Training Adversarially Robust Sparse Networks via Bayesian Connectivity Sampling

Ozan Özdenizci [1 2] **Robert Legenstein** [1]

Abstract

Deep neural networks have been shown to be susceptible to adversarial attacks. This lack of adversarial robustness is even more pronounced when models are compressed in order to meet hardware limitations. Hence, if adversarial robustness is an issue, training of sparsely connected networks necessitates considering *adversarially robust sparse learning*. Motivated by the efficient and stable computational function of the brain in the presence of a highly dynamic synaptic connectivity structure, we propose an intrinsically sparse rewiring approach to train neural networks with state-of-the-art robust learning objectives under high sparsity. Importantly, in contrast to previously proposed pruning techniques, our approach satisfies global connectivity constraints throughout robust optimization, i.e., it does not require dense pre-training followed by pruning. Based on a Bayesian posterior sampling principle, a network rewiring process simultaneously learns the sparse connectivity structure and the robustness-accuracy trade-off based on the adversarial learning objective. Although our networks are sparsely connected throughout the whole training process, our experimental benchmark evaluations show that their performance is superior to recently proposed robustness-aware network pruning methods which start from densely connected networks.

1. Introduction

Despite their widely-acknowledged success and deployment in various application fields, deep neural networks (DNNs) are known to be highly susceptible to intentionally crafted adversarial examples that cause incorrect decision making. Seminal work by (Szegedy et al., 2013) showed that such adversarial examples can be created via perturbations that are hardly perceptible to humans, which exposed important weaknesses of standard deep learning algorithms. Numerous studies explored adversarial defense methods to such threats. Notably successful approaches rely on harnessing adversarial examples during model training (Goodfellow et al., 2015; Madry et al., 2018), and its immediate extensions with robust training losses using regularization schemes to diminish the generalization gap based on an inherent robustness-accuracy trade-off (Tsipras et al., 2019; Zhang et al., 2019; Wang et al., 2020).

Recent work further suggests better robustness with increasing network width and complexity (Madry et al., 2018; Nakkiran, 2019; Wu et al., 2020). Deployment of such large models, however, is challenging in resource-constrained settings. Thus, under consideration of memory and computational demand concerns, this highlights a need to consider achieving model compactness and sparsity simultaneously with adversarial robustness in DNNs.

There has been a growing interest in tackling the problem of achieving robustness against adversarial attacks with very sparsely connected neural networks (cf. Section 2). Success was so far demonstrated by robustness-aware pruning of adversarially trained dense networks (Sehwag et al., 2019; 2020). Importantly these studies only considered naive "end-to-end sparse learning" baseline comparisons with a random and static sparse network initialization. Subsequently, these intrinsically sparse models were found to yield inferior robustness than compressed models obtained with robustness-aware pruning methods. However pruning an adversarially trained DNN does not allow robust training under strict sparsity constraints. To date, no effective method existed for robust end-to-end sparse training to meet such limitations, where the challenge is to enable sparse network connections to rearrange during training such that a well-performing robust and sparse model can be configured.

In this paper we present a method for end-to-end sparse training of neural networks with robust adversarial training objectives. Our approach is motivated by the dynamic synaptic connectivity structure in the brain, which maintains its stable computational function in the presence of an under-

[1]Graz University of Technology, Institute of Theoretical Computer Science, Graz, Austria [2]Silicon Austria Labs, TU Graz - SAL Dependable Embedded Systems Lab, Graz, Austria. Correspondence to: Ozan Özdenizci <ozan.ozdenizci@igi.tugraz.at>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

lying synaptic rewiring process. We consider robust neural network training, in which we allow the network to self-construct its sparse connectivity during training analogous to such a process. We formulate the neural network training problem as estimating a posterior distribution that combines the robust training objective with a sparse connectivity prior on the network parameters, in a Bayes optimal manner. During robust, sparse neural network training we sample the network parameters simultaneously with the sparse connectivity structure from this posterior, hence performing *Bayesian connectivity sampling* during model training with robust learning objectives.

Our work conceptually differs from existing studies at the intersection of model sparsity and adversarial robustness by combining two important concepts that were not yet considered together: (1) enabling online robust training of DNNs that are initialized with very sparse connectivity constraints (i.e., robust end-to-end sparse learning), and (2) its compatibility to state-of-the-art robust training objectives beyond standard adversarial training heuristics.

Contributions of this work are summarized as follows:

- We demonstrate for the first time that a sparsely initialized neural network can be adversarially trained end-to-end for improved robustness under strict connectivity constraints throughout training.
- Our *Bayesian connectivity sampling* approach is agnostic to the robust training objective, and it allows for training the network connectivity structure simultaneously with the robustness-accuracy trade-off imposed by the objective during training.
- We empirically show in benchmark evaluations that our approach yields state-of-the-art performance against robustness-aware pruning methods that are based on robust pre-training of densely connected networks.

2. Preliminaries and Related Work

2.1. Adversarial Robustness

Over the past decade, a wide range of defense methods against adversarial threats have been proposed. Important insights by (Athalye et al., 2018), however, revealed emerging weaknesses in several recent defense proposals and highlighted the need for thorough robustness evaluations (Carlini et al., 2019). To date, defense methods that rely on harnessing adversarial examples during training have found to be the most effective (Athalye et al., 2018).

Notation: Given a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, we consider neural networks f with learnable parameters $\boldsymbol{\theta}$ of the form $f_{\boldsymbol{\theta}}(x) = \arg\max_{y \in \mathcal{Y}} p(y|x, \boldsymbol{\theta})$. Here the standard maximum likelihood learning rule corresponds to: $\arg\min_{\boldsymbol{\theta}} \mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathcal{L}_{\text{natural}}(\boldsymbol{\theta}, x, y)]$, with the loss function $\mathcal{L}_{\text{natural}} = -\log p(y|x, \boldsymbol{\theta})$.

Adversarial training (AT) relies on injecting adversarial examples to the training set at every step of the optimization process in order to robustify the learned decision boundary (Goodfellow et al., 2015; Madry et al., 2018). The learning problem in *robust training objectives* is defined as:

$$\arg\min_{\boldsymbol{\theta}} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\tilde{x} \in \mathcal{B}_\epsilon^p(x)} \mathcal{L}_{\text{robust}}(\boldsymbol{\theta}, \tilde{x}, y) \right], \quad (1)$$

with the inner maximization obtaining adversarial examples in $\mathcal{B}_\epsilon^p(x) := \{\tilde{x} : \|\tilde{x} - x\|_p \leq \epsilon\}$, defined as the l_p-norm ball around samples x with a perturbation strength of $\epsilon > 0$.

