}	$0.534_{(0.027)}$	$0.102_{(0.011)}$	$0.137_{(0.006)}$	$0.123_{(0.004)}$	$0.658_{(0.015)}$	$0.871_{(0.011)}$	$0.966_{(0.003)}$	$0.852_{(0.002)}$
5	c log-log	10^{-4}	$0.520_{(0.006)}$	$0.067_{(0.008)}$	$0.123_{(0.003)}$	$0.104_{(0.001)}$	$0.697_{(0.006)}$	$0.842_{(0.008)}$	$0.961_{(0.003)}$	$0.851_{(0.002)}$
5	logit	10^{-2}	$0.416_{(0.041)}$	$0.095_{(0.029)}$	$0.175_{(0.021)}$	$0.162_{(0.018)}$	$0.563_{(0.054)}$	$0.762_{(0.040)}$	$0.908_{(0.026)}$	$0.807_{(0.029)}$
5	logit	10^{-3}	$0.554_{(0.013)}$	$0.093_{(0.009)}$	$0.137_{(0.003)}$	$0.123_{(0.003)}$	$0.660_{(0.008)}$	$0.802_{(0.005)}$	$0.936_{(0.004)}$	$0.853_{(0.005)}$
5	logit	10^{-4}	$0.520_{(0.003)}$	$0.063_{(0.004)}$	$0.122_{(0.002)}$	$0.102_{(0.001)}$	$0.706_{(0.005)}$	$0.823_{(0.004)}$	$0.949_{(0.003)}$	$0.862_{(0.003)}$
5	probit	10^{-2}	$0.460_{(0.048)}$	$0.079_{(0.046)}$	$0.197_{(0.064)}$	$0.182_{(0.053)}$	$0.504_{(0.167)}$	$0.808_{(0.034)}$	$0.927_{(0.073)}$	$0.689_{(0.240)}$
5	probit	10^{-3}	$0.564_{(0.018)}$	$0.099_{(0.013)}$	$0.147_{(0.018)}$	$0.132_{(0.014)}$	$0.636_{(0.045)}$	$0.822_{(0.040)}$	$0.939_{(0.020)}$	$0.840_{(0.015)}$
5	probit	10^{-4}	$0.523_{(0.005)}$	$0.067_{(0.012)}$	$0.122_{(0.002)}$	$0.105_{(0.001)}$	$0.701_{(0.006)}$	$0.823_{(0.002)}$	$0.953_{(0.002)}$	$0.860_{(0.003)}$
10	c log-log	10^{-2}	$0.423_{(0.239)}$	$0.062_{(0.051)}$	$0.127_{(0.017)}$	$0.120_{(0.014)}$	$0.684_{(0.046)}$	$\mathbf{0.894}_{(\mathbf{0.062})}$	$\mathbf{0.986}_{(\mathbf{0.012})}$	$0.832_{(0.020)}$
10	c log-log	10^{-3}	$\mathbf{0.582}_{(\mathbf{0.016})}$	$0.102_{(0.006)}$	$0.128_{(0.003)}$	$0.115_{(0.002)}$	$0.680_{(0.007)}$	$\mathit{0.880}_{(\mathit{0.004})}$	$0.972_{(0.003)}$	$0.861_{(0.004)}$
10	c log-log	10^{-4}	$0.537_{(0.010)}$	$0.064_{(0.004)}$	$\mathit{0.116}_{(\mathit{0.001})}$	$0.096_{(0.001)}$	$0.717_{(0.003)}$	$0.837_{(0.002)}$	$0.971_{(0.001)}$	$0.860_{(0.002)}$
10	logit	10^{-2}	$0.531_{(0.031)}$	$0.107_{(0.008)}$	$0.151_{(0.010)}$	$0.140_{(0.008)}$	$0.623_{(0.025)}$	$0.802_{(0.022)}$	$0.934_{(0.013)}$	$0.838_{(0.014)}$
10	logit	10^{-3}	$0.579_{(0.009)}$	$0.096_{(0.012)}$	$0.127_{(0.005)}$	$0.113_{(0.004)}$	$0.686_{(0.013)}$	$0.817_{(0.006)}$	$0.954_{(0.005)}$	$0.861_{(0.002)}$
