Dynamic Sparse Adversarial Training for Robust Image Classification

Abstract—**Adversarial training has proven effective against white-box attacks. However, adversarial training is prone to overfitting on the adversarial samples during training. Dynamic sparsity in neural networks has shown to effectively reduce parameter count and regularize networks while maintaining high accuracy. To mitigate the issue of overfitting in adversarial training, this empirical research explores the interplay between adversarial robustness and dynamic sparsity in image classification. Our results show that dynamic sparse models generally achieve higher performance against adversarial attacks compared to dense counterparts. Our investigations indicate that adversarial attacks achieve misclassification by shifting away the attention of models towards regions less relevant for classifications, and that the attention of dynamic sparse models is more robust against being directed away from parts relevant for classification by adversarial attacks. This research is a contribution to the understanding of the emergent effect of robustness in dynamic sparse neural networks against adversarial attacks.**

I. Introduction

Adversarial attacks undermine the prediction performance of deep neural networks (DNN) [1]. In adversarial attacks, gradient-based perturbations are added to input data to cause miss-classification of the adversarial examples. One of the most prominent defence methods, adversarial training, prepares the model for adversarial attacks by using both unperturbed and perturbed images during training [2]. Unfortunately, adversarial training tends towards overfitting on the attack type during training time [3]–[5].

Artificial neural networks (ANN) typically feature fully connected layers, meaning each neuron of one layer is connected to every other neuron in the next layer. Inspired by neural pruning in the biological brain [6], research has documented that with reduction of parameter count (sparsifying), neural networks are still able to achieve similar performance while decreasing computational requirement [7]–[11]. The training of sparse neural networks, can be subdivided into static and dynamic training.

Sparse Evolutionairy Training (SET) [12], [13] and Rigging the Lottery (RigL) [14] are dynamic sparse training method that actively add and remove connections during training to reduce parameter count and achieve increased regularization. Near zero weights are removed after each cycle and replaced with new connections. The amount of connections is fixed throughout training.

It has been shown that dynamic sparse training methods effectively reduce overfitting while reducing parameter count and maintaining high accuracy [13]. It is however yet to be shown if the regularization effects dynamic sparse training apply to adversarial training. To diminish overfitting in adversarial training, this research explores the effects of dynamic sparsity on adversarial training. Therefore this paper focuses on the following research question:

Does dynamic sparse training produce robustness against adversarial examples?

This paper answers this question affirmatively and shows through its results that:

- Dynamic Sparse models generally achieve similar or higher accuracy against adversarial attacks with less than 50% of the connection-count compared to fully connected models.
- Adversarial attacks shift away the attention of the model towards sections in the image irrelevant towards correct classification.
- The attention of the dynamic sparse models is less shifted away from the relevant parts of the image during adversarial attacks.

With these results, this research contributes to the understanding of the emergence of robustness through dynamic sparse training in artificial neural networks and underlines the importance of sparse neural networks to the implementation of machine learning in the context of limited computing devices, and subject to adversarial attacks. Furthermore this work sheds light on how adversarial attacks and training shift the attention of neural networks in image classification.

Section II **Background**, introduces the research used and leading up to this paper. Section III **Related works** discusses contemporary state-of-the-art research at the intersection of adversarial training and sparsity. Section IV **Methodology** describes model usage, hyperparameter settings and visualisation techniques. Section V **Results** discusses experimental results. Section VI **Discussion**, discusses the results in relation to the research question and Section VII **Conclusion**, concludes the paper.

II. Background

Adversarial attacks prove to be a serious risk to Deep Neural Networks (DNN). Many neural networks are designed in linear ways to facilitate easier optimization. Consequently, they are more susceptible to small perturbations that lead to misclassification of the input image. An adversarial attack might be targeted to fool the network $f : X \rightarrow Y$ into classifying an input $x \in X$ as a specific target label $t \in Y$ such that $f(x + \eta) = t$, where η is the perturbation. An attack might also be untargeted with the only aim to let the network misclassify the input as any other label such that $f(x+\eta) \neq f(x)$. Furthermore, the adversarial example should be indistinguishable from the original input to the human eye and therefore, the perturbations should be as small as possible. Hence there exist a set of allowed perturbations defined by the maximum possible distance of the perturbation $D(x, x_adv) \leq \epsilon$, restricting the distance between an input x and its adversarial counterpart x_{adv} to a manipulation budget ϵ.

Fast Gradient Sign Method (FGSM) is the earliest attack method used for creating adversarial images. The method was first introduced in the paper “Explaining and harnessing adversarial examples” by Goodfellow et al. in 2014 and can be used for both targeted and untargeted attacks [1]. FGSM aims to maximize the value of the networks loss function by adding the sign of the gradient of the loss to the original input image in one step. The following equation creates an untargeted adversarial input with the FGSM method:

$$x^{adv} = x + \epsilon * sign(\nabla_x J(\theta, x, y)) \tag{1}$$

where x is the input to the network, y the correct target associated with x, θ the parameters of the network and J(θ, x, y) the cost function used to train the neural network. The sign of the gradient of the loss function is multiplied with hyperparameter ϵ and added to the original input in one step, to ensure that the adversarial input x_{adv} satisfies the distance requirements with regards to the original input.

Projected Gradient Descent (PGD) is a multi step variant of FGSM [15]. Rather than applying FGSM only one time, it applies it several times. Applying FGSM naively many times would increase the distance between the original input image and the generated adversarial counterpart to be greater than ϵ, possibly making the perturbations visible. Since we are constrained to let the generated adversarial image x^{adv} be in $[x-\epsilon, x+\epsilon]$ range, it is necessary to bound the possible set of generated adversarial images. The distance between the input and adversarial image is usually defined as a l_∞ norm. All adversarial images should lie inside the l_∞ norm-ball. If the generated example is located outside the norm constrained, it is projected into the area where the constrain is satisfied.

More formally, the PGD attack is defined as follows:

$$x_{i+1}^{adv} = \Pi_{x+S}\left[x_i^{adv} + \alpha \operatorname{sign}\left(\nabla_x L(x, y, \theta)\right)\right] \tag{2}$$

where $x + S$ is the set of perturbation that are inside the l_∞ norm ball around the input x. Π_{x+S} is the projection operation in the PGD algorithm, and α is the step size of the gradient descent.

Adversarial Training is not only one of the most popular and intuitive defence method against adversarial attacks but is also one of the most effective defence methods that achieves state-of-the-art accuracies [15], [16]. Adversarial training augments benign training data by creating adversarial examples and adding them to the training pool. This defence increases the robustness of the trained models against adversarial attacks and can be formulated as a min-max expression:

$$\min_\theta \max_{D(x,x')<\eta} J\left(\theta, x', y\right) \tag{3}$$

where x is the input with adversarial counterpart x', θ is the network weights and y is the ground truth label. $J\left(\theta, x', y\right)$ is the loss function with adversarial input, while $D\left(x, x'\right) < \eta$ is a distance metric between x and x'. The maximization of the adversarial loss serves the purpose of finding the best adversarial examples, which can be created by adversarial attacks mentioned previously like FGSM or PDG. The outer minimization represents the training that minimizes the loss. The goal is to achieve robustness of the model against the adversarial attacks. [17]–[19] highlight that the robustness of adversarial training is achieved by over-fitting. It does not generalize well to other data sets or different attack methods than it was initially trained on.

Sparsity is the absence of weights in an artificial neural network. In general, artificial neural networks have fully connected layers with every neuron $n \in N_l$ in layer l is connected to each node $n \in \{N_{l-1}, N_{l+1}\}$. High amounts of parameters may lead to overparameterization and result in overfitting and poor generalization [20]–[22]. Many weights in fully connected layers are near-zero valued and can be removed to reduce computing power during forward- and backward-propagation while maintaining performance accuracy [7]–[11]. Sparse neural networks are fully connected neural networks with one or more connections removed. Sparsity in neural networks has shown to be a competent tactic to counter overparametrization [7]–[11]. Consequently, over the past years it has attracted increasing interest [23]. The training of sparse neural networks is generally split into two categories: pruning and dynamic training.

Pruning can be subdivided into dense-to-sparse and sparse-to-sparse training. In dense-to-sparse training, a fully connected neural network is initially trained regularly to find weights that minimize the loss. Afterwards it is pruned on near-zero weights. The term pruning has been introduced in 1989 by Mozer and Smolensky [24] and in 1991 by LeCun et al. [8] in Optimal Brain Damage (OBD). OBD was the first sparse training method using dense-to-sparse training.

In response to the decreased accuracy of dense-to-sparse training, Frankle and Carbin (2019) introduced the Lottery Ticket Hypothesis, stating that the remaining weights after pruning might be reset to the value at the first training initialization [25]. Then training this sub-network, it is possible to achieve equal performance compared to the initial network with at most an equal number of training iterations. The resulting sub-network is called the winning lottery ticket. In contrast to dense-to-sparse training, sparse-to-sparse training directly initializes a neural network with reduced weight count. Initial sparse-to-sparse training methods used methodlogies from Restricted Boltzmann Machines (RBMs) [26]. The first sparse-to-sparse training method was introduced in Complex Boltzmann Machines (XBM) by Mocanu et al., outperforming many sparse-to-sparse training methods of the time. In 2017, Bourely et al. [27] showed that arbitrary initialization of a fixed sparse topology can achieve higher accuracy than fully connected neural networks. In the Lottery Ticket Hypothesis winning tickets were found with fully

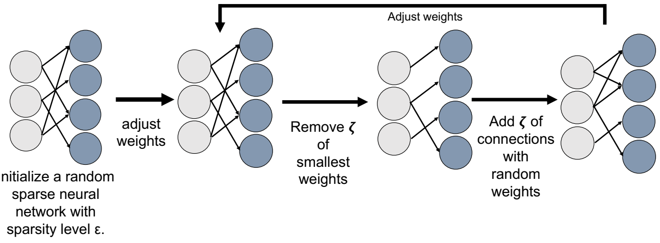
Adjust weights
adjust weights
Remove ζ of smallest weights
Add ζ of connections with random weights
Initialize a random sparse neural network with sparsity level ε.

Fig. 1: The training cycle for Sparse Evolutionary Training (SET) and Rigging the Lottery (RigL)

trained neural networks. In 2020, You et al. [28] proposed that a fully trained neural network is not required because special subnetworks (Early Tickets) can be found in initial training stages to significantly reduce training cost.

Dynamic sparse training removes connections during training. Pruning has proven to effectively reduce training cost of neural networks while maintaining high performance. Because the sparse topology of the network is static, it has to be found outside of training. To overcome static sparse topologies, research started shifting towards dynamic training methods. Dai et al. (2018) [11] coined NeST, a dynamic training method that introduces connections during training to reduce the loss function and subsequently reduces near zero weights. Dettmerse et al. (2019) [10] proposed Sparse Momentum, a sparse training method that distributes removed weights over current weights.

Based on the reduction of neurons in the biological brain [6], Mocanu et al. [13] introduced Sparse Evolutionary Training (SET), a sparse training method that removes and subsequently reintroduces a constant percentage of near-zero weights. The location of connectivity regrowth is random. The topology of the sparse neural network is initialized through Erdős– Rényi sparse weight masks with a uniform distribution. The percentage of near-zero weights is usually refered to as ζ. In 2021, Evci et al. [14] introduced Rigging the Lottery (RigL), a sparse training method focused on finding winning lottery tickets without initially training a fully connected neural network. Similar to SET, RigL uses dynamic removal and adding of weights during training. While SET reintroduces weights and random positions, RigL grows the connections with the highest magnitude gradient.

