Calibrated Prediction Intervals for Neural Network Regressors

GIL KEREN[1], NICHOLAS CUMMINS[1], (Member, IEEE), AND BJÖRN SCHULLER[1,2], (Fellow, IEEE)

[1]ZD.B Chair of Embedded Intelligence for Health Care and Wellbeing, University of Augsburg, 86159 Augsburg, Germany
[2]Group on Language, Audio & Music, Imperial College London, London SW7 2AZ, U.K.

Corresponding author: Gil Keren (gil.keren@informatik.uni-augsburg.de)

This work was supported in part by the European Union's Seventh Framework Programme under Grant 338164 (ERC StG iHEARu) and in part by the Horizon 2020 Programme under Grant 688835 (RIA DE-ENIGMA).

ABSTRACT Ongoing developments in neural network models are continually advancing the state-of-the-art in terms of system accuracy. However, the predicted labels should not be regarded as the only core output; also important is a well-calibrated estimate of the prediction uncertainty. Such estimates and their calibration are critical in many practical applications. Despite their obvious aforementioned advantage in relation to accuracy, contemporary neural networks can, generally, be regarded as poorly calibrated and as such do not produce reliable output probability estimates. Furthermore, while post-processing calibration solutions can be found in the relevant literature, these tend to be for systems performing classification. In this regard, we herein present two novel methods for acquiring calibrated predictions intervals for neural network regressors: empirical calibration and temperature scaling. In experiments using different regression tasks from the audio and computer vision domains, we find that both our proposed methods are indeed capable of producing calibrated prediction intervals for neural network regressors with any desired confidence level, a finding that is consistent across all datasets and neural network architectures we experimented with. In addition, we derive an additional practical recommendation for producing more accurate calibrated prediction intervals. We release the source code implementing our proposed methods for computing calibrated predicted intervals.

INDEX TERMS Machine learning, artificial neural networks.

I. INTRODUCTION

Deep learning has undoubtedly improved the state-of-the-art performance of machine learning models across a variety of machine learning applications, in terms of overall system accuracy. In addition, there is an increasing research attention within the deep learning community on estimating prediction uncertainty, i. e., recognizing and quantifying when an output may be incorrect. The estimation of uncertainty can indeed be crucial for a wide range of applications. For example, the decisions made by neural network technology deployed in healthcare settings could have life-threatening consequences. Uncertainty information could therefore act as a guide for clinicians or doctors to seek a potentially life saving advice.

For a regression problem, uncertainty of a model output can be estimated using *prediction intervals* – estimates of the interval in which the target label is expected to lie within a prescribed probability. Standard neural network regressors output a point estimation [1]–[3], from which the estimation of calibrated prediction intervals is a non-trivial task. Other neural network regressors use a technique which poses the regression task as a classification task, with a softmax output that produces a posterior distribution over the output space [4], [5]. Using this method, one could compute prediction intervals for a given confidence level α, simply by taking an interval in the output space that contains α of the posterior probability mass, as illustrated in Figure 1.

However, an interval in the output space that contains α of the posterior probability mass does not have to correspond to a probability of α that the label will fall within this interval's boundaries. For example, a neural network making overconfident predictions may tend to concentrate α of the posterior probability mass in small intervals of the output space,

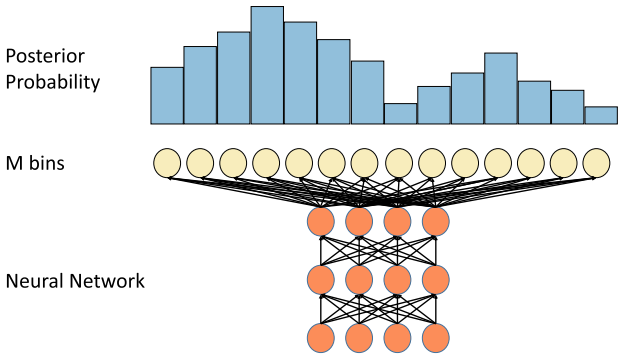

FIGURE 1. A neural network regressor designed as a softmax classifier. By binning the output space into *M* bins, one can design a neural network regressor as a softmax classifier over *M* classes, and derive a posterior distribution over the output space instead of a single point estimate. This allows emitting prediction intervals that contain a prescribed amount of the posterior probability mass. However, we show that the resulting prediction intervals will normally be miscalibrated, i. e., will not correspond to the desired confidence level.

while the probability that these intervals contain the actual labels can be considerably lower. In this case, we say that the prediction intervals are miscalibrated. Recent work has shown that the outputs of modern neural network classifiers are miscalibrated in the sense that posterior class probabilities do not reflect actual correctness probabilities [6]. Therefore, when using neural network regressors that are designed as such classifiers, we expect the resulting prediction intervals to be miscalibrated as well.

Neural network models have not always been considered miscalibrated. Indeed, work presented in [6] and [7] identified pre-modern neural network models as a good learning paradigm in terms of producing well-calibrated probabilities for binary classification tasks. It has been demonstrated that the poor calibration levels observed in more contemporary deep topologies have come about through recent changes in network architecture and training procedures [6]. For example, miscalibration has been associated with increases in model capacity, and has also been observed in networks trained with batch normalization or a minimal amount of weight decay [6].

Despite network calibration being a more recent problem for neural nets, calibration and confidence estimation themselves are not new problems, e. g., [8]–[15]. More recently, a plethora of calibration and uncertainty quantification approaches have been proposed and developed for contemporary neural networks in the wider machine learning community. Bayesian Neural Networks produce a probabilistic relationship between the network input and output [16], [17], but often suffer from tractability issues. Ensemble based approaches, bootstrapping, and Monte Carlo based approaches have also been proposed, for example [18]–[21]. While such approaches can produce calibrated prediction intervals, they often require training and testing a multitude of different individual networks which considerably increases the associated time and computational costs [22]. Closely related to the current work, a range of post-processing calibration tasks of neural network classifiers were evaluated for a range of different networks topologies [6]. The authors found some of the evaluated methods to successfully calibrate the outputs of the classification models, but counterpart methods for producing calibrated prediction intervals for neural network regressors are still absent.

Motivated by the above, in this work we present two novel methods for producing calibrated prediction intervals for neural network regressors, at any desired confidence level. Both of our proposed methods are performed as post-processing of the outputs of a the trained regression model that uses a softmax classification layer, therefore do not require retraining of the model, and are very fast to compute. Our first proposed method, *empirical calibration*, assesses the amount of the model's posterior probability mass that corresponds in practice to the desired confidence level. Our second proposed method, *temperature scaling*, is an adaptation of a related method proposed in [6] for calibrating classification models, to the regression and prediction intervals setting. Temperature scaling tunes the smoothness of the model's output distribution, to find a balance that results in calibrated prediction intervals.

We corroborate our proposed methods in experiments with four regression tasks from the audio and computer vision domains. We first find that as expected, using neural network classifiers to perform regression, and obtaining prediction intervals by taking an interval in the output space that contains the desired posterior probability mass, results in prediction intervals that are poorly calibrated. On the contrary, we find that applying our proposed calibration methods yields prediction intervals that are considerably better calibrated, a finding that is consistent across all datasets, neural network architectures and confidence levels we experimented with.

Further, we find that when splitting the output space into a finite number of bins, using a larger number of bins and applying our proposed methods results in calibrated prediction intervals that are tighter, i.e., a more accurate estimation of the range in which the label may fall. Finally, we validate that using neural network classifiers to perform regression does not cause any degradation in regression performance, as measured by mean squared error. We conclude that both of are proposed methods are appropriate for emitting calibrated prediction intervals for neural network regressors. We make the source code using empirical calibration and temperature scaling for computing calibrated predicted intervals publicly available.[1]

The rest of this paper is laid out as follows. In Section II we present how regression can be performed using neural network classifiers, and how (not calibrated) prediction intervals can be obtained; Section II-B then presents the two proposed calibration methods. The experimental results on the different tasks are presented in Section III, and finally

[1] code available in http://github.com/cruvadom/prediction_intervals

a brief conclusion and our future work plans are given in Section IV.

II. ACQUIRING PREDICTION INTERVALS

A. POSTERIOR PREDICTION INTERVALS

We consider neural network regressors that process an input $x \in \mathbb{R}^n$ with an associated label $y \in \mathbb{R}$. For a regression task, the standard neural network contains a top layer with only one unit [1]–[3]. The single value in the output of this top layer is then used together with the ground-truth label to compute the mean squared error, which is the training objective of the network. Using this standard design, the network only outputs a single point estimate, and there is no obvious way to use the network's output for computing prediction intervals.

In contrast, a natural approach for designing neural networks regressors from which prediction intervals can be derived, is to construct a regressor that emits a probability distribution p_x over the real numbers, and use this distribution to define intervals with a certain level of probability mass. Denoting $\hat{Y}_x$ as a real-valued random variable that is distributed according to p_x, we define the notion of posterior prediction intervals:

Definition 1: The interval (u_x, v_x) is called *posterior α-prediction interval* if

$$\mathbb{P}[u(x) < \hat{Y}_x < v(x)] = \alpha \quad \text{and} \quad u(x) < \mathbb{E}[\hat{Y}_x] < v(x).$$

The posterior prediction interval (u_x, v_x) is simply an interval around the expected prediction of the network $\mathbb{E}[\hat{Y}_x]$ that is designed to contain a probability mass of α from the network's output distribution p_x. We refer to α as the *confidence level* of the interval.

With the aim of emitting a probability distribution over the real numbers, neural network regressors can be designed similarly to conventional neural network classifiers, as was done in [4] and [5]. The real numbers are divided into a finite number of bins M with edges

$$-\infty = a_0 < a_1 < \ldots < a_M = \infty,$$

and for the training procedure each real-valued label y is replaced with the appropriate class label $t \in \{0, \ldots, M-1\}$ such that

$$a_t \leq y < a_{t+1}.$$

The single unit top layer of the standard neural network regressors is replaced with a layer of $M-1$ units. Softmax normalization is then applied on the output of the top layer and the network is trained as a standard neural network classifier with the cross-entropy loss. The output of such a neural network is a vector of class probabilities

$$(pr_0, \ldots, pr_{M-1}).$$

To emit a probability distribution over the real numbers, we can distribute each class probability uniformly, or according to the distribution of training set real-valued labels, between the class's bin boundaries.

Posing the regression problem as a classification problem allows the network to emit a distribution over the real numbers instead of a point estimate, which in turn can be used to calculate posterior α-prediction intervals.