10	logit	10^{-4}	$0.539_{(0.007)}$	$0.074_{(0.013)}$	$0.126_{(0.005)}$	$0.095_{(0.002)}$	$0.707_{(0.010)}$	$0.823_{(0.007)}$	$0.957_{(0.005)}$	$0.858_{(0.004)}$
10	probit	10^{-2}	$0.508_{(0.037)}$	$0.088_{(0.044)}$	$0.145_{(0.018)}$	$0.137_{(0.014)}$	$0.639_{(0.045)}$	$0.835_{(0.015)}$	$0.960_{(0.008)}$	$0.829_{(0.020)}$
10	probit	10^{-3}	$0.558_{(0.034)}$	$\mathbf{0.111}_{(\mathbf{0.005})}$	$0.134_{(0.003)}$	$0.120_{(0.002)}$	$0.666_{(0.008)}$	$0.831_{(0.007)}$	$0.955_{(0.001)}$	$0.863_{(0.003)}$
10	probit	10^{-4}	$0.541_{(0.010)}$	$0.076_{(0.006)}$	$0.119_{(0.002)}$	$0.098_{(0.001)}$	$0.712_{(0.005)}$	$0.828_{(0.003)}$	$0.961_{(0.002)}$	$0.862_{(0.001)}$
15	c log-log	10^{-2}	$0.564_{(0.016)}$	$0.108_{(0.014)}$	$0.143_{(0.006)}$	$0.134_{(0.005)}$	$0.640_{(0.015)}$	$0.879_{(0.011)}$	$0.972_{(0.005)}$	$0.851_{(0.006)}$
15	c log-log	10^{-3}	$0.559_{(0.026)}$	$\mathit{0.111}_{(\mathit{0.008})}$	$0.127_{(0.004)}$	$0.113_{(0.003)}$	$0.682_{(0.010)}$	$0.871_{(0.008)}$	$\mathit{0.974}_{(\mathit{0.002})}$	$\mathbf{0.868}_{(\mathbf{0.002})}$
15	c log-log	10^{-4}	$0.538_{(0.009)}$	$0.054_{(0.003)}$	$\mathbf{0.115}_{(\mathbf{0.002})}$	$\mathit{0.093}_{(\mathit{0.001})}$	$0.720_{(0.006)}$	$0.835_{(0.007)}$	$0.970_{(0.003)}$	$0.860_{(0.006)}$
15	logit	10^{-2}	$0.551_{(0.020)}$	$0.104_{(0.008)}$	$0.139_{(0.011)}$	$0.129_{(0.009)}$	$0.654_{(0.027)}$	$0.815_{(0.017)}$	$0.948_{(0.016)}$	$0.856_{(0.015)}$
15	logit	10^{-3}	$0.551_{(0.010)}$	$0.106_{(0.016)}$	$0.129_{(0.008)}$	$0.114_{(0.005)}$	$0.680_{(0.019)}$	$0.818_{(0.008)}$	$0.952_{(0.007)}$	$\mathit{0.866}_{(\mathit{0.001})}$
15	logit	10^{-4}	$0.543_{(0.008)}$	$0.056_{(0.003)}$	$0.121_{(0.004)}$	$\mathbf{0.090}_{(\mathbf{0.001})}$	$\mathbf{0.723}_{(\mathbf{0.004})}$	$0.833_{(0.004)}$	$0.964_{(0.003)}$	$0.862_{(0.004)}$
15	probit	10^{-2}	$0.534_{(0.032)}$	$0.104_{(0.013)}$	$0.148_{(0.015)}$	$0.138_{(0.014)}$	$0.631_{(0.038)}$	$0.845_{(0.030)}$	$0.964_{(0.010)}$	$0.852_{(0.010)}$
15	probit	10^{-3}	$\mathit{0.580}_{(\mathit{0.021})}$	$0.104_{(0.016)}$	$0.129_{(0.008)}$	$0.116_{(0.005)}$	$0.680_{(0.018)}$	$0.832_{(0.010)}$	$0.959_{(0.007)}$	$0.866_{(0.003)}$
15	probit	10^{-4}	$0.533_{(0.004)}$	$0.065_{(0.005)}$	$0.117_{(0.002)}$	$0.094_{(0.001)}$	$\mathit{0.721}_{(\mathit{0.004})}$	$0.832_{(0.002)}$	$0.964_{(0.001)}$	$0.863_{(0.001)}$

