Pattern Recognition Letters

Authorship Confirmation

Please save a copy of this file, complete and upload as the "Confirmation of Authorship" file.

As corresponding author I, Jordi de la Torre, hereby confirm on behalf of all authors that:

1. This manuscript, or a large part of it, has not been published, was not, and is not being submitted to any other journal.
2. If presented at or submitted to or published at a conference(s), the conference(s) is (are) identified and substantial justification for re-publication is presented below. A copy of conference paper(s) is(are) uploaded with the manuscript.
3. If the manuscript appears as a preprint anywhere on the web, e.g. arXiv, etc., it is identified below. The preprint should include a statement that the paper is under consideration at Pattern Recognition Letters.
4. All text and graphics, except for those marked with sources, are original works of the authors, and all necessary permissions for publication were secured prior to submission of the manuscript.
5. All authors each made a significant contribution to the research reported and have read and approved the submitted manuscript.

Signature____ Date______________________

List any pre-prints:

Relevant Conference publication(s) (submitted, accepted, or published):

Justification for re-publication:

Graphical Abstract (Optional)

To create your abstract, please type over the instructions in the template box below. Fonts or abstract dimensions should not be changed or altered.

Type the title of your article here
Author's names here

This is the dummy text for graphical abstract. This is the dummy text for graphical abstract.

Research Highlights (Required)

To create your highlights, please type the highlights against each `\item` command.

It should be short collection of bullet points that convey the core findings of the article. It should include 3 to 5 bullet points (maximum 85 characters, including spaces, per bullet point.)

- Formulation of a new loss function for the optimization of multi-class classification of ordinal data based on the weighted kappa index to be used in supervised deep learning architectures.
- Comparison of the performance achieved between the new loss and the standard logarithmic loss function using a diabetic retinopathy image classification problem.
- Check the stability of the new loss function using different input and batch sizes.
-
-

Pattern Recognition Letters
journal homepage: www.elsevier.com

Weighted kappa loss function for multi-class classification of ordinal data in deep learning

Jordi de la Torre[a], Domenec Puig[a], Aida Valls[a,**]

[a]*Departament d'Enginyeria Informàtica i Matemàtiques*
Universitat Rovira i Virgili
Avinguda Països Catalans, 26
ES-43007 Tarragona

ABSTRACT

Weighted Kappa is a index of reference used in many diagnosis systems to compare the agreement between different raters. This index can be also used to compare the goodness of a machine-learning based classification method against the results gotten from a consensus expert group. On the other hand, deep learning has achieved in the last years a great importance as a machine learning method for designing classification algorithms also for medical diagnosis. In this paper we explore the direct use of a Weighted Kappa loss function for the optimization of classification ratings, where one ore more underlying causes induce some pr-established ordering of the classes to predict. We show that for the case of diabetic retinopathy image classification, better results can be obtained from the direct optimization of Kappa over the results obtained from the usage of a multi-class logarithmic loss function.

1. Introduction

Deep Learning methods have been used extensively in the last years for many automatic classification tasks. The scheme used for images, is based on extracting the important features with a set of convolutional layers and after that make a final classification with them using a set of fully connected layers. A last soft-max output layer gives as a result the predicted output probabilities of the set of classes the model. During training the model parameters are changed using a gradient based optimization algorithm, minimizing a predefined loss function. For multi-class classification the standardized loss function to use is the logarithmic loss (log-loss).

On the other hand, weighted kappa index is used in medical diagnosis systems for measuring the level of agreement between raters (Hripcsak and Heitjan, 2002), in the applications where some underlying factors present on images or data, need to be interpreted in order to infer from them a diagnose. In this interpretation normally is present some level of subjectivity that make sometimes the conclusions of different experts to differ. Weighted kappa (κ) is able to measure the level of discrepancy of a set of diagnosis made by different raters over the same population (Viera et al., 2005). Depending on the value of the index, the strength of agreement between the raters can be evaluated (see table 1).