B. CALIBRATED PREDICTION INTERVALS

In Section II-A we described how neural network regressors can be designed in a manner that allows emitting posterior α-prediction intervals. However, the confidence level α does not guarantee that the label is likely to fall within its appropriate posterior prediction interval with probability α. For example, consider the case of a neural network that produces overly confident predictions. In this case, the output probability distribution p_x will have most of its mass concentrated in a small region, therefore prediction intervals containing a mass of α of the network's output probability distribution will be very narrow. However, despite the confident predictions, the actual ground-truth labels might fall within the boundaries of those intervals on average only α_0 of the times, with $\alpha_0 < \alpha$. Equivalently, the confidence level α may also not represent the actual probability of the label falling within the prediction interval's boundaries in the case of network predictions that are not confident enough.

Ideally, one would aspire to obtain prediction intervals with a confidence level of α, such that α is the actual probability of the label falling within the prediction interval's boundaries. We define the notion of calibrated prediction intervals:

Definition 2: A set of intervals $\{(u_x, v_x)\}_{x \in X}$ is considered as *calibrated α-prediction intervals* if

$$\mathbb{P}_{x,y \sim X,Y}[u_x < y < v_x] = \alpha,$$

where X, Y corresponds to the joint distribution of inputs and labels of the given regression task.

We refer to α as the *confidence level* of the calibrated prediction intervals. In regression analysis, a calibrated prediction interval is an estimate of an interval in which the label will lie, with a certain probability α. Calibrated prediction intervals capture information about the uncertainty of the predicted value across the output space, and convey information that is absent from a single point estimate of the label, that might be critical for a wide range of practical applications.

In recent work, it was shown that modern neural network *classifiers* tend to produce non-calibrated outputs, i. e., the posterior probability assigned to predictions does not correspond to the actual ground-truth accuracy of these predictions [6]. Therefore, when using neural network regressors that are constructed as classifiers, and using those to emit posterior α-prediction intervals, we cannot expect those posterior α-prediction intervals to be calibrated α-prediction intervals. In Section III we show that indeed in practice, the obtained posterior α-prediction intervals are not calibrated α-prediction intervals.

Below we present the main novel contribution of this work, two methods for computing calibrated α-prediction intervals for neural network regressors. Consider the neural network regressors designed as classifiers described

in Section II-A. Recall that in this setting we divide the real numbers into M bins, and given an input x, a regressor emits a categorical probability distribution over the different bins: $(pr_0, \ldots, pr_{M-1})$.

We compute the network's real-valued prediction (point estimate) as the expected prediction with respect to the emitted class probability distribution:

$$\hat{y} = \sum_{i=0}^{M-1} pr_i * c_i, \tag{1}$$

where c_i is the mean of real-valued labels of all training examples with class label i. We denote the class that contains $\hat{y}$ (according to its bin's edges) with $\hat{t}$:

$$\hat{t} = r \quad \text{s.t. } a_r \leq \hat{y} \leq a_{r+1}. \tag{2}$$

For computing posterior α-prediction intervals, we take the smallest symmetric interval around $\hat{t}$ that contains α of the neural network's posterior probability mass. Formally, we take the posterior α-prediction interval to be

$$(u_x^\alpha, v_x^\alpha) = (a_{\hat{t}-i}, a_{\hat{t}+1+i}), \tag{3}$$

such that i is the minimal non-negative integer (possibly zero) for which

$$pr_{\hat{t}-i} + \ldots + pr_{\hat{t}+i} \geq \alpha. \tag{4}$$

Note that we restrict the endpoints of the interval to be the discrete bins edges, therefore the condition from Definition 1 only holds approximately.

In the rest of this section we describe our two proposed novel methods for calibrating the prediction intervals. Both methods apply post-processing to the outputs of a trained neural network, and do not require retraining the neural network. The hyperparameters of the methods are to be chosen using a validation set, and the chosen values should then be used when applying the methods to the test set predictions.

1) EMPIRICAL CALIBRATION

We first observe that posterior α_0-prediction intervals $(u_x^{\alpha_0}, v_x^{\alpha_0})$ as defined according to (3) are actually calibrated α_1-prediction intervals for

$$\alpha_1 = \mathbb{P}_{x,y\sim X,Y}[u_x^{\alpha_0} < y < v_x^{\alpha_0}]. \tag{5}$$

This holds because for every set of prediction intervals, there is an actual probability of the label falling within the boundaries of those intervals. Therefore by definition those are calibrated prediction intervals with this probability as their confidence level.

When calibrating the prediction intervals empirically, we want to find α_0 such that the posterior α_0-prediction intervals are calibrated α-prediction intervals, for a desired confidence level α. Note that $\mathbb{P}_{x,y\sim X,Y}[u_x^{\alpha_0} < y < v_x^{\alpha_0}]$ is increasing in α_0 with fixed points in 0 and 1, since larger posterior prediction intervals necessarily mean that the label is more likely to fall within the intervals' boundaries.

Therefore, our empirical calibration method is comprised of a binary search along different values of $\alpha_0 \in [0, 1]$ to find α_0 such that $|\mathbb{P}_{x,y\sim X,Y}[u_x^{\alpha_0} < y < v_x^{\alpha_0}] - \alpha| < \epsilon$ on the validation set, for a given error tolerance ϵ. In our experiments we use $\epsilon = 0.001$. The error tolerance is necessary, since for a finite validation set finding calibrated prediction intervals with confidence level *exactly* α may be impossible. The α_0 that we end up with is the one that is used for computing prediction intervals on for the test set.