TABLE IV
ADIENCE TEST RESULTS. BS STANDS FOR BATCH SIZE, LF FOR LINK FUNCTION AND LR FOR LEARNING RATE.

BS	LF	LR	$\overline{\text{QWK}}_{(SD)}$	$\overline{\text{MS}}_{(SD)}$	$\overline{\text{MAE}}_{(SD)}$	$\overline{\text{MSE}}_{(SD)}$	$\overline{\text{CCR}}_{(SD)}$	$\overline{\text{Top-2}}_{(SD)}$	$\overline{\text{Top-3}}_{(SD)}$	$\overline{\text{1-off}}_{(SD)}$
64	c log-log	10^{-2}	$0.808_{(0.025)}$	$0.086_{(0.041)}$	$0.147_{(0.008)}$	$0.129_{(0.007)}$	$0.415_{(0.031)}$	$0.677_{(0.024)}$	$0.798_{(0.036)}$	$0.804_{(0.015)}$
64	c log-log	10^{-3}	$0.873_{(0.006)}$	$0.144_{(0.057)}$	$\mathbf{0.124}_{(\mathbf{0.003})}$	$0.101_{(0.003)}$	$\mathbf{0.519}_{(\mathbf{0.014})}$	$\mathit{0.764}_{(\mathit{0.010})}$	$0.861_{(0.019)}$	$0.886_{(0.006)}$
64	c log-log	10^{-4}	$0.799_{(0.010)}$	$0.000_{(0.000)}$	$0.174_{(0.001)}$	$0.100_{(0.002)}$	$0.324_{(0.015)}$	$0.616_{(0.020)}$	$0.795_{(0.012)}$	$0.771_{(0.014)}$
64	logit	10^{-2}	$0.778_{(0.019)}$	$0.074_{(0.041)}$	$0.159_{(0.006)}$	$0.137_{(0.007)}$	$0.366_{(0.025)}$	$0.636_{(0.015)}$	$0.785_{(0.010)}$	$0.775_{(0.015)}$
64	logit	10^{-3}	$\mathbf{0.881}_{(\mathbf{0.005})}$	$\mathit{0.178}_{(\mathit{0.023})}$	$\mathit{0.126}_{(\mathit{0.001})}$	$0.098_{(0.003)}$	$\mathit{0.518}_{(\mathit{0.008})}$	$\mathbf{0.765}_{(\mathbf{0.015})}$	$\mathbf{0.902}_{(\mathbf{0.005})}$	$\mathbf{0.894}_{(\mathbf{0.005})}$
64	logit	10^{-4}	$0.784_{(0.011)}$	$0.000_{(0.000)}$	$0.180_{(0.001)}$	$0.108_{(0.004)}$	$0.318_{(0.026)}$	$0.621_{(0.034)}$	$0.772_{(0.024)}$	$0.731_{(0.030)}$
64	probit	10^{-2}	$0.836_{(0.005)}$	$0.135_{(0.021)}$	$0.134_{(0.002)}$	$0.121_{(0.002)}$	$0.468_{(0.011)}$	$0.720_{(0.009)}$	$0.861_{(0.009)}$	$0.829_{(0.005)}$
64	probit	10^{-3}	$\mathit{0.874}_{(\mathit{0.004})}$	$0.134_{(0.012)}$	$0.126_{(0.003)}$	$0.105_{(0.003)}$	$0.511_{(0.014)}$	$0.756_{(0.009)}$	$\mathit{0.895}_{(\mathit{0.003})}$	$\mathit{0.889}_{(\mathit{0.003})}$
64	probit	10^{-4}	$0.805_{(0.004)}$	$0.000_{(0.000)}$	$0.170_{(0.001)}$	$0.100_{(0.002)}$	$0.360_{(0.011)}$	$0.653_{(0.011)}$	$0.809_{(0.009)}$	$0.790_{(0.009)}$
128	c log-log	10^{-2}	$0.832_{(0.013)}$	$0.123_{(0.031)}$	$0.135_{(0.004)}$	$0.117_{(0.002)}$	$0.463_{(0.013)}$	$0.705_{(0.019)}$	$0.813_{(0.025)}$	$0.832_{(0.006)}$
128	c log-log	10^{-3}	$0.873_{(0.006)}$	$\mathbf{0.185}_{(\mathbf{0.029})}$	$0.128_{(0.002)}$	$0.100_{(0.001)}$	$0.513_{(0.007)}$	$0.758_{(0.008)}$	$0.870_{(0.011)}$	$0.880_{(0.009)}$
128	c log-log	10^{-4}	$0.659_{(0.025)}$	$0.000_{(0.000)}$	$0.190_{(0.002)}$	$0.125_{(0.004)}$	$0.235_{(0.026)}$	$0.466_{(0.031)}$	$0.640_{(0.030)}$	$0.536_{(0.041)}$