III. Related Works

There exists extensive research on robustness training and network sparsity, yet little research is conducted at the intersection of these two topics. There are some recent works that study the relationship between weight sparsity of DNNs and adversarial robustness [29] [19] [30] [31] [32]. Guo et. al. [29] are among the first ones to present the intrinsic relationship between sparsity of DNNs and their adversarial robustness on a theoretical and empirical level. They consider sparsity as the sparsity of weights among neurons but also sparsity of neuron activation's in linear and non-linear DNNs. The experiments use FGSM and DeepFool [33], while using a progressive pruning strategy [34] that prunes a portion of weights with the smallest magnitudes after each iteration. These weights are not to be activated again and remain at value zero. Their theoretical and empirical experiments show that sparse nonlinear DNN's have the ability for higher adversarial robustness than their dense counterpart, but that pruning too much of the networks weights had detrimental effects on the performance on both benign and adversarial classification accuracy.

Ye et al. [32] integrate weight pruning into adversarial training and explore if training a smaller model from scratch would yield similar results. They use the Alternating Direction Method Of Multipliers (ADMM) pruning method, which supports irregular and regular pruning schemes that uniformly prune every layer by the same ratio. They show that pre-pruning a model before adversarial training achieves worse accuracy on both adversarial and benign examples. It is also harder to prune an adversarial model than a benign model as adversarial models have fewer zero weights. However, concurrent pruning with adversarial training achieves adversarial robustness. The authors conclude that the initial capacity of the network in the adversarial setting is of importance for robustness, but that it is still possible to achieve adversarial robustness and high standard accuracy when removing unimportant weights from a network with large capacity.

Sehwag et. al. [31] propose a novel pruning technique that is aware of the robust training objective. The robust training objective decides which weights to prune. Rather than to remove weights with the least magnitude, they perform an architecture search with the desired pruning ratio that has the least drop in accuracy. This is achieved by considering the robust training objective as a empirical risk minimization and solving it with the use of stochastic gradient descent. This approach, coined HYDRA, showcases state of the art performance in benign and adversarial accuracy.

Vemparala et al. [30] study the robustness of uncompressed, distilled, pruned and binarized CNN's against white and black box adversarial attacks. The CNNs, a ResNet20 and ResNet56, are trained and evaluated on the CIFAR10 and ImageNet dataset without any adversarial examples in the training set. The distiled CNN's utilize Knowledge Distillation [35] to transfer knowledge from the bigger teacher network to a smaller student network. They use a learning based pruning method that leverages a reinforcement learning agent which learns sparsity potential in each layer and prunes based on an l_1-norm heuristic [36]. By modifying the agent, Vemparala et al. experiment with weight-wise, kernel-wise, filter-wise and channel-wise pruning of the CNN. Despite binarized CNN's representing the strongest form of quantization, with the network weights and activation constrained to [-1;+1], they achieve the least amount of accuracy degradation with regards to their baseline compared to all models. All pruning techniques perform worse compared to their vanilla versions against PGD attacks. The authors conclude that while the pruned networks can show robustness against certain attacks, all pruned networks break against PGD attacks in the experiment.

Vivek et al. [19] observe that models trained with adversarial examples generated from single step attack methods are pseudo robust and as a reaction propose a new adversarial training method involving single step adversarial samples with dropout scheduling. Pseudo robust models are robust against single step white box attacks, but susceptible to single step black box attacks. Furthermore, they are not robust against multi-step attacks in white and black box settings. During their experiments, the adversarial training method with FGSM is not able to produce adversarial examples that maximize the loss after the initial training iterations due to the fact that it overfits on the training data. The multi-step adversarial training with PGD does not showcase overfitting nor the loss of the ability to produce adversarial samples that maximize the loss. Therefore, single-step models learn to prevent the generation of single-step adversaries. To mitigate the overfitting of these single-step models, the authors propose to add a droupout layer after non-linear layers, starting with a high dropout probability and decaying it with the training iterations. With the proposed Single Step Adversarial Training with Dropout Scheduling (SADS) they achieve on par performance with the multi step adversarial training (PGD) while being computationally more efficient on both single and multi step attack in white and black box settings.

The former stated research suggests the effectiveness of reducing neural connectivity through sparsity and drop-out. While some research focuses on the effects of pruning on adversarial training, the combination of sparse- and adversarial training appears wholly novel.

IV. Methodology

For this research robustness is defined in terms of the accuracy performance on clean and adversarial data. The experiments are subdivided into three parts. First the fully connected and sparse models are trained on ResNet50 to compare their performance. Second, the ability of the models to achieve high accuracy in sub-optimal training environments with lower parameter count and fewer epochs is tested. Third, the layer activations throughout the models are visualised to complement the quantitative results.

In part I, models are trained as ResNet50. Nine models are trained on Cifar10 as the combination of fully-connected (FC), SET and RigL and three different adversarial intensities. The learning-rate is 0.001 and momentum is 0.9. Sparsity in the sparse models is simulated through sparse weight masks over the network layers. Mask connectivity is initialized as an Erdős–Rényi random graph [37]. Connection removal-rate is set to $\zeta = 0.3$ and the the remove-growth cycle is activated every 100 gradient steps. Models are initially trained clean and consecutively trained adversarilly. Clean training is done for 100 epochs and robust training for 250 epochs. Both clean training and adversarial training are optimized with Adaptive Momentum Estimation (ADAM). To test robustness, every 10 epochs the model is attacked with an adversarial attack that corresponds to the training type. Three different adversarial intensities are used: $\epsilon = 0.3$ for both attack and defence; $\epsilon = 0.3$ for defence and $\epsilon = 0.1$ for attack to test unequal epsilons; and random sampling from Norm(0,0.25) for both attack and defence to account for the fact that it is unknown to the creator of the model what attack intensity will be used by an attacker. The random sampling occurs at every weight update and at every attack.

