DEEP METAMEMORY - A GENERIC FRAMEWORK FOR STABILIZED ONE-SHOT CONFIDENCE ESTIMATION IN DEEP NEURAL NETWORKS AND ITS APPLICATION ON COLORECTAL CANCER LIVER METASTASES GROWTH PREDICTION

Alexander Katzmann[⋆‡] *Alexander Mühlberg*[⋆] *Michael Sühling*[⋆]
Dominik Nörenberg[†] *Horst-Michael Groß*[‡]

[⋆]Siemens Healthcare GmbH, Department CT R&D Image Analytics
91301 Forchheim, Germany

[†]University Hospital Großhadern, Ludwig-Maximilians-University Munich, Department of Radiology
81377 Munich, Germany

[‡]University of Technology, Neuroinformatics and Cognitive Robotics Lab
98693 Ilmenau, Germany

ABSTRACT

With the rise of deep learning within medical applications, questions about classification confidence become of major interest as misclassifications might have serious impact on human health. While multiple ways of confidence estimation have been proposed, most of them suffer from computational inefficiency or low statistical accuracy. We utilize a modified version of the method introduced by DeVries et al. for one-shot confidence estimation and show its application for colorectal cancer liver metastases growth prediction. Furthermore, we propose a psychologically motivated generalized training framework called "deep metamemory" comparable to the idea of curriculum learning, which utilizes confidence estimation for efficient training augmentation with improved classification performance on unseen data.

***Index Terms*—** deep learning, certainty estimation, curriculum learning, colorectal cancer, tumor growth prediction

1. INTRODUCTION

Due to their paramount performance in a wide range of tasks, deep neural networks have been applied for highly demanding tasks like autonomous driving [1], earth-quake prediction [2], or medical image analysis [3][4]. All these domains set hard limits for the maximum misclassification rate. Within the medical domain, this issue is of particular importance, as the consequences of false classifications might lead to no or mistreatment and subsequently even death. Big and comprehensive datasets are rare, leading to a situation where the base population is often not sufficiently represented within the training dataset. In turn, this leads to a potentially higher misclassification rate as it becomes more likely that a classifier is applied to cases outside the known problem space.

Uncertainty estimation describes the process of giving a profound guess on the probability of an appropiate classification resulting in a grounded estimate $c_i \in \{0, 1\} \in \mathcal{R}$ for sample i with a known maximum misclassification rate of $m_i = (1 - c_i)$.

Uncertainty estimation for neural networks has recently shown to be of emerging interest. Most work on this topic focusses on the utilization of approaches of variational inference, e.g. Monte-Carlo Dropout [5], Stochastic Batch Normalization [6], or model ensemble techniques [7]. Variational inference methods, i.e. approximating probability densities through optimization [8], are mostly based on the idea of finding specific parameters of an assumed well-known probability distribution over the network's posteriors to give a statistically founded estimate on its variation. The majority of the mentioned techniques require either the network to be run numerous times or to provide a multitude of instances of the same or a partly modified network, resulting in significantly higher computational effort.

Late research has also focussed on a special type of probabilistic networks, called Bayesian Neural Networks (BNNs) [9]. BNNs differ in that way as they inherently provide a probabilistically well-grounded framework for uncertainty.

This work has received funding from the German Federal Ministry of Education and Research as part of the PANTHER project under grant agreement no. 13GW0163A. The concepts and information presented in this article are based on research and are not commercially available.

This is implemented by keeping track of the networks training process through maintaining not only one fixed weight for each connection, but by modelling each weight as a probabilistic function. Unfortunately, BNNs are thus computationally inefficient with respect to the classification goal and their application is restricted by time-consumption in training and application phase.

A popular way to gain confidence estimates is to interpret the output probabilities themselves, following the assumption that higher output activations tend to be associated with a higher probability of a confident estimate [10]. However, while this is often the case when a classification criterion is appropriate, it may not necessarily be applicable to any case.

