Hofmann, Martin; Mäder, Patrick

Synaptic scaling - an artificial neural network regularization inspired by nature

Original published in: IEEE transactions on neural networks and learning systems / Institute of Electrical and Electronics Engineers. - [New York, NY] : IEEE. - 33 (2022), 7, p. 3094-3108.

Original published: 2021-01-27

ISSN: 2162-237X

DOI: 10.1109/TNNLS.2021.3050422

[*Visited:* 2023-03-16]

Synaptic Scaling—An Artificial Neural Network Regularization Inspired by Nature

Martin Hofmann and Patrick Mäder

***Abstract*— Nature has always inspired the human spirit and scientists frequently developed new methods based on observations from nature. Recent advances in imaging and sensing technology allow fascinating insights into biological neural processes. With the objective of finding new strategies to enhance the learning capabilities of neural networks, we focus on a phenomenon that is closely related to learning tasks and neural stability in biological neural networks, called homeostatic plasticity. Among the theories that have been developed to describe homeostatic plasticity, synaptic scaling has been found to be the most mature and applicable. We systematically discuss previous studies on the synaptic scaling theory and how they could be applied to artificial neural networks. Therefore, we utilize information theory to analytically evaluate how mutual information is affected by synaptic scaling. Based on these analytic findings, we propose two flavors in which synaptic scaling can be applied in the training process of simple and complex, feedforward, and recurrent neural networks. We compare our approach with state-of-the-art regularization techniques on standard benchmarks. We found that the proposed method yields the lowest error in both regression and classification tasks compared to previous regularization approaches in our experiments across a wide range of network feedforward and recurrent topologies and data sets.**

***Index Terms*— Computational neuroscience, homeostatic plasticity, information bottleneck, mutual information, neural network, regularization, synaptic scaling.**

I. Introduction

In 1943, McCulloch and Pitts [1] gained insights into brain function by formalizing the neuron concept describing the activity of nerve cells (McCulloch-Pitts-Cell). Later, machine learning concepts were frequently inspired by nature. Examples are Hebb's learning rule [2] that inspired learning algorithms [3] since 1949 and receptive fields in the visual cortex that inspired the convolution concept [4], [5], improving the accuracy of neural networks and reducing the computational cost due to a significant reduction of a network's parameters making neural networks suitable for a wide variety of hardware, including mobile devices. Even sexual procreation [6] inspired researchers to propose the dropout [7] regularization technique. In the meantime, neural networks exceeded human abilities in simple tasks, such as image recognition, and complicated tasks, such as the ancient game of Go. Prominent remaining problems are relatively slow learning and a high demand for labeled input data to train artificial neural networks (ANNs), while the human brain is able to learn from few examples and constantly learns from trial and error. Especially in edge and mobile computing scenarios with limited computational and memory resources, such as autonomous driving [8] and digital assistants [9], [10], methods improving generalization are a constant sought after. Aiming for advances in these problems, we study parallels and differences between natural and ANNs.

An important difference between artificial and natural neural networks is sleep. Sleep has many functions, is especially relevant for learning [11], and is, therefore, our key inspiration for further improving the generalization of ANN's. Natural learning and its link to sleep have been the subjects of a great variety of experiments discovering mechanisms that enable brains to learn faster from fewer examples. Initial experiments have shown that sleep greatly impacts humans' ability to build and save memories making the study of sleep phases an important research topic. A mechanism observed at sleep phase IV is slow wave sleep (SWS) [12], [13]. SWS is a brain state dominated by low strength noise resulting in slow and low-amplitude brain waves. Human subjects [14] and mice [15] not reaching this sleep phase suffer severe memory problems. This observation is linked to the role of homeostatic plasticity [16] in memory consolidation. Researchers observed that memories learned in connection with a special odor are better recognized later on if the learner is exposed to the flavor during the SWS sleep. No effect was found when the flavor was exposed in another sleep phase highlighting the importance of the SWS phase [17], [18].

Homeostatic plasticity [19] is considered a key concept during SWS and refers to a process of auto regulating synaptic connection strength. Today, the most mature and applicable theory describing homeostatic plasticity is synaptic scaling [20], [21]. Tetzlaff *et al.* [22] propose a mathematically stable description of natural synaptic scaling. The authors build on the theory that firing rates in natural neural networks are regulated through synaptic weights w that are scaled proportionally to the difference between the actual activity and certain target activity. They demonstrate that synapses are strictly stabilized in an input-determined way and also found that natural neural networks still learn when synaptic scaling takes place, even for random or recurrent network topologies.

In this article, we study the scaling of artificial neurons' activity and propose two methods for incorporating the scaling approach in a network's training process. We denote the first

Manuscript received June 7, 2019; revised January 23, 2020 and July 29, 2020; accepted November 7, 2020. Date of publication January 27, 2021; date of current version July 7, 2022. This work was supported in part by the German Federal Ministry for the Environment, Nature Conservation, Building and Nuclear Safety (BMUB) under Grant 3514685C19, Grant 3519685A08, and Grant 67KI2086A; in part by the German Ministry of Education and Research (BMBF) under Grant 01LC1319, Grant 01IS20062, and Grant 16PGF0304; in part by the Stiftung Naturschutz Thüringen (SNT) under Grant SNT-082-248-03/2014; in part by the Thuringian Ministry for Environment, Energy and Nature Conservation under Grant 68678; in part by the Friedrich Naumann Stiftung für die Freiheit PhD scholarship no. ST7847/P622; and in part by a Nvidia GPU Grant. *(Corresponding author: Martin Hofmann.)*

The authors are with the Department of Computer Science and Automation, Technische Universität Ilmenau, 98693 Ilmenau, Germany (e-mail: martin.hofmann@tu-ilmenau.de; patrick.maeder@tu-ilmenau.de).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2021.3050422.

Digital Object Identifier 10.1109/TNNLS.2021.3050422

method SynScaW, performing layerwise update according to Tetzlaf *et al.*'s rule. The second method, SynScaL, is a loss function that penalizes the difference between the mean output of a network and a specific target activation.

II. Mutual Information in Neural Networks

Mutual information is a fundamental concept in information theory and is, e.g., used to find parameters for optimal learning [23] or maximized in order to perform unsupervised learning [24]. Ben-David *et al.* [25] argue that compression and learning are equivalent through a proof that learnability of the family of sets $F*$ over the class of probability distributions $P*$ is undecidable. The authors describe learning as a game between compressor and reconstructor. While the compressor tries to find more efficient representations, the constructor tries to make sense of them. The construction is bound by the efficiency of the representation and the remaining amount of important information. Since the amount of information that can be passed to the reconstructor is limited, the compressor has to neglect less important information. Mutual information is a measure for the information of the input distribution that is passed to the reconstructor and has, therefore, been used to study the superior generalization of deep neural network topologies over shallow ones before [26], [27]. More precisely, mutual information $I(X;Y)$ measures the dependence of two random variables quantifying the information on a variable that is obtained if the other is known [28] and is formally defined as

$$I(X;Y) = I(Y;X) = \int_X \int_Y p(z,y) \log \frac{p(x,y)}{p(x)p(y)} dxdy. \quad (1)$$

A simple example illustrating the concept is rolling a die with six sides. As we roll the die, the number on the upper side is odd or even and, at the same time, prime or not prime. If someone, for example, tells us that she or he rolled prime, we know that the number is either 2, 3, or 5 and that the probability for being even is 33%. No matter what number is rolled and what property of the number we observe, we gain information on the other property too. Given x being prime or not prime and y being even or odd, the amount of information shared is the mutual information, which can be calculated for this example as

$$\begin{aligned} I(X;Y) &= \sum_{n=0}^{1} \sum_{m=0}^{1} p(x_n, y_m) \log \frac{p(x_n, y_m)}{p(x_n)p(y_m)} \\ &= 2 \times -0.097 + 2 \times 0.138 = 0.0817. \end{aligned} \quad (2)$$

In our example, two bits of information are needed to accurately describe each possible outcome: one bit distinguishing odd and even, and one bit distinguishing prime and not prime. The computed mutual information is a measure for how much information about one bit is gained if the other is observed.

Tishby and Zaslavsky [26] and Shwartz-Ziv and Tishby [29] discuss how mutual information can be used for examining the capabilities of neural networks. Fig. 1 illustrates how a single layer can be understood as a probability process, with mutual information being the difference between the input entropy $H(X)$ and the lost entropy $H(X|Z)$. Shwartz-Ziv and Tishby [29] discuss two major sources of entropy induced by a network's training process. The first source is randomly created training batches that only represent an estimated input distribution based on a small sample, leading to a distortedly learned target distribution and added noise in a network's weights. The second source is randomly

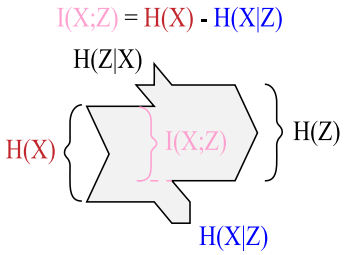

Fig. 1. Entropy $H(X)$ diminished by the conditional entropy $H(X|Z)$ is the mutual information $I(X;Z)$ that passes through a probability process, for example, a signal process or a layer of a neural network. Meanwhile, the conditional entropy $H(Z|X)$ is inserted into the process leading to the entropy $H(Z)$.

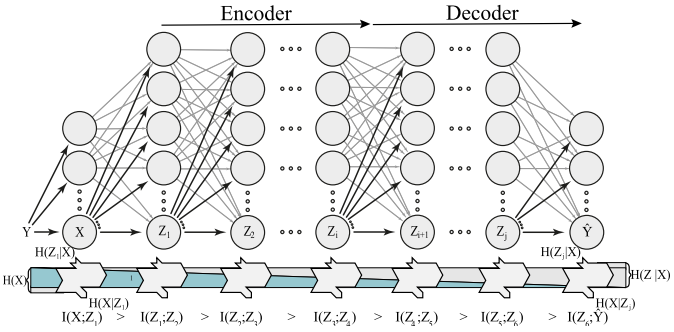

Fig. 2. Mutual information flow through a neural network in encoder–decoder scheme. The blue bar illustrates the loss of mutual information on the input from layer to layer. The true label Y forms the input X through a probability process. The mutual information on the input X in a hidden layer H_n decreases from layer to layer.

initialized weights and the noise that they induce into a network from the beginning of the training process. The authors further describe an ANN as a cascade of such probabilistic functions Z that processes a signal from the input X to the output $\hat{Y}$. Based on the process per layer, Fig. 2 shows that new entropy is added in every layer of a network, while the other entropy is filtered out leading to decreased mutual information on the input with every succeeding layer. When new input information passes through these partly noisy layers, additional entropy is added to them, and this entropy accumulates over a network's depth.

Tishby and Zaslavsky [26] systematically analyze ANNs' learning process using the mutual information concept. More specifically, the authors study how deep learning ANNs are able to generalize and to converge faster than shallow learning ANNs while having a comparable number of parameters. Thereby, the information about the true label $I(X;Y)$ corresponds to a network's generalization error [29]. They found that, in successfully trained networks, mutual information inserted into the network $\hat{X}$ is decreasing from layer to layer. Each succeeding layer aims to maximize the utilization of the remaining information toward predicting an output label $\hat{Y}$. Therefore, an efficient training process needs to train layers toward finding a minimum sufficient statistic. This goal is formulated as an information bottleneck objective

$$\mathcal{L}[p(z|x)] = I(Z;Y) - \beta I(Z;X) \quad (3)$$

where β is a positive parameter representing a trade-off between the complexity of the representation $R = I(Z;X)$ and the amount of relevant information preserved $I_Y = I(Z;Y)$.

