Sparse Artificial Neural Networks: Adaptive Performance-based Connectivity inspired by Human-Brain processes

Viktoriia Lapshyna
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
v.lapshyna@student.utwente.nl

ABSTRACT

Artificial Neural Networks are powerful machine learning systems. However, a high number of weights close to zero make networks unnecessary large and heavy. Sparse models remove redundant weights, aiming to decrease the number of parameters with minimal loss in accuracy. Sparse Evolutionary Training procedure adaptively evolves weights of the Artