1 Introduction

...

2 Related work

...

3 Experiments

3.1 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 and 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, illumination conditions and 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 [1] 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 256 by 256 pixels. Data augmentation techniques 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 [2], so that the images are 256px 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.

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

3.2 The model

A convolutional neural network (CNN) has been used for both datasets. The architecture of this CNN is presented in the Table 1.

Layer	**Output shape**
Conv_32_3x3	254x254x32
Conv_32_3x3	252x252x32
MaxPool_2x2	126x126x32
Conv_64_3x3	124x124x64
Conv_64_3x3	122x122x64
MaxPool_2x2	61x61x64
Conv_128_3x3	59x59x128
Conv_128_3x3	57x57x128
MaxPool_2x2	28x28x128
Conv_128_3x3	26x26x128
Conv_128_3x3	24x24x128
MaxPool_2x2	12x12x128
Conv_128_4x4	9x9x128
Dense_1_output	1

Table 1: Description of the architecture used in the experiments. For convolutional layers, Conv_N_WxH, where N is the number of filters, W the filter width and H the filter height. Stride is 1 for every convolutional layer. For max pool layers, MaxPool_SxS, where S is the pool size.

Every convolutional layer is followed by an ELU activation layer [3] and a batch normalization [4]. At the output, a Proportional Odds Model (POM) is used with different link functions [5]. 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.

- *Logit.* Logit link function is the most widely used function for Proportional Odds Models. These kind of models are also called Cummulative Logit Models. The logit link is shown in

Eq. 1.

$$logit[P(Y_i \leq j)] = \alpha_j + \beta' x_i, \quad j = 1, ..., c-1 \quad (1)$$

- *Probit.* Probit link function is the inverse of the standard normal cummulative distribution function (cdf). Its expression is shown in Eq. 2.

$$\Phi^{-}1[P(Y \leq j)] = \alpha_j + \beta' x, \quad j = 1, ..., c-1 \quad (2)$$

- *Complementary log-log.* Unlike logit and probit, complementary log-log function is not symmetric. With a continuous predictor x, for example, $P(Y \leq j)$ approaches 0 at a different rate than it approaches 1. Complementary log-log expression is shown in Eq. 3.

$$log[-log[1-P(Y \leq j)]] = \alpha_j + \beta' x, \quad j = 1, ..., c-1 \quad (3)$$

3.3 Procedure

The model is optimized using a batch based first-order optimization algorithm called Adam [6]. 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.

Quadratic Weighted Kappa Loss, that J. de la Torre described in previous work, is considered as loss function for this optimizer as it gives better performance for ordinal classification problems.

Both datasets have been artificially equalised using data augmentation techniques [7][8] based on image cropping and zooming, horizontal and vertical flipping, brightness adjustment and random rotations. 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) [9]. This evaluation measure gives a higher weight to the errors that are further from the correct class.

3.4 Parameters

Three different parameters 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 [10][11]. 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: 5, 10 and 15.

- *Link function.* Different link functions have been used for the POM at the last layer output: logit, probit and complementary log-log.

4 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 average quadratic weighted kappa across these five executions for validation and test values.

4.1 Diabetic Retinopathy

Detailed results for the Diabetic Retinopathy dataset are presented in Table 2.

BS	LR	LF	κ_{val}	κ_{test}
5	10^{-2}	Logit	0.44888	0.4163
		Probit	0.46724	0.45972
		c log-log	0.42854	0.41448
	10^{-3}	Logit	0.56496	0.55356
		Probit	0.57084	0.56392
		c log-log	0.54796	0.53436
	10^{-4}	Logit	0.52884	0.52032
		Probit	0.53802	0.52302
		c log-log	0.53824	0.51974
10	10^{-2}	Logit	0.5497	0.53142
		Probit	0.52124	0.50806
		c log-log	0.4276	0.4227
	10^{-3}	Logit	0.58036	0.57686
		Probit	0.56536	0.5581
		c log-log	**0.5883**	**0.5815**
	10^{-4}	Logit	0.53158	0.5385
		Probit	0.5456	0.54082
		c log-log	0.53928	0.53742
15	10^{-2}	Logit	0.55828	0.55076
		Probit	0.54112	0.5343
		c log-log	0.56676	0.56428
	10^{-3}	Logit	0.55748	0.55106
		Probit	0.58182	0.57894
		c log-log	0.5634	0.55938
	10^{-4}	Logit	0.54686	0.54342
		Probit	0.53258	0.53278
		c log-log	0.539	0.53828

Table 2: Diabetic Retinopathy results. BS stands for Batch Size, LR for Learning Rate and LF for link function.

BS	LR	LF	κ_{val}	κ_{test}
5	10^{-2}	Logit	–	–
		Probit	–	–
		c log-log	–	–
	10^{-3}	Logit	–	–
		Probit	–	–
		c log-log	–	–
	10^{-4}	Logit	–	–
		Probit	–	–
		c log-log	–	–
10	10^{-2}	Logit	–	–
		Probit	–	–
		c log-log	–	–
	10^{-3}	Logit	–	–
		Probit	–	–
		c log-log	–	–
	10^{-4}	Logit	–	–
		Probit	–	–
		c log-log	–	–
15	10^{-2}	Logit	–	–
		Probit	–	–
		c log-log	–	–
	10^{-3}	Logit	–	–
		Probit	–	–
		c log-log	–	–
	10^{-4}	Logit	–	–
		Probit	–	–
		c log-log	–	–

Table 3: Adience results. BS stands for Batch Size, LR for Learning Rate and LF for link function.

4.2 Adience

...

4.3 Statistical analysis

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

The ANalysis Of the VAriance (ANOVA) [12] 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 optimisation algorithm (10^{-2}, 10^{-3} and 10^{-4}) and the batch size (5, 10 and 15).

Following the setup of the previous study, we performed an ANOVA III analysis and multiple comparison tests. The tests show that there is no batch size whose results are significatively better than the results of all other batch sizes. This does not mean

that these differences could not exist for specific numbers of samples per batch. So, in order to determine for each type of function whether a batch size is better than the others, we have performed an ANOVA I analysis - where the only factor is the batch size - and multiple comparison tests.

We denote by $W_{i,j,k}(i = 1, ..., 3; j = 1, ..., 3; k = 1, ..., 3)$ 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 two factors do not act independently and therefore there exists an interaction between them. In this case, the observations fit Eq. 4.

$$\begin{aligned} W_{i,j,k} =& \mu + L_i + P_j + B_k + LP_{i,j} + LB_{i,k} \\ &+ PB_{j,k} + LPB_{i,j,k} + \epsilon_{i,j,k} \end{aligned} \tag{4}$$

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 (10^{-4}, 10^{-3}, 10^{-2}) and B_k is the effect associated with the *k-th* level of the size batch factor (5, 10, 15). The term $LP_{i,j}$ denotes the joint 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 $\epsilon_{i,j,k}$ is the influence on the result of everything that could not be assigned or of random factors. $W_{i,j,k}$ is the quadratic weighted kappa measure, the response variable used to perform the statistical analysis.

References

[1] 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.

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

[3] 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.

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

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

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

[7] 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.

[8] 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.

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

[10] 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.

[11] 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.

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