Examples of the usage of the κ index for measuring inter-rater agreement are the measure of reliability in ultrasound scans interpretation (Hintz et al., 2007), evaluation of expert agreement in diagnosis of glaucoma (Varma et al., 1992), evaluation of reliability of radiographic assessment (Günther and Sun, 1999), inter-observer agreement evaluation in diabetic retinopathy detection (Patra et al., 2009), between many others. κ takes into account the pr-established ordering of the classes and penalizes the erroneous predictions in function of the distance. In that way, a failure in a prediction that is close to the real category is considered better than a prediction that is farthest.

This index can be used also to measure the goodness of the prediction given by a machine learning method. The values of the prediction can be compared against the correct values reported by a human experts consensus group. In that way, the index compares the values predicted by the model with the considered "true value" coming from the consensus of a human experts group.

In this paper, we study the direct optimization of κ, using it not only as a evaluation function but also as loss function. We use a diabetic retinopathy image classification problem to

[**] Corresponding author: Tel.: +34-977-559-708; fax: +34-977-559-699; *e-mail:* aida.valls@urv.cat (Aida Valls)

Table 1. Table interpretation of Weighted Kappa, after Landis & Koch (1977)

QWK	Strength of agreement
<0.20	Poor
0.21-0.40	Fair
0.41-0.60	Moderate
0.61-0.80	Good
0.81-1.00	Very good

check its stability and to compare the performance of the results obtained from the use of the standard logarithmic loss against the results obtained from the optimization of κ.

The study is organized as follows: we present the background of this work showing the standardized loss function used for classification, next we propose the new cost function for multi-class classification of ordinal data (ordinal regression) with all the mathematical equations required for the optimization, we define the experiments, the results obtained and finally present the conclusions.

2. Deep learning method

Supervised deep learning techniques are used extensively for many automatic classification tasks. In the case of images, most of the state of the art methods are based on the use of deep convolutional neural networks (CNN). These techniques are focused on learning multiple levels of representation and abstraction that help to make sense of the hidden information in data such as images. In this way, having a complete set of correctly classified images and without any a priori understanding of the features, the system is able to learn the properties of the image that minimize a defined cost function that is direct or indirectly related with the classification score index to optimize (see fig. 2).

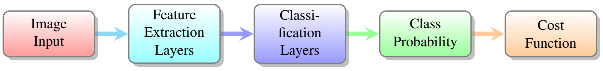

Fig. 1. High level description of a deep learning image classification scheme

For image classification a convolutional neural network is used for making the automatic feature extraction followed by a one or more classification layers (normally fully connected layers) and a final output layer formed by as many outputs as classes to predict. It is usual to have in such an output layer a soft-max function that represents the output probability of every class to predict. Normally, the chosen class is the one with the highest value of probability in the output layer. This neural network architecture has a set of parameters on every layer that have to be optimized to achieve the most accurate prediction of the data. This is done by optimizing a function of the output variables, called loss function. This function depends on the output probabilities given by the model and is defined in a way that minimizing it maximizes the probability of the correct class. If the defined function is differentiable with respect to the output variables then is possible to apply a gradient descent based algorithm to optimize the function. In every optimization step the classification derivative of the loss function is calculated and back-propagated through the network. The parameters are updated accordingly to the optimization algorithm in order to reduce de discrepancy between the output of the model and the true value given by the known data.

Multi-class classification is addressed mainly by the optimization of the logarithmic loss function (see eq. 1). This function is very easy to optimize using gradient descent methods due to the simplicity of its derivatives, its numerical stability and experimental tested validity (Goodfellow et al., 2016). Additionally, the logarithmic loss has a very robust probabilistic foundation: minimizing it is the same as minimizing the logarithmic likelihood, that is equivalent to do a maximum likelihood estimation (MLE) or equivalently, to find the maximum a posteriori probability (MAP), given a uniform prior (Murphy, 2012). This function does not encode any prior information about the classes and this fact make this function a general purpose function that performs well in many applications.

$$\mathscr{L} = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} t_i \log y_i^{(c)} \tag{1}$$

Where:

N: is the number of samples

t_i: is 1 for the correct class of sample i and 0 elsewhere

C: is the number of classes

On the other hand, there are applications where we have a priori information about some property of the classes. For example, in the case of ordinal regression, the different categories are sorted in a predefined way that intends to classify into categories the gradation of some intrinsic information. When the log-loss is applied for the classification in this problems, it has to learn from the data such a pr-established ordering. In those cases, its general capability can become disadvantage. A loss that encodes such a prior information about the intrinsic property of the classes can perform better or faster due to the fact that does not require to learn it from the available data.