The authors observed that ANNs are trained in two phases. In a memorization phase, weights are initially filled, whereby $R = I(Z; X)$ and I_Y are increasing until the network's degree of freedom in terms of parameters is reached. In a subsequent forgetting phase, the input X is compressed, and R gradually decreases due to a lack of storage left in the weights. Shwartz-Ziv and Tishby [29] report that, in the forgetting phase, the weight gradients' standard deviation is larger than their mean. The authors argue that gradients begin to behave like Gaussian noise with small means, meaning that random noise is added to the weights under the label constraint. Since random noise is not compressible, the information on the input left in the weights must be further compressed in order to store additional information. In practice, for example, the smaller a training batch size is chosen, the shorter the memorization phase will be, and the forgetting phase will be more distinctive. Worse gradient estimations will sooner lead to learning superimposed by induced random entropy and subsequent less comprehensively learned representation since the level of random entropy dominates the information present in the data set. A large batch size, however, will lower the influence of the entropy induced by the gradient descent and will lead to a longer memorization phase and a later forgetting phase [30]. Shwartz-Ziv and Tishby [29] found no significant dependence between mutual information of a layer and its width measured as the number of neurons (see [29, Fig. 5]), but they did find a significant dependence on the depth of a network measured as the number of layers leading to a pronounced forgetting phase (see [29, Fig. 5]).

Tishby and Zaslavsky [26] argue that compression behavior leads to special generalization bounds explaining the superior convergence of deep neural networks compared to shallow ones. The authors discuss that the generalization error ϵ decreases with increasing partition of the input space's entropy X_ϵ. They further argue that the cardinality of distribution Z_ϵ represented by an ANN is two to the power of the mutual information of itself on X ($|Z_\epsilon| \sim 2^{I(Z_\epsilon;X)}$). Furthermore, they state that H_ϵ, the entropy of the input covering partition covered on X, is similar to two to the power of the cardinality of X_ϵ and conclude that this entropy is similar to the cardinality of Z_ϵ ($|H_\epsilon| \sim 2^{|X|} \to 2^{|Z_\epsilon|}$). Eventually, Tishby *et al.* [31] conclude that the conventional generalization bounds

$$\epsilon^2 \leq \frac{\log |H_\epsilon| + \log \frac{1}{\delta}}{2m}$$

are updated in deep learning to

$$\epsilon^2 \leq \frac{2^{I(Z_\epsilon;X)} + \log \frac{1}{\delta}}{2m} \tag{4}$$

with ϵ being the generalization error, δ being the confidence, and m being the number of training examples. That is, the compression of a network's information on the input leads to a reduction in the number of necessary training samples for reaching the same generalization error ϵ by the power of two [32], [33].

The information theoretical analysis of ANNs has also been discussed controversially. For example, Saxe *et al.* [34] criticize the forgetting phase concept and argue that, in their experiments, nonlinearities were causing compression rather than noise induced by stochastic gradient descent in combination with minibatch entropy. However, Yu *et al.*'s [35] follow-up study supports Tishby *et al.*'s initial findings and shows that saturating nonlinearities were not causing compression. Therefore, we consider the information-theoretical approach valid for exploring synaptic scaling, a novel regularization approach that we propose in this article. In addition, we evaluate the application of synaptic scaling in a series of experiments.

III. Regularization Approaches for ANNs

Regularization aims to improve the generalization abilities of ANNs. In recent years, various regularization techniques have been proposed. A prominent approach is dropout [7] and its successor DropConnect [36] that disable each neuron's output with a certain probability, thereby introducing noise into a network's hidden units. Another regularization concept is label smoothing [37], [38] where the training labels' distribution and, accordingly, the learned output distribution are smoothed. All these techniques have been shown to reduce generalization error. Another research direction is the loss function that also aims to reduce generalization error. Alemi *et al.* [39] propose the variational information bottleneck (VIB) approach that utilizes variational embedding. Pereyra *et al.* [40] build upon this work and propose a technique called confidence penalty that adds the output entropy to the cross-entropy loss. They demonstrate that their approach outperforms previous regularization techniques.

In the following, we discuss relevant approaches in detail. Dropout [7] adds artificial noise into the training process of neural networks by randomly deactivating neurons. The approach stabilizes the training process and also scales the nondeactivated neurons according to the dropout rate for achieving a constant average activity. Training a neural network with dropout can be seen as training a large ensemble of different neural networks that share most of their weights. In the inference phase, dropout is not used and, therefore, allows the network to make a joined decision in the virtual ensemble that has been shown to reduce generalization error.

The remaining regularization techniques, which we want to discuss, operate by changing the loss function of neural networks. For a better understanding how all these techniques work, we first briefly introduce a major problem of the common cross-entropy loss function and then explain how the proposed approaches aim to mitigate this problem. The cross-entropy loss function is defined as

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^{N} [H(p(y|y_n), p(y|x_n]))] \tag{5}$$

where $(1/N) \sum_{n=1}^{N}$ is the mean over the batch with N being the number of training samples. Cross-entropy $H(p, q)$ is defined as

$$H(p, q) = -\sum_{y} p_y \log q_y. \tag{6}$$

The output distribution $q_y = p(y|x_n)$ of a neural network is typically given by logsoftmax of its output. The first argument p_y of the cross-entropy is the input distribution $p(y|y_n)$, which is a discrete probability distribution of only one nonzero probability. Hence, it is a one-hot encoding of the target label $p(y|y_n) = \delta_{y_n}(y)$ and trained to be as discriminative as the target's output distribution. Therefore, the network is trained to emphasize single low entropy outputs that are highly discriminative even if the decision is uncertain.

Label smoothing [37] is a simple attempt to conquer this behavior by filling the probability distribution $p(y|y_n)$ with

random nonzero elements $0 < e < \epsilon$ and by normalizing $p(y|y_n)$ afterward. Confidence penalty [40] is an approach that adds negative entropy of the output distribution to the cross-entropy loss in order to penalize models that have low entropy output distributions. The proposed confidence penalty loss function is defined as

$$\mathcal{L} = \frac{1}{N}\sum_{n=1}^{N}[H(p(y|y_n), p(y|x_n)) - \beta H(p(y|x_n))] \quad (7)$$

where $H(p(y|y_n), p(y|x_n))$ refers to the cross-entropy and $H(p(y|x_n))$ denotes the entropy. With this loss function, a network will be trained to prefer high entropy distributions without changing the target distribution, thereby overcoming the limitations of label smoothing.

The VIB approach [39] utilizes variational encoding, which adds the mutual information on the input to the cross-entropy loss. VIB was demonstrated to improve predictive accuracy while hardening the network against adversarial attacks. The proposed loss function is directly related to the information bottleneck objective [see (3)]. Due to the missing conditional output distribution $p(y|z)$, the exact calculation of mutual information is not affordable. The authors conquer this problem by using a variational approximation $q(y|z)$ for the mutual information [see 1]

$$I(Z;Y) = \int dydzp(y,z)\log\frac{q(y|z)}{p(y)}. \quad (8)$$

Based on this representation of mutual information, an upper bound for $I(Z;Y)$ can be derived as

$$I(Z;Y) \geq \int dydzp(y,z)\log q(y|z) + H(Y). \quad (9)$$

$H(Y)$ can be neglected since it is not dependent on the input. An upper bound can be computed since the Kullback–Leibler divergence between the true distribution $p(Y|Z)$ and its approximation $q(Y|Z)$ is always positive. Treating the distributions of X, Y, and Z as a Markov chain ($Y \leftrightarrow X \leftrightarrow Z$) allows to formulate the upper bound in a form that is solely dependent on distributions that are readily available

$$I(Z;Y) \geq \int dxdydzp(x)p(y|x)p(z|x)\log q(y|z). \quad (10)$$

Alemi *et al.* [39] substitute the unknown marginal distribution $p(z)$ with a variational approximation $r(z)$ to compute mutual information on the output $I(Z;X) = \int dz\,dx p(x,z)\log(p(z|x)/p(z))$. Analogously, the upper bound can be formulated as

$$I(Z;X) \leq \int dxdyp(x)p(z|x)\log\frac{p(z|x)}{r(z)}. \quad (11)$$

Utilizing the derived bounds, the VIB loss function is defined as

$$\begin{aligned}\mathcal{L} &= I(Z;Y) - \beta I(Z;X)\\ &\geq \int dxdydzp(x)p(y|x)p(z|x)\log q(y|z)\\ &\quad -\beta\int dydzp(x)p(z|x)\log\frac{p(z|x)}{r(z)}. \end{aligned} \quad (12)$$

The authors use a multilayer perceptron to approximate the distribution $p(z|x)$. To approximate the marginal $r(z)$, they chose a normal distribution with a zero mean and a scale of 1.0

$$\begin{aligned}\mathcal{L} = \frac{1}{N}\sum_{n=1}^{N}&\mathbb{E}_{\epsilon\sim p(\epsilon)}[-\log q(y_n|f(x_n,\epsilon))]\\ &+\beta KL[p(Z|x_n), r(Z)]. \end{aligned} \quad (13)$$

In conclusion, Alemi *et al.* [39] show that an information theory-based approach yields improvements to both the understanding of neural networks and its quality measures.

IV. Synaptic Scaling Concept

Bi and Poo [41], Sjöström *et al.* [42], and Froemke *et al.* [43] observed that synaptic weights in natural neural networks only increase in particular directions and with a reduced growth of strong neurons. Researchers refer to this activity of neurons as homeostatical regulation [20], [21], [44]. Homeostatically regulated neurons downscale strong weights in dependence to their activity resulting in a competition that improves synaptic sensitivity [45], [46]. Neurophysiologists studied natural neural networks and observed higher neural sensitivity when synaptic weights were regulated by a process following homeostatic plasticity [47]. Other researchers found the existence of homeostatic plasticity to be linked with the success of natural learning processes [48]–[54]. We hypothesize that synaptic scaling, a process that leads to homeostatic plasticity in natural neurons, could also support the decision on which information to remember and which to forget in ANNs.

Tetzlaff *et al.* [22] define a synaptic scaling mechanism that regulates a living neuron's activity and maintains a stable network, not being dominated by a single or a small group of neurons, as $H = \gamma(v_T - F(u))w^n$, where v_T refers to the target activation that we denote as z_T and $F(u,w)$ refers to the nonlinear activation (see [21, eq. (4)], [22, eq. (4)], and [52, eq. (4)]). Derived from this definition, we denote a scaling term for ANNs as

$$w_{\text{scaled}} = w + w^2\hat{z}\gamma \quad (14)$$

where the scaling rate γ adjusts the impact of the scaling, while $\hat{z}$ is the difference of the mean target activity $\bar{z}_T$ and a neuron's actual mean activation $\bar{z}$. The mechanism regulates strong neurons based on the difference between the actual and the target activity. That is synaptic scaling changes the distribution of synaptic weights, which may also be a viable approach for ANNs.

Teramae and Fukai [55] studied synaptic weight distributions in natural neurons and found that positive weights in the neocortex and hippocampus, where the learning is located, are log-normal distributed. The analysis is restricted to positive weights as only those are measurable today. The authors hypothesize that the discovered log-normal distribution could be the reason for the superior computational capabilities of natural neural networks over artificial ones. Buzsáki and Mizuseki [56] describe brain regions where activation patterns of principal cells are long-tail, typically log–normal distributed as a result of the weights' distribution. Another interesting finding is that not only the analyzed natural neural networks show this log-normal distribution of weights but also the ever-present noise in natural networks is considerably stronger than in natural networks with not log-normal distributed weights. This finding is interesting since it has been demonstrated that training noise significantly improves

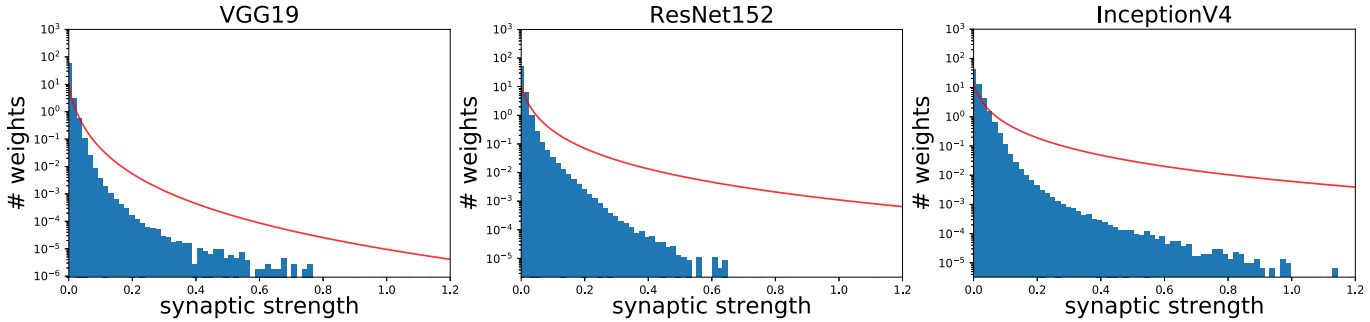

Fig. 3. Strength of synaptic weights for three popular topologies of ANNs. All three topologies have an over proportional number of weights close to zero. The line denotes a log-normal distribution.

on the generalization error of ANNs [57]. The other finding that effective natural networks show a log-normal distribution of weights is no less interesting and stimulated our initial study of weight distributions in trained ANNs.