In contrast, this work tries to provide a generalized framework for a one-shot uncertainty estimate for deep neural networks, and furthermore utilizes this estimate for knowledge gain within training with a focus on small datasets as typically faced within clinical studies. Using a reinterpretation of meta-knowledge in terms of neural networks, we propose a novel neural network design implementing probability estimation with inherent knowledge of error in an integrated training process called **deep metamemory**. Our approach provides an efficiency enhanced training and an improved overall classification performance through uncertainty-driven knowledge acquisition. It does not require an extensive redesign of the network architecture but only minor modifications and is applicable to nearly arbitrary network designs.

2. BACKGROUND

Estimation of classifier knowledge can be seen as second-order knowledge, as it is based on knowledge itself. Within human learning, cognitive psychology deals with this topic within the scope of *metacognition* (a term for *thinking about thinking*). Within this domain, *knowledge about knowledge* is referred to as **metamemory** [11].

Meta-knowledge may intuitively be assumed to be difficult to estimate, as the estimation of the confidence might seem to require the actual knowledge about the sample's label *as well as* the belonging to the familiar population. However, research in psychology shows that this is not necessarily the case. While humans might not be able to correctly reply to a given question, they are reasonably well in estimating whether they can find a fairly justified answer, i.e. give an estimate on problem difficulty with respect to current knowledge [12].

3. BASE APPROACH

Commonly, learning in neural networks is modeled using numerical optimization over the network weights **w** with respect to the network loss $\mathcal{L}$ as a function of the network's output p and the ground truth y, i.e. to solve $\min \mathcal{L}(p, y)$. The proposed approach (see Sec. 4) is based on the confidence estimation technique of DeVries et al.[13]. Within their work, they propose the augmentation of the original network with another network for confidence estimation. This additional network has a single-neuron output representing the confidence estimate $c_i(x_i) \in \mathcal{R} : 0 \leq c_i \leq 1$ for sample i with input x_i based on the output of one of the original network's hidden layers (e.g. the pre-output layer). DeVries et al. introduce a derived output probability function p_i' employing the current confidence estimate c_i for inducing hints y_i (i.e. the ground-truth) into the training procedure:

$$p_i' = c_i \cdot p_i + (1 - c_i) \cdot y_i \tag{1}$$

The classifer is trained using the binary crossentropy for the derived probability estimate:

$$\mathcal{L}_p = -\sum_{k=0}^{m} \log(p_{i,m}') \cdot y_{i,m} \tag{2}$$

with m being the number of outputs. However, this procedure can be applied to arbitrary loss functions. A separate loss function is introduced for assigning hints with costs:

$$\mathcal{L}_c = -\sum \log(c) \cdot \lambda \tag{3}$$

where λ is a scaling parameter dynamically adapted while training to enforce the classifier to always produce the same overall confidence loss β for hinting independent of the network's overall confidence:

$$\mathcal{L}_c \approx \beta \tag{4}$$

This is necessary to preserve learning, as it otherwise converges to a solution where the classifer does *a)* take hints for every decision while certainty converges to 0 ($\forall x_i : c_i \to 0$), thus not learning the actual problem, or reversed, *b)* $\forall x_i : c_i \to 1$, so no confidence loss is produced (i.e. $\mathcal{L}_c = 0$, independent of the actual input). Case *b* reduces the problem to the original one with no (reasonable) confidence estimate, while *a* prevents the network from learning the actual training task itself. The parameter β therefore is an additional design parameter and has to be chosen before training. However, DeVries and Taylor show that the actual quality of the estimates does not highly depend on the concrete choice of β within a wide range of reasonable values [13]. The overall loss is formed as:

$$\mathcal{L} = \mathcal{L}_p + \mathcal{L}_c \tag{5}$$

enforcing a simultaneous optimization of the training as well as the confidence estimation task.