Standard AT by (Madry et al., 2018) iteratively generates adversarial examples via projected gradient descent (PGD)[1] using cross-entropy loss, and replaces the training mini-batch with adversarial samples at each iteration, i.e., $\mathcal{L}_{\text{AT}} = -\log p(y|\tilde{x}, \boldsymbol{\theta})$. While yielding high robustness, accuracies on clean test samples for such adversarially trained models degrade. To achieve better test performance (Goodfellow et al., 2015) and (Kurakin et al., 2017) proposed mixed-batch AT by using a mixture of benign and adversarial samples per iteration. Subsequently, several robust training objectives were introduced to address this robustness-accuracy trade-off (Tsipras et al., 2019).

State-of-the-art robust learning objectives mainly rely on the TRADES loss proposed by (Zhang et al., 2019) with $\mathcal{L}_{\text{robust}}$ in Eq. (1) defined via the regularized loss:

$$\mathcal{L}_{\text{TRADES}} = \mathcal{L}_{\text{natural}} + \beta \cdot D_{\text{KL}}(p(y|x, \boldsymbol{\theta}) || p(y|\tilde{x}, \boldsymbol{\theta})), \quad (2)$$

where β is the robustness-accuracy tradeoff parameter and $\tilde{x}$ is obtained by PGD using $D_{\text{KL}}(p(y|x, \boldsymbol{\theta}) || p(y|\tilde{x}, \boldsymbol{\theta}))$ as the loss to be maximized in the inner optimization step.

Later (Wang et al., 2020) studied the TRADES loss within a misclassification aware robust training (MART) scheme. MART explicitly differentiates misclassified examples during training, emphasizes learning with the misclassified samples through the KL-regularizer, and uses a boosted cross-entropy loss on the adversarial samples with an improved decision margin. A widely acknowledged semi-supervised learning method to improve robustness of DNNs has recently been introduced by (Carmon et al., 2019). The proposed robust self-training (RST) scheme allows training with the heuristic TRADES loss using mini-batches that contain pseudo-labeled additional data samples. Importantly, RST revealed that robustness benefits come with additional data information rather than precise labels.

[1] We focus on l_∞-norm adversarial robustness in this paper. For $\tilde{x} \in \mathcal{B}_\epsilon^\infty(x)$ this yields the well-known k-step l_∞-PGD examples: $\tilde{x}_{k+1} = \Pi_{\mathcal{B}_\epsilon^\infty(x)}(\tilde{x}_k + \epsilon_k \cdot \text{sign}(\nabla_{\tilde{x}_k} \mathcal{L}(\boldsymbol{\theta}, \tilde{x}_k, y)))$ with $\tilde{x}_0 = x$ and $\Pi_{\mathcal{B}_\epsilon^\infty(x)}(.)$ is the projection/clipping onto the ϵ-ball around x.

Our work embraces these state-of-the-art robust learning objectives within a sparse network training framework, and incorporates the robustness-accuracy trade-offs imposed by the adversarial training objective to the learning process.

2.2. Sparsity in Deep Neural Networks

Learning efficient, sparse neural networks have generally been of significant interest under the consideration that such models would be deployed to hardware with memory and/or computation capability limitations. Most of these compression methods either operate on a dense DNN after it was trained without global connectivity constraints (Han et al., 2015; Zhu & Gupta, 2017), or by dense training with sparsity-inducing regularizers (Collins & Kohli, 2014; Louizos et al., 2018). Such model compression methods are mainly designed for natural DNN training (i.e., to minimize cross-entropy loss on training samples), and explore weight pruning methods to obtain what (Frankle & Carbin, 2019) proposed as "winning lottery tickets" (i.e., efficient sub-networks within DNNs that can reach performances similar to the original network when trained in isolation). A long-standing standard to achieve sparsity has been the iterative train-prune-retrain approach from (Han et al., 2015), where the weights with least magnitude are pruned.

More closely related to our work, recently emerging methods to train compressed neural networks with a sparse connectivity initialization provide on-hardware learning capabilities for DNNs. Along this line, (Mocanu et al., 2018) proposed *sparse evolutionary training* to train a sparsely initialized DNN by dynamically changing the connectivity with a magnitude-based pruning and random growth strategy. A similar approach proposed by (Bellec et al., 2018) is *deep rewiring*, which trains sparse neural networks with stochastic parameter updates, by sampling the sparse connectivity pattern based on a posterior which is theoretically shown to converge to a stationary distribution. Subsequently (Mostafa & Wang, 2019) proposed *dynamic sparse reparameterization* to train sparse neural networks via adaptive threshold based pruning, and (Dettmers & Zettlemoyer, 2019) introduced a sparse learning method based on re-growing pruned weights according to their momentum. These works did not consider robust training objectives.

2.3. Adversarial Robustness and Sparsity

Recent work grew interest in exploring adversarial robustness in the context of sparse neural networks. In an earlier study (Guo et al., 2018) theoretically demonstrated that adversarial robustness can be improved with an appropriately higher sparsity within a neural network, whereas this robustness tend to be negatively impacted with very large sparsity in a DNN. (Wang et al., 2018) similarly concluded that adversarial robustness decreases with high model sparsity. Subsequently, several studies aimed to tackle the question on learning compressed neural networks while preserving their robustness. Following the pioneering defense view on harnessing adversarial samples during training by (Madry et al., 2018), (Ye et al., 2019) proposed a concurrent adversarial training and weight pruning scheme by approximately solving a constrained optimization problem with alternating direction method of multipliers (ADMM). Their Adv-ADMM optimization approach is in some ways similar to the work by (Gui et al., 2019), where adversarial training and pruning is also combined with factorization and quantization. (Rakin et al., 2019) proposed l_1-regularized adversarial training and subsequent weight pruning. However their approach does not generalize to very sparse models or target sparsity constraints. From an adversarial defense perspective (Madaan et al., 2020) proposed to suppress the latent feature level vulnerability in DNNs by gradually pruning the network during training based on a regularized loss function. While improving robustness, their structured pruning approach does not yield very sparse models. Recently (Kundu et al., 2020) introduced a dynamic pruning method within adversarial training by exploiting the momentum based weight re-growing approach from (Dettmers & Zettlemoyer, 2019). Similarly their approach did not impose global connectivity constraints but gradually shrink the network size, and was also not compatible with recent robust training objectives besides standard adversarial training.

On another line of work, (Sehwag et al., 2019) proposed to prune and fine-tune pre-robustly-trained networks using the heuristic least weight magnitude criterion. Later outperforming this approach (Sehwag et al., 2020) introduced *HYDRA*, where a robustness-aware importance score determines which weights to be pruned from a robustly pre-trained network. HYDRA exploits the importance score learning approach by (Ramanujan et al., 2020) to discover efficient sub-networks, with a robustness criterion. Importantly their approach demonstrates compatibility to various robust training objectives, which constitutes the current state-of-the-art method in robustness-aware pruning of neural networks towards model compression. While their approach is currently the only very high model compression method that offers such a compatibility, necessitating a pre-trained dense robust network and the fine-tuning step introduces an additional computational overload.

Recently successful methods in this context (Ye et al., 2019; Sehwag et al., 2019; 2020) evaluated robust pruning in comparison to a naive "end-to-end sparse learning" baseline using a random, static and sparse initialization. Our work proposes a new state-of-the-art in that sense, and differs from existing studies on training adversarially robust sparse networks by (1) enabling online robust training of DNNs that are initialized with very sparse connectivity constraints (i.e., robust end-to-end sparse learning), and (2) its compat-

ibility to state-of-the-art robust training objectives beyond standard adversarial training heuristics.

3. Training Adversarially Robust Networks with Sparse Connectivity

3.1. Design Motivation

The brain has a highly dynamic synaptic connectivity structure whilst maintaining a stable and efficient computational function (Holtmaat et al., 2005; Stettler et al., 2006). This underlying synaptic rewiring process is also shown to serve an important role in learning (Peters et al., 2014). We consider sparse neural network training analogous to such biological learning processes and combine it with state-of-the-art robust training objectives. Unlike traditional DNNs with a pre-defined static connectivity, i.e., a temporal snapshot of the rewiring process, we train robust sparse neural networks by sampling its connectivity from a learned posterior. Our approach allows the network to self-construct its connectivity in a Bayes optimal manner, such that a well-performing sparse network can be configured while the network gains its robustness via the adversarial training objective.

3.2. Bayesian Connectivity Sampling

The standard supervised learning goal in neural networks is generally defined by finding the parameters that maximize the likelihood $p(y|x, \boldsymbol{\theta})$, via a negative log-likelihood loss. For robust end-to-end sparse network training under a strict connectivity constraint we can state the problem as:

$$\min_{\boldsymbol{\theta}} \mathbb{E}\left[\max_{\tilde{x}\in\mathcal{B}_\epsilon^p(x)} \mathcal{L}_{\text{robust}}(\boldsymbol{\theta}, \tilde{x}, y)\right] \quad s.t. \quad \|\boldsymbol{\theta}\|_0 = \kappa, \quad (3)$$

where κ defines the sparsity constraint on the network parameters[2], and $\mathbb{E}[.]$ is stochastically estimated over training batches. We approach this problem from a Bayesian perspective by incorporating parameter sparsity as a prior belief $p(\boldsymbol{\theta})$ for network parameters $\boldsymbol{\theta}$, and explore the robust training objective via the posterior: $p(\boldsymbol{\theta}\,|x, y) \propto p(\boldsymbol{\theta}) \cdot p(y|x, \boldsymbol{\theta})$. More formally, we optimize the neural networks with a *negative log-posterior* loss which combines: (1) a robust training loss function $\mathcal{L}_{\text{robust}}$ that is dependent on the learned likelihood distribution $p(y|x, \boldsymbol{\theta})$, and (2) a prior distribution $p(\boldsymbol{\theta})$ for parameters that imposes a sparse connectivity constraint. At each iteration of robust training, parameter values as well as its connectivity with regards to the constraint is updated within a stochastic optimization scheme. This scheme ensures that during training we are sampling from the posterior $p(\boldsymbol{\theta}\,|x, y)$ which assigns high probability to robust and sparse connectivity patterns, hence the name *Bayesian connectivity sampling*.

[2]We define sparsity in terms of the number of zero weights in learned weight kernels of dense and convolutional layers.

Similar Bayesian posterior sampling approaches were previously explored in the context of standard neural network training (Welling & Teh, 2011; Kappel et al., 2015; Chen et al., 2016; Bellec et al., 2018). It was shown by (Welling & Teh, 2011) and (Sato & Nakagawa, 2014) that injecting noise into model parameters with an annealed step size during stochastic mini-batch optimization allows the trajectory of sampled parameters converge to sampling from their posterior distribution, known as the *stochastic gradient Langevin sampling dynamics*. Accordingly we combine gradient descent with stochastic parameter updates that enables sampling sparse connectivity patterns with a lower $\mathcal{L}_{\text{robust}}$ from the posterior $p(\boldsymbol{\theta}\,|x, y) \propto p(\boldsymbol{\theta}) \cdot p(y|x, \boldsymbol{\theta})$, by moving from low to high probability regions on the posterior loss landscape throughout optimization. We perform the parameter updates in Eq. (4) while $\|\boldsymbol{\theta}\|_0 = \kappa$ is satisfied:

$$\Delta\,\boldsymbol{\theta}_k = \eta_t\Big(\nabla\Omega(\boldsymbol{\theta}_k) + \nabla\,\mathbb{E}\left[\,\mathcal{L}_{\text{robust}}(\boldsymbol{\theta}_k, \tilde{x}, y)\right]\Big) + \zeta_t, \quad (4)$$

where η_t is the learning step size at iteration t, $\zeta_t \sim \mathcal{N}(0, \sigma\eta_t)$ is Gaussian noise, and σ is a constant scaling factor. In the context of the Bayesian perspective discussed above, $\nabla\Omega(.)$ is the gradient of the log-prior $\log p(\boldsymbol{\theta})$ (e.g., an additional l_2-regularization constraint on the parameter values) and $\nabla\,\mathcal{L}_{\text{robust}}$ is the gradient of the data log-likelihood $\log p(y|x, \boldsymbol{\theta})$. Note that independently one can incorporate robust regularization terms within $\mathcal{L}_{\text{robust}}$ (e.g., the D_{KL} term in Eq. (2) for TRADES loss).

Incorporating the Sparse Connectivity Prior: We impose sparsity as a prior belief for the network parameters by using a reparametrization trick. We perform a simple mapping between each learnable network parameter $\boldsymbol{\theta}_k$ and weight matrices/kernels of dense and convolutional layers that are subject to the constraint. During forward and backward passes our network uses the weights:

$$\boldsymbol{w}_k = \gamma_k \cdot \max\{0, \boldsymbol{\theta}_k\} \quad s.t. \quad \gamma_k \in \{-1, 1\} \quad (5)$$

to optimize parameters $\boldsymbol{\theta}_k$, where each sign γ_k is uniformly sampled once the corresponding connection is introduced. If a parameter $\boldsymbol{\theta}_k$ obtains a negative value at any iteration, since this connectivity will fade, we uniformly sample another connection $\boldsymbol{\theta}_j$ and *activate* it with a value of 10^{-12}. This naturally yields a number of parameters to be dynamically rewired at each training step, and results in a different sparse connectivity structure during each iteration. Since we initialize the parameters with a random sparse connectivity that satisfies the constraint, this reparametrization and rewiring scheme ensures $\|\boldsymbol{\theta}\|_0 = \kappa$ throughout training.