128	logit	10^{-2}	$0.781_{(0.041)}$	$0.096_{(0.059)}$	$0.153_{(0.007)}$	$0.125_{(0.007)}$	$0.398_{(0.031)}$	$0.638_{(0.033)}$	$0.790_{(0.025)}$	$0.779_{(0.020)}$
128	logit	10^{-3}	$0.865_{(0.005)}$	$0.127_{(0.026)}$	$0.134_{(0.001)}$	$0.099_{(0.003)}$	$0.497_{(0.009)}$	$0.754_{(0.008)}$	$0.882_{(0.009)}$	$0.874_{(0.008)}$
128	logit	10^{-4}	$0.586_{(0.008)}$	$0.000_{(0.000)}$	$0.196_{(0.001)}$	$0.151_{(0.005)}$	$0.192_{(0.001)}$	$0.364_{(0.060)}$	$0.581_{(0.034)}$	$0.396_{(0.002)}$
128	probit	10^{-2}	$0.849_{(0.005)}$	$0.132_{(0.010)}$	$0.131_{(0.001)}$	$0.115_{(0.001)}$	$0.479_{(0.004)}$	$0.728_{(0.007)}$	$0.854_{(0.009)}$	$0.847_{(0.007)}$
128	probit	10^{-3}	$0.866_{(0.002)}$	$0.124_{(0.043)}$	$0.130_{(0.002)}$	$0.100_{(0.003)}$	$0.505_{(0.006)}$	$0.750_{(0.010)}$	$0.882_{(0.004)}$	$0.873_{(0.006)}$
128	probit	10^{-4}	$0.718_{(0.015)}$	$0.000_{(0.000)}$	$0.185_{(0.001)}$	$0.110_{(0.002)}$	$0.300_{(0.031)}$	$0.575_{(0.015)}$	$0.733_{(0.010)}$	$0.640_{(0.033)}$
256	c log-log	10^{-2}	$0.853_{(0.004)}$	$0.157_{(0.024)}$	$0.130_{(0.002)}$	$0.110_{(0.001)}$	$0.485_{(0.009)}$	$0.744_{(0.006)}$	$0.842_{(0.016)}$	$0.858_{(0.004)}$
256	c log-log	10^{-3}	$0.840_{(0.017)}$	$0.095_{(0.017)}$	$0.144_{(0.005)}$	$0.097_{(0.004)}$	$0.456_{(0.021)}$	$0.720_{(0.022)}$	$0.840_{(0.018)}$	$0.842_{(0.018)}$
256	c log-log	10^{-4}	$0.552_{(0.010)}$	$0.000_{(0.000)}$	$0.199_{(0.001)}$	$0.165_{(0.004)}$	$0.187_{(0.001)}$	$0.368_{(0.022)}$	$0.475_{(0.025)}$	$0.387_{(0.001)}$
256	logit	10^{-2}	$0.764_{(0.102)}$	$0.077_{(0.067)}$	$0.155_{(0.020)}$	$0.125_{(0.015)}$	$0.387_{(0.083)}$	$0.632_{(0.103)}$	$0.790_{(0.077)}$	$0.783_{(0.065)}$
256	logit	10^{-3}	$0.851_{(0.008)}$	$0.100_{(0.030)}$	$0.147_{(0.003)}$	$\mathbf{0.094}_{(\mathbf{0.002})}$	$0.449_{(0.015)}$	$0.726_{(0.015)}$	$0.861_{(0.006)}$	$0.850_{(0.008)}$
256	logit	10^{-4}	$0.558_{(0.008)}$	$0.000_{(0.000)}$	$0.202_{(0.001)}$	$0.191_{(0.002)}$	$0.187_{(0.002)}$	$0.206_{(0.007)}$	$0.395_{(0.046)}$	$0.389_{(0.003)}$
256	probit	10^{-2}	$0.858_{(0.005)}$	$0.164_{(0.033)}$	$0.130_{(0.002)}$	$0.112_{(0.002)}$	$0.486_{(0.007)}$	$0.741_{(0.008)}$	$0.867_{(0.008)}$	$0.862_{(0.005)}$
256	probit	10^{-3}	$0.850_{(0.008)}$	$0.111_{(0.040)}$	$0.144_{(0.002)}$	$\mathit{0.095}_{(\mathit{0.001})}$	$0.460_{(0.011)}$	$0.732_{(0.006)}$	$0.865_{(0.006)}$	$0.853_{(0.007)}$
256	probit	10^{-4}	$0.565_{(0.010)}$	$0.000_{(0.000)}$	$0.196_{(0.001)}$	$0.150_{(0.005)}$	$0.189_{(0.001)}$	$0.409_{(0.014)}$	$0.602_{(0.022)}$	$0.392_{(0.002)}$