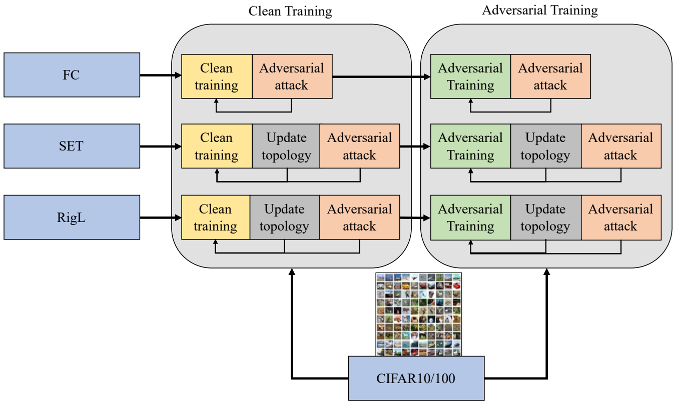

Fig. 2: Model training in step I and II. Models are first trained on clean data and subsequently on adversarial data. During training, the robustness of the models is routinely tested with adversarial attacks and the topology of the sparse models is routinely updated.

In part II, the generalisation ability of sparse- and fully-connected models in adversarial setting is tested by putting it under strain; training on smaller models and with fewer epochs. Models are trained as ResNet34. Six models are trained as the combination of two datasets cifar10/cifar100 and the three training methods: FC, SET and RigL. The training is performed with 50 epochs of clean training followed by 50 epochs of adversarial training. The final test accuracy and connection count will be used as performance metrics. All other hyperparameters are equivalent to those in part I. The training process for part I and II is visualised in Fig. 2

In part III, visualisation of layer activations will be used to qualitatively asses the differences in robustness of the six models. The first layer is visualised by displaying the gradients of the first layer of the neural network as a heatmap with color intensity decided by weight-value. Gradient-weighted Class Activation Mapping (Grad-CAM) [38] is used to retrieve the gradients flowing into the target class from the last convolutional layer to create a rough localization map highlighting regions in the image relevant for classification.

Robust Training	Pertubation	Model	Testing Accuracy		Connection Count	
			CIFAR10	CIFAR100	CIFAR10	CIFAR100
None	None	FC	89.86%	67.02%	21.33E+6	21.33E+6
None	None	SET	89.12%	64.19%	10.79E+6	10.80E+6
None	None	RIGL	89.54%	65.26%	10.80E+6	10.82E+6
None	FGSM	FC	2.54%	1.54%	21.33E+6	21.33E+6
None	FGSM	SET	5.33%	1.96%	10.79E+6	10.80E+6
None	FGSM	RIGL	4.89%	1.86%	10.80E+6	10.82E+6
None	PGD	FC	2.44%	0.1%	21.33E+6	21.33E+6
None	PGD	SET	1.97%	0.11%	10.79E+6	10.80E+6
None	PGD	RIGL	1.84%	0.1%	10.80E+6	10.82E+6
FGSM	None	FC	85.14%	58.32%	21.33E+6	21.33E+6
FGSM	None	SET	92.34%	63.73%	9.16E+6	8.07E+6
FGSM	None	RIGL	91.35%	62.92%	10.64E+6	7.85E+6
FGSM	FGSM	FC	79.97%	38.29%	21.28E+6	21.33E+6
FGSM	FGSM	SET	97.59%	72.09%	9.16E+6	8.07E+6
FGSM	FGSM	RIGL	97.96%	72.12%	10.64E+6	7.85E+6
PGD	None	FC	75.71%	12.41%	21.28E+6	21.28E+6
PGD	None	SET	78.98%	37.83%	8.80E+6	8.80E+6
PGD	None	RIGL	74.09%	40.29%	10.24E+6	10.24E+6
PGD	PGD	FC	45.51%	8.25%	21.28E+6	21.28E+6
PGD	PGD	SET	44.47%	9.10%	8.80E+6	8.10E+6
PGD	PGD	RIGL	44.08%	8.79%	10.24E+6	7.85E+6

TABLE I: Accuracy and connection count after various forms of adversarial attacks through pertubated images for ResNet34 with initial clean training and various combinations of FGSM, PGD and no adversarial training. The clean training has been performed with a small number of epochs of 50. Connection death-rate is set to 0.3.

V. Results

Figure 3 shows the results of part I. The six graphs are the combination of the two attack/training types FGSM/PGD and the three epsilon settings. Each graph shows the validation accuracy during training. The dotted vertical line at epoch 100 denotes the transition from training on clean data to training on adversarial data. The straight plots denote the validation accuracy on clean data and the dotted plots denote the validation accuracy on adversarial data. In general, once the adversarial training starts after 100 epochs of clean training, the accuracy on the clean data decreases, while the accuracy on adversarial data increases. After the initial drop in accuracy for clean data samples, the accuracy mostly recovers and remains mostly stable. For PGD, the drop is far greater and recovery is only a fraction of the original accuracy. In the case of PGD defense/attack ϵ at 0.3, the clean accuracy does not recover at all.

The graphs show that training with clean data creates small but significant amount of robustness against FGSM attacks. The dynamic sparse training methods consistently achieve higher accuracy than fully connected training when trained on clean data and tested on adversarial data. In accordance with previous research, this clean training introduces no significant robustness against PGD attacks.