4. METHODS

4.1. Metamemory Importance Sampling

Humans actively utilize metamemory for selective learning (Sec. 2). This process, interpretable as information weight-

ing, can directly be integrated into the neural network training process in a bootstrapping-like manner. Given a confidence estimation method, we propose the following cyclic algorithm:

1. Predict classifier confidences $C = c_0, ..., c_n \forall x_i \in X$ with $c_i = c(x_i)$ and samples X
2. Calculate confidence based sampling probabilities $\Pi = \pi_0, ...\pi_n$.
3. Set importance weights for training sample generator G according to the sample importances Π
4. Train network for one epoch with generator G

Sampling probabilities π_i for samples x_i are calculated as the normalized inverse classifier certainty estimate:

$$\pi_i = \frac{1 - c_i}{\sum\limits_{k=0,..,|X|} (1 - c_i)} \tag{6}$$

To stabilize the training, we transform sampling probabilities π_i' into a logarithmic equal distribution within an interval $I = [0.1, 1]$ using inverse transform sampling $RS_{log_{eq}}(\pi_i)$, prohibitting very high or low sampling probabilities:

$$\pi_i' = RS_{log_{eq}} \left(\frac{1 - c_i}{\sum\limits_{k=0,..,|X|} (1 - c_i)} \right) \rightarrow [0.1, 1] \tag{7}$$

4.2. Model augmentation

The network gets augmented by an additional confidence estimation network with one sigmoid confidence output appendend after the pre-output fully-connected layer.

In contrast to the approach from DeVries[13], we integrate the choice of β into our network. The network receives an additional β-estimation lane, consisting of a constant input of 1 and an additional layer with one neuron, whose scalar product output $\hat{y}_\beta = \hat{\beta}$ is interpreted as the current estimate of β. This route is trained together with the actual training time optimization. $\hat{y}_\beta$ is adapted with respect to the confidence estimation error using the mean squared error between the estimated and the actual β parameter:

$$\mathcal{L}_\beta = ||\hat{y}_\beta - \beta||^2 \tag{8}$$

We start with a target value of $\beta = 1$ and adjust β after each epoch according to:

$$\beta' = \frac{1}{n} \sum_{i}^{n} \mathcal{L}_{p,i} + \mathcal{L}_{c,i} \quad , \quad \beta \leftarrow \beta' \tag{9}$$

to not significantly exeed the average loss induced by training and confidence estimation. Adjusting β to be directly correlated to the task loss adjusts inter-class-margins' width to be correlated to the actual classifier performance and, thus, should lead to a more grounded confidence estimate.

The weight $w_{\hat{\beta}}$ can directly be interpreted as the confidence loss weighting variable λ. This is a remarkable difference to the approach from [13], as λ is optimized together with the actual training task. As already mentioned, the proposed augmentation can be applied to any existing or pretrained network.

As the models confidence guess is based on an inner layer, the confidence task could also interfer with the actual training task. To circumvent this, the confidence estimation is trained after an initial cooldown period as we experienced major interferences to mainly occur within the very early training.

5. EXPERIMENTS

We train our model given the above definition using the Adam optimizer [14] with an initial learning rate of $lr = 1 \cdot 10^{-3}$. We reduce lr by half at plateaus of at least 15 epochs with no improvement. The training is stopped after 35 epochs with no improvement.

5.1. Metrics

We chose to use the Matthew's correlation coefficient Φ on the validation data set as the evaluation metric, as it is zero-centered and balanced, making it the perfect measure when no class weight choice is possible at training time. We furthermore provide the values of accuracy (ACC), F1 score, true positive rate (TPR or sensitivity), true negative rate (TNR or specifity), positive predictive value (PPV) and negative predictive value (NPV), and area under the ROC curve (AUC). Significance tests were done using two-tailed z-test with 10,000 iterations of bootstrapping [15].