effect of the presence of level i of the first factor and level j of the second one; this, therefore, is denominated the interaction term between L and P factors. The same interaction effect is appreciated on $LB_{i,k}$, $PB_{j,k}$ and $LPB_{i,j,k}$. The term $\epsilon_{i,j,k,l}$ is the influence on the result of everything that could not be assigned or of random factors.

TABLE V
ANOVA III FOR THE ANALYSIS OF THE MAIN FACTORS IN THE DESIGN OF A CONVOLUTIONAL ORDINAL NEURAL NETWORK FOR THE RETINOPATHY DATASET.

Response variable QWK

Source	S.S.	D.F.	M.S.	F-ratio	Sig.
Model	37.860	9	4.207	1562.840	0.000
L factor	0.057	2	0.029	10.646	0.000
P factor	0.121	2	0.060	22.468	0.000
LP factors	0.057	4	0.014	5.261	0.001
Error	0.339	126	0.003		
Total	38.199	135			

$QWK_{i,j,k,l}$ is the quadratic weighted kappa measure, the response variable used to perform the statistical analysis.

We consider some hypotheses testing where the null hypothesis is proposed that each term of the above equation is independent of the levels involved. The hypotheses for the levels of the L factor are

$$H_0 \equiv L_1 = L_2 = L_3$$

$$H_1 \equiv \text{ some } L_i \text{ is different}$$

The same hypotheses are made for the other factors. In this way, we test in the null hypothesis that all of the population means are equal against an alternative hypothesis that there is at least one mean that is not equal to the others.

The hypothesis associated with the interaction between L and P is

$$H_0 \equiv LP_{i,j} = 0 \quad \forall i, j$$

$$H_1 \equiv \exists LP_{i,j} \neq 0$$

Similar hypotheses can be assumed for the interaction between the other factors.

The analysis of variance table containing the sum of squares, degrees of freedom, mean square, test statistics and significance level represent the initial study in a compact form, where non-significative factors and interactions have been removed (p-value > 0.05). These factors and interactions take part of the error component now.

The results of the ANOVA III test for the Diabetic Retinopathy dataset are resumed in Table V. There are significative differences in the means of QWK depending on the link function and also depending on the learning rate for $\alpha = 0.05$ (p-value $= 0.000$). Moreover, an interaction between the link function and the learning rate can be recognised (p-value $= 0.001$). It means that the learning rate and the link functions have a significative impact on the optimization algorithm results.

As Table V showed that there exist significative differences between the means, we are now analysing those differences. A post-hoc multiple comparison test has been performed on the mean QWK obtained. An HSD Tukey's test [30] was made under the null hypothesis that the variance of the error of the dependent variable is the same between groups. The results over the test set show that the best link function is the complementary log-log, though it doesn't have significative differences with the probit function (p-value $= 0.248$). There are differences between the complementary log-log and the probit function concerning the logit function (p-values 0.000 and 0.011).

TABLE VI
ANOVA III FOR THE ANALYSIS OF THE MAIN FACTORS IN THE DESIGN OF A CONVOLUTIONAL ORDINAL NEURAL NETWORK FOR THE ADIENCE DATASET.