3. QWK Loss Function

In this section we present our contribution to the optimization of CNN using weighted kappa. Weighted Kappa (see eq. 2) is normally used as an index to measure the inter-rating agreement between a set of different classes where these categories have a predefined ordering in such a way that the property that we want to categorize is a high level abstraction of the level of some sort of intrinsic information that we want to extract from the categorization.

$$\kappa = 1 - \frac{\sum_{i,j}\omega_{i,j}O_{i,j}}{\sum_{i,j}\omega_{i,j}E_{i,j}} \tag{2}$$

Where:

C: is the number of classes

$i, j \in \{1, 2, ..., C\}$

$O_{i,j}$: the number of observations that received a rating of i by observer A (prediction) and a rating of j by observer B ("true value").

$E_{i,j}$: outer product between each rater's histogram vector of ratings (prediction and "true value"), normalized such that E and O have the same sum.

$\omega_{i,j}$: weight penalization for every pair of classes. Frequently used $\omega_{i,j} = \frac{(i-j)^n}{(C-1)^n}$. For linear penalization $n = 1$. For quadratic penalization (more commonly used and the studied in this paper): $n = 2$.

This index establishes a penalization when there is a discrepancy between the raters that depends on the distance between both predictions. In the case of the Quadratic Weighted Kappa (QWK) the penalization of the discrepancy grows quadratically with the distance between the two ratings. If the predicted classes of both raters is the same, we say the there is an *absolute* concordance between both raters and no penalization is applied. When the predicted classes are different, we say that the there is *relative* concordance between both raters and there is a penalization in the calculation of the inter-rating index that is proportional, in the case of QWK, to the square of the distance between both predictions. This discrepancies are calculated for all the N items of the k mutually exclusive categories and summed over the N items. The penalizing term is normalized dividing their value by the expected discrepancy, obtaining as a result a value between -1 and 1. A value of 1 of the index would indicate a perfect agreement between both raters, -1 a perfect symmetric disagreement between the classes and 0 a random evaluation method.

3.1. Mathematical foundation

The optimization problem that we want to solve in order to maximize the probabilities of the true classes of our model using the κ function is presented in equation 3

$$\underset{x}{\text{maximize}} \quad \kappa = 1 - \frac{\sum_{i,j} \omega_{i,j} O_{i,j}}{\sum_{i,j} \omega_{i,j} E_{i,j}} \quad \kappa \in [-1, 1] \tag{3}$$

This problem can be redefined in the form of equation 4 in order change the maximization problem to a minimization one. Loss function are defined as minimization problems, that's why we redefine the problem to solve it as a minimization one. Taking the log we change the output range and increment the numerical stability of the problem.

$$\underset{x}{\text{minimize}} \quad \mathscr{L} = \log(1-\kappa) \quad \mathscr{L} \in (-\infty, \log 2] \tag{4}$$

As we want to use the index as a optimization function, we have to take into account that our model will not report a predicted class as an output but a probability for every class. That's why the observed values given by our model will not be integer values but real ones. In eq. 5 we show the expression of the numerator of the κ function for taking into account the probabilities of every prediction. In eq. 6 we also show the redefinition of the denominator of the same index in order to take into account the probabilities given by the model.

$$\mathscr{N} = \sum_{i,j} \omega_{i,j} O_{i,j} = \sum_{k=1}^{N} \sum_{c=1}^{C} \omega_{t_k,c} P_c(X_k) \tag{5}$$

$$\mathscr{D} = \sum_{i,j} \omega_{i,j} E_{i,j} = \sum_{i=1}^{C} \hat{N}_i \sum_{j=1}^{C} \left(\omega_{i,j} \sum_{k=1}^{N} P_j(X_k) \right) \tag{6}$$

Where:

X_k: input data of sample k

$E_{i,j} = \frac{N_i \sum_{k=1}^{N} P_j(X_k)}{N} = \hat{N}_i \sum_{k=1}^{N} P_j(X_k)$

N: number of samples

N_i: number of samples of class i

$\hat{N}_i = \frac{N_i}{N}$

t_k: correct class number for sample k

$P_c(X_k)$: probability that sample k is of class C given that the true class in t_k

3.2. Partial derivatives of the QWK loss function

For solving this optimization problem using any gradient descent based algorithm, we need to derive the partial derivatives of the loss function with respect to the output variables of the network.

For the case minimizing the loss function $\mathscr{L} = \log \frac{\mathscr{N}}{\mathscr{D}}$, the derivative takes the next form:

$$\frac{\partial \mathscr{L}}{\partial y_m} = \frac{1}{\mathscr{N}} \frac{\partial \mathscr{N}}{\partial y_m} - \frac{1}{\mathscr{D}} \frac{\partial \mathscr{D}}{\partial y_m} \tag{7}$$

And $\frac{\partial \mathscr{N}}{\partial y_m}$ and $\frac{\partial \mathscr{D}}{\partial y_m}$ can be calculated with the next expressions:

$$\frac{\partial \mathscr{N}}{\partial y_m(X_k)} = \omega_{t_k m} \tag{8}$$

$$\frac{\partial \mathscr{D}}{\partial y_m(X_k)} = \sum_{i=1}^{C} \hat{N}_i \omega_{i,m} \tag{9}$$

Where:

$m \in \{1, 2, ..., C\}$

That in its array form can be rewritten as:

$$\frac{\partial \mathscr{N}}{\partial y_m} = \begin{pmatrix} \omega_{t_1,1} & \omega_{t_1,2} & \cdots & \cdots & \omega_{t_1,C} \\ \omega_{t_2,1} & \omega_{t_2,2} & \cdots & \cdots & \omega_{t_2,C} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \omega_{t_N,1} & \omega_{t_N,2} & \cdots & \cdots & \omega_{t_N,C} \end{pmatrix} \tag{10}$$

$$\frac{\partial \mathscr{D}}{\partial y_m} = \begin{pmatrix} \sum_{i=1}^{C} \hat{N}_i \omega_{1,i} & \cdots & \cdots & \cdots & \sum_{i=1}^{C} \hat{N}_i \omega_{C,i} \\ \sum_{i=1}^{C} \hat{N}_i \omega_{1,i} & \cdots & \cdots & \cdots & \sum_{i=1}^{C} \hat{N}_i \omega_{C,i} \\ \cdots & ::: & \cdots & \cdots & \cdots \\ \sum_{i=1}^{C} \hat{N}_i \omega_{1,i} & \cdots & \cdots & \cdots & \sum_{i=1}^{C} \hat{N}_i \omega_{C,i} \end{pmatrix} \tag{11}$$

With the definition of the loss function and its derivatives we have all the equations required to apply to our model any first order optimization algorithm to fit the data.

4. Experiments

The finality of the experiments of this paper is to test the proposed optimization loss function against a real example. We have chosen a diabetic retinopathy image classification task. Here below, we define the problem, describe the data and finally define the procedure the we have used to test the performance of the new loss.

4.1. Classification Problem Definition

Diabetic Retinopathy (DR) is a leading disabling chronic disease and one of the main causes of blindness and visual impairment in developed countries for diabetic patients. Studies reported that 90% of the cases can be prevented through early detection and treatment Torrents-Barrena et al. (2015). Eye screening through retinal images is used by physicians to detect the lesions related with this disease. Due to the increasing number of diabetic people, the amount of images to be manually analyzed is becoming unaffordable. Moreover, training new personnel for this type of image-based diagnosis is long, because it requires to acquire expertise by daily practice.