2) TEMPERATURE SCALING

When training the neural network for the classification task, class probabilities $(pr_0, \ldots, pr_{M-1})$ are computed from the output of the top layer $(z_0, \ldots, z_{M-1})$ using the softmax function:

$$pr_i = \frac{\exp(z_i/T)}{\sum_{j=0}^{M-1} \exp(z_j/T)}, \tag{6}$$

where T is called the softmax temperature. During training, the default temperature $T = 1$ is used. Equation 6 can be written as

$$pr_i = \frac{1}{\sum_{j=0}^{M-1} \exp((z_j - z_i)/T)}, \tag{7}$$

that shows that the output of the softmax normalization depends only on the the temperature T and the differences between the output values $(z_0, \ldots, z_{M-1})$. Therefore, scaling the outputs of the top layer before applying the softmax function affects the smoothness of the output probability distribution. Specifically, using a lower temperature $0 < T < 1$ makes the probability distribution ''pointier'', i. e., more probability mass is given to the classes with higher z values. On the contrary, using a larger temperature $1 < T < \infty$ tends towards distributing the probability mass more evenly between the different classes.

Using this property of the softmax normalization function, temperature scaling uses a different temperature at evaluation time for computing class probabilities. A network that produces overconfident predictions, will result in posterior α-prediction intervals that are too narrow, i. e., $\mathbb{P}_{x,y\sim X,Y}[u_x^\alpha < y < v_x^\alpha] < \alpha$. In this case, temperature scaling with a temperature $T > 1$ can be applied to reduce the network's confidence, and increase the width of the posterior prediction intervals. Equivalently, a low temperature $0 < T < 1$ should be used to increase the network's confidence and decrease the width of posterior prediction intervals.

More generally, we define

$$F_\alpha(T) = \mathbb{P}_{x,y\sim X,Y}[u_x^\alpha < y < v_x^\alpha] \tag{8}$$

where u_x^α and v_x^α are the posterior α-prediction intervals that now depend also on T. As increase in T increases the width of the posterior prediction intervals, the function $F_\alpha(T)$ is continuous and monotonic increasing in T, with

$\lim_{T\to 0} F_\alpha(T) = 0$ and $\lim_{T\to\infty} F_\alpha(T) = 1$. Therefore, there must be a temperature T such that $F_\alpha(T) = \alpha$.

Motivated by the above theoretical properties, our temperature scaling method is comprised of a binary search along different values of T to find the temperature value such that

$$|F_\alpha(T) - \alpha| < \epsilon \tag{9}$$

on the validation set, for the desired confidence level α and a given error tolerance of ϵ. In our experiments we use an error tolerance $\epsilon = 0.001$ that is again necessary, since for a finite validation set finding calibrated prediction intervals with confidence level exactly α may be impossible. The temperature T that is chosen using the validation set is then used for computing prediction intervals for the test set. Temperature scaling was used in [6] for calibrating the output probabilities of neural network *classifiers*, and here we extend this method to the regression and prediction intervals setting.

III. EXPERIMENTS

We evaluated our two proposed calibration methods for prediction intervals on four different regression tasks from the audio and computer vision domains.

A. DATASETS AND TASKS

We describe the four regression tasks and datasets we used in our experiments.

1) AGE PREDICTION (AUDIO)

The first task we consider is the prediction speakers' age based on a recording of their speech, using the aGender corpus [23], [24]. The aGender corpus contains audio recordings of predefined utterances and natural speech, annotated for the speakers' age and gender. We split the corpus into speaker independent training, validation and test sets, according to the split used in [25]. In total, the three sets contain more than 38 hours of audio, in more than 53,000 utterances. The total number of speakers is 611, such that 331 speakers are assigned to the training set, 140 to the validation set, and 299 to the test set. We extracted Mel-Frequency Cepstrum Coefficients (MFCC) features from each recordings, using frames of 25 ms shifted by 10 ms. From every frame 13 features were extracted. We applied mean and standard deviation normalization across features and time, for every recording separately.

2) SNR PREDICTION

The second regression task from the audio domain we experimented with is prediction of Signal-to-Noise Ratio (SNR) of speech audio utterances with background noise. For constructing this task's corpus, we used clean speech utterances from the degree of nativeness corpus from the INTERSPEECH 2016 computational paralinguistics challenge [26], [27] and background noise recordings from the CHiME-4 challenge [28]. The native language corpus contains more than 64 hours of clean speech utterances from 5,132 speakers of 11 different native languages, split into speaker independent training, validation, and test sets. The background noises are recordings of four different environments: bus, café, pedestrian area, street junction, and are 14 hours in total. For creating the training set, training speech utterances were mixed with random segments of the background noises according to a random SNR in the range $[0, 25]$. The SNR was then used as the real-valued label for the regression task. The validation and test set were created in a similar manner, using the corresponding clean utterances from the native language corpus and dedicated portions of the noise recordings. We applied a short-time Fourier transform (STFT) on every recording to extract 201 magnitude spectrogram features from every 25 ms frame, where frames are shifted 10 ms. The magnitude spectrogram features were then normalized across features and time, for every utterance separately, to have a mean of 0 and a standard deviation of 1.

3) AGE PREDICTION (IMAGES)

The first dataset we experimented with in the computer vision domain is the Wikipedia faces dataset [29]. The dataset contains 62,359 images of people (one image per person) crawled from Wikipedia, labeled with the age of each person at the time the picture was taken. Since the dataset has no official training/validation/test split, we randomly allocated 60% of the examples to the training set, 20% to the validation set and 20% to the test set. As the dimensions of the different images vary, we resized every image to 224×224 pixels before feeding it to the neural network. In addition, we normalized pixel values for every image separately, to have a mean of 0 and a standard deviation of 1.