Fig. 3 illustrates the distribution of positive weights for three common deep neural network topologies (VGG19, ResNet152, and Inception V4)[1] trained on the ImageNet [58] 1000-class data set. All three topologies show a similar weight distribution that appears almost log-normal distributed (see the red line in the figure), but statistical tests show significant differences in all three cases. Discovering this difference in weight distribution between trained artificial networks and their natural counterpart, we hypothesize that an artificial synaptic scaling mechanism that actively influences weight distribution may yield improved generalization error of an ANN much like its natural counterpart.

We propose a synaptic scaling technique for ANNs that follow its natural counterpart [22] and study two alternative ways of implementing this approach. For the SynScaW realization, the mean target activity $\bar{z}$ is calculated as activation of a uniformly distributed random batch of inputs. The weights are updated by iterating over the layers in between training steps

$$\bar{z} := \frac{1}{N}\sum_{n=0}^{N} f(x_n) \text{ with } f(x) = \phi(wx - b) \tag{15}$$

where ϕ denotes the activation function of one layer. Afterward, we calculate the difference of the mean activation of the layer $\hat{z}$ and the mean activation of the random input $\bar{z}_{\text{target}}$ as $\hat{z} = \bar{z}_{\text{target}} - \bar{z}$ with

$$\bar{z}_{\text{target}} = \frac{1}{N}\sum_{n}^{N} \phi(u_n), u_n \leftarrow \mathcal{U}(1, -1). \tag{16}$$

As another way to achieve synaptic scaling, we propose a synaptic scaling loss function SynScaL. The loss function encourages optimizers to train neural networks by exciting neurons that are less active than the mean and inhibiting overly active neurons. Propagating this error back through the network's layers efficiently affects every neuron without updating every single weight separately. For classification tasks, we assume training batches with equally distributed labels and denote the target mean activation as $E_T = 1/\#\text{classes}$. In the case of regression, the target activation is calculated as mean of the regression target, i.e., $z_T = 1/N \sum_n^N y_n$, where y_n is the nth sample regression target. The target mean activation for regression tasks is the mean of the output scalar. We jointly denote the synaptic scaling loss function for classification and regression cases as

$$\mathcal{L}' = \frac{1}{N}\sum_{1}^{N}\left[\mathcal{L} + \gamma\,\mathcal{L}_{\text{scal}}\right] \tag{17}$$

$$\mathcal{L}_{\text{scal}} = \begin{cases} \left(\dfrac{1}{\#\text{classes}} - p(x,z)\right)^2, & \text{classification} \\ \left(\left(\dfrac{1}{N}\displaystyle\sum_{m=0}^{N} y_m - \hat{y}\right)\right)^2, & \text{regression.} \end{cases} \tag{18}$$

The two proposed synaptic scaling methods act as regularizers by reducing overfitting. SynScaW and SynScaL penalize neurons with disproportionately high and low activity stimulating them to decrease or increase their activity, respectively. In this process, initially less important features become more relevant having a regulatory effect on the model's training.

A. Convergence Properties

In order to ensure that our synaptic scaling methods do not destabilize the training process, we evaluated their convergence properties. Sangari and Sethares [59] examined the convergence of mean squared error logistic regression and showed that it almost certainly converges. The logistic regression criterion ex $p(x) - p(\hat{x}) = (e_i^x / \sum e_i^x) - p(\hat{x})$ corresponds to our SynScaL method with a constant target probability of $E_T = \hat{x}$ giving us confidence in claiming that our method does not negatively influence convergence of a network's training. In contrast, we are not able to discuss the convergence of SynScaW in the same manner since a unique solution for a quadratic stochastic differential equation cannot be easily found. Alternative methods to examine convergence depend on distinct properties [60], [61] that are violated for the problem addressed in our case. However, a linearization leads to fixed-point solutions that have previously been presented by Tetzlaff *et al.* [22] to demonstrate convergence. We consider a profound proof of convergence further exercise and beyond the scope of this publication.

B. Analytical Evaluation on SynScaW

To initially explore the effect of synaptic scaling, we use a simple theoretic evaluation based on information theory. We formally describe the mutual information of a single layer

[1] https://pytorch.org/docs/stable/torchvision/models.html

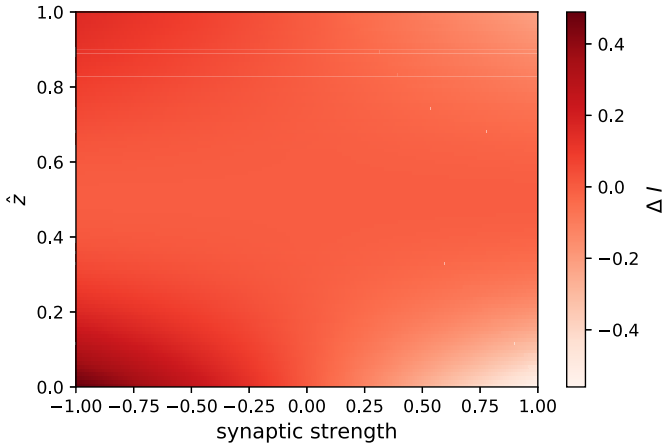

Fig. 4. Influence of synaptic scaling on the mutual information in a single-neuron layer and sigmoid activation. The abscissa shows the synaptic strength, i.e., the synaptic weight, the ordinate, and the difference between target and actual activities $\hat{z}$. Color indicates the change in mutual information ΔI with values between -1 and 1.

containing a single neuron with and without synaptic scaling and then compare both formulae. In this analysis, we use a sigmoid activation function and define the output distribution of the neuron analogous to [26] as

$$p(z|x) = \frac{1}{1 + e^{(-w\ p(x|z)-b)}}. \tag{19}$$

We have already defined mutual information using a joint probability distribution [see (1)]. Since an ANN exposes a conditional distribution, we reformulate (1) using Bayes' theorem into

$$I(X;Z) = I(Z;X) = \sum_{x\in X, z\in Z} p(z,x) \log\left(\frac{p(z|x)}{p(z)}\right) \tag{20}$$

which is the definition of the Kullback–Leibler divergence $I(X;Z) = I(Z;X) = D_{KL}[p(z,x)||p(z)p(x)]$. We can now insert the *sigmoid* activated example [see (19)] into this formula resulting in

$$I(Z;X) = \sum_{x\in X, z\in Z} p(z,x) \log\left(\frac{1}{p(z)(1 + e^{(-w\ p(x|z)-b)})}\right). \tag{21}$$

Equation 21 represents a simple neuron without scaling. To introduce synaptic scaling into this model we incorporate 14 as follows:

$$I(Z;X)_{\text{scaled}} = \sum_{x\in X, z\in Z} p(z,x) \times \log\left(\frac{1}{p(z)(1 + e^{-p(x|z)(w\ +w^2\hat{z}\gamma)-b}}\right). \tag{22}$$

We compare the example with and without scaling by computing $\Delta I := I(Z;X)_{\text{scaled}} - I(Z;X)$. Appendix VII-A provides additional details on the computations.

Fig. 4 shows the of plots ΔI in relation to synaptic strength and $\hat{z}$. For this plot, we set $p(x|z)$ to 0.5 because changing $p(x|z)$ results just in a scaled absolute value. For $p(x|z) = 0$, the ΔI shrinks to 0. We also set z_{target} to 0.5 since the mean activation of a sigmoid activated neuron with uniform random inputs between -1 and 1 is 0.5. The plot shows that the way synaptic scaling affects the mutual information depends on $\hat{z}$, the divergence of the actual activation $\bar{z}$ from the expected mean activation $\bar{z}_{\text{target}}$, and whether the synaptic strength is positive or negative. Negative values of ΔI indicate a reduction of mutual information when using synaptic scaling. To further explore the figure, we divide the discussion into two cases. The first case refers to positive synaptic strengths and corresponds to the first and the fourth quadrants of the plot. The second case refers to negative synaptic strengths and corresponds to the second and third quadrants of the plot. Studying the first case, we find that ΔI is negative over the whole two quadrants. This means that scaling neurons with positive synaptic strength leads to a reduction in mutual information in our setup. The opposite is true for negative strengths and the second case. Since the minimum value in the first case is lower than the maximum in the second case, our conclusion is that synaptic scaling at least lowers the average mutual information. Scaling only positive weights will lead to a reduction of mutual information in this scenario. Changing mutual information while training ANNs is assumed to alter the training process and the generalization error.

C. Network's Mutual Information With SynScaW

For a better understanding of how synaptic scaling affects ANNs, we study its effect on the mutual information captured in two trained ANN topologies. We considered two classes of algorithms: sample-based methods and variational approximation to estimate this mutual information. Sample-based methods record neurons' activations across many samples in order to estimate mutual information. Marginal distributions are then derived by binning recorded activations making the estimation highly dependent on the way bins to be formed [62]. Variational approximation aims to overcome this problem by learning the distribution of layer activations through an additional VIB layer consisting of n neurons representing means and n neurons representing variances of the desired marginal distribution $p(z)$ [39]. We decided to obtain variational approximations of mutual information [see (1) and (8)] by adding VIB layers to the networks under inspection. During training, we compute a loss for each VIB layer inserted into a network's topology [see (13)] and add this loss to the output loss of the network. Each VIB layer consists of $2 \times N$ neurons encoding N means and N variances. $N \times 12$ samples are drawn from the encoded distributions with the reparameterization trick [63] and fed into a decoder. The decoder consists of one neuron per target class. For the calculation of the upper bound of mutual information, the Kullback–Leibler divergence between the sample and the normal distribution is calculated. Since the variational encoder requires its own loss function [see (3)], we restrict this analysis to SynScaW.

In the first experiment, we apply synaptic scaling to a simple MLP topology with 784-1024-1024-10 neurons that have also been used by Alemi *et al.* [39] and train it on the MNIST data set. In this setup, the bounds of mutual information compute to $I(Z;X) = 55.99$ on the input and $I(Z;Y) = 3.22$ on the output at an error rate of 1.17% without synaptic scaling and to $I(Z;X) = 28.38$ on the input and $I(Z;Y) = 3.19$ on the output at an error rate of 1.14% with synaptic scaling. In conclusion, we observe that the information on the input drops by half with synaptic scaling, while the information on the output remains almost unchanged. At the same time, the error rate decreases with synaptic scaling from 1.17% to 1.14%. This observation supports the hypothesis that lower mutual information on the input, i.e., a higher compression,

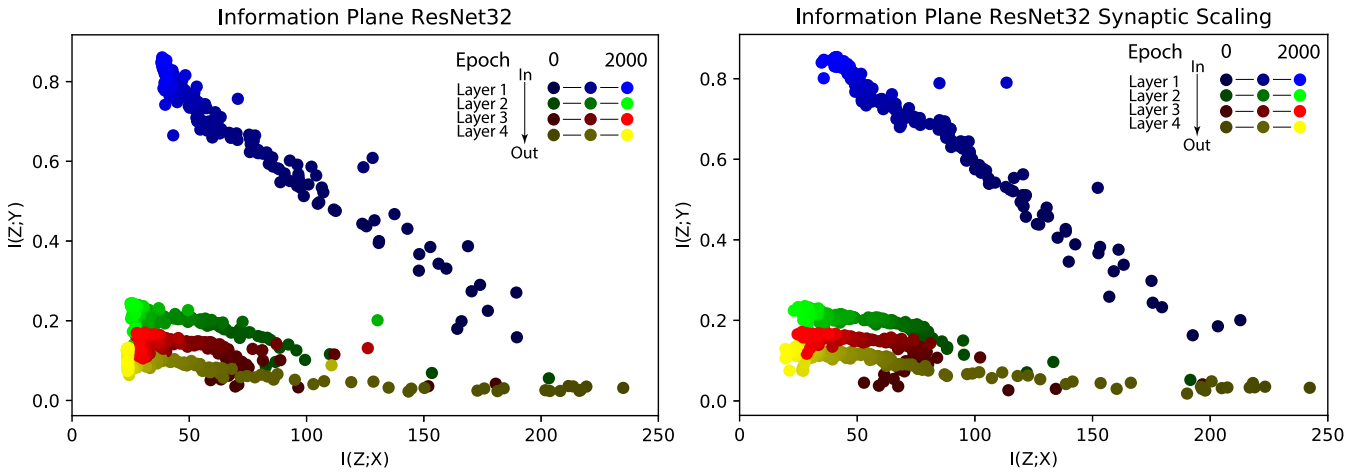

Fig. 5. Visualization of the information plane for two training setups. Darker colors indicate earlier epochs. Blue indicates the first layer, yellow the last. The yellow layer learns the least bits of information on the output and the input, while the blue layer learns the most. Synaptic scaling leads to superior compression [lower $I(Z;X)$] and equal $I(Z;Y.)$. The accuracies are 64.48% without Synaptic Scaling and 71.39% with scaling every epoch. Every dot is a mean of 50 runs and 10 epochs. The most information on the output is shown by the ResNet32 setting without synaptic scaling with $I(Z;Y) = 38.38$ in its last layer. The highest and fastest compression is shown by the last layer of ResNet32 scaling, with $I(Z;X) = 34.31$.

yields a lower generalization error and that synaptic scaling improves compression.