In 2003 the medical community established a standardized classification based on four severity stages CP et al. (2003) determined by the type and number of lesions (as microaneurysms, hemorrhages and exudates): class 0 referring to no apparent retinopathy, class 1 as a Mild Non-Proliferative Diabetic Retinopathy (NPDR), class 2 as Moderate NPDR, class 3 as a Severe NPDR and class 4 as a Proliferative DR.

The problem consist on optimize a deep convolutional neural network to maximize the classification rates over a test set of never seen before images. The generalization capability will be scored against the quadratic weighted kappa over the test set.

4.2. Data

The dataset used in this work consists of two independent high resolution image sets (train and test). For every patient right and left eye images are reported. All the images are classified by ophthalmologists according to the standard severity scale presented before CP et al. (2003). The images are taken in variable conditions: by different cameras, illumination conditions and resolutions.

The training set contains a total of 35.126 images; 25.810 of class 0, 2.443 of class 1, 5.292 of class 3, 873 of class 3 and 708 of class 4. The test set contains a total of 53.576 images; 39.533 of class 0, 3.762 of class 1, 7.861 of class 2, 1.214 of class 3 and 1.206 of class 4. Notice that it is highly imbalanced.

4.3. The models

In this section we define the models that we have used to compare the performance of the to losses. Four different models have been used, one for every different image size. The idea was to first design smaller models based on restricted image sizes in order to have a small training time that could allow to test the maximum amount of experiments. This big set of previous experiments would allow to restrict the hyper-parameter space. Bigger models, more slower to train, would then be trained with the better selection of hyper-parameters.

All the models follow the scheme defined in fig. 2. The feature extraction layers use all 3x3 convolutions with padding of 1 and stride 1, followed by a batch normalization(Ioffe and Szegedy, 2015) and a ReLU activation function (Dahl et al., 2013). Every two feature layers a 2x2 max-pooling layer of stride 2 is applied for dimensionality reduction. The number of feature layers vary depending on the image size. All the models use an unique classification layer followed by the final output soft-max layer. The number of layers and the total number of parameters of each model are summarized in table 3.

The model used for the 128x128 image case has 1.16 million of parameters, 10 feature lavers of 32/32, 64/64, 128/128, 128/128 filters in every convolution (/ separate the filters for every map size, the commas indicate a 2x2 max-pooling operation) and a 4x4 convolution classification layer of 128 filters. The model used for the 256x256 image case has 1.44 million of parameters, 12 feature lavers of 32/32, 64/64, 128/128, 128/128, 128/128 filters in every convolution and a 4x4 convolution classification layer of 128 filters. The model used for the 384x384 image case has 1.77 million of parameters, 12 feature lavers of 32/32, 64/64, 128/128, 128/128, 128/128 filters in every convolution and a 6x6 convolution as a classification layer of 128 filters. Finally, the model used for the 512x512 image case has 11.3 million of parameters, 12 feature lavers of 16/16, 32/32, 64/64, 128/128, 256/256, 512/512 filters in every convolution, a 5x5 convolution as classification layer of 512 filters, followed by a 4x4 average pooling and a dropout layer (ratio = 0.3).

4.4. Procedure

We define a set of experiments to compare the performance of the model when its parameters are optimized using the log-loss and the QWK loss. We want to check if the encoding of the ordering of the classes helps in the improvement of the generalization results. In order to make the comparison possible, every model will be optimized with both functions. The final generalization capability of the same model optimized with both functions will be compared.

As deep learning is a computer intensive task, the experiments have been done using models of different input resolution images: 128x128, 256x256, 384x384 and 512x512. The lower time required to train low resolution images allow to run more experiments and serve to not only make a first validation of the new cost function but also to make a first hyper-parameters selection that can serve later on to validate the new cost function against higher resolution images using less time consuming experiments.

The original training set is split in two random subsets: one with 90% of the data and other with 10%. The last one is used as a validation set for hyper-parameter selection. The value of QWK is calculated every epoch either for the training or the validation set.

Notice that the image set is highly imbalanced. In order facilitate the training the training set is artificially equalized using data augmentation techniques based on 0-360° random rotation, X and Y mirroring and contrast and brightness random sampling.