4) ISO SPEED PREDICTION

The second images dataset we experimented with is the MIRFLICKR-25000 dataset. The MIRFLICKR-25000 dataset consists of 25,000 images downloaded from the social photography site Flickr through its public API [30]. In addition to images, the dataset contains additional metadata on every image, such as the ISO speed, that measures the sensitivity of the camera's film or sensor to light. The ISO speed affects the brightness of photos, therefore a regression task for predicting the ISO speed of given images is sensible. We split the dataset and extracted features in the same way as described in Section III-A3.

B. NEURAL NETWORKS

As described in Section II, we learn the regression tasks using a classification neural network, where the real numbers are split into M bins. For the audio experiments, the network we used is comprised of two long short-term memory (LSTM) layers, each with 512 units. The output of the last time step in the top layer is fed into the fully-connected output layer, with the number of units equal to the number of bins we use. Softmax normalization is applied to the output layer's units.

For the computer vision experiments, we used a convolutional neural net (CNN) that is comprised of 8 residual blocks [2]. Each residual block first applies a convolutional layer on the input, followed by batch normalization [31] and

a rectified linear activation function. A second convolutional layer is then applied on the output of the rectified linear activation, and the output is added to the block's input. Batch normalization and another rectified linear activation are then applied, to emit the output of the residual block. Before applying the residual blocks, a convolutional layer with a 7×7 kernel is applied on the network's input, with a 2×2 stride and 64 feature maps. The output of this convolutional layer is fed to the a sequence of 8 residual blocks, all using convolutional kernel size of 3×3 and 64,64,128,128,256,256,512,512 feature maps (one value for each residual block). A 2×2 stride is applied for residual blocks number 3, 5 and 7. A global average pooling is applied on the output of the last residual block, to average each of the 512 feature maps across all spatial locations. Similarly to the audio experiments, a fully-connected layer is then applied to project the 512 dimensional vector to the relevant number of bins, and a softmax normalization is applied.

In all experiments, the training objective is the standard cross-entropy, and model parameters are learnt using the Adam optimiser [32] with default β_1, β_2 values and a learning rate of 0.001. We experimented with binning the real numbers into $M = 10, 30, 60$ bins to demonstrate that our method can operate successfully regardless of the number of bins, and to study the differences between the resulting prediction intervals with different number of bins. For a given number of classes M, we set class boundaries $a_0, \ldots, a_M$ to be equally spaced between the minimum and maximum real-valued label values in the training set, and then set $a_0 = -\infty$ and $a_M = \infty$.

C. CALIBRATION RESULTS

For the main results of this work, we evaluated each of the two proposed calibration methods from Section II-B on the different regression tasks, with different neural network architectures and different number of bins. For each task, we trained three neural networks with 10, 30 and 60 bins. Each of the proposed calibration methods was applied to the outputs of each trained network using confidence levels of 66%, 80% and 90%. For each calibration method and dataset, the associated hyperparameters were chosen using the validation set, then we applied this calibration method to the test set using the chosen hyperparameters. All results we report are on the test set.

The aim of each calibration method is to produce calibrated α-prediction intervals. To assess the level in which this goal was achieved, we measure the *calibration error*, which is the absolute difference between the desired confidence level α and the actual probability of the label falling within the boundaries of the acquired prediction intervals. Mathematically, the calibration error is defined as

$$|\mathbb{P}_{x,y\sim X,Y}[u(x) < y < v(x)] - \alpha|, \tag{10}$$

where $(u(x), v(x))$ is the prediction interval emitted by the calibration method for example x, and X, Y are distributed uniformly over the test set examples.

TABLE 1. A comparison of test set calibration error ([%]) before ('Posterior' column) and after applying each of the the two proposed calibration methods for the different regression tasks. 'Empirical', 'Temp' and 'Confidence' columns represent empirical calibration, temperature scaling and the prediction intervals' confidence level respectively. In all cases, both of the proposed methods manage to considerably reduce the calibration error of prediction intervals, compared to prediction intervals based on the networks' posterior distribution (smaller numbers on the right side of the dashed line). Both of the proposed methods yield comparable performance. This result holds when training the network with either 10, 30 or 60 bins, with no clear advantage for a specific number of bins.

Dataset	Confidence	Bins	Posterior	Empirical	Temp'
Age (Audio)	66%	10	7.63	0.60	0.09
		30	12.40	2.25	1.80
		60	11.95	0.82	0.35
	80%	10	10.78	0.64	1.16
		30	15.37	2.63	3.13
		60	13.74	1.45	2.69
	90%	10	9.64	1.81	2.44
		30	11.83	2.53	3.13
		60	10.97	1.95	2.16
SNR	66%	10	22.11	6.44	7.22
		30	19.67	1.78	1.78
		60	12.22	0.11	0.78
	80%	10	14.56	1.67	0.56
		30	12.67	2.89	1.78
		60	9.22	1.44	1.78
	90%	10	7.56	0.89	0.44
		30	6.11	2.78	2.78
		60	4.78	0.00	0.11
Age (Images)	66%	10	0.29	0.01	0.14
		30	4.50	0.14	0.26
		60	13.43	0.22	0.29
	80%	10	3.90	0.15	0.05
		30	2.71	0.26	0.14
		60	14.98	0.22	0.63
	90%	10	4.46	0.11	0.08
		30	0.73	0.08	0.05
		60	14.34	0.25	0.02
Iso Speed	66%	10	17.39	0.76	0.82
		30	6.06	0.06	0.70
		60	7.21	0.21	0.58
	80%	10	6.58	0.15	0.67
		30	6.58	0.15	0.67
		60	4.45	0.03	0.06
	90%	10	3.55	0.21	0.24
		30	4.06	0.73	0.27
		60	3.70	0.24	0.91