In the second experiment, we visualize the effects of synaptic scaling in the information plane, as proposed by Shwartz-Ziv and Tishby [29]. The information plane shows the information on the output $I(Z;Y)$ over the information on the input $I(Z;X)$ per epoch of a network's training process. This kind of analysis allows additional insight into the training process of a network. Accordingly, we apply the same method as in the previous experiment for estimating the bounds of mutual information [39]. We insert three VIB layers between bottleneck blocks into a residual network with 32 layers [64]. We study a ResNet topology since it makes use of state-of-the-art techniques, such as residuals and batch normalization in contrast to the simple MLP topology studied above. We train these networks for 2000 epochs. Fig. 5 shows that the mutual information changes quickly in the first of the 2000 epochs of the training and then slowly but steadily further changes into the same direction. Finally, mutual information follows the training behavior of the neural network. In comparison, both ResNet32 configurations show similar behavior but different levels of compression and accuracies. All layers of the setting without synaptic scaling show higher information on both the input and the output, while the accuracy of the setup with synaptic scaling is considerably higher 71.39% compared to the one without 68.48%.

Concluding from these evaluations, we found that layers trained with the proposed synaptic scaling approach show less information on the input compared to regularly trained ones. Our observations are also conclusive to the hypothesis of Tishby *et al.* that lower information on the input causes a lower generalization error [see (4)]. In Section V, we will, therefore, study whether and how synaptic scaling impacts the generalization error of trained networks and compare these results to other regularization techniques.

D. Network's Weights' and Activations' Distribution

The observation that synaptic weights are log-normal distributed in biological neural networks raises the question of how synaptic weights in ANNs are distributed and how synaptic scaling affects their distribution. Therefore, we exemplarily evaluate the Jensen–Shannon divergence of the synaptic weights' distribution of an ANN from the log-normal distribution under different scaling rates and during training. We report the Jensen–Shannon divergence, essentially a two-sided Kullback–Leibler divergence, since it can be interpreted as s metric. Specifically, we evaluated the test error and the synaptic weights and the synaptic activations distributions' Jensen–Shannon divergence from the log-normal distribution averaged over ten training runs of a two-layer MLP on the MNIST data set [see Fig. 6(a) and (c)]. To evaluate the synaptic activity, we recorded all activations for the whole test set of the experiment. We observe that the divergence decreases over time until it starts to alternate. At this time, the testing error is already saturated. This might indicate the compression phase described by Tishby *et al.*, an observation that could be further explored in the future. Especially, when comparing the Jensen–Shannon divergence of activations with and without synaptic scaling, we observe substantially more log-normal distributed activations with our proposed methods. Fig. 6(b) and (d) presents test error rates and the synaptic weights and synaptic activation distributions' divergence from the log-normal distribution for a parameter variation of the scaling rate γ. We observe that the scaling rate influences divergence and test error in our example. The lowest test error was achieved at a scaling rate of 0.085 for SynScaW and 0.74 for SynScaL.

V. Evaluation

In this section, we discuss multiple empirical studies that we conducted in order to evaluate the effect of SynScaW and SynScaL on different network topologies trained with different data sets and varying parameterization.[2]

A. Experimental Setup

1) Studied Data Sets: We utilize four data sets for our experimentation: MNIST [65], CIFAR-10 [66],

[2]Example: https://github.com/SECSY-Group/SynapticScaling

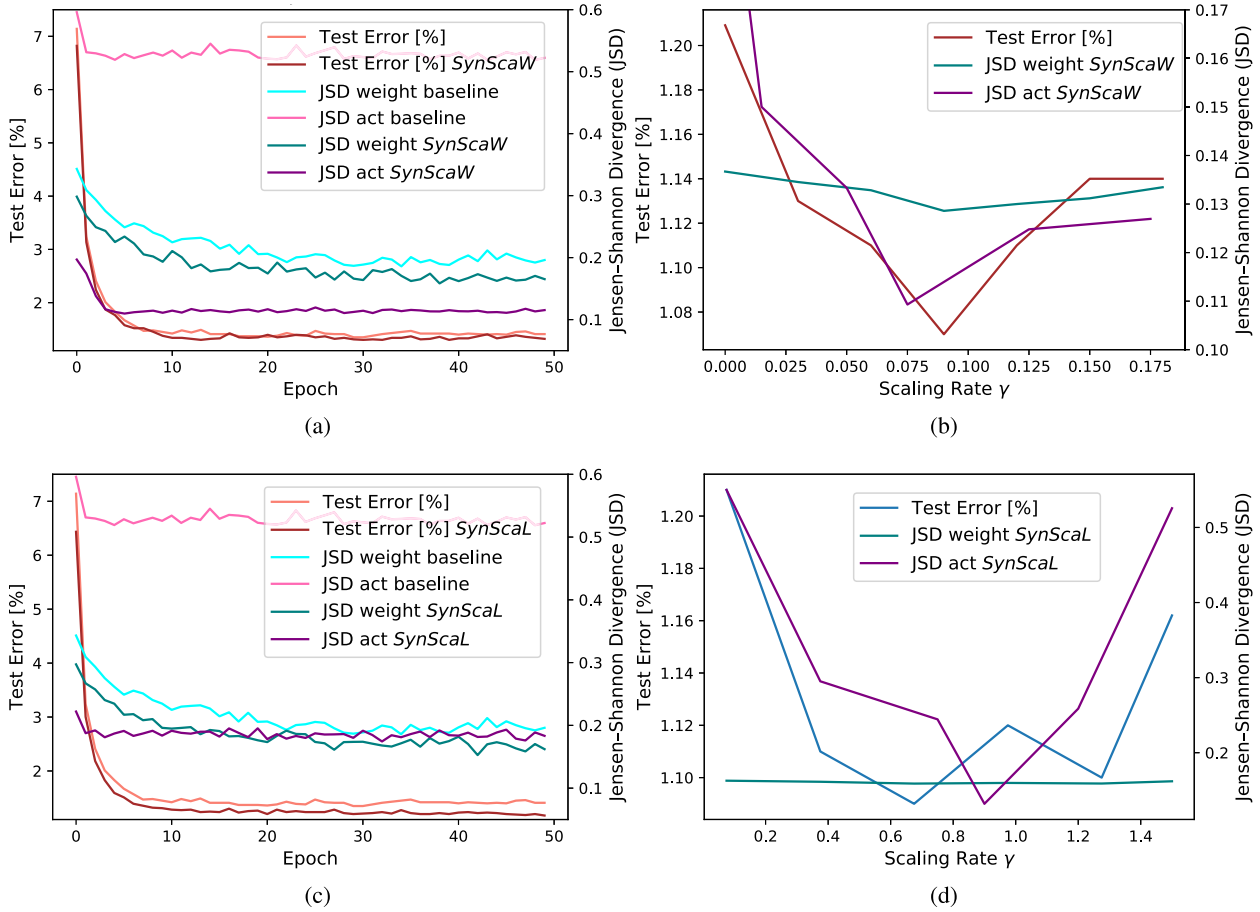

Fig. 6. Observed Jensen–Shannon divergence from the log-normal distribution shown on the primary ordinate and test error shown on the secondary ordinate. Figures at the top show results for SynScaL and SynScaW at the bottom. Figures on the left-hand side present changes during training. On the right-hand side, results for different scaling rates γ are shown. All values are averaged over ten training runs on the MNIST data set. (a) Classification error and Jensen–Shannon divergence during training with SynScaW. (b) Classification error and Jensen–Shannon divergence for different scaling rates γ and SynScaW. (c) Classification error and Jensen–Shannon divergence during training with SynScaL. (d) Classification error and Jensen–Shannon divergence for different scaling rates γ and SynScaL.

CIFAR-100 [66], and ImageNet [58]. These data sets are characterized by increasing complexity in terms of classes to be separated. MNIST, the smallest data set, consists of 70 000 square, gray-scale images of handwritten digits with a size of 28×28 pixels, and a depth of one byte, split into 60 000 training and 10 000 test images. The CIFAR-10 and CIFAR-100 data sets are nonoverlapping labeled subsets of the 80 million tiny images data set [67] with ten and 100 classes, respectively. They contain 60 000 square color images with a size of 32×32 pixels and a depth of 3 bytes split into 50 000 training and 10 000 test images. Accordingly, CIFAR-100 contains exactly 500 training images per class and CIFAR-10 exactly 5000. ImageNet, the largest data set, consists of 1 281 167 high-resolution images downloaded from the internet belonging to 1000 classes that were filtered and afterward labeled by a human. All four are well-known benchmarks and, therefore, allow for easy comparison of results with other authors.

2) Studied Network Topologies: We study the synaptic scaling approach on three network topologies with increasing complexity. As the most simple topology, we study an MLP consisting of two dense hidden layers in a topology of 784-1024-1024-10 neurons. We include this configuration to gain comparable results especially to Alemi *et al.* [39] and Pereyra *et al.* [40]. To provide results on a common und broadly available topology, we choose two versions of the ResNet architecture [64] that incorporates convolution [68], batch normalization [69], and residuals [64]. In order to evaluate the VIB method [39], we modified the topologies by replacing the last dense layer with a variational encoder block. In both cases, MLP and ResNet32, neither input augmentation nor input normalization, were utilized. The last topology that we use is the original ResNet50 [64] topology, including input normalization and input augmentation. Since all four data sets pose classification problems, we use a cross-entropy loss function (see Appendix VII-B).

The PyTorch framework was used for all experimentation. For various configurations, we discovered subtle differences between previously published and our results possibly due to nonpublished hyperparameters in the original study. In order to facilitate overall comparability, we decided to rely on our replicated results.