Relation to Synaptic Sampling: How the posterior distribution of weights can be learned was also addressed in spiking neural networks to understand brain plasticity (Kappel et al., 2015; 2018). The presented *synaptic sampling* framework defines stochastic plasticity rules for network

parameters to achieve a Bayesian posterior learning goal from observations x via the parameter dynamics:

$$d\,\boldsymbol{\theta}_k = \eta \left[\frac{\partial}{\partial\,\boldsymbol{\theta}_k} \log p(\boldsymbol{\theta}\,|x) \right] dt + \sqrt{2\eta T}\, d\mathcal{W}_k, \qquad (6)$$

where this continuous time differential equation considers an additive stochastic Wiener process term $\mathcal{W}_k$ with a learning rate scale η. For $T = 1$ one recovers a unique stationary distribution for the posterior (Kappel et al., 2015). Discrete time approximation replaces $\mathcal{W}_k$ by Gaussian noise, resembling to posterior sampling through Langevin dynamics. This framework was later extended to supervised learning with hard posterior constraints by (Bellec et al., 2018).

Our training procedure is outlined in Algorithm 1. One important cornerstone is also the use of the regularization scheme to push weights towards zero in order to promote sampling of novel connectivity patterns, especially at early stages of training. We realize this via decoupled weight decay regularization (Loshchilov & Hutter, 2019) since it offers (1) stronger regularization of variables than l_2-penalty alone which yields better generalization (also differently than a standard sparsity-promoting l_1-regularization (Bellec et al., 2018; Rakin et al., 2019), see Appendix B.2 for comparisons), and (2) compatibility for our algorithm to have a similar training pipeline that one uses for its fully connected counterpart with robust training objectives, since the state-of-the-art in image classification considers DNNs trained with momentum SGD and weight decay. We scale the decoupled weight decay in sync with the learning rate and additive noise, hence guiding the sampling process with standard optimization hyper-parameter schedulers. After training, the learned network structure is fixed and used for testing without stochasticity. Hence these models do not relate to possible gradient obfuscation related robustness fallbacks (Athalye et al., 2018).

Algorithm 1 Robust end-to-end sparse training

Input: Dataset $\mathcal{D}$, neural network f, parameters $\boldsymbol{\theta}$, iterations T, batch size τ, $\mathcal{L}_{\text{robust}}$ and $\mathcal{L}_{\text{PGD}}$, PGD iterations K, perturbation strength ϵ and step size α, sparsity constraint κ, noise scaling factor σ, learning rate λ_l, weight decay factor λ_w, hyper-parameter scheduler $\phi(.)$
Initialize: $\boldsymbol{\theta}$ with $\boldsymbol{\theta}_k \geq 0$ such that $\|\boldsymbol{\theta}\|_0 = \kappa$
Initialize: Sample $\gamma_k \in \{-1, 1\}$ uniformly $\forall\, \boldsymbol{\theta}_k > 0$
for $t = 1$ **to** T **do**
  Sample a mini-batch $\{(x_i, y_i)\}_{i=1}^{\tau} \subset \mathcal{D}$
  $\lambda_{l_t}, \lambda_{w_t} \leftarrow \phi(\lambda_l, t), \phi(\lambda_w, t)$
  Compute weights $\boldsymbol{w}_k = \gamma_k\, \boldsymbol{\theta}_k,\ \forall\, \boldsymbol{\theta}_k \neq 0$
  for $i = 1$ **to** τ **do**
    for $k = 1$ **to** K **do**
      Compute $\mathcal{L}_{\text{PGD}}$ via $f_{\boldsymbol{w}}(\tilde{x}_i^k)$ and y_i
      $\tilde{x}_i^{k+1} \leftarrow \Pi_\epsilon\big(\tilde{x}_i^k + \alpha \cdot \text{sign}(\nabla_{\tilde{x}_i^k}\, \mathcal{L}_{\text{PGD}})\big)$
    end for
  end for
  Compute $\mathbb{E}[\mathcal{L}_{\text{robust}}]$ via $\{f_{\boldsymbol{w}}(\tilde{x}_i)\}_{i=1}^{\tau}$ and $\{y_i\}_{i=1}^{\tau}$
  for all $\boldsymbol{\theta}_k \neq 0$ **do**
    Sample noise $\zeta_t \leftarrow \mathcal{N}(0, \sigma\lambda_{l_t})$
    $\boldsymbol{\theta}_k \leftarrow \boldsymbol{\theta}_k - \lambda_{l_t}\big(\nabla_{\boldsymbol{\theta}_k}\, \mathbb{E}[\mathcal{L}_{\text{robust}}]\big) - \lambda_{w_t}\, \boldsymbol{\theta}_k + \zeta_t$
    $\boldsymbol{\theta}_k \leftarrow \max\{0, \boldsymbol{\theta}_k\}$
  end for
  while $\|\boldsymbol{\theta}\|_0 < \kappa$ **do**
    Uniformly sample a parameter index j
    if $\boldsymbol{\theta}_j = 0$ **then**
      $\boldsymbol{\theta}_j \leftarrow 10^{-12}$
      Sample $\gamma_j \in \{-1, 1\}$ unless assigned before
    end if
  end while
end for

4. Experiments

We compare our approach with sparse learning baselines in Section 4.1, and state-of-the-art robustness-aware pruning methods in Section 4.2. Dataset and model specifications, as well as training and evaluation details are described below.

Datasets & Model Architectures: We perform experiments with three benchmark datasets: CIFAR-10 and CIFAR-100 (Krizhevsky, 2009), and SVHN (Netzer et al., 2011) (see Appendix A.1 for further details). In our main evaluations we used VGG-16 (Simonyan & Zisserman, 2015), ResNet-18 (He et al., 2016), and Wide-ResNet-28-4 (Zagoruyko & Komodakis, 2016) architectures. In what follows, we also report additional evaluations with variants of ResNet and Wide-ResNet models.