5.2. Cifar-10

We first evaluated our approach using the Cifar 10 dataset [16]. Therefore we chose a convolutional neural network with four blocks of 3x3 convolutions (32, 64, 128 and 196 filters), batch normalization, leaky ReLU activation and 2x2 max-pooling, followed by two additional fully-connected layers with 128 neurons each, followed by a 10-neuron softmax output layer. We compared training with augmentation and confidence importance sampling (*metamemory*) versus training without confidence augmentation as a baseline (*BL*). We also trained one model using the confidence augmentation but without confidence importance sampling (*conf. only*) for an approximation of the error induced to the training problem by an additional confidence estimation task. We used micro averaging and 1-vs-all classification (which obviously leads to higher accuracy values). We did not provide PPV and NPV as they reduce to TPR and TNR with 1-vs-all classification:

	BL	metamemory	conf. only	sig.
ACC	.952	.955	.917	***
F1	.763	.774	.576	***
TPR	.759	.775	.579	***
TNR	.973	.975	.952	***
MCC	.737	.746	.533	***
AUC	.971	.969	.909	***

BL and importance sampled metamemory outperformed confidence augmentation only with respect to all tested metrics and p-values markedly below $1 \cdot 10^{-3}$ (***). Although a general trend can be seen, there were no significant differences between the baseline and the proposed metamemory approach ($\alpha = .05$). Spearman's rank correlation coefficient between confidence and target class probability was measured

$$r_S = \frac{cov(rg_c, rg_{p_y})}{\sigma_{rg_c}\sigma_{rg_{p_y}}} = .508 \tag{10}$$

5.3. Cifar-100

We also evaluated our approach on the more complex Cifar-100 dataset.

	BL	metamemory	sig.
ACC	.986	**.988**	
F1	.334	**.385**	*
TPR	.336	**.384**	*
TNR	.993	**.994**	
MCC	.332	**.380**	*
AUC	.568	**.592**	*

The effects of metamemory importance sampling over the baseline approach were significant for F1, TPR, MCC and AUC ($\alpha < .05$ *). As expected, confidence estimation accuracy was correlated to problem difficulty with $r_S = .225$, which is significantly lower than r_S on the Cifar-10 dataset.

5.4. Radiologic image data

We expect our method to be especially beneficial when training data is rare, which is typical for clinical studies. We applied our method to an enlarged set of the data from [17] with masked 2D-views of colorectal cancer liver metastases in computed tomography images:

samples	lesions	scans	patients
592	320	138	75

As we have few data, we employ a very simple convolutional neural network with batch normalization, one 8-neuron ReLU-layer and an additional softmax output with a total of 79,127 parameters with 4-fold cross-validation. The network is pretrained using an autoencoder architecture to reduce the possible parameter space. Tumor growth classification was done using the *Response Evaluation Criteria in Solid Tumors* (RECIST) lesion assessment, which utilizes the maximum diameter Ø within one slice. For sample i, we used the lesion baseline $x_{i,0}$ as well as one followup scan $x_{i,t}$ to predict the RECIST lesion progression status $y_{i,t+1}$:

$$y_{i,t+1} = \begin{cases} 1 & \text{if } \text{Ø}_{i,t+1}/\text{Ø}_{i,t} \geq 1.2 \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

which equals the lesion progression status according to the RECIST progessive disease lesion assessment criterion. As the test set was markedly smaller than for Cifar-10 and Cifar-100, we also report standard deviations:

	BL	Metamemory	sig.
ACC	$\mathbf{.736 \pm .018}$	$.712 \pm .020$	
F1	$.298 \pm .040$	$\mathbf{.341 \pm .037}$	
TPR	$.408 \pm .053$	$\mathbf{.543 \pm .054}$	**
TNR	$\mathbf{.787 \pm .018}$	$.739 \pm .019$	**
PPV	$.234 \pm .036$	$\mathbf{.248 \pm .032}$	
NPV	$.893 \pm .014$	$\mathbf{.911 \pm .014}$	
MCC	$.157 \pm .047$	$\mathbf{.213 \pm .045}$	
AUC	$.604 \pm .037$	$\mathbf{.675 \pm .036}$	*

Significant differences were noted with stars ($p < .05*$, $.02 **$). TPR and TNR differences reflect class preference, while AUC shows superiority for the metamemory approach. Averaged Spearman's rank correlation between confidence and target class probability over all folds was $r_S = .251$.