3) Hyperparameter Search: To identify appropriate values for the hyperparameters learning rate and scaling rate γ, we performed a parameter search [70] trained on a 20% split of the MNIST training set while validating on the remaining 80%. We found that a learning rate of 0.0005 and an exponential decay of 94%, every 5 epochs yielded the best results. We used

TABLE I

CLASSIFICATION ERROR OF SYNAPTIC SCALING WEIGHTS (SYNSCAW) AND SYNAPTIC SCALING LOSS (SYNSCAL) ON THE MNIST, CIFAR-10, CIFAR-100, AND IMAGENET DATA SETS IN COMPARISON TO THE BASELINE WITHOUT SYNAPTIC SCALING AND OTHER REGULARIZATION METHODS

	MLP with 2 hidden units			**ResNet32**			**ResNet50**
Regularization	MNIST Err. (Δ)[%]	CIFAR-10 Err. (Δ)[%]	CIFAR-100 Err. (Δ)[%]	MNIST Err. (Δ)[%]	CIFAR-10 Err. (Δ)[%]	CIFAR-100 Err. (Δ)[%]	ImageNet Err. (Δ)[%]
– baseline without regularization –	1.21	41.47	68.58	0.33	12.50	40.85	24.76
Dropout	1.27 (-4.1)	41.77 (-0.7)	70,78 (-3.1)	–*	–*	–*	–*
Confidence Penalty	1.14 (5.8)	41.26 (0.5)	68.96 (-0.6)	0.30 (9.1)	12.20 (2.4)	38.64 (5.4)	–+
Variational Information Bottleneck	1.17 (3.3)	41.17 (0.7)	69.27 (-1.0)	0.31 (6.1)	13.24 (5.9)	51.91 (-27.1)	–+
$SynScaW$	1.14 (5.8)	41.15 (0.8)	**67.88** (1.0)	0.33 (0.0)	**11.72** (6.2)	**38.07** (6.8)	**24.01** (3.0)
$SynScaL$	1.12 (7.4)	**40.80** (1.6)	68.55 (0.0)	**0.29** (12.1)	12.28 (1.8)	40.88 (-0.1)	24.63 (0.5)
$SynScaW$ + Confidence Penalty	1.11 (8.3)	41.13 (0.8)	68.82 (-0.3)	0.35 (-6.1)	12.47 (0.2)	40.84 (0.0)	–+
$SynScaL$ + Confidence Penalty	**1.05** (13.2)	41.26 (0.5)	68.96 (-0.6)	0.33 (0.0)	11.98 (4.2)	38.97 (4.6)	–+

(*) Dropout cannot be applied on the ResNet topology (cp. He et al. [65]) due to a phenomenon called neural shift (cp. Li et al. [72]). (+) Confidence penalty and variational information bottleneck could not be computed on ResNet50 due to their special loss functions making a training of a large network unstable without extensive fine tuning.

TABLE II

JENSEN–SHANNON DIVERGENCE OF THE TRAINED WEIGHTS DISTRIBUTION PER REGULARIZATION–DATA SET–TOPOLOGY COMBINATION FROM THE IDEAL LOG-NORMAL DISTRIBUTION (SMALLER IS BETTER)

	MLP with 2 hidden units			**ResNet32**		
Regularization	MNIST JSD	CIFAR-10 JSD	CIFAR-100 JSD	MNIST JSD	CIFAR-10 JSD	CIFAR-100 JSD
– baseline without regularization –	0.206	0.043	0.036	0.062	0.051	0.044
Dropout	0.240	0.034	0.056	–*	–*	–*
Confidence Penalty	0.185	0.043	0.035	0.080	0.095	0.068
Variational Information Bottleneck	0.273	0.136	0.105	0.062	0.050	0.045
$SynScaW$	0.125	**0.030**	**0.025**	**0.044**	**0.046**	**0.043**
$SynScaL$	0.226	0.037	0.042	0.056	0.048	0.059
$SynScaW$ + Confidence Penalty	**0.112**	0.033	0.038	0.080	0.076	0.107
$SynScaL$ + Confidence Penalty	0.124	0.037	0.031	0.067	0.055	0.047

(*) Dropout cannot be applied on the ResNet topology (cp. He et al. [65]) due to a phenomenon called neural shift (cp. Li et al. [72]).

this learning rate for the MLP and ResNet32 topologies. For the ResNet50, we followed He *et al.*'s [64] training procedure and hyperparameters as closely as possible, initializing the learning rate to 0.1 and divided by 10 on plateauing error. The ResNet50 model was trained for 600 000 iterations with a weight decay of 0.0001 and a momentum of 0.9.

We discovered the optimal scaling rates for SynScaW and SynScaL as 0.01 and 1.0, respectively. In these experiments, γ remained constant throughout an entire training procedure. Both configurations use the ADAM optimizer [63], and training was terminated after 200 epochs, making results comparable to those of [39] and [40]. The batch size was set to 100.

B. Synaptic Scaling of Weights SynScaW

Table I summarizes the results for the examined combinations of network topologies and data sets (columns) and regularization approaches (rows). The table's cells show the absolute classification error and the relative change as a percentage (green—improvement; red—degradation) compared to the baseline without regularization (first row).

The SynScaW approach lowers the classification error compared to the baseline on all evaluated combinations though, for one combination, it remains almost identical. The MNIST classification error shrinks from 1.21% to 1.14% for the MLP being a 5.8% relative improvement, while it remains almost unchanged for the ResNet32. The CIFAR-10 classification error shrinks relatively by 0.8% for the MLP and by 6.2% for the ResNet32, the latter being the largest improvement for this data set in our study. The CIFAR-100 classification error shrinks from 68.58% to 67.88% for the MLP being a 1.0% relative improvement and from 40.85% to 38.07% for the ResNet32 being a 6.8% relative improvement. Both are the highest measured improvements for the respective data set in our study.

After training with SynScaW, we observe, for five out of six combinations, the lowest divergence of the synaptic weights' distribution from a log-normal distribution, the only exception being the MLP trained on MNIST (see Table II). The neural activations' divergence was observed the lowest, training with SynSvaW in every experiment and decreasing from CIFAR-10 to CIFAR-100 (see Table III).

When comparing the SynScaW results to previously proposed regularization approaches, we observe that the proposed approach results in lower classification error for the MLP and the ResNet32 trained on CIFAR-10 and CIFAR-100. For the MLP trained on MNIST, confidence penalty provides equal results; for the ResNet32 trained on MNIST, confidence penalty and VIB achieve higher improvements in classification error. We also evaluated the confidence penalty in combination with SynScaW since these are applicable at the same time. However, only for the smallest combination of the MLP trained on MNIST, we observe an improvement over SynScaW by itself.

Applying SynScaW while training a ResNet50 on the ImageNet data set (see Table I) results in a classification error that is relatively 3.0% lower than the baseline (see Fig. 7). Fig. 7 compares classification error throughout the training process for the baseline and two proposed synaptic scaling approaches. The figure shows the classification error

TABLE III

JENSEN–SHANNON DIVERGENCE OF THE TEST ACTIVATION DISTRIBUTION PER REGULARIZATION–DATA SET–TOPOLOGY COMBINATION FROM THE IDEAL LOG-NORMAL DISTRIBUTION (SMALLER IS BETTER)

	MLP with 2 hidden units			**ResNet32**		
Regularization	MNIST JSD	CIFAR-10 JSD	CIFAR-100 JSD	MNIST JSD	CIFAR-10 JSD	CIFAR-100 JSD
– baseline without regularization –	0.551	0.172	0.182	0.148	0.126	0.691
Dropout	0.474	0.217	0.366	–*	–*	–*
Confidence Penalty	0.348	0.352	0.182	0.041	0.063	0.057
Variational Information Bottleneck	0.262	0.295	0.306	0.115	0.141	0.098
$SynScaW$	**0.121**	**0.042**	**0.019**	**0.031**	**0.027**	**0.003**
$SynScaL$	0.152	0.048	0.058	0.037	0.064	0.072
$SynScaW$ + Confidence Penalty	0.171	0.113	0.076	0.094	0.063	0.092
$SynScaL$ + Confidence Penalty	0.147	0.103	0.094	0.127	0.078	0.622

(*) Dropout cannot be applied on the ResNet topology (cp. He et al. [65]) due to a phenomenon called neural shift (cp. Li et al. [72]).

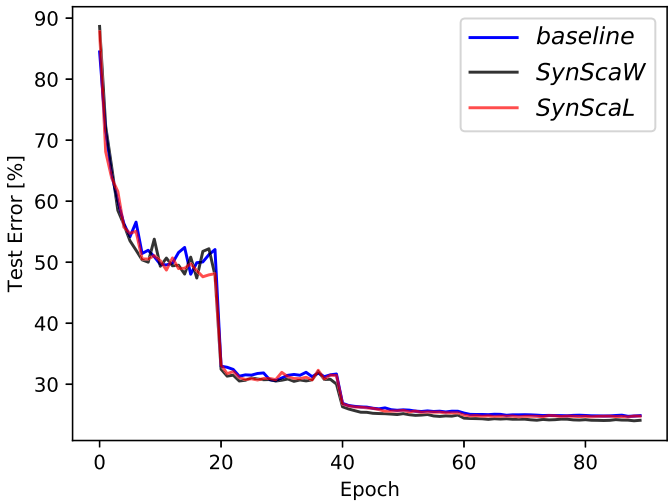

Fig. 7. Test error for a ResNet50 trained on the ImageNet data set without synaptic scaling (baseline), with SynScaW, and with SynScaL.

continuously lower than the baseline, especially in the later training epochs.

C. Synaptic Scaling Loss Function SynScaL

Table I also shows results for the proposed SynScaL approach evaluated on the same combinations of network topologies and data sets. Similar to SynScaW, the SynScaL approach lowers the classification error compared to the baseline on all but one of the evaluated combinations. We observe improvements in terms of reduced classification error, especially for the MNIST and CIFAR-10 data sets. The MNIST classification error shrinks from 1.21% to 1.12% for the MLP being a 7.4% relative improvement and from 0.33% to 0.29% for the ResNet32 being a 12.1% relative improvement. The CIFAR-10 classification error shrinks from 41.47% to 40.80% for the MLP being a 1.6% relative improvement and from 12.50% to 12.28% for the ResNet32 being a 1.8% relative improvement. For CIFAR-100, the classification error remains almost identical compared to the baseline. For the MLP trained on the MNIST data set, we observe even better results when combining SynScaL with confidence penalty yielding the lowest overall error of 1.05% for this topology.

Evaluations of the Jensen–Shannon divergence (see Table II) between the synaptic weights' distribution of the network trained with SynScaL and a log-normal distribution show lower divergence than the baseline for only two of the six trained combinations. The activations' divergence was observed lower than the baseline and increasing from CIFAR-10 to CIFAR-100 (see Table III).

Applying SynScaL while training a ResNet50 on the ImageNet data set (see Table I) results in a classification error that is relatively 0.5% lower than the baseline. Fig. 7 compares classification error throughout the training process for the baseline and the two proposed synaptic scaling approaches. The figure shows that the classification error is continuously lower than the baseline, especially, in the later training epochs, but higher than the SynScaW classification error.

VI. DISCUSSION

We evaluated the concept of synaptic scaling and its effect on the training of ANNs by assessing: 1) the flow of mutual information throughout a network's layers; 2) the distribution of trained weights; and 3) the accuracy of trained classifiers in an experimental setup with different topologies and data sets.

In an analytical evaluation, we found that the biologically inspired synaptic scaling approach is able to influence mutual information in artificial neurons. We further explored this result and exemplarily visualized mutual information in the information plane. We found that layers in networks trained with synaptic scaling show less mutual information on the input and conclude that potentially more generalizing feature representations can be trained with these networks ultimately resulting in higher classification accuracies.

Biological observations show that synaptic weights and the activity of natural neurons in the cortex are log-normal distributed. We, therefore, evaluated the divergence of trained weights' and activations' distribution from a log-normal distribution and found that, throughout the training with synaptic scaling, the distribution of weights becomes more and more log-normal distributed. Once converged, the network's activations that are trained with synaptic scaling (SynScaW and SynScaL) are more log-normal distributed than the traditionally trained counterpart suggesting a connection analogous to the observation from neurobiology. We also observed that measured accuracies tend to correlate with the divergence of neurons' activations from a log-normal distribution, and this divergence is dependent on the scaling rate.

Experiments with benchmark data sets and common network topologies suggest that a regularization through SynScaW has a positive effect on the classification accuracy of the trained classifier in general. We observe, however, that the positive effect of SynScaW on the error tends to increase with the number of classes in the classification problem.