A comparison of the calibration error when using the posterior prediction intervals, and after applying each of the two proposed calibration methods is given in Table 1. First, we observe that the posterior prediction intervals, without applying a calibration method, generally yield a large calibration error. This finding is consistent with findings from [6] regarding the miscalibration of modern neural network classifiers. Second, we see that in all cases, both the empirical calibration and temperature scaling methods manage to considerably reduce the calibration error, eliminating the calibration error to small levels of normally around 0%-2%. These results indicate that using these methods, calibrated prediction intervals for neural network regressors can indeed be acquired. Moreover these findings hold across all datasets, confidence levels, and number of bins used for training the networks.

TABLE 2. A comparison of the test set average width of prediction intervals using the two proposed calibration methods, empirical calibration and temperature scaling. 'Empirical', 'Temperature' and 'Confidence' columns represent empirical calibration, temperature scaling and the prediction intervals' confidence level respectively. For all datasets except 'Age (Audio)', training the network with more bins generally results in tighter prediction intervals, since the network can learn a more precise distribution of posterior probability (numbers in the 30 and 60 bins rows are generally smaller than in the 10 bins rows). The width of the intervals is comparable between the two calibration methods and naturally grows with the confidence level. Finally, the width of the intervals naturally depends on the performance of the neural network in the regression task.

Dataset	Confidence	Bins	Empirical	Temperature
Age (Audio)	66%	10	34.84	35.70
		30	34.05	36.35
		60	36.16	38.59
	80%	10	44.55	44.79
		30	43.84	44.13
		60	45.23	45.41
	90%	10	52.77	52.68
		30	52.69	52.30
		60	53.84	54.32
SNR	66%	10	2.60	2.60
		30	1.97	1.91
		60	1.64	1.58
	80%	10	3.49	3.31
		30	2.50	2.47
		60	2.22	2.15
	90%	10	4.74	4.63
		30	3.11	3.04
		60	2.96	2.94
Age (Images)	66%	10	20.99	20.92
		30	19.66	19.11
		60	18.71	19.80
	80%	10	27.48	27.65
		30	28.83	25.53
		60	25.71	25.81
	90%	10	35.60	35.43
		30	33.51	33.58
		60	34.63	33.72
Iso Speed	66%	10	2.23	2.20
		30	1.94	1.80
		60	2.21	2.08
	80%	10	3.22	3.23
		30	2.94	3.02
		60	2.90	2.91
	90%	10	4.35	4.44
		30	4.09	4.05
		60	3.83	3.79

However, even when using one of the two proposed calibration methods, calibration error does not vanish completely. The reason for this is that calibration hyperparameters were chosen on the validation set, and do not generalize perfectly to the test set. Nevertheless, a calibration error of 1%-2% is sufficiently enough for the majority of applications (e.g., a confidence level of 81% instead of a desired 80% will not make a large difference in most applications). Both calibration methods yield comparable performance, and are fast to execute, typically around 1-3 seconds for a test set of 10000 examples, depending on the number of bins used.

Further, we compare the width of the emitted prediction intervals for the empirical calibration and temperature scaling methods. Table 2 contains the average width of the prediction intervals for test sets of the different regression tasks. Posterior prediction intervals were above to be poorly calibrated, therefore their width is not meaningful with respect to the desired confidence level, and we omit them from Table 2. We first observe that naturally, the width of the interval grows with the desired confidence level. The main conclusion that can be derived from these results is that networks trained using a larger number of bins tend to produce tighter prediction intervals. Specifically, for all tasks except age prediction from audio signal, the width of the resulting calibrated prediction intervals is generally smaller when using 30 or 60 bins, compared to 10 bins. The reason for this phenomenon is that a larger number of bins allows the network a more precise allocation of posterior probability mass.

Additionally, we find that the two calibration methods produce prediction intervals of a comparable width, with no prominent advantage for neither of the two methods. This result indicates that both methods can be interchangeably used to produce calibrated prediction intervals of the same quality. Lastly, we note that width of the prediction intervals is closely affected by the quality of the regressor that they are based on. A better neural network regressor is one that assigns a higher probability mass around the correct labels, which will in turn result in tighter prediction intervals.

TABLE 3. Performance in the different regression tasks as measured by the root MSE, for a standard neural network regressor and a neural network classifier with different number of classes. The performance of the standard regressors is comparable to the performance of the models performing regression using a classification models. This indicates that using neural network classifiers to perform regression task, that allow emitting calibrated prediction intervals, does not cause any degradation in the regression performance.