Robust Training Settings: We demonstrate the compatibility of our method with the following state-of-the-art robust training objectives: Standard AT, Mixed-batch AT, TRADES, MART, RST. We chose the trade-off parameter as $\beta = 6$ for TRADES loss, and 4 for MART. We used half-benign half-adversarial batches for mixed-batch AT. We implemented RST using the same pseudo-labeled 500K TinyImages dataset shared by the authors[3]. To craft adversarial examples at each mini-batch during training (i.e., inner maximization step of the objective) we used 10 PGD steps with random starts, a maximum perturbation budget of $\epsilon = 8/255$, and a perturbation step size of $2/255$ as suggested by (Madry et al., 2018). For TRADES and RST, the inner maximization PGD was performed on $\mathcal{L}_{\text{PGD}} = D_{\text{KL}}(p(y|x, \boldsymbol{\theta})||p(y|\tilde{x}, \boldsymbol{\theta}))$, whereas other methods perform PGD during training on a cross-entropy loss.

White box Threat Methods: We follow the conventional settings for l_∞-norm bounded white box robustness evaluations. Perturbation budget for all datasets and adversarial attacks is $\epsilon = 8/255$. For the baseline comparisons in

[3] https://github.com/yaircarmon/semisup-adv

Table 1. Evaluations of sparse networks learned with various robust training objectives on CIFAR-10 with VGG-16. Standard VGG-16 indicates the models that are trained with full connectivity (0% sparsity). All other networks are trained with a sparse connectivity from scratch. Evaluations are presented as clean/robust accuracy (%). Robust accuracy is evaluated via PGD^{50} with 10 restarts ($\epsilon = 8/255$).

	Standard VGG-16	90% Sparsity			99% Sparsity		
		Random	*Fixed*	**Ours**	*Random*	*Fixed*	**Ours**
Natural Training	93.2/0.0	90.4/0.0	90.6/0.0	**91.8/0.0**	56.9/0.0	86.2/0.0	**87.7/0.0**
Standard AT (Madry et al., 2018)	78.4/44.9	73.9/43.3	75.8/42.6	**78.3/44.5**	42.0/27.0	64.6/39.3	**69.8/42.1**
Mixed-batch AT (Kurakin et al., 2017)	84.0/41.1	78.8/33.8	81.3/39.2	**83.0/40.2**	67.3/29.7	72.7/33.9	**77.8/37.6**
TRADES (Zhang et al., 2019)	80.0/46.1	75.5/43.1	76.0/44.3	**78.2/45.7**	49.1/30.8	68.6/38.2	**72.4/41.7**
MART (Wang et al., 2020)	75.3/46.8	72.8/42.2	73.4/44.3	**76.0/45.2**	48.0/34.7	63.9/42.4	**68.2/45.4**
RST (Carmon et al., 2019)	83.1/52.1	77.0/46.0	78.1/46.8	**80.9/49.6**	54.4/32.2	69.9/38.5	**74.0/42.3**

Section 4.1 we evaluate the models against PGD attacks with random starts (Madry et al., 2018) using 50 iterations (PGD^{50}) which is stronger than the PGD that was used to craft adversarial examples during training (see Appendix B.3 for further comparisons). PGD attacks were performed with 10 restarts, i.e., we run full PGD with 50 steps initialized from 10 different random starts and consider an attack unsuccessful if the model was not fooled by any of these 10 attacks. For PGD attacks we determined the step size by the $2.5 * \epsilon / \#steps$ rule of thumb (Madry et al., 2018). In Section 4.2 we evaluate the models with a wider range of attacks: fast gradient sign method (FGSM, a single gradient step attack), PGD^{50}, PGD with 100 iterations and 20 restarts (PGD^{100}), and the Brendel & Bethge attack ($B\&B_\infty$) (Brendel et al., 2019) with 100 steps for VGG-16 and 500 steps for WideResNet-28-4 models. All adversarial attacks were evaluated using the implementations from the Foolbox Native benchmark (Rauber et al., 2020). We also present robust accuracies under an ensemble of four l_∞-norm bounded perturbation attacks via the AutoAttack benchmark (AA_∞) (Croce & Hein, 2020).

Black box Threat Methods: White box adversarial threats constitute a strong robustness evaluation method, but are generally unrealistic when the model parameters are unknown to the attacker. Hence we also consider black box threats where the attacker has no knowledge of the architecture or parameters, but only has access to send a limited number of queries to the model. We use Square Attack for these evaluations which was recently shown as a powerful query-based black box threat (Andriushchenko et al., 2020).

Implementations: Optimization for all models was performed using SGD with momentum and decoupled weight decay (Loshchilov & Hutter, 2019). All models were trained for 200 epochs with a batch size of 128. Only for models trained with RST the batch size was set to 256, while keeping the total number of iterations the same. We used piecewise constant decay learning rate and weight decay schedulers. Initial learning rates were set to 0.1 and were divided by 10 at 100th and 150th epochs. Network weights were initialized via Kaiming initialization (He et al., 2015). Details on optimization hyper-parameter specifications and sparse connectivity initialization schemes are provided in Appendix A.2 and A.3. Our code is available at: https://github.com/IGITUGraz/SparseAdversarialTraining.

Table 2. Clean/robust accuracy (%) evaluations with ResNet-18 on CIFAR-100 and WideResNet-28-4 on SVHN. Standard methods indicate models trained with full connectivity. Robust accuracy is evaluated via PGD^{50} with 10 restarts ($\epsilon = 8/255$).

	Sparsity	& Method	Natural Training	Mixed-batch AT	TRADES
CIFAR-100	0%	Standard	74.2/0.0	60.5/22.1	56.0/27.1
	90%	*Fixed*	70.2/0.0	59.4/22.1	53.9/26.3
		Ours	**71.1**/0.0	**61.8/23.4**	**55.2/27.2**
	99%	*Fixed*	58.9/0.0	45.1/17.6	43.5/19.3
		Ours	**61.5**/0.0	**53.1/20.2**	**47.7/22.2**
SVHN	0%	Standard	96.5/0.0	97.1/47.7	92.5/56.6
	90%	*Fixed*	96.2/0.0	97.0/51.2	89.2/**55.8**
		Ours	**96.4**/0.0	97.0/**51.5**	**92.8**/55.6
	99%	*Fixed*	95.2/0.0	91.9/**44.5**	87.4/48.3
		Ours	**95.7**/0.0	**95.7**/43.1	**89.5/52.7**

4.1. Bayesian Connectivity Sampling Enables Sparse Learning with Robust Training Objectives

We initially evaluate our method with baseline comparisons in the light of recent work in this domain. To date, *sparse and robust training from scratch* was (yet) only represented and evaluated via DNNs trained with a static sparse connectivity initialization. Accordingly we compare our approach with the following baselines to train a sparse model with different robust training objectives:

Random: We train a sparse network from scratch with a randomly initialized connectivity that is kept static during optimization. In this case, an equal fraction of connections

Table 3. Comparisons with the current state-of-the-art robustness-aware pruning method HYDRA. CIFAR-10 evaluations are performed based on RST, and SVHN evaluations are based on TRADES adversarial training for consistency with the original evaluation checkpoints by (Sehwag et al., 2020). All attacks are evaluated via Foolbox and AutoAttack (AA) for l_∞-perturbations with $\epsilon = 8/255$.