A random initialization based in the Kaiming&He approach (He et al., 2015) is used for all the networks. All models are optimized using Adam (Kingma and Ba, 2014) over batches. We study different learning rates in order to find the optimal one for each loss function. We use a batch-based optimization algorithm. QWK loss function uses a normalization term that normalizes over the sample set. Presumably this loss function has to be more sensible to small batches that the logarithmic loss due to this normalization term, that's why different batch sizes are tested. For every batch, the images are chosen randomly from the training set, with repetition. Data augmentation techniques are applied to augment the diversity of the classes (random rotations and brightness and contrast modifications). The epoch size is set to maintain fixed the total number of images per epoch to 100.000. This value is approximately the number of images required to sample to ensure that all of them has been selected every epoch. Also, using different batch sizes, the number of updates per epoch of the parameters of the neural network change. In the case of small batch sizes the number of updates per epoch is greater than the case of big batch sizes. Studying different batch sizes we would also explore different number of parameter updates. All the tests were run for 100 epochs.

5. Results

Table 2 shows the experiments using both losses with different learning rates and batch sizes. QWK score is calculated over the validation set for comparison between the results obtained from the direct optimization of the function and the standardized indirect method of using the log-loss function.

In figure 3 we show a graphical representation of the maximum value of QWK on the test set for the different tested configurations. We can see that directly optimizing QWK gives consistently better results than optimizing log-loss. Only in the case of very small batch sizes (in this case, BS = 5) log-loss performs better than QWK. This is probably due to the fact that QWK uses a normalization term in the denominator that with not big enough batch sizes could cause instabilities in the gradient that affect the performance. In any case, even in those cases the results obtained with the log-loss are worse that those achieved using the best QWK configuration.

The optimal batch size for solving this diabetic retinopathy classification is of 5 in the case of using the log-loss and 15 in the case of the QWK-loss. For greater values of this hyperparameter, the increase of precision in the calculation of the gradient does not gives any advantage in the optimization. In the case of the log-loss a lower precision in the calculation of the gradient works better. This fact could probably is due to the fact that this imprecision in the calculation increases the stochasticity of the prediction and due to the fact that we are not optimizing directly the metrics index this make possible to explore adjacent zones that could work better for improving the metrics index that the ones that specifically improve the log-loss. Additionally, although smaller batch sizes give worse approximations of the gradient, the number of updates per epoch of the parameters of the model is greater. This seems to be an advantage in this case.

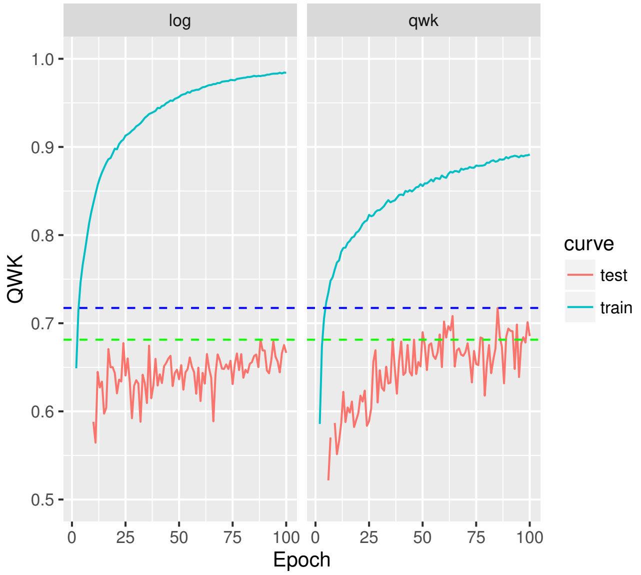

Fig. 2. Evolution of QWK values during training for logarithmic and QWK loss functions (512x512 input image case)

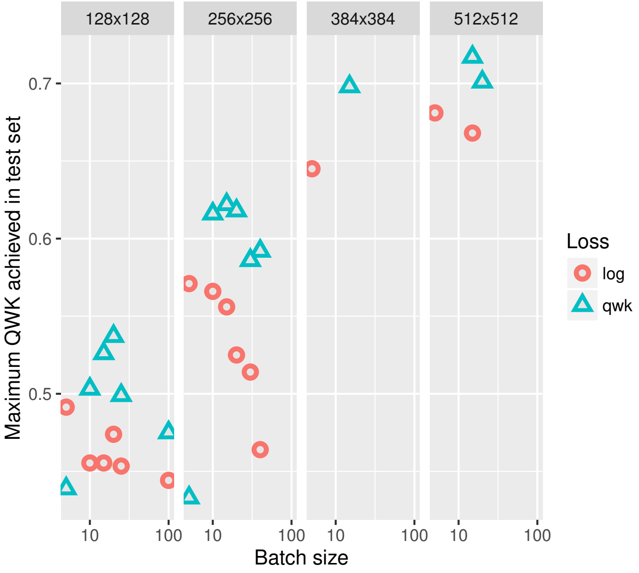

Fig. 3. Best classification results achieved over test set for each loss function

6. Conclusions

We presented a new loss function for ordinal regression problems based on the optimization of the weighted kappa index. In contrast to the logarithmic loss that uses a uniform prior over the set of classes, this new loss defines a penalization over the discrepancy that is proportional to a power of the distance (power of 2 in the case of quadratic weighted kappa) that allows to encode the prior known information about the predefined ordering of the classes.

We checked the performance of this new loss function against a diabetic retinopathy multi-class classification problem. The results presented in this paper show that with the direct optimization of the QWK index consistently better generalization

Table 2. Results achieved in different experiments

Input	BS	Loss	LR	$\kappa_{train} 10^3$	$\kappa_{test} 10^3$	Gap	Epoch	Updates 10^{-3}
128	5	log	10^{-5}	771	418	353	78	1560
			10^{-4}	851	**491**	360	73	1460
			10^{-3}	676	418	258	29	580
		qwk	5×10^{-5}	545	402	143	50	1000
			10^{-5}	646	439	207	70	1400
			10^{-4}	497	326	171	31	620
	10	log	10^{-5}	797	397	400	82	820
			10^{-4}	874	455	419	81	810
			10^{-3}	514	336	178	57	570
		qwk	10^{-5}	774	476	298	82	820
			10^{-4}	755	503	252	84	840
			10^{-3}	596	289	307	95	950
	15	log	10^{-5}	803	368	435	79	527
			10^{-4}	899	458	441	95	633
			10^{-3}	868	447	421	80	533
		qwk	5×10^{-5}	715	491	224	77	513
			10^{-4}	800	526	274	77	513
			5×10^{-4}	823	523	300	72	480
	20	log	10^{-4}	896	474	422	79	395
		qwk	10^{-4}	835	**537**	298	93	465
	25	log	10^{-5}	821	315	506	96	384
			10^{-4}	913	453	460	93	372
			10^{-3}	849	382	467	70	280
		qwk	10^{-5}	808	423	385	95	380
			10^{-4}	824	499	325	65	260
			10^{-3}	655	447	208	80	320
	100	log	10^{-4}	929	377	552	98	98
			10^{-3}	947	444	503	99	99
			10^{-2}	842	412	430	67	67
		qwk	10^{-4}	879	450	429	93	93
			10^{-3}	798	455	343	71	713
			10^{-2}	-	-	-	-	-
256	5	log	10^{-4}	871	**571**	300	52	1040
		qwk	10^{-4}	605	433	172	15	300
	10	log	10^{-4}	903	566	337	75	750
		qwk	10^{-4}	832	616	216	70	700
	15	log	10^{-4}	925	556	369	98	653
		qwk	10^{-4}	878	**622**	256	93	620
	20	log	10^{-4}	923	525	398	97	485
		qwk	10^{-4}	891	618	273	97	485
	30	log	10^{-4}	925	514	411	93	310
		qwk	10^{-4}	900	586	314	98	327
	40	log	10^{-4}	922	464	458	93	233
		qwk	10^{-4}	894	592	302	78	195
384	5	log	10^{-4}	863	**641**	222	38	760
	15	qwk	10^{-4}	889	**698**	191	93	620
512	5	log	10^{-4}	980	**681**	299	88	1760
	15	log	10^{-4}	978	668	310	94	626
		qwk	10^{-4}	884	**717**	167	86	573
	20	qwk	10^{-4}	903	701	202	89	445

Table 3. Results achieved with every tested model optimizing with log-loss and the new qwk-loss

Input	Total Layers	Feature Layers	Classific. Layers	Params 10^{-6}	κ_{test}^{qwk}	κ_{test}^{log}	Δ
128x128	12	10	1	1.16	0.537	0.491	9.3 %
256x256	14	12	1	1.44	0.622	0.571	8.9 %
384x384	14	12	1	1.77	0.698	0.663	5.3 %
512x512	14	12	1	11.3	0.717	0.681	5.3 %

results can be achieved over the standard use of the logarithmic loss. Logarithmic loss has to learn the pre-set order of the classes from data and it seems that this comes to be a disadvantage. Results showed that more than a 5% of increment of QWK_{test} scores can be obtained from the direct optimization of the function.

One minor drawback of the new loss is its low performance with very small batch sizes. The experiments show that for the retinopathy classification problem with batch sizes of 5 the performance of the function is lower than using the logarithmic loss. This parameter is for sure problem depend-ant and has to be taken into account as an important hyper-parameter to check in other classification tasks.

Acknowledgments

This work is supported by the URV grant 2014PFR-URV-B2-60 and the Spanish research projects PI15/01150 and PI12/01535 (Instituto de Salud Carlos III). The authors would like to thank to the California Healthcare Foundation and Eye-PACS for providing the images used in this study.

References

CP, W., 3rd, F.F., RE, K., PP, L., CD, A., M, D., D, D., A, K., R, P., JT, V., 2003. Proposed international clinical diabetic retinopathy and diabetic macular edema disease severity scales. Ophthalmology 110(9), 1677–82.

Dahl, G.E., Sainath, T.N., Hinton, G.E., 2013. Improving deep neural networks for LVCSR using rectified linear units and dropout, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8609–8613.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press. `http://www.deeplearningbook.org`.

Günther, K.P., Sun, Y., 1999. Reliability of radiographic assessment in hip and knee osteoarthritis. Osteoarthritis and Cartilage 7, 239–246.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1026–1034.

Hintz, S.R., Slovis, T., Bulas, D., Van Meurs, K.P., Perritt, R., Stevenson, D.K., Poole, W.K., Das, A., Higgins, R.D., Network, N.N.R., et al., 2007. Interobserver reliability and accuracy of cranial ultrasound scanning interpretation in premature infants. The Journal of pediatrics 150, 592–596.

Hripcsak, G., Heitjan, D.F., 2002. Measuring agreement in medical informatics reliability studies. Journal of biomedical informatics 35, 99–110.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: Deep Learning Symposium (NIPS 2015).

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. CoRR abs/1412.6980. URL: `http://arxiv.org/abs/1412.6980`.

Murphy, K.P., 2012. Machine Learning: A Probabilistic Perspective. The MIT Press.

Patra, S., Gomm, E., Macipe, M., Bailey, C., 2009. Interobserver agreement between primary graders and an expert grader in the bristol and weston diabetic retinopathy screening programme: a quality assurance audit. Diabetic Medicine 26, 820–823.

Torrents-Barrena, J., Melendez, J., Valls, A., Romero, P., Puig, D., 2015. Screening for diabetic retinopathy through retinal colour fundus images using convolutional neural networks, in: Artificial Intelligence Research and Development - Proceedings of the 18th International Conference of the Catalan Association for Artificial Intelligence. IOSPress, pp. 259–262.

Varma, R., Steinmann, W.C., Scott, I.U., 1992. Expert agreement in evaluating the optic disc for glaucoma. Ophthalmology 99, 215–221.

Viera, A.J., Garrett, J.M., et al., 2005. Understanding interobserver agreement: the kappa statistic. Fam Med 37, 360–363.