6. DISCUSSION

As shown in Sec. 5.2, our proposed metamemory importance sampling seems to reduce interferences of the confidence estimation with the classification task, as it performed significantly better than the confidence estimation network only, while preserving a significant correlation to test time classification performance. The results in Sec. 5.3 show that the approach might also be beneficial with respect to the final classification performance. This improvement becomes even clearer for smaller dataset sizes which are typically found in clinical studies, as shown in Sec. 5.4, although due to few data significance could not be shown with respect to all metrics.

The process of metamemory importance sampling might seem similar to the idea of **curriculum learning** [18]. However, there are major differences. First, sample difficulty is determined not by a human expert or heuristically but deduced within the training process. Difficulty might vary with respect to the current learning state of the network, just as human learning difficulty depends on prior knowledge and understanding. Also, curriculum learning modifies the training time working set based on a monotonically increased difficulty function. In contrast, our approach is cyclic and defines difficulty based on the network's current state, meaning samples get omitted and re-introduced as needed. Future research will further analyze the training time reduction potential as well as the improvements for the classification performance.

7. REFERENCES

[1] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2722–2730.

[2] Bertrand Rouet-Leduc, Claudia Hulbert, Nicholas Lubbers, Kipton Barros, Colin J Humphreys, and Paul A Johnson, "Machine learning predicts laboratory earthquakes," *Geophysical Research Letters*, vol. 44, no. 18, pp. 9276–9282, 2017.

[3] Nima Tajbakhsh, Jae Y Shin, Suryakanth R Gurudu, R Todd Hurst, Christopher B Kendall, Michael B Gotway, and Jianming Liang, "Convolutional neural networks for medical image analysis: Full training or fine tuning?," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1299–1312, 2016.

[4] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen AWM van der Laak, Bram Van Ginneken, and Clara I Sánchez, "A survey on deep learning in medical image analysis," *Medical image analysis*, vol. 42, pp. 60–88, 2017.

[5] Yarin Gal and Zoubin Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, 2016, pp. 1050–1059.

[6] Andrei Atanov, Arsenii Ashukha, Dmitry Molchanov, Kirill Neklyudov, and Dmitry Vetrov, "Uncertainty estimation via stochastic batch normalization," *arXiv preprint arXiv:1802.04893*, 2018.

[7] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems*, 2017, pp. 6402–6413.

[8] David M Blei, Alp Kucukelbir, and Jon D McAuliffe, "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[9] Alex Kendall and Yarin Gal, "What uncertainties do we need in bayesian deep learning for computer vision?," in *Advances in neural information processing systems*, 2017, pp. 5574–5584.

[10] Dan Hendrycks and Kevin Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," *arXiv preprint arXiv:1610.02136*, 2016.

[11] Thomas O Nelson, "Metamemory: A theoretical framework and new findings," in *Psychology of learning and motivation*, vol. 26, pp. 125–173. Elsevier, 1990.

[12] William L Kelemen, "Metamemory cues and monitoring accuracy: Judging what you know and what you will know.," *Journal of Educational Psychology*, vol. 92, no. 4, pp. 800, 2000.

[13] Terrance DeVries and Graham W Taylor, "Learning confidence for out-of-distribution detection in neural networks," *arXiv preprint arXiv:1802.04865*, 2018.

[14] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[15] Bradley Efron, *The jackknife, the bootstrap, and other resampling plans*, vol. 38, Siam, 1982.

[16] Alex Krizhevsky and Geoffrey Hinton, "Learning multiple layers of features from tiny images," Tech. Rep., Citeseer, 2009.

[17] Alexander Katzmann, Alexander Muehlberg, Michael Suehling, Dominik Noerenberg, Julian Walter Holch, Volker Heinemann, and Horst-Michael Gross, "Predicting lesion growth and patient survival in colorectal cancer patients using deep neural networks," 2018.

[18] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 41–48.