Dataset	Standard	10 classes	30 classes	60 classes
Age (Audio)	20.07	19.73	19.95	20.04
SNR	1.32	1.21	1.41	1.30
Age (Images)	11.48	11.55	11.36	11.47
ISO Speed	1.29	1.28	1.28	1.29

D. REGRESSION RESULTS

For studying the the effect of performing the regression tasks using neural network classifiers, we additionally train a standard neural network regressor for each of the tasks. For each task the standard neural network regressor is trained with an identical architecture to the corresponding neural network classifier for this task, except the topmost layer that contains only a single unit, as described in Section II. The regressor is trained with the same optimiser as the classifiers to minimize the mean squared error (MSE) between the network's predictions and the labels. For the classification models, MSE is computed using the prediction $\hat{y}$ defined in Eq. 1.

The root MSE on the test set for the different models is found in Table 3. The results in the table show that regression performance of the standard regressor and the classifiers is generally comparable on all tasks. We therefore conclude that training neural network regressors using neural network classifiers, that allow emitting calibrated prediction

intervals, does not cause any degradation in the regression task performance.

IV. CONCLUSIONS

The output of contemporary neural networks, despite being highly accurate in many circumstances, can be considered miscalibrated, thereby producing unreliable output probability estimates [6]. This issue is exacerbated in regression, in which the output of a standard neural network regressor is a point estimate of the predicted values.

By posing neural network regression as a multi-class classification problem and introducing two novel post-processing calibration methods, we demonstrated that it is possible to produce well-calibrated prediction intervals for neural network regression, that can be critical for a large variety of real-world application. We find that our proposed methods were fast to execute and produce calibration prediction intervals for any desired confidence level, across a variety of regression tasks from the audio and computer vision domains and different neural network architectures. In addition, we found that using a larger number of classification bins generally resulted in tighter prediction intervals, and importantly, that using our proposed methods does not cause any degradation in regression performance, as measured by the mean squared error.

Future work includes exploring alternative training mechanisms that will lead to tighter calibrated prediction intervals [33], [34], embedding the calibrated outputs into the decision making process of more complex models such as [35], and applying the proposed methods to a variety of applications such as computational paralinguistics [36]–[38]. Further, given the complication when performing regression fusion associated with effects such as multicollinearity, we also plan to test our approach to aid late fusion of multiple neural network regressors.

REFERENCES

[1] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *Proc. CVPR*, Columbus, OH, USA, Jun. 2014, pp. 1653–1660.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[3] S. Yao *et al.*, "RDeepSense: Reliable deep mobile computing models with uncertainty estimations," in *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 1, no. 4, Dec. 2017, Art. no. 173.

[4] A. Van Den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *Proc. ICML*, New York, NY, USA, 2016, pp. 1747–1756.

[5] A. van den Oord *et al.*, "Wavenet: A generative model for raw audio," in *Proc. 9th ISCA Speech Synth. Workshop*, Sunnyvale, CA, USA, 2016, p. 125.

[6] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. ICML*, Sydney, NSW, Australia, 2017, pp. 1321–1330.

[7] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Proc. ICML*, Bonn, Germany, 2005, pp. 625–632.

[8] A. P. Dawid, "The well-calibrated Bayesian," *J. Amer. Stat. Assoc.*, vol. 77, no. 379, pp. 605–610, 1982.

[9] M. Weintraub, F. Beaufays, Z. Rivlin, Y. Konig, and A. Stolcke, "Neural-network based measures of confidence for word recognition," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Munich, Germany, vol. 2, Apr. 1997, pp. 887–890.

[10] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Adv. Large Margin Classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[11] F. Wessel, R. Schluter, K. Macherey, and H. Ney, "Confidence measures for large vocabulary continuous speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 9, no. 3, pp. 288–298, Mar. 2001.

[12] H. Jiang, "Confidence measures for speech recognition: A survey," *Speech Commun.*, vol. 45, no. 4, pp. 455–470, 2005.

[13] N. Brummer and D. A. Van Leeuwen, "On calibration of language recognition scores," in *Proc. IEEE Odyssey– Speaker Lang. Recognit. Workshop*, San Juan, Puerto Rico, Jun. 2006, pp. 1–8.

[14] D. Yu, J. Li, and L. Deng, "Calibration of confidence measures in speech recognition," *IEEE Trans. Audio, Speech, Language Process.*, vol. 19, no. 8, pp. 2461–2473, Nov. 2011.

[15] J. Deng and B. Schuller, "Confidence measures in speech emotion recognition based on semi-supervised learning," in *Proc. INTERSPEECH*, Portland, OR, USA, 2012, pp. 2226–2229.

[16] R. M. Neal, *Bayesian Learning for Neural Networks*, vol. 118. Springer, 2012.

[17] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *Proc. NIPS*, Long Beach, CA, USA, 2017, pp. 5574–5584.

[18] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Proc. NIPS*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Long Beach, CA, USA, 2017, pp. 6402–6413.

[19] A. Khosravi, S. Nahavandi, D. Srinivasan, and R. Khosravi, "Constructing optimal prediction intervals by using neural networks and bootstrap method," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 8, pp. 1810–1815, Aug. 2015.

[20] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. ICML*, New York, NY, USA, 2016, pp. 1050–1059.

[21] A. Naumov, V. Spokoiny, and V. Ulyanov. (2017). "Bootstrap confidence sets for spectral projectors of sample covariance." [Online]. Available: https://arxiv.org/abs/1703.00871

[22] H. Li, X. Wang, and S. Ding, "Research and development of neural network ensembles: A survey," *Artif. Intell. Rev.*, vol. 49, no. 4, pp. 455–479, Apr. 2018.