The accuracy on the clean data samples is always higher than the accuracy on the adversarial data for both FGSM and PGD attacks, except when the models are PGD attacked with a higher ϵ. In this case, the accuracy on adversarial samples is higher than on the clean data samples. There is a substantial difference between the accuracy on clean data samples and adversarial data samples for all models, yet the the difference is smaller with the PGD attack then the FGSM attack. Sparse models mostly achieve higher accuracy on clean data compared to fully connected counterpart, especially in PGD attacks. Unlike pruning that achieves lower performance with PGD [30], dynamic sparse training mostly achieves slightly higher accuracy on adversarial data. the fully connected model achieves significantly higher accuracy on FGSM with defense/attack ϵ at 0.3/0.1.

Table I highlights the generalisation ability of sparse and fully connected models under only a small number of training epochs and with reduced connection count compared to the experiments showcased in Figure 3. The table shows that SET and RigL achieve higher accuracy than fully connected models, especially in cases of robust training with FGSM, in which the difference can be up to 17% for CIFAR 10 and up to 34% for CIFAR 100. In cases where the fully connected model performs better, the difference in performance between the sparse models is marginal and only occurs for CIFAR 10, while the fully connected performance for CIFAR 100 is always lower except for the scenario with no perturbation and no robust training.

The performance of sparse models seems to always be higher or almost equal to the fully connected model, dispite the sparse models having a 50% lower connection count. Although the differences are marginal, the values show that the SET models perform slightly better than the RigL models in most cases. The PGD attack and training create a much greater reduction in accuracy compared to no attack/training than the FGSM attack.

Fig. 4 shows a visual representation of the image features that activate the neurons for the first layer through Guided Backpropagation (GPB) [39], for the three models under the three robust training regimens. Without robust training, there is no significant difference between the three models. For SET, activations are slightly more concentrated on the center of the foreground.

During FGSM training, there are clear visual differences between FC, FGSM and RigL models. Compared to no robust training, the activations are in FC and RigL are reduced and more scattered around the main object of the picture. The activations of SET are even lower but also very focused on the object in the image. The FC model does contain significantly more noise than the sparse models especially in the background area of the image. With close observation, it is possible to see the outline of the boat in the activations.

The PGD attack clearly shows its impact on the first layer activation. The activations are greatly reduced in intensity and are more focused on the object in the picture. Only few parts of the background excite activation. In the fully connected model, the PGD attack moves the attention of the activation towards the left side of the boat, along the horizon that splits the sky from the sea. The sparse models seem to mitigate