We also observe a positive effect on the classification error for the SynScaL regularization, which, however, tends to decrease with a growing number of classes in the problem, i.e., CIFAR-100 and ImageNet. For a large number of classes, the effect of synaptic scaling, when applied via the loss function, is distributed across more classes resulting in an overall lower target activation, i.e., E_T [see (17)] decreases with a growing number of classes. In contrast to previous regularization approaches, we found especially SynScaW to yield superior results. A combination of our synaptic scaling approaches with the confidence penalty approach was only beneficial for the smallest evaluated combination of an MLP trained on MNIST. We also found that the scaling rates that we had determined for the MLP trained on MNIST were transferable to other topologies and data sets leaving room for further improved results in a data set topology-specific hyperparameter search. We successfully applied synaptic scaling to a larger classification problem by training a ResNet50 on the ImageNet data set with augmentation. We did by purpose not tune any hyperparameter and transferred the scaling rate from a much smaller data set and topology. Nonetheless, we found that both synaptic scaling regularizations yielded improved accuracy compared to our baseline, which followed the initially published training procedure as close as possible. Furthermore, we performed an extensive evaluation of a variety of smaller data sets, topologies, and tasks (see Appendix VII-B) to further substantiate our results. In this evaluation, SynScaW showed the lowest test errors across various classification and regression tasks and across feedforward and recurrent network topologies.

VII. Conclusion

Inspired by homeostatic plasticity observed as an essential mechanism in vertebrates' sleep to realize effective learning processes, we found synaptic scaling to be a concept that describes its origin from the perspective of neurons' interconnections. We successfully transferred the concept of synaptic scaling to ANNs and studied its effect on network weight distribution. Our comparative experiments demonstrated beneficial effects in every studied configuration ranging from small data sets and simplistic network topologies to large data sets and complex topologies. Synaptic scaling also showed superior regularization compared to other techniques in both feedforward and recurrent topologies for either classification or regression.

Further research is needed to investigate how synaptic scaling affects the derivative of the loss function to allow the optimizer to reach a lower minimum. A good candidate regarding this analysis is the exploration of singularities. It is also highly interesting to find convergence proof for the algorithm that resulted from Tetzlaff *et al.* [22]'s rule since the proposed fixed-point solutions are not considered final. In addition, the influence of synaptic scaling on the convergence rate should be further examined. A useful method for convergence analysis come from dynamic programming–based methods outstandingly demonstrated by Arora *et al.* [72] and Reddy *et al.* [73] evaluating learning schemes. Another interesting point is that homeostatic plasticity and synaptic scaling are but two observations in natural neural networks connected to sleep. Further research could examine additional processes appearing during sleep. Another direction is the visualization and description of changes and states in neural networks with other techniques. Further research will also pay more attention to synaptic activations and what similarities and differences are observable in natural and artificial models and how synaptic scaling changes these similarities.

Appendix

A. Calculus Concerning (25)

We did a short analytical examination of synaptic scaling (see Section IV-B). The starting point was the definition of the mutual information [see (20)]. We inserted a sigmoid activation function [see (19)] resulting in

$$I(Z;X)=\sum_{x\in X,z\in Z}p(z,x)\log\left(\frac{1}{p(z)(1+e^{-w\ p(x|z)-b})}\right). \quad (23)$$

Accordingly, we formulated the mutual information in our example neuron, by inserting the scaled weight [see (14)]

$$I(Z;X)_{\text{scaled}}=\sum_{x\in X,z\in Z}p(z,x)\times\log\left(\frac{1}{p(z)(1+e^{-p(x|z)(w\ +w^2\bar{z}\gamma-b)})}\right). \quad (24)$$

We substitute $(1+e^{-w\ p(x|z)-b})$ with ξ for simplification

$$I(Z;X)=\sum_{x\in X,z\in Z}p(z,x)\log\left(\frac{1}{p(z)\xi}\right)$$

and $(1+e^{-w\ p(x|z)+w^2\bar{z}\gamma-b})$ with δ to receive

$$I(Z;X)_{\text{scaled}}=\sum_{x\in X,z\in Z}p(z,x)\log\left(\frac{1}{p(z)\delta}\right).$$

Now, we can calculate the difference between mutual information with and without synaptic scaling

$$\begin{aligned}I(Z;X)_{\text{scaled}}-I(Z;X)&:=\Delta I\\&=\sum_{x\in X,z\in Z}p(z,x)\log\left(\frac{1}{p(z)\delta}\right)\\&\quad-\sum_{x\in X,z\in Z}p(z,x)\log\left(\frac{1}{p(z)\xi}\right).\end{aligned}$$

Since the same elements are summed up, we can conclude

$$\Delta I=\sum_{x\in X,z\in Z}p(z,x)\left[\log\left(\frac{1}{p(z)\delta}\right)-\log\left(\frac{1}{p(z)\xi}\right)\right].$$

Using $-\log(1/x)=\log x$ results in

$$\left[\sum_{x\in X,z\in Z}p(z,x)\log\left(\frac{1}{p(z)\delta}\right)+\log(p(z)\xi)\right].$$

Using $\log a-\log b=\log(a/b)$ results in

$$\Delta I=\sum_{x\in X,z\in Z}p(z,x)\log\left(\frac{p(z)\xi}{p(z)\delta}\right).$$

Applying back substitution, the eventual result becomes

$$\Delta I=\sum_{x\in X,z\in Z}p(z,x)\log\left(\frac{\xi}{\delta}\right).$$

TABLE IV

OVERVIEW OF THE STUDIED BENCHMARK DATA SETS. FOR CLASSIFICATION TASKS, WE LIST THE ACCURACY OF THE REFERENCE IMPLEMENTATION. FOR THE REGRESSION TASK, I.E., THE PARKINSON DATA SET, WE REPORT THE MEAN ABSOLUTE ERROR (MAE). THE LETTERS *s* AND *c* DENOTE SCALAR AND CATEGORICAL VALUES, RESPECTIVELY, WHILE CLASS AND REG REFER TO CLASSIFICATION AND REGRESSION TASKS

		Input variables						
Dataset	Task/ #Classes	#/Type	Spatial Variability	#Samples	Balance	Acc[%]/ MAE	Validation	Ref.
Thyroid Disease	class/3	21s	invariant	7,200	low	97.35	3-fold cross-validation	[75], [76]
Epileptic Seizure+	class/5	4,097s	variate	500	high	100.00	10-fold cross-validation (classes D,E)	[77], [76]
Breast Cancer	class/2	1s+8c	invariant	286	medium	74.90	80/20, 100x random subsampling	[78], [76]
Breast Cancer Wisc.	class/2	10s	invariant	699	medium	98.50	5-fold cross-validation	[79], [76]
IRIS	class/3	4s	invariant	150	high	100.00	80/20, 100x random subsampling	[80], [76]
Mesothelioma	class/2	34s	invariant	324	medium	96.30	3-fold cross-validation	[81]
Parkinson	reg/2	16s	invariant	5,923	–	5.80,7.50	90/10, 1,000x random subsampling	[82], [76]
Red Wine	reg/1	11s	invariant	4,897	–	0.45	5-fold cross-validation	[83], [76]
White Wine	reg/1	11s	invariant	1,598	–	0.45	5-fold cross-validation	[83], [76]
Diabetes	reg/1	10s	invariant	441	–	38.90	5-fold cross-validation	[79], [76]
Abalone	reg/1	8s	invariant	4,177	–	–	10-fold-cross-validation	[84], [76]
Life Expectancy	reg/1	19s	invariant	2,937	–	–	–	–
IMDB	seq. class/2	19s	invariant	2,937	high	97.4	5-fold cross-validation	[51]
Sequential MNIST	seq. class/10	784s	variant	70,000	high	97.0	10,000/60,000-split	[85]

In conclusion, the term for the difference in mutual information is

$$\begin{aligned}\Delta I &= \sum_{x\in X, z\in Z} p(z,x)[I(Z;X)_{\text{scaled}} - I(Z;X)] \\ &= \sum_{x\in X, z\in Z} p(z,x)\log \\ &\quad \times\left(\frac{1+e^{-w\ \ p(x|z)-b}}{1+\exp\left(-p(x|z)(w\ \ +w^2\bar{z}\gamma)-b\right)}\right). \quad (25)\end{aligned}$$

B. Additional Evaluation on Various Benchmark Data Sets

In this section, we present further experiments with the proposed synaptic scaling approaches on a variety of benchmark data sets.

1) Studied Data Sets: We selected a variety of data sets, originating mostly from life sciences, which are frequently used for evaluation and for which previous results are available. Table IV provides an overview of the seven utilized data sets and their characteristics. The data sets vary in the spatial input, the task, the number of classes, the balance of training data, the number of input variables and their type, and the number of training samples and, therefore, provide a rich test set for the proposed synaptic scaling approach. The Thyroid Disease data set consists of 7200 samples in three highly imbalanced classes and a medium number of variables [85]. It represents the most imbalanced data set in our selection. The Epileptic Seizure data set consists of five classes with 100 samples each [86] and represents a large number of samples and spatially variant features. We augment the samples following scheme 1 proposed by Tzallas *et al.* [76], reducing the dimensionality by slicing the 4097 scalar values per sample in chunks of 512 with a stride of 64. Samples for testing are obtained by slicing the 4097 scalar values into four separately evaluated chunks. The medium-sized Breast Cancer data set consists of 286 samples belonging to two classes.[3] Input data contain eight categorical variables inflated into 32 binary variables. Low reference results [77] suggest that the data lack important information about the classes, making it a challenging test task. The medium-sized Breast Cancer Wisconsin data set is imbalanced but includes a higher number of data points to facilitate higher classification results. The very small and highly balanced IRIS data set [87] consists of three classes and four features with 50 samples per class. Finally, the Mesothelioma [80] data set consists of a medium number of samples belonging to two fairly imbalanced classes and a high number of features. In total, these data sets cover combinations from high dimensionality with limited samples to low dimensionality with a high number of samples.

In the second set, we selected four regression data sets. The Parkinson Tele Monitoring data set [81] poses a regression task of two targets and provides a fairly large number of samples with a medium number of input variables. Furthermore, we selected the red and white wine data sets [82] offering fewer samples and fewer variables. A further data set [88] is the Abalone data set, which consists of 4177 samples with eight attributes. The last data set is a Life Expectation data set [89] that consists of 2937 samples with 19 scalar variables mined from WHO data. The data set is provided by Kaggle.[4] Another data set is the diabetes disease progression data set [90] that comes with 441 samples and ten scalar attributes. In addition, we used the sequential MNIST data set [68] that consists of a train and a test split with 60 000 samples and 10 000, respectively, with 784 scalar variables and the IMDB sentiment analysis data set [91], which consists of 2937 samples with 19 scalar variables.

2) Studies Network Topologies: We evaluated all data sets, except for the Epileptic Seizure data set, with the same network topology consisting of four hidden layers with the same number of neurons per layer. Thereby, we calculated the number of neurons, in relation to the problem, like ten times the number of input features times the number of classes. We used Leaky ReLU as the activation function. For the Epileptic Seizure data set, we propose a novel topology that first decomposes the input into its constituting frequencies using stacks of dilated convolutions and then extracts features using 2-D convolutions. The input is fed into every dilated convolution in the stack at the same time and stacked to a

[3]The data are provided by M. Zwitter and M. Soklic from the University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia.