Response variable QWK

Source	S.S.	D.F.	M.S.	F-ratio	Sig.
Model	84.372	27	3.125	4414.006	0.000
L factor	0.156	2	0.078	110.103	0.000
P factor	0.925	2	0.462	653.163	0.000
B factor	0.040	2	0.020	28.218	0.000
LP factors	0.284	4	0.071	100.118	0.000
LB factors	0.008	4	0.002	2.837	0.028
PB factors	0.026	4	0.007	9.267	0.000
LPB factors	0.021	8	0.003	3.728	0.001
Error	0.076	108	0.001		
Total	84.449	135			

The results have been obtained following a similar methodology with the different values of the learning parameter including the HSD Tukey's test and the learning parameter ranking. Significative differences can be observed in the mean values for $\eta = 10^{-3}$ with respect to the other values of the parameter (p-values 0.023 for 10^{-4} and 0.000 for 10^{-2}). Also, there are differences for $\eta = 10^{-4}$ concerning $\eta = 10^{-2}$ (p-value $= 0.000$).

The results of the ANOVA III test for the Adience dataset are shown in Table VI. First, we observe that there exist significative differences in the mean QWK concerning the three factors (p-value $= 0.000$). Secondly, we found interactions between all the pairs of factors and between all the three factors together (p-values 0.000, 0.000, 0.000 and 0.001). It means that the joint action of two or three factors significatively affects the results obtained by the algorithm.

As we did for the DR dataset, a post-hoc multiple comparison test has been performed on the average QWK obtained for Adience. Under the null hypothesis that the variance of the error of the dependent variable is the same between groups, an HSD Tukey's test has been done. The results showed that the best link function is the logit one, in this case, with 0.827 as mean QWK value. Also, there are significative differences between this function and the probit link (p-value $= 0.000$), that has a mean QWK value of 0.781. Both links functions present significative differences concerning the complementary log-log function, that has a mean value of 0.743.

The same methodology was followed for the learning rate and the batch size factors. The test reported significative differences between $\eta = 10^{-3}$ (mean value 0.864) and the rest of values of this parameter (p-value $= 0.000$ for both values), being this one the best value for this factor. The best value after 10^{-3} is 10^{-2}, which have a mean QWK value of 0.818. Both values present significative differences with $\eta = 10^{-4}$, that has a mean QWK value of 0.670 (p-value $= 0.000$).

Lastly, the best value for the batch size parameter is 128 (mean value 0.802) as it has significative differences with the rest of values (p-values 0.026 and 0.000). The second best value is 256 (mean 0.788), and it has significative differences with 64 (p-value = 0.000), that has a mean value of 0.761.

However, the interactions between these factors made the configuration that uses a logit link, $\eta = 10^{-3}$ and batch size of 64, the best configuration. It obtained a mean QWK value of 0.940 for validation and 0.881 for test. The same parameters, but using the probit link, achieves the second best result (0.874). The standard deviation is very low for both cases.

To sum up, the results showed that the best parameter configuration depends on the problem that is being solved. The optimal value for the batch size and the optimal link function are not the same for Retinopathy and Adience datasets. These results highlight the importance of adjusting the hyper-parameters for each problem that is treated instead of trying to find an optimal configuration for all the datasets. However, the best learning rate for both datasets were 10^{-3}. It is recommended to use this value for future datasets. The best batch size for DR was 10 while the best value for Adience was 128. Both values are the intermediate value that we have tried. Finally, there are more interactions between the three factors for the Adience dataset than for the Diabetic Retinopathy one. This highlights the importance of making experimental designs associated with each dataset to determine the best value for each factor.

D. Comparison with nominal method and previous works

The same experiments described in Section V were repeated using the cross-entropy instead of the QWK as loss function for the optimizer and the softmax function instead of the Proportional Odds Model for the output of the network. The evaluation metric remains the same in order to be able to compare. There are some parameter configurations where the training process gets stagnated and a very low QWK is obtained. As we saw in Sections VI-A and VI-B, this problem is not present when using the ordinal method.

For the Diabetic Retinopathy dataset, the best mean value of QWK was 0.497 and was obtained when using a batch size of 10 and a learning rate of 10^{-4}.

In the case of Adience dataset, the highest QWK was 0.787 and was achieved with a batch size of 64 and a learning rate of 10^{-3}.

Finally, a comparison of the best results for each dataset for ordinal and nominal cases and previous works is shown in tables VII and VIII. Also, it is compared with previous works. The proposed ordinal model outperforms all the other alternatives in terms of QWK. The performance increase of POM over softmax reaches 16.8% for Diabetic Retinopathy and 11.9% for Adience dataset. The enhancement of the ordinal method for Retinopathy dataset is more notable than the rise for Adience dataset. It seems that the method proposed in this work offers a more significant improvement as the given problem complexity increases: it is harder to detect the diabetes disease from a retina image than predicting the age range of a person.