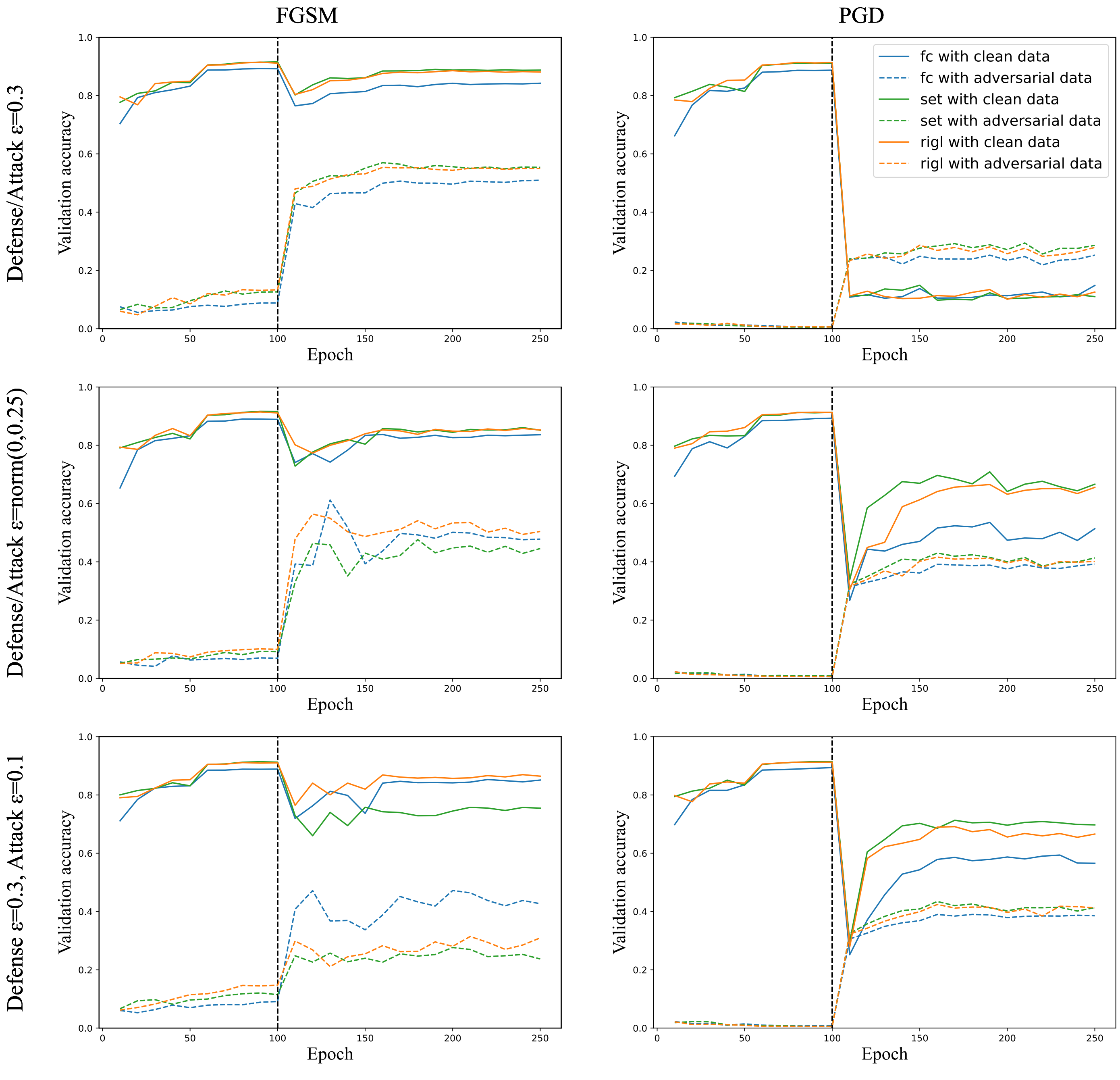

Fig. 3: Validation accuracy of ResNet50 trained on Cifar10. In epoch 0-100, the model is trained with clean training. In epoch 101-250, the model is trained with adversarial data. The straight plots denote accuracy on clean validation data and the dotted plots denote accuracy on adversarial validation data. The training and attacks in the left and right column are respectively FSM and PGD.

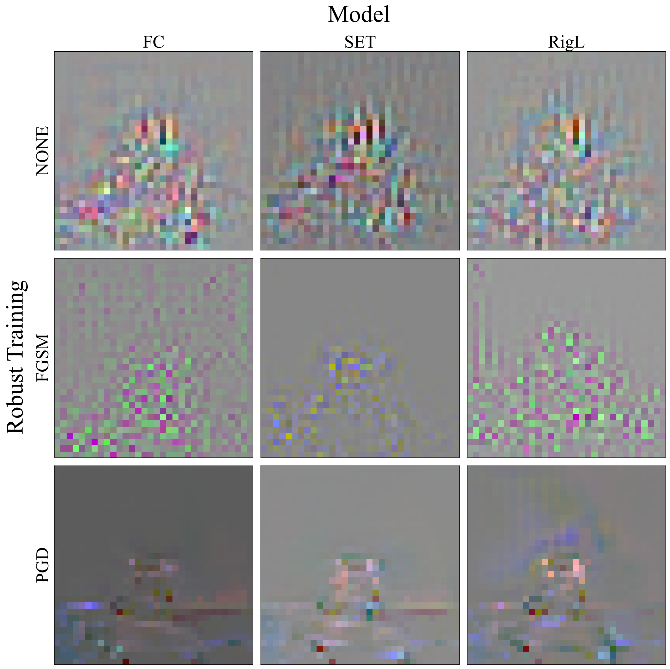

Fig. 4: Visualisation of the first layer of ResNet34 trained on CIFAR10 as fully connected model (FC), sparse model with sparse evolutionary training (SET) and sparse model with Rigging the Lottery (RigL) by means of Guided Back Propagation (GBP). The clean training has been performed in 100 epochs and was followed by various robust training (None, FGSM and PGD).

this move. SET has greater focus towards the middle of the picture and RigL around the boat.

To complement Fig. 4 and explore in more detail the differences between sparse and fully connected models, Fig. 5 showcases the regions of the image most important for the classification by means of Grad-CAM [38]. Similar to the observations from Fig. 4, the sparse method are more concentrated and have less regions that are important for the classification than the fully connected model. With no robust training, the pruning of unimportant weights for the sparse models can be visually inspected, by comparing the increased segments of the boat that have no or small impact on the classification. In case of FGSM with sparse models, it can be observed that the background sky is much less important than without any robust training or compared to the fully connected model.

The robust training with PGD shows that the regions of interest are moved from the structure of the boat towards the sea at the bottom of the picture for the fully connected model and towards the edge of the sky for the sparse models. The sparse models seem to be able to capture the structure of the boat more than the fully connected model while still having attention towards certain parts of the background.

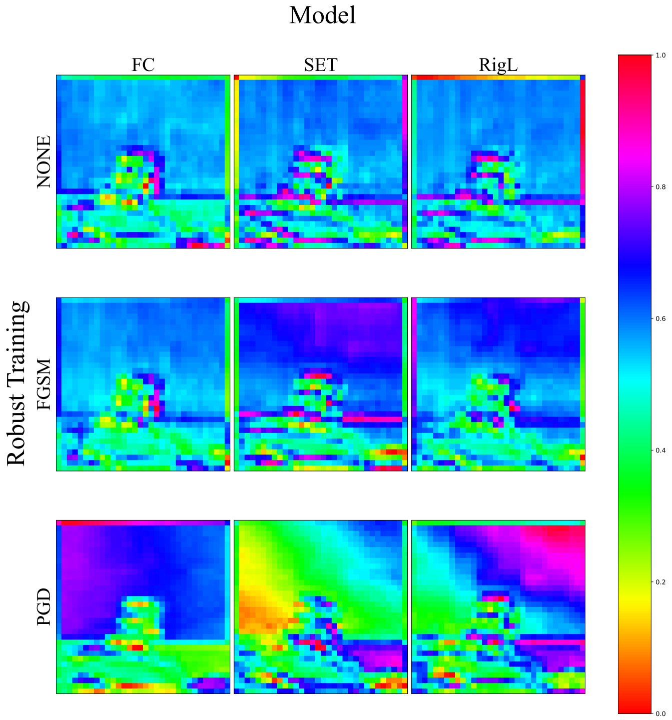

Fig. 5: Visualisation of the region in the image most relevant for classifying. Gradients are retrieved through Grad-CAM from ResNet34 trained on CIFAR10 as fully connected model (FC) sparse model with sparse evolutionairy training (SET) and sparse model with Rigging the Lottery (RigL). The clean training has been performed in 100 epochs and was followed by various robust training (None, FGSM and PGD).