[4]https://www.kaggle.com/kumarajarshi/life-expectancy-who/metadata

TABLE V

EVALUATION RESULTS FOR THE STUDIED BENCHMARK DATA SETS. SIGNIFICANCE LEVELS ARE DENOTED BY "*," "'**," AND "***" REFERRING TO A STUDENT'S T-TEST WITH A P-VALUE EQUAL OR BELOW 0.1, 0.01, AND 0.001, RESPECTIVELY, WHILE *−* DENOTES NO SIGNIFICANT DIFFERENCE BETWEEN THE VALUE AND THE BASELINE. + REFERS TO THE RELATIVE ERROR MAE/$\text{MAD}_{\text{MEDIAN}}$

	Baseline		Dropout		Confidence Penalty		$SynScaW$		$SynScaL$	
Dataset	Acc[%]/MAE	Epoch	Acc[%]/MAE	Epoch	Acc[%]/MAE	Epoch	Acc[%]/MAE	Epoch	Acc[%]/MAE	Epoch
Classification tasks with MLP										
Thyroid Disease	98.24±0.1	65	96.39±0.2*	75**	97.89±0.3-	65-	**98.64** ±0.1*	**56***	**98.64**±0.1*	64-
Epileptic Seizure	89.55±0.2	99	38.50±0.7***	99-	89.45±0.1-	99-	**90.00** ±0.1*	**81****	89.70±0.1-	99-
Breast Cancer	74.90±0.8	12	71.55±0.7*	20***	75.26±0.8-	**9***	**77.66**±0.6***	20***	76.89±0.6***	11-
Breast Cancer Wisc.	97.90±0.1	15	97.57±0.2*	**9***	97.99±0.2-	**9***	**98.14**±0.2*	**9***	**98.14**±0.2*	11-
IRIS	98.67±0.4	8	97.33±0.1*	8-	98.67±0.1-	8-	**100.00** ±0.0-	**7***	**100.00**±0.0-	8-
Mesothelioma	100.00±0.0	3	**100.00**±0.0-	4-	98.14±0.1*	56**	**100.00** ±0.0-	3-	**100.00**±0.0-	**2***
Regression tasks with MLP										
Parkinson motor	6.20±0.1	66	6.50±0.1*	**28*****	–		**6.10**±0.1-	41**	6.40±0.1*	45**
Parkinson total	5.90±0.1	64	6.10±0.1*	**26*****	–		**5.80**±0.1-	43**	5.90±0.1-	43**
Red Wine	0.10±0.0	94	0.11±0.0*	77**	–		**0.09**±0.0*	**70***	**0.09**±0.0*	**70***
White Wine	0.11±0.0	78	0.12±0.0-	74-	–		**0.10**±0.0-	**60***	**0.10**±0.0-	62*
Diabetes	38.52±0.4	99	42.54±2.3*	**98-**	–		**37.23**±0.3*	99-	41.62±1.3+	99-
Abalone+	0.67±0.0	93	0.80±0.1-	50*	–		**0.65**±0.0*	83*	0.67±0.0-	**31*****
Life Expectancy	0.01±0.0	97	0.01±0.0*	85*	–		**0.00**±0.0**	97-	0.00±0.0-	**59*****
Classification tasks with sequential data										
IMDB GRU	82.26±1.3	5	80.76±0.1**	5-	82.76±0.2*	5-	**82.98**±0.2*	5-	82.32±0.0*	4-
IMDB LSTM	87.57±0.9	4	84.39±0.2**	5-	86.01±0.2*	5-	**88.05**±0.1*	5-	87.60±0.0-	4-
seq. MNIST GRU	98.32±0.5	2	97.62±0.1*	3-	98.14±0.1*	2-	**98.49**±0.1**	2-	98.37±0.0*	2-
seq. MNIST LSTM	98.83±0.3	3	98.53±0.2-	4-	98.91±0.1-	2-	**98.90**±0.0*	2-	98.83±0.0*	2-

2-D representation afterward. To make sure that the outputs that are stacked together share the same length, we limit the length of the input that is fed into every convolution separately. We refer to the network topology as the frequency decomposition network (FDN). For the Epileptic Seizure data set, we used 128 dilated convolutions with a kernel size of 2 and a dilation ranging from 1 to 128. The stacked outputs yield a 128×128 feature map with three channels. On top of this, we perform a common convolutional feature extraction consisting of four stacks of two 2-D convolutional layers separated by max-pooling layers. The convolutional layers have a kernel size of three each and a stride of one, while the maximum pooling layers have a kernel size of two and a stride of two. The classifier is one single linear classifier. All outputs except the last are batch normalized and Leaky ReLU activated. In total, the FDN has 1.9 million parameters.

3) Experimental Setup: We used Adam as an optimizer for all data sets with a learning rate of 0.01, betas of 0.9 and 0.999, ϵ of $1E-08$, and a weight decay of 0. We use a cross-entropy loss function weighted by the classes' prevalence. The learning rate is decayed by 10% every 40 epochs. The scaling rate for SynScaW was set to 0.01 and 0.5 for SynScaL following a hyperparameter search on a randomly chosen 10% validation split of the Epileptic Seizure data set. To further stabilize the training process, we decayed the SynScaW scaling rate analogous to the learning rate. For the dropout experiments, probabilities were set to 0.2 for input neurons and 0.5 for all other neurons. For the confidence penalty experiments, the parameter β was set to 1.0. All networks are trained for 100 epochs. We perform tenfold cross-validation to evaluate test accuracy as a fair trade between classifier stability and data set variance estimation [92], [93]. Especially, to be able to differentiate results in case of 100% accuracy, we also analyze and report the average number of epochs resulting in the highest accuracy.

4) Evaluation Results: Table V presents results of our experiments either in terms of computed accuracy (classification task) or mean absolute error (regression task) and standard deviation across the ten-fold cross-validation. In a second column per evaluated regularization approach, we report the highest accuracy averaged over all runs and the average epoch in which it was observed. We statistically tested the results of all methods against the respective baseline and report the computed significance level next to the result. Per data set (rows), we report from left to right results for the baseline without regularization, for the dropout approach [7], the confidence penalty approach [40], the proposed SynScaW approach, and the proposed SynScaL approach.

SynScaW yields superior classification and regression results for all seven data sets with accuracies significantly above the baseline for four data sets. SynScaL yields similar results that are also above the baseline for the classification tasks but is comparable or worse than the baseline for the regression tasks. Dropout achieves the worst MAE on the Parkinson data set and also delivers results below the baseline for the other data sets. Confidence Penalty provides results comparable to the baseline for the classification tasks but does not support regression tasks. For the regression experiments, we observe that synaptic scaling leads to lower errors. We noticed that SynScaL reduced the epochs in most cases and that SynScaW did not. Finally, we gained a few observations on recurrent networks. In these recurrent setups, SynScaW produced the highest accuracies among the methods.

ACKNOWLEDGMENT

Open access of this paper is sponsored by the BMBF Open Access Post Grant grant no. 16PGF0304.

REFERENCES

[1] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, Dec. 1943.

[2] D. G. Hebbs, *The Organization of Behavior*. New York, NY, USA: Wiley, 1949.

[3] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[4] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *J. Physiol.*, vol. 148, no. 3, pp. 574–591, Oct. 1959.

[5] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *J. Physiol.*, vol. 195, no. 1, pp. 215–243, Mar. 1968.

[6] A. Livnat, C. Papadimitriou, N. Pippenger, and M. W. Feldman, "Sex, mixability, and modularity," *Proc. Nat. Acad. Sci. USA*, vol. 107, no. 4, pp. 1452–1457, 2010.

[7] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[8] V. R. Kumar *et al.*, "FisheyeDistanceNet: Self-supervised scale-aware distance estimation using monocular fisheye camera for autonomous driving," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 574–581.

[9] M. Hofmann, M. Seeland, and P. Mäder, "Efficiently annotating object images with absolute size information using mobile devices," *Int. J. Comput. Vis.*, vol. 127, no. 2, pp. 207–224, Feb. 2019, doi: 10.1007/s11263-018-1093-3.

[10] M. Rzanny, P. Mäder, A. Deggelmann, M. Chen, and J. Wäldchen, "Flowers, leaves or both? How to obtain suitable images for automated plant identification," *Plant Methods*, vol. 15, no. 1, p. 77, Dec. 2019, doi: 10.1186/s13007-019-0462-4.

[11] G. Yang, C. S. W. Lai, J. Cichon, L. Ma, W. Li, and W.-B. Gan, "Sleep promotes branch-specific formation of dendritic spines after learning," *Science*, vol. 344, no. 6188, pp. 1173–1178, Jun. 2014.

[12] S. R. Williams, T. I. Tóth, J. P. Turner, S. W. Hughes, and V. Crunelli, "The 'window' component of the low threshold Ca^{2+} current produces input signal amplification and bistability in cat and rat thalamocortical neurones," *J. Physiol.*, vol. 505, no. 3, pp. 689–705, Dec. 1997.

[13] R. Marrocco, "Arousal systems," *Current Opinion Neurobiol.*, vol. 4, no. 2, pp. 166–170, 1994.

[14] M. P. Walker, "The role of slow wave sleep in memory processing," *J. Clin. Sleep Med.*, vol. 5, no. 2, p. 6, Apr. 2009.

[15] V. Ego-Stengel and M. A. Wilson, "Disruption of ripple-associated hippocampal activity during rest impairs spatial learning in the rat," *Hippocampus*, vol. 20, no. 1, pp. 1–10, 2010.

[16] G. Wang, B. Grone, D. Colas, L. Appelbaum, and P. Mourrain, "Synaptic plasticity in sleep: Learning, homeostasis and disease," *Trends Neurosci.*, vol. 34, no. 9, pp. 452–463, Sep. 2011.

[17] B. Rasch, C. Buchel, S. Gais, and J. Born, "Odor cues during slow-wave sleep prompt declarative memory consolidation," *Science*, vol. 315, no. 5817, pp. 1426–1429, Mar. 2007.

[18] S. Gais *et al.*, "Sleep transforms the cerebral trace of declarative memories," *Proc. Nat. Acad. Sci. USA*, vol. 104, no. 47, pp. 18778–18783, Nov. 2007.

[19] K. B. Hengen, A. T. Pacheco, J. N. McGregor, S. D. Van Hooser, and G. G. Turrigiano, "Neuronal firing rate homeostasis is inhibited by sleep and promoted by wake," *Cell*, vol. 165, no. 1, pp. 180–191, Mar. 2016.

[20] D. Stellwagen and R. C. Malenka, "Synaptic scaling mediated by glial TNF-alpha," *Nature*, vol. 440, no. 7087, pp. 1054–1059, 2006.

[21] G. G. Turrigiano and S. B. Nelson, "Homeostatic plasticity in the developing nervous system," *Nature Rev. Neurosci.*, vol. 5, no. 2, pp. 97–107, Feb. 2004.

[22] C. Tetzlaff, "Synaptic scaling in combination with many generic plasticity mechanisms stabilizes circuit connectivity," *Frontiers Comput. Neurosci.*, vol. 5, p. 47, Nov. 2011.

[23] J. Barbier, F. Krzakala, N. Macris, L. Miolane, and L. Zdeborová, "Optimal errors and phase transitions in high-dimensional generalized linear models," *Proc. Nat. Acad. Sci. USA*, vol. 116, no. 12, pp. 5451–5460, Mar. 2019.

[24] R. D. Hjelm *et al.*, "Learning deep representations by mutual information estimation and maximization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019, pp. 1–24. [Online]. Available: https://openreview.net/forum?id=Bklr3j0cKX

[25] S. Ben-David, P. Hrubeš, S. Moran, A. Shpilka, and A. Yehudayoff, "Learnability can be undecidable," *Nature Mach. Intell.*, vol. 1, no. 1, pp. 44–48, Jan. 2019.

[26] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Apr. 2015, pp. 1–5.

[27] L. Paninski, "Estimation of entropy and mutual information," *Neural Comput.*, vol. 15, no. 6, pp. 1191–1253, Jun. 2003.

[28] T. M. Cover and J. A. Thomas, *Elements of Information Theory* (Wiley Series in Telecommunications and Signal Processing), 2nd ed. Hoboken, NJ, USA: Wiley, 2006.

[29] R. Shwartz-Ziv and N. Tishby, "Opening the black box of deep neural networks via information," 2017, *arXiv:1703.00810*. [Online]. Available: http://arxiv.org/abs/1703.00810

[30] F. Liu, "Implementation and verification of the information bottleneck interpretation of deep neural networks," M.S. thesis, School Elect. Eng. Comput. Sci., KTH Roy. Inst. Technol., Stockholm, Sweden, 2018.

[31] V. N. Vapnik, *Statistical Learning Theory*. New York, NY, USA: Wiley, 1998. [Online]. Available: http://www.loc.gov/catdir/description/wiley032/97037075.html

[32] O. Shamir, S. Sabato, and N. Tishby, "Learning and generalization with the information bottleneck," *Theor. Comput. Sci.*, vol. 411, nos. 29–30, pp. 2696–2711, Jun. 2010.