[23] F. Burkhardt, M. Eckert, W. Johannsen, J. Stegmann, and D. Telekom, "A database of age and gender annotated telephone speech," in *Proc. Int. Conf. Lang. Resour. Eval. (LREC)*, Valletta, Malta, 2010.

[24] B. Schuller *et al.*, "The INTERSPEECH 2010 paralinguistic challenge," in *Proc. INTERSPEECH*, Makuhari, Japan, 2010, pp. 2794–2797.

[25] G. Keren and B. Schuller, "Convolutional RNN: An enhanced model for extracting features from sequential data," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Vancouver, BC, Canada, Jul. 2016, pp. 3412–3419.

[26] B. Schuller *et al.*, "The INTERSPEECH 2016 computational paralinguistics challenge: Deception & sincerity," in *Proc. INTERSPEECH*, San Francsico, CA, USA, 2016, pp. 2001–2005.

[27] G. Keren, J. Deng, J. Pohjalainen, and B. Schuller, "Convolutional neural networks with data augmentation for classifying speakers' native language," in *Proc. INTERSPEECH*, San Francsico, CA, USA, 2016, pp. 2393–2397.

[28] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Comput. Speech Lang.*, vol. 46, pp. 535–557, Nov. 2017.

[29] R. Rothe, R. Timofte, and L. Van Gool, "Dex: Deep expectation of apparent age from a single image," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Santiago, Chile, Dec. 2015, pp. 252–257.

[30] M. J. Huiskes, B. Thomee, and M. S. Lew, "New trends and ideas in visual concept detection: The MIR flickr retrieval evaluation initiative," in *Proc. ACM Int. Conf. Multimedia Inf. Retr. (MIR)*, New York, NY, USA, 2010, pp. 527–536.

[31] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, Lille, France, 2015, pp. 448–456.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, 2015.

[33] G. Keren, S. Sabato, and B. Schuller, "Tunable sensitivity to large errors in neural network training," in *Proc. AAAI*, San Francisco, CA, USA, 2017, pp. 2087–2093.
[34] G. Keren, S. Sabato, and B. Schuller. (2017). "Fast single-class classification and the principle of logit separation." [Online]. Available: https://arxiv.org/abs/1705.10246
[35] G. Keren, M. Schmitt, T. Kehrenberg, and B. Schuller. (2018). "Weakly supervised one-shot detection with attention similarity networks." [Online]. Available: https://arxiv.org/abs/1801.03329
[36] J. Deng, X. Xu, Z. Zhang, S. Frühholz, and B. Schuller, "Exploitation of phase-based features for whispered speech emotion recognition," *IEEE Access*, vol. 4, pp. 4299–4309, Jul. 2016.
[37] J. Deng, S. Frühholz, Z. Zhang, and B. Schuller, "Recognizing emotions from whispered speech based on acoustic feature transfer learning," *IEEE Access*, vol. 5, no. 1, pp. 5235–5246, Dec. 2017.
[38] E. Marchi, S. Frühholz, and B. Schuller, "The effect of narrow-band transmission on recognition of paralinguistic information from human vocalizations," *IEEE Access*, vol. 4, pp. 6059–6072, Aug. 2016.

GIL KEREN received the bachelor's and master's degrees in mathematics and psychology from Ben Gurion University, Israel. He is currently pursuing the Ph.D. degree with the ZD.B Chair of Embedded Intelligence for Health Care and Wellbeing, University of Augsburg. He is involved in a number of research projects with the University of Augsburg. His current research interests include artificial intelligence, neural networks, and computational cognition.

NICHOLAS CUMMINS received the bachelor's degree (Hons.) and the Ph.D. degree in electrical engineering from UNSW, Australia, in 2011 and 2016, respectively. He is currently pursuing the Habilitation degree with the ZD.B Chair of Embedded Intelligence for Health Care and Wellbeing, University of Augsburg. His Ph.D. dissertation investigated whether the voice can be used as an objective marker in the diagnosis and monitoring of clinical depression. He is involved in Horizon 2020 projects DE-ENIGMA, RADAR-CNS, and TAPAS with the University of Augsburg. He has authored regularly in the field of depression detection since 2011; these papers have attracted significant attention and citations. His current research areas include behavioral signal processing with a focus on the automatic multisensory analysis and understanding of different health states.

BJÖRN SCHULLER received the Diploma degree in electrical engineering and information technology, the Ph.D. degree in electrical engineering and information technology, with a focus on automatic speech and emotion recognition, and the Habilitation degree in electrical engineering and information technology from TUM, Munich, Germany, in 1999, 2006, and 2012, respectively. He is currently a Reader of machine learning with the Department of Computing, Imperial College London, U.K.; a Full Professor and the Head of the Chair of Embedded Intelligence for Health Care and Wellbeing, University of Augsburg, Germany; and a CEO of audEERING, Audio Intelligence company. He has authored or co-authored over six books and 750 publications in peer-reviewed books, journals, and conference proceedings leading to more than overall 19000 citations (h-index $= 66$). He is an Elected Member of the IEEE Speech and Language Processing Technical Committee and a President Emeritus of AAAC. In 2012, he received the Adjunct Teaching Professorship in the area of signal processing and machine intelligence from TUM. He is a Co-Program Chair of Interspeech 2019, a repeated Area Chair of ICASSP, and an Editor-in-Chief of the IEEE TRANSACTIONS ON AFFECTIVE COMPUTING. He holds associate and guest editor roles and functions in technical and organizational committees.