VI. Discussion

The observations show that sparse models are more robust than fully connected models especially when the models undergo adversarial training and/or are only trained for a short period of time. In both clean and adversarial accuracy, the sparse models perform better or on an equal level. The difference is especially visible in PGD attacks, for which the sparse models SET and RigL have a significantly better performance on clean data. In the few cases where the fully connected model performs better than the sparse models, the difference in accuracy is often only marginal. The fully connected model performed better than the sparse models on adversarial accuracy when the attack was simple (FGSM) and the intensity low, shown in Fig. 3. With decreasing adversarial intensity, the fully connected model performs increasingly better compared to the dynamic sparse model. The fully connected model has higher capacity to capture more patterns in the images than sparse models. However sparse models have the advantage in adversrial attacks since they do not model as many patterns as the fully connected models but only the most important ones, making them more robust to attacks. Sparse models are less performant compared to fully connected models models in clean samples since they do not benefit from the increased robustness in clean data compared to when they are attacked.

Although the connection count is almost 2 times lower for the sparse models, they achieve better or on par performance in general, which indicates their increased generializability and ability to counter over-fitting in normal and adversarial training. If the sparse models would have the same number of connections as the fully connected model, it is possible that they would perform significantly better. The better robustness of sparse models is supported by the observation that they seem to be less susceptible to attacks due to their higher concentration and resilience to noise, although PGD does fool the sparse models effectively as well.

The adversarial attacks have shown to shift the attention of the model towards different parts of the picture for both fully connected and sparse models. While especially the PGD attack is still successful in shifting the focus of all models, which is defined by the activation and the most relevant regions for classification, the sparse models showcase a stronger resilience in retaining the important activations that were also generated without any adversarial samples. This resilience could be the reason for the better performance of sparse models not only in adversarial accuracy, but also the accuracy on clean data samples. It also seems that the weaker the intensity of the attack is, the less dominant the sparse models are, but with increasing strength of the attack, the difference becomes more visible until a certain point for PGD, where all models perform poorly due to the strength of the attack.

The results of the research give preliminary insight into the possible potential of sparse models for increasing the accuracy even under adversarial attacks and the potential for mitigating the over-fitting drawback of adversarial training. In order to derive a more generalisable conclusion, several limitations of this study have to be taken into account for future research. First, it is necessary to perform the experiment on multiple different data-sets with more repetitions to mitigate any variance between training models with equivalent hyperparameters and to ensure a higher confidence in the results under different scenarios. Second, previous research has shown that pruning performs worse than fully connected models on PGD, while this research shows that dynamic sparse training performs better or on par to fully connected models on PGD. This should be taken as an indication rather than a conclusion as the results are from different models in different contexts. Adding pruning techniques to the experiments, such that results are actually comparable, could yield more insights into the comparative performance of different sparsity techniques and highlight possible reasons on the higher robustness of dynamic sparse models. Third, since the analysis of possible reasons for the increased performance of sparse models are only performed based on visual clues from the Grad-CAM and first layer activations, it is not possible to confidently generalize its conclusions. A more formal mathematical analysis of the possible effects of pruning on adversarial attack would be a next step to explore this phenomena further. Finally, as mentioned previously, training the sparse model to have the same connection count as their fully connected counterpart after the pruning procedure, could showcase even higher performances by the sparse models.

VII. Conclusion

In this study we show that dynamic sparse training can increase the accuracy of a model that is under attack by adversarial examples. Specifically, we study the behaviour of two dynamic sparse training procedures SET and RigL, with adversarial training and compared it to a fully connected model on CIFAR 10 and CIFAR 100. Dynamic sparse training performs either on par or better than their fully connected counterpart, while containing less than two times the amount of connections between neurons. Through a visual analysis of the first layer activations and Grad-Cam, we observe that adversarial training tries to shift the attention of the model away from the foreground of the image, leading to increased focus on regions unrelated to the subject of the image such that it leads to misclassification. Dynamic sparse training mitigates this effect to some extent by retaining the focus on the important parts of the object and giving less importance to possible noise in the image.

Future work can focus on experimenting under more diverse scenarios, including more data-sets, a broad region of attack intensities, more attack types and comparing the sparse training methods with pruning methods. Furthermore, training the sparse models to have the same number of connection as the fully connected model, could showcase the possible potential of the sparse training methods. Lastly, future work would benefit from approaching a more formal mathematical explanation on possible impacts of adversarial training in combination with sparse training procedures.

References

[1] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[2] Alexey Kurakin, Ian Goodfellow, Samy Bengio, Yinpeng Dong, Fangzhou Liao, Ming Liang, Tianyu Pang, Jun Zhu, Xiaolin Hu, Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, Alan Yuille, Sangxia Huang, Yao Zhao, Yuzhe Zhao, Zhonglin Han, Junjiajia Long, Yerkebulan Berdibekov, Takuya Akiba, Seiya Tokui, and Motoki Abe. Adversarial Attacks and Defences Competition. pages 195–231, 2018. Series Title: The Springer Series on Challenges in Machine Learning.

[3] Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8093–8104. PMLR, 13–18 Jul 2020.

[4] Peilin Kang and Seyed-Mohsen Moosavi-Dezfooli. Understanding catastrophic overfitting in adversarial training. *CoRR*, abs/2105.02942, 2021.

[5] Vivek B. S. and R. Venkatesh Babu. Single-step adversarial training with dropout scheduling, 2020.

[6] Gal Chechik, Isaac Meilijson, and Eytan Ruppin. Synaptic Pruning in Development: A Computational Account. *Neural Computation*, 10(7):1759–1777, October 1998. Conference Name: Neural Computation.