[33] N. Tishby. *Opening the Black Box of Deep Neural Networks Via Information: A Deeper Theory and Some New Algorithms*. Accessed: Apr. 30, 2017. [Online]. Available: http://icri-ci.technion.ac.il/files/2017/05/ICRI-CI-retreat-2017-Abstra%cts-160430.pdf

[34] A. M. Saxe *et al.*, "On the information bottleneck theory of deep learning," *J. Stat. Mech., Theory Exp.*, vol. 2019, no. 12, Dec. 2019, Art. no. 124020.

[35] S. Yu, K. Wickstrom, R. Jenssen, and J. Principe, "Understanding convolutional neural networks with information theory: An initial exploration," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 435–442, Jan. 2021.

[36] L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in *Proc. 30th Int. Conf. Mach. Learn.*, vol. 28, S. Dasgupta and D. McAllester, Eds. Atlanta, GA, USA: PMLR, 2013, pp. 1058–1066. [Online]. Available: http://proceedings.mlr.press/v28/wan13.html

[37] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[38] L. Xie, J. Wang, Z. Wei, M. Wang, and Q. Tian, "DisturbLabel: Regularizing CNN on the loss layer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4753–4762.

[39] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," 2016, *arXiv:1612.00410*. [Online]. Available: http://arxiv.org/abs/1612.00410

[40] G. Pereyra, G. Tucker, J. Chorowski, L. Kaiser, and G. E. Hinton, "Regularizing neural networks by penalizing confident output distributions," *CoRR*, vol. abs/1701.06548, pp. 1–5, Dec. 2017. [Online]. Available: https://openreview.net/pdf?id=HyhbYrGYe

[41] G.-Q. Bi and M.-M. Poo, "Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type," *J. Neurosci.*, vol. 18, no. 24, pp. 10464–10472, Dec. 1998.

[42] P. J. Sjöström, G. G. Turrigiano, and S. B. Nelson, "Rate, timing, and cooperativity jointly determine cortical synaptic plasticity," *Neuron*, vol. 32, no. 6, pp. 1149–1164, Dec. 2001.

[43] R. C. Froemke, M.-M. Poo, and Y. Dan, "Spike-timing-dependent synaptic plasticity depends on dendritic location," *Nature*, vol. 434, no. 7030, pp. 221–225, Mar. 2005.

[44] G. G. Turrigiano and S. B. Nelson, "Hebb and homeostasis in neuronal plasticity," *Current Opinion Neurobiol.*, vol. 10, no. 3, pp. 358–364, Jun. 2000.

[45] N. Kasthuri and J. W. Lichtman, "The role of neuronal identity in synaptic competition," *Nature*, vol. 424, pp. 426–430, Jul. 2003.

[46] S. Song, P. J. Sjöström, M. Reigl, S. Nelson, and D. B. Chklovskii, "Highly nonrandom features of synaptic connectivity in local cortical circuits," *PLoS Biol.*, vol. 3, no. 3, p. e68, Mar. 2005.

[47] S. Song, K. D. Miller, and L. F. Abbott, "Competitive hebbian learning through spike-timing-dependent synaptic plasticity," *Nature Neurosci.*, vol. 3, no. 9, pp. 919–926, Sep. 2000.

[48] A. A. Minai and W. B. Levy, "Sequence learning in a single trial," in *Proc. INNS World Congr. Neural Netw.*, 1993, pp. 505–508.

[49] L. F. Abbott and K. I. Blum, "Functional significance of long-term potentiation for sequence learning and prediction," *Cerebral Cortex*, vol. 6, no. 3, pp. 406–416, 1996.

[50] K. I. Blum and L. F. Abbott, "A model of spatial map formation in the hippocampus of the rat," *Neural Comput.*, vol. 8, no. 1, pp. 85–93, Jan. 1996.

[51] M. R. Mehta, M. C. Quirk, and M. A. Wilson, "Experience-dependent asymmetric shape of hippocampal receptive fields," *Neuron*, vol. 25, no. 3, pp. 707–715, Mar. 2000.

[52] L. F. Abbott and S. B. Nelson, "Synaptic plasticity: Taming the beast," *Nature Neurosci.*, vol. 3, no. S11, pp. 1178–1183, Nov. 2000.

[53] R. P. N. Rao and T. J. Sejnowski, "Predictive sequence learning in recurrent neocortical circuits," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 164–170.

[54] S. Bogdanovich *et al.*, "Functional improvement of dystrophic muscle by myostatin blockade," *Nature*, vol. 420, no. 6914, pp. 418–421, Nov. 2002.

[55] J.-N. Teramae and T. Fukai, "Computational implications of lognormally distributed synaptic weights," *Proc. IEEE*, vol. 102, no. 4, pp. 500–512, Apr. 2014.

[56] G. Buzsáki and K. Mizuseki, "The log-dynamic brain: How skewed distributions affect network operations," *Nature Rev. Neurosci.*, vol. 15, no. 4, pp. 264–278, Apr. 2014.

[57] C. M. Bishop, "Training with noise is equivalent to Tikhonov regularization," *Neural Comput.*, vol. 7, no. 1, pp. 108–116, Jan. 1995.

[58] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[59] A. Sangari and W. Sethares, "Convergence analysis of two loss functions in soft-max regression," *IEEE Trans. Signal Process.*, vol. 64, no. 5, pp. 1280–1288, Mar. 2016.

[60] J. A. Bucklew, T. G. Kurtz, and W. A. Sethares, "Weak convergence and local stability properties of fixed step size recursive algorithms," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 966–978, May 1993.

[61] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of ADAM and beyond," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–23. [Online]. Available: https://openreview.net/forum?id=ryQu7f-RZ

[62] B. Poole, S. Ozair, A. Van Den Oord, A. Alemi, and G. Tucker, "On variational bounds of mutual information," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, vol. 97, K. Chaudhuri and R. Salakhutdinov, Eds. Long Beach, CA, USA: PMLR, Jun. 2019, pp. 5171–5180. [Online]. Available: http://proceedings.mlr.press/v97/poole19a.html

[63] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2014, pp. 1–14.

[64] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[65] Y. LeCun, C. Cortes, and C. Burges. *Mnist Handwritten Digit Database*. Accessed: Jan. 6, 2021. [Online]. Available: http://yann.lecun.com/exdb/mnist

[66] A. Krizhevsky, "Learning multiple layers of features from tiny images," M.S. thesis, Dept. Comput. Sci., Univ. Tronto, Toronto, CA, USA, 2009.

[67] A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images: A large data set for nonparametric object and scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 1958–1970, Nov. 2008.

[68] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Dec. 1998.

[69] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, vol. 37, F. Bach and D. Blei, Eds., Jul. 2015, pp. 448–456.

[70] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, Feb. 2012.

[71] X. Li, S. Chen, X. Hu, and J. Yang, "Understanding the disharmony between dropout and batch normalization by variance shift," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2677–2685.

[72] V. Arora, L. Behera, T. K. Reddy, and A. P. Yadav, "HJB equation based learning scheme for neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 2298–2305.

[73] T. Kumar Reddy, V. Arora, and L. Behera, "HJB-equation-based optimal learning scheme for neural networks with applications in brain–computer interface," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 4, no. 2, pp. 159–170, Apr. 2020.

[74] I. Ionita, "Prediction of thyroid disease using data mining techniques," *Broad Res. Artif. Intell. Neurosci.*, vol. 7, pp. 115–124, Dec. 2016.

[75] D. Dua and C. Graff. (2017). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[76] A. T. Tzallas, M. G. Tsipouras, and D. I. Fotiadis, "Epileptic seizure detection in EEGs using time–frequency analysis," *IEEE Trans. Inf. Technol. Biomed.*, vol. 13, no. 5, pp. 703–710, Sep. 2009.

[77] I. Fischer and J. Poland, "Amplifying the block matrix structure for spectral clustering," in *Proc. 14th Annu. Mach. Learn. Conf. Belgium Netherlands*, 2005, pp. 21–28.

[78] K. D. Humbird, J. L. Peterson, and R. G. Mcclarren, "Deep neural network initialization with decision trees," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1286–1295, May 2019.

[79] M. M. Akhlagh, S. Chiang Tan, and F. Khak, "Temporal data classification and rule extraction using a probabilistic decision tree," in *Proc. Int. Conf. Comput. Inf. Sci. (ICCIS)*, Jun. 2012, pp. 346–351.

[80] O. Er, A. C. Tanrikulu, A. Abakay, and F. Temurtas, "An approach based on probabilistic neural network for diagnosis of Mesothelioma's disease," *Comput. Electr. Eng.*, vol. 38, no. 1, pp. 75–81, Jan. 2012.

[81] A. Tsanas, M. A. Little, P. E. McSharry, and L. O. Ramig, "Accurate telemonitoring of Parkinson's disease progression by noninvasive speech tests," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 4, pp. 884–893, Apr. 2010.

[82] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis, "Modeling wine preferences by data mining from physicochemical properties," *Decis. Support Syst.*, vol. 47, no. 4, pp. 547–553, Nov. 2009.

[83] Ä. Uysal and H. A. Gávenir, "Instance-based regression by partitioning feature projections," *Int. J. Speech Technol.*, vol. 21, no. 1, pp. 57–79, Jul. 2004.

[84] Q. Le, N. Jaitly, and G. Hinton, "A simple way to initialize recurrent networks of rectified linear units," *CoRR*, vol. abs/1504.00941, pp. 1–9, Apr. 2015.

[85] J. R. Quinlan, P. J. Compton, K. A. Horn, and L. Lazarus, "Inductive knowledge acquisition: A case study," in *Proc. 2nd Austral. Conf. Appl. Expert Syst.*, 1987, pp. 137–156.

[86] R. G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. E. Elger, "Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 64, no. 6, Nov. 2001, Art. no. 061907.

[87] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugenics*, vol. 7, no. 2, pp. 179–188, Sep. 1936.

[88] P. Shannon, "Cytoscape: A software environment for integrated models of biomolecular interaction networks," *Genome Res.*, vol. 13, no. 11, pp. 2498–2504, Nov. 2003.

[89] E. Jaba *et al.*, "Statistical evaluation of the influence of determining factors of life expectancy," *Anal. Stiintifice Univ. Alexandru Ioan Cuza Iasi-Stiinte Economice, Alexandru Ioan Cuza Univ., Fac. Econ. Bus. Admin.*, vol. 2011, pp. 215–223, Dec. 2011.

[90] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Ann. Statist.*, vol. 32, no. 2, pp. 407–499, 2004.

[91] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol.* Portland, OR, USA: Association for Computational Linguistics, 2011, pp. 142–150. [Online]. Available: https://www.aclweb.org/anthology/P11-1015

[92] C. Beleites *et al.*, "Variance reduction in estimating classification error using sparse datasets," *Chemometric Intell. Lab. Syst.*, vol. 79, nos. 1–2, pp. 91–100, Oct. 2005.

[93] J.-H. Kim, "Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap," *Comput. Statist. Data Anal.*, vol. 53, no. 11, pp. 3735–3745, Sep. 2009.

Martin Hofmann received the master's degree from Technische Universität Ilmenau, Ilmenau, Germany, in 2015.

He has been inventing and building a high-speed grain analysis machine working at the Steinbeis Qualitätssicherung und Bildverarbeitung, Ilmenau, under the supervision of Prof. Notni and Prof. Linß. After joining the Flora Incognita Group of Prof. Mäder at TU Ilmenau in 2016, he continued working in computer vision moving on to neural networks and computational neuroscience improving neural architectures with methods borrowed from nature.

Patrick Mäder received the Diploma degree in industrial engineering and the Ph.D. degree (Hons.) in computer science from Technische Universität Ilmenau, Ilmenau, Germany, in 2003 and 2009, respectively.

He was a Consultant for EXTESSY AG, Wolfsburg, Germany, a Post-Doctoral Fellow with the Institute for Systems Engineering and Automation (SEA), Johannes Kepler University Linz, Linz, Austria, and a Post-Doctoral Researcher with the Software and Requirements Engineering Center, DePaul University, Chicago, IL, USA. He is currently a Professor with Technische Universität Ilmenau, Ilmenau, Germany, where he has been heading the Endowed Chair on Software Engineering for Critical Systems. His research interests include machine learning, software engineering, and safety engineering.