		VGG-16						WideResNet-28-4					
		90% Sparsity			99% Sparsity			90% Sparsity			99% Sparsity		
		HYDRA	**Ours**	Δ	HYDRA	**Ours**	Δ	HYDRA	**Ours**	Δ	HYDRA	**Ours**	Δ
CIFAR-10	Clean	80.5	80.9	**+0.4**	73.2	74.0	**+0.8**	83.7	84.8	**+1.1**	75.6	76.9	**+1.3**
	FGSM	55.6	55.3	-0.3	46.5	46.5	0.0	61.1	60.0	-1.1	51.0	49.5	-1.5
	PGD^{50}	50.0	49.6	-0.4	41.9	42.3	**+0.4**	55.6	54.0	-1.6	47.4	45.1	-2.3
	PGD^{100}	49.9	49.5	-0.4	41.8	42.1	**+0.3**	55.5	53.9	-1.6	47.3	44.9	-2.4
	$B\&B_\infty$	48.1	47.7	-0.4	39.1	40.0	**+0.9**	53.8	52.2	-1.6	45.2	42.9	-2.3
	AA_∞	45.46	44.98	-0.48	37.18	37.45	**+0.27**	51.74	49.78	-1.96	42.80	40.18	-2.62
SVHN	Clean	89.2	89.4	**+0.2**	84.4	86.4	**+2.0**	94.4	92.8	-1.6	88.9	89.5	**+0.6**
	FGSM	63.1	64.5	**+1.4**	57.1	58.4	**+1.3**	88.8	70.0	-18.8	74.3	63.1	-11.2
	PGD^{50}	52.8	53.7	**+0.9**	47.8	48.7	**+0.9**	43.9	55.6	**+11.7**	39.1	52.7	**+13.6**
	PGD^{100}	52.4	53.3	**+0.9**	47.5	48.3	**+0.8**	38.3	55.1	**+16.8**	36.5	52.4	**+15.9**
	$B\&B_\infty$	48.9	49.8	**+0.9**	43.7	45.0	**+1.3**	36.5	52.1	**+15.6**	32.3	49.9	**+17.6**
	AA_∞	45.51	44.88	-0.63	38.80	40.78	**+1.98**	30.60	47.00	**+16.40**	26.66	45.78	**+19.12**

was randomly discarded at each layer during initialization, i.e., global sparsity was equal to layer-wise sparsities, similar to the baselines in (Sehwag et al., 2019; 2020).

Fixed: We train a sparse network from scratch with a fixed connectivity where the number of connections at each layer were chosen equal to the number of connections that our method was found to converge at. In this case, this initialized fixed connectivity is also kept static during optimization.

Table 1 and Table 2 depicts our evaluations for different models with 90% and 99% sparsity constraints. Results highlight that Bayesian connectivity sampling enables learning with robust training objectives under sparse connectivity from scratch. Furthermore, our method scales to various datasets (see Appendix B.4 for additional experiments), network architectures, robust training objectives and regularization schemes in a similar manner. Our approach with 90% sparse models yields clean and robust accuracies similar to their fully-connected counterparts, simultaneously. Particularly at a very high 99% sparsity the naive *Random* approach fails to benefit from adversarial training, which is consistent with the recent evaluations (Sehwag et al., 2019; 2020). This points to the importance of the dynamic rewiring aspect of our approach, to let the network sample its own robust connectivity within the learning process. Accordingly with *Fixed*, learning can be performed to an even larger extent than naively considered. Note that since the *Fixed* method which uses learned per-layer connectivity distributions was found to perform better than *Random*, we did not include experiments with *Random* connectivity in Table 2. Overall these results highlight the functional importance of Bayesian connectivity sampling for sparse networks during robust training.

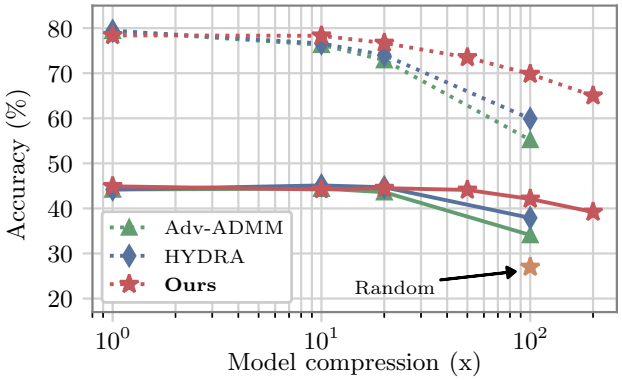
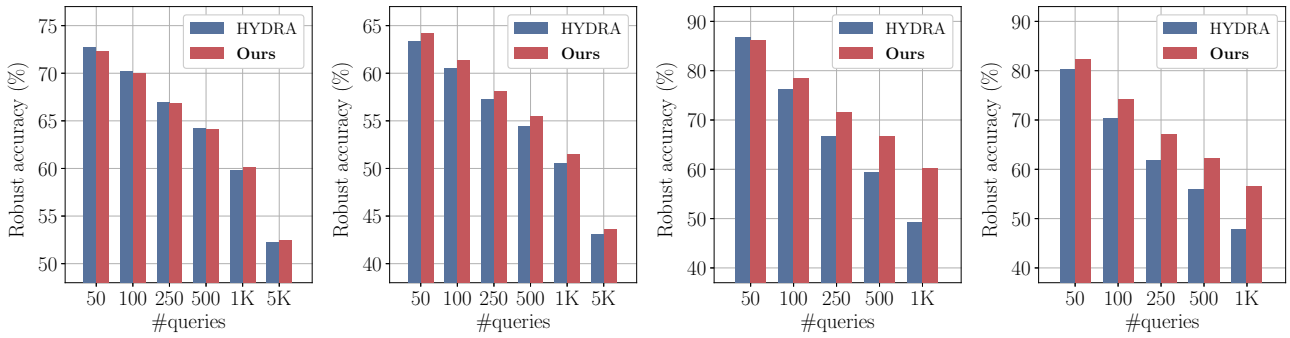


Figure 1. Standard AT on CIFAR-10 with varying VGG-16 model compression. Above dotted lines show clean, and below solid lines show robust accuracies. Orange marker shows the robust accuracy at 1% with a *Random* static sparse initialization. Accuracies at 100x model compression are indicated in the first row of Table 4.