[7] Misha Denil, Babak Shakibi, Laurent Dinh, Marc'Aurelio Ranzato, and Nando de Freitas. Predicting Parameters in Deep Learning. *arXiv:1306.0543 [cs, stat]*, October 2014. arXiv: 1306.0543.

[8] Yann LeCun, John Denker, and Sara Solla. Optimal Brain Damage. In *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann, 1990.

[9] Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both Weights and Connections for Efficient Neural Networks. *arXiv:1506.02626 [cs]*, October 2015. arXiv: 1506.02626.

[10] Tim Dettmers and Luke Zettlemoyer. Sparse Networks from Scratch: Faster Training without Losing Performance. *arXiv:1907.04840 [cs, stat]*, August 2019. arXiv: 1907.04840.

[11] Xiaoliang Dai, Hongxu Yin, and Niraj K. Jha. NeST: A Neural Network Synthesis Tool Based on a Grow-and-Prune Paradigm. *IEEE Transactions on Computers*, 68(10):1487–1497, October 2019. Conference Name: IEEE Transactions on Computers.

[12] D. C Mocanu, A Liotta, and G Exarchakos. *Network computations in artificial intelligence*. PhD thesis, Technische Universiteit Eindhoven, Eindhoven, 2017. ISBN: 9789038643052 OCLC: 993672622.

[13] Decebal Constantin Mocanu, Elena Mocanu, Peter Stone, Phuong H. Nguyen, Madeleine Gibescu, and Antonio Liotta. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature Communications*, 9(1):2383, June 2018. Bandiera_abtest: a Cc_license_type: cc_by Cg_type: Nature Research Journals Number: 1 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Complex networks;Computer science;Machine learning Subject_term_id: complex-networks;computer-science;machine-learning.

[14] Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: Making all tickets winners.

[15] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[16] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 501–509, 2019.

[17] Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pages 8093–8104. PMLR, 2020.

[18] Peilin Kang and Seyed-Mohsen Moosavi-Dezfooli. Understanding catastrophic overfitting in adversarial training. *arXiv preprint arXiv:2105.02942*, 2021.

[19] BS Vivek and R Venkatesh Babu. Single-step adversarial training with dropout scheduling. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 947–956. IEEE, 2020.

[20] Yann LeCun. Generalization and network design strategies. *Connectionism in perspective*, 19:143–155, 1989. Publisher: Elsevier Zurich, Switzerland.

[21] John Denker, Daniel Schwartz, Ben Wittner, Sara Solla, Richard Howard, Larry Jackel, and John Hopfield. Large automatic learning, rule extraction, and generalization. *Complex Systems*, 1, January 1987.

[22] Eric Baum and David Haussler. What Size Net Gives Valid Generalization? In *Advances in Neural Information Processing Systems*, volume 1. Morgan-Kaufmann, 1989.

[23] Lucas Souza. The Case For Sparsity in Neural Networks, Part 2: Dynamic Sparsity, October 2020.

[24] MICHAEL C. MOZER and PAUL SMOLENSKY. Using Relevance to Reduce Network Size Automatically. *Connection Science*, 1(1):3–16, January 1989. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/09540098908915626.

[25] Jonathan Frankle and Michael Carbin. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. *arXiv:1803.03635 [cs]*, March 2019. arXiv: 1803.03635.

[26] Guido Montufar. Restricted Boltzmann Machines: Introduction and Review. *arXiv:1806.07066 [cs, math, stat]*, June 2018. arXiv: 1806.07066.

[27] Alfred Bourely, John Patrick Boueri, and Krzysztof Choromonski. Sparse Neural Networks Topologies. *arXiv:1706.05683 [cs, stat]*, June 2017. arXiv: 1706.05683.

[28] Haoran You, Chaojian Li, Pengfei Xu, Yonggan Fu, Yue Wang, Xiaohan Chen, Richard G. Baraniuk, Zhangyang Wang, and Yingyan Lin. Drawing early-bird tickets: Towards more efficient training of deep networks. *arXiv:1909.11957 [cs, stat]*, August 2020. arXiv: 1909.11957.

[29] Yiwen Guo, Chao Zhang, Changshui Zhang, and Yurong Chen. Sparse dnns with improved adversarial robustness. *Advances in neural information processing systems*, 31, 2018.

[30] Manoj-Rohit Vemparala, Alexander Frickenstein, Nael Fasfous, Lukas Frickenstein, Qi Zhao, Sabine Kuhn, Daniel Ehrhardt, Yuankai Wu, Christian Unger, Naveen-Shankar Nagaraja, et al. Breakingbed: Breaking binary and efficient deep neural networks by adversarial attacks. In *Proceedings of SAI Intelligent Systems Conference*, pages 148–167. Springer, 2021.

[31] Vikash Sehwag, Shiqi Wang, Prateek Mittal, and Suman Jana. Hydra: Pruning adversarially robust neural networks. *Advances in Neural Information Processing Systems*, 33:19655–19666, 2020.

[32] Shaokai Ye, Kaidi Xu, Sijia Liu, Hao Cheng, Jan-Henrik Lambrechts, Huan Zhang, Aojun Zhou, Kaisheng Ma, Yanzhi Wang, and Xue Lin. Adversarial robustness vs. model compression, or both? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 111–120, 2019.

[33] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[34] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.

[35] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

[36] Yihui He, Ji Lin, Zhijian Liu, Hanrui Wang, Li-Jia Li, and Song Han. Amc: Automl for model compression and acceleration on mobile devices. In *Proceedings of the European conference on computer vision (ECCV)*, pages 784–800, 2018.

[37] P. Erdös and A. Rényi. On random graphs i. *Publicationes Mathematicae Debrecen*, 6:290, 1959.

[38] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. 128(2):336–359.

[39] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.