TABLE VII
COMPARISON BETWEEN THE BEST RESULTS OF NOMINAL, ORDINAL AND PREVIOUS WORKS FOR THE DIABETIC RETINOPATHY DATASET.

Method	$\overline{QWK}_{(SD)}$	$\overline{CCR}_{(SD)}$	$\overline{\text{1-off}}_{(SD)}$
Ordinal proposed	$0.582_{(0.016)}$	$0.681_{(0.067)}$	$0.748_{(0.071)}$
Nominal proposed	$0.498_{(0.011)}$	$0.692_{(0.012)}$	$0.854_{(0.006)}$
J. Torre et al. [18]	$0.537_{(-)}$	-	-
À. Nebot et al. [41]	$0.555_{(-)}$	-	-

TABLE VIII
COMPARISON BETWEEN THE BEST RESULTS OF NOMINAL, ORDINAL AND PREVIOUS WORKS FOR THE ADIENCE DATASET.

Method	$\overline{QWK}_{(SD)}$	$\overline{CCR}_{(SD)}$	$\overline{\text{1-off}}_{(SD)}$
Ordinal proposed	$0.881_{(0.005)}$	$0.519_{(0.013)}$	$0.894_{(0.005)}$
Nominal proposed	$0.787_{(0.004)}$	$0.458_{(0.008)}$	$0.800_{(0.007)}$
E. Eidinger et al. [32]	-	$0.451_{(0.026)}$	$0.807_{(0.011)}$
J.-C. Chen et al. [33]	-	$0.529_{(0.060)}$	$0.885_{(0.022)}$
G. Levi et al. [34]	-	$0.507_{(0.051)}$	$0.847_{(0.022)}$

In the case of DR, the nominal method obtained a higher CCR and 1-off accuracy value because of the important dataset unbalance. It is possible to achieve a high accuracy classifying all the examples in the majority class. However, as we are working with an ordinal problem, it is preferable to achieve a good QWK value rather than a high accuracy. This effect is not seen in the Adience dataset as it is less unbalanced.

VII. CONCLUSIONS

In this section, the conclusions obtained from this work are exposed. The first thing that we have noticed is that the optimal values for the different parameters considered are problem-dependant.

The complementary log-log function offers the best results in Diabetic Retinopathy dataset while the logit link is the best result for the Adience dataset. These results provide an opportunity for exploring new generalised link functions that can suit any problem.

The best value for the learning rate parameter for both datasets is $\eta = 10^{-3}$. It can be considered a good value for this parameter when training the model with new datasets.

Both datasets have obtained the best performance with the intermedium batch size: 10 for Diabetic Retinopathy and 128 for Adience.

Also, the statistical tests reported that there are relevant interactions between the three factors that we have take into account. The results highlight the importance of making an experimental design were all of these parameters are adjusted for each problem.

The proposed POM has improved the performance of the deep network compared to the model that uses the softmax function and the models proposed in previous works. Also, it reduces the chance that the model gets stuck when training with some parameter configurations.

So, the most significant improvements of these link functions are the performance increase, the reduction of the number of parameters configurations that should be tried to find the best one and the prevention of the over-fitting and the stagnation.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[2] D. Cireşan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," *arXiv preprint arXiv:1202.2745*, 2012.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[5] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal processing magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[6] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[7] X. Jiang, Y. Pang, X. Li, and J. Pan, "Speed up deep neural network based pedestrian detection by sharing features across multi-scale models," *Neurocomputing*, vol. 185, pp. 163–170, 2016.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[9] J. Yu, B. Zhang, Z. Kuang, D. Lin, and J. Fan, "iprivacy: image privacy protection by identifying sensitive objects via deep multi-task learning," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 5, pp. 1005–1016, 2017.

[10] C. Hong, J. Yu, J. Wan, D. Tao, and M. Wang, "Multimodal deep autoencoder for human pose recovery," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5659–5670, 2015.

[11] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[12] Z. Li and J. Tang, "Weakly supervised deep metric learning for community-contributed image retrieval," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 1989–1999, 2015.

[13] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European conference on computer vision*. Springer, 2014, pp. 184–199.

[14] Y. Sun, X. Wang, and X. Tang, "Deep convolutional network cascade for facial point detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 3476–3483.