4.2. Comparisons with Robustness-Aware Pruning

We mainly compare our approach with the current state-of-the-art robustness-aware neural network pruning method HYDRA. In order to present fair and identical comparisons relying on the same methods used by the authors, we particularly report comparisons for the VGG-16 and WideResNet-28-4 models, on CIFAR-10 with RST and on SVHN with TRADES as indicated in (Sehwag et al., 2020). We evaluate all models in comparison to our approach using the exact model specifications and provided model checkpoints[4]. Accordingly with (Sehwag et al., 2020) we also report the best test robustness and benign accuracy that was achieved

[4] https://github.com/inspire-group/hydra



(a) VGG-16 (90%) (b) VGG-16 (99%) (c) WideResNet-28-4 (90%) (d) WideResNet-28-4 (99%)

Figure 2. Black box Square Attack (Andriushchenko et al., 2020) evaluations with limited queries. Maximum l_∞-norm perturbation strength is $\epsilon = 8/255$. Sub-labels indicate the architecture (sparsity) for each evaluation. VGG-16 models are trained with RST on CIFAR-10, and WideResNet-28-4 models trained with TRADES loss on SVHN, which are also the models from Table 3.

across checkpoints. In Table 3 we validate their clean test set accuracies that was reported, and evaluate robustness of all models under several white box attacks, as well as the AutoAttack ensemble benchmark. Overall, our method achieves comparable benign and robust test accuracies even though HYDRA needs robust pre-trained dense networks in comparison to our robust end-to-end sparse training approach. While for the WideResNet-28-4 on CIFAR-10, HYDRA performs somewhat better, robust Bayesian connectivity sampling performs comparable or slightly better with VGG-16 on both datasets, and considerably better for WideResNet-28-4 models on SVHN.

Earlier work by (Ye et al., 2019; Gui et al., 2019) on combining model compression with adversarial robustness focused on standard AT by (Madry et al., 2018). Since these methods are not compatible with more recent robust learning approaches (Zhang et al., 2019; Wang et al., 2020), so far we only considered HYDRA as the robustness-aware pruning approach that is most related to our work. However for completeness we also performed evaluations with standard AT, presented in Figure 1 in comparison to the results that were previously obtained with Adv-ADMM (Ye et al., 2019) and HYDRA for varying compression levels on VGG-16 with CIFAR-10. To train and evaluate our models we use the same AT configuration, as well as the same robustness evaluations reported in their work. Figure 1 depicts clean and robust accuracy differences in favor of our approach, extending towards 200x compression (99.5% sparsity; note that here we rely on the accuracies reported in (Sehwag et al., 2020) which did not include 99.5% sparsity for standard AT). Going further, Table 4 presents similar results at 99% sparsity for different residual network architectures. Results show compatibility of our method with standard AT to better explore sparse and robust network configurations, yielding superior results simultaneously across all architectures under high compression.

Table 4. CIFAR-10 evaluations at 99% sparsity for standard AT. Clean/robust accuracies (%) for Adv-ADMM and HYDRA are retrieved from (Sehwag et al., 2020), and we used the same training and robustness evaluations with their reported configurations.

	Adv-ADMM	HYDRA	**Ours**
VGG-16	55.2/34.1	59.9/37.9	**69.8/42.1**
ResNet-18	58.7/36.1	69.0/41.6	**72.1/44.8**
ResNet-34	68.8/41.5	71.8/44.4	**73.3/44.7**
ResNet-50	69.1/42.2	73.9/45.3	**75.0/46.9**
WideResNet-28-2	48.3/30.9	54.2/34.1	**60.2/38.6**

4.3. Evaluating Black Box Robustness

Figure 2 represents our black box evaluations for the trained sparse networks in comparison to adversarially pruned networks with HYDRA. We show robust accuracies under attacks with different query access limits ranging from 50 to 5000, for VGG-16 networks trained on CIFAR-10 with RST and a larger network WideResNet-28-4 trained on SVHN with TRADES loss. We observe in Figures 2(b) and 2(d) that black box robustness gap between these methods increased particularly at 99% sparsity, especially for the WideResNet-28-4 model. One fundamental difference between these two methodologies is also that our approach enables training these neural networks on-hardware under strict sparsity constraints, whereas HYDRA requires robust pre-training of a densely connected neural network to be pruned and fine-tuned (cf. Appendix B.4 for time cost discussions).

5. Conclusion

We propose a *Bayesian connectivity sampling* approach for robust end-to-end sparse training of neural networks. Our method simultaneously explores robust connectivity patterns and network weights via rewiring during training with

state-of-the-art robust training objectives. While the network gains robustness via the adversarial training objective, its combination with our brain-inspired rewiring approach enables for the first time sparsity and robustness to be simultaneously achieved by end-to-end training.

Our approach provides an efficient end-to-end robust and sparse training procedure. In practical deployment scenarios, models may need to be fine-tuned on customized datasets or require online updating, e.g., on specialized low-power hardware. In this context the pre-trained sparse network becomes an initialization that rewiring can proceed from under a robust training objective, where non-active connections also do not add computational burden (i.e., gradient computation is not needed for those parameters).

Robust adversarial training methods demonstrated large empirical success so far, yet it was shown that provable robustness guarantees can be gained by restricting the global Lipschitz constant of neural networks (Hein & Andriushchenko, 2017; Cisse et al., 2017). Our approach remains open to such methods, since one can impose local Lipschitzness to network layers through regularization while learning a posterior by rewiring parameters. In this study we demonstrate that translating neuroscientific evidences can help us construct tools for emerging problems in machine learning, and we argue that brain-inspired computing has a fundamental prospect towards developing powerful methods for contemporary artificial intelligence.

Acknowledgements

We thank all reviewers for their careful evaluations that help to improve the manuscript. This work has been supported by the "University SAL Labs" initiative of Silicon Austria Labs (SAL) and its Austrian partner universities for applied fundamental research for electronic based systems. This work is also partially supported by the Austrian Science Fund (FWF) within the ERA-NET CHIST-ERA programme (project SMALL, project number I 4670-N).

References

Andriushchenko, M., Croce, F., Flammarion, N., and Hein, M. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, pp. 484–501, 2020.

Athalye, A., Carlini, N., and Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, pp. 274–283, 2018.

Bellec, G., Kappel, D., Maass, W., and Legenstein, R. Deep rewiring: Training very sparse deep networks. In *International Conference on Learning Representations (ICLR)*, 2018.

Brendel, W., Rauber, J., Kümmerer, M., Ustyuzhaninov, I., and Bethge, M. Accurate, reliable and fast robustness evaluation. In *Advances in Neural Information Processing Systems*, pp. 12861–12871, 2019.

Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., Goodfellow, I., Madry, A., and Kurakin, A. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

Carmon, Y., Raghunathan, A., Schmidt, L., Duchi, J. C., and Liang, P. S. Unlabeled data improves adversarial robustness. In *Advances in Neural Information Processing Systems*, pp. 11192–11203, 2019.

Chen, C., Carlson, D., Gan, Z., Li, C., and Carin, L. Bridging the gap between stochastic gradient MCMC and stochastic optimization. In *Artificial Intelligence and Statistics*, pp. 1051–1060, 2016.

Cisse, M., Bojanowski, P., Grave, E., Dauphin, Y., and Usunier, N. Parseval networks: improving robustness to adversarial examples. In *International Conference on Machine Learning*, pp. 854–863, 2017.

Collins, M. D. and Kohli, P. Memory bounded deep convolutional networks. *arXiv preprint arXiv:1412.1442*, 2014.

Croce, F. and Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning*, pp. 2206–2216, 2020.

Dettmers, T. and Zettlemoyer, L. Sparse networks from scratch: Faster training without losing performance. *arXiv preprint arXiv:1907.04840*, 2019.

Frankle, J. and Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

Gui, S., Wang, H. N., Yang, H., Yu, C., Wang, Z., and Liu, J. Model compression with adversarial robustness: A unified optimization framework. In *Advances in Neural Information Processing Systems*, pp. 1285–1296, 2019.

Guo, Y., Zhang, C., Zhang, C., and Chen, Y. Sparse DNNs with improved adversarial robustness. In *Advances in Neural Information Processing Systems*, pp. 242–251, 2018.

Han, S., Pool, J., Tran, J., and Dally, W. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, pp. 1135–1143, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1026–1034, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Hein, M. and Andriushchenko, M. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *Advances in Neural Information Processing Systems*, pp. 2266–2276, 2017.

Holtmaat, A. J., Trachtenberg, J. T., Wilbrecht, L., Shepherd, G. M., Zhang, X., Knott, G. W., and Svoboda, K. Transient and persistent dendritic spines in the neocortex in vivo. *Neuron*, 45(2):279–291, 2005.

Kappel, D., Habenschuss, S., Legenstein, R., and Maass, W. Synaptic sampling: a Bayesian approach to neural network plasticity and rewiring. In *Advances in Neural Information Processing Systems*, volume 28, pp. 370–378, 2015.

Kappel, D., Legenstein, R., Habenschuss, S., Hsieh, M., and Maass, W. A dynamic connectome supports the emergence of stable computational function of neural circuits through reward-based learning. *Eneuro*, 5(2), 2018.

Krizhevsky, A. Learning multiple layers of features from tiny images. *Technical Report, University of Toronto*, 2009.

Kundu, S., Nazemi, M., Beerel, P. A., and Pedram, M. A tunable robust pruning framework through dynamic network rewiring of DNNs. *arXiv preprint arXiv:2011.03083*, 2020.

Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial machine learning at scale. In *International Conference on Learning Representations (ICLR)*, 2017.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Louizos, C., Welling, M., and Kingma, D. P. Learning sparse neural networks through l_0 regularization. In *International Conference on Learning Representations (ICLR)*, 2018.

Madaan, D., Shin, J., and Hwang, S. J. Adversarial neural pruning with latent vulnerability suppression. In *International Conference on Machine Learning*, pp. 6575–6585, 2020.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

Mocanu, D. C., Mocanu, E., Stone, P., Nguyen, P. H., Gibescu, M., and Liotta, A. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature Communications*, 9 (1):1–12, 2018.

Mostafa, H. and Wang, X. Parameter efficient training of deep convolutional neural networks by dynamic sparse reparameterization. In *International Conference on Machine Learning*, pp. 4646–4655, 2019.

Nakkiran, P. Adversarial robustness may be at odds with simplicity. *arXiv preprint arXiv:1901.00532*, 2019.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning, 2011.

Peters, A. J., Chen, S. X., and Komiyama, T. Emergence of reproducible spatiotemporal activity during motor learning. *Nature*, 510(7504):263–267, 2014.

Rakin, A. S., He, Z., Yang, L., Wang, Y., Wang, L., and Fan, D. Robust sparse regularization: Simultaneously optimizing neural network robustness and compactness. *arXiv preprint arXiv:1905.13074*, 2019.

Ramanujan, V., Wortsman, M., Kembhavi, A., Farhadi, A., and Rastegari, M. What's hidden in a randomly weighted neural network? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11893–11902, 2020.

Rauber, J., Zimmermann, R., Bethge, M., and Brendel, W. Foolbox native: Fast adversarial attacks to benchmark the robustness of machine learning models in PyTorch, Tensorflow, and JAX. *Journal of Open Source Software*, 5(53):2607, 2020.

Sato, I. and Nakagawa, H. Approximation analysis of stochastic gradient Langevin dynamics by using Fokker-Planck equation and Ito process. In *International Conference on Machine Learning*, pp. 982–990, 2014.

Sehwag, V., Wang, S., Mittal, P., and Jana, S. Towards compact and robust deep neural networks. *arXiv preprint arXiv:1906.06110*, 2019.

Sehwag, V., Wang, S., Mittal, P., and Jana, S. HYDRA: Pruning adversarially robust neural networks. *Advances in Neural Information Processing Systems*, 7, 2020.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

Stettler, D. D., Yamahachi, H., Li, W., Denk, W., and Gilbert, C. D. Axons and synaptic boutons are highly dynamic in adult visual cortex. *Neuron*, 49(6):877–887, 2006.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. Robustness may be at odds with accuracy. In *International Conference on Learning Representations (ICLR)*, 2019.

Wang, L., Ding, G. W., Huang, R., Cao, Y., and Lui, Y. C. Adversarial robustness of pruned neural networks. In *ICLR Workshop Track*, 2018.

Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., and Gu, Q. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations (ICLR)*, 2020.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient Langevin dynamics. In *International Conference on Machine Learning*, pp. 681–688, 2011.

Wu, B., Chen, J., Cai, D., He, X., and Gu, Q. Do wider neural networks really help adversarial robustness? *arXiv preprint arXiv:2010.01279*, 2020.

Ye, S., Xu, K., Liu, S., Cheng, H., Lambrechts, J.-H., Zhang, H., Zhou, A., Ma, K., Wang, Y., and Lin, X. Adversarial robustness vs. model compression, or both? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 111–120, 2019.

Zagoruyko, S. and Komodakis, N. Wide residual networks. In *British Machine Vision Conference*, 2016.

Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., and Jordan, M. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pp. 7472–7482, 2019.

Zhu, M. and Gupta, S. To prune, or not to prune: exploring the efficacy of pruning for model compression. *arXiv preprint arXiv:1710.01878*, 2017.