[15] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[16] J. Leonard and M. Kramer, "Improvement of the backpropagation algorithm for training neural networks," *Computers & Chemical Engineering*, vol. 14, no. 3, pp. 337–341, 1990.

[17] X.-H. Yu, G.-A. Chen, and S.-X. Cheng, "Dynamic learning rate optimization of the backpropagation algorithm," *IEEE Transactions on Neural Networks*, vol. 6, no. 3, pp. 669–677, 1995.

[18] J. de la Torre, D. Puig, and A. Valls, "Weighted kappa loss function for multi-class classification of ordinal data in deep learning," *Pattern Recognition Letters*, vol. 105, pp. 144–154, 2018.

[19] A. Senior, G. Heigold, K. Yang *et al.*, "An empirical study of learning rates in deep neural networks for speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6724–6728.

[20] L. N. Smith, "Cyclical learning rates for training neural networks," in *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*. IEEE, 2017, pp. 464–472.

[21] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[22] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," *arXiv preprint arXiv:1609.04836*, 2016.

[23] P. M. Radiuk, "Impact of training set batch size on the performance of convolutional neural networks for diverse datasets," *Information Technology and Management Science*, vol. 20, no. 1, pp. 20–24, 2017.

[24] M. Cruz-Ramírez, C. Hervás-Martínez, J. Sánchez-Monedero, and P. A. Gutiérrez, "Metrics to guide a multi-objective evolutionary algorithm for ordinal classification," *Neurocomputing*, vol. 135, pp. 21–31, 2014.

[25] N. Mehdiyev, D. Enke, P. Fettke, and P. Loos, "Evaluating forecasting methods by considering different accuracy measures," *Procedia Computer Science*, vol. 95, pp. 264–271, 2016.

[26] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and psychological measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[27] P. A. Gutierrez, M. Perez-Ortiz, J. Sanchez-Monedero, F. Fernandez-Navarro, and C. Hervas-Martinez, "Ordinal regression methods: survey and experimental study," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 1, pp. 127–146, 2016.

[28] A. Agresti, *Analysis of ordinal categorical data*. John Wiley & Sons, 2010, vol. 656.

[29] R. G. Miller Jr, *Beyond ANOVA: basics of applied statistics*. Chapman and Hall/CRC, 1997.

[30] J. W. Tukey, "Comparing individual means in the analysis of variance," *Biometrics*, pp. 99–114, 1949.

[31] C. Beckham and C. Pal, "Unimodal probability distributions for deep ordinal classification," *arXiv preprint arXiv:1705.05278*, 2017.

[32] E. Eidinger, R. Enbar, and T. Hassner, "Age and gender estimation of unfiltered faces," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 12, pp. 2170–2179, 2014.

[33] J.-C. Chen, A. Kumar, R. Ranjan, V. M. Patel, A. Alavi, and R. Chellappa, "A cascaded convolutional neural network for age estimation of unconstrained faces," in *Biometrics Theory, Applications and Systems (BTAS), 2016 IEEE 8th International Conference on*. IEEE, 2016, pp. 1–8.

[34] G. Levi and T. Hassner, "Age and gender classification using convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 34–42.

[35] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[36] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning*, 2013, pp. 1310–1318.

[37] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[38] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.

[39] P. McCullagh, "Regression models for ordinal data," *Journal of the royal statistical society. Series B (Methodological)*, pp. 109–142, 1980.

[40] R. Herbrich, "Large margin rank boundaries for ordinal regression," *Advances in large margin classifiers*, pp. 115–132, 2000.

[41] À. Nebot *et al.*, "Diabetic retinopathy detection through image analysis using deep convolutional neural networks," in *Artificial Intelligence Research and Development: Proceedings of the 19th International Conference of the Catalan Association for Artificial Intelligence, Barcelona, Catalonia, Spain, October 19-21, 2016*. IOS press, 2016, p. 58.

[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[43] D. A. Van Dyk and X.-L. Meng, "The art of data augmentation," *Journal of Computational and Graphical Statistics*, vol. 10, no. 1, pp. 1–50, 2001.

[44] A. Ben-David, "Comparison of classification accuracy using cohen's weighted kappa," *Expert Systems with Applications*, vol. 34, no. 2, pp. 825–832, 2008.

[45] Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it? a new look at signal fidelity measures," *IEEE signal processing magazine*, vol. 26, no. 1, pp. 98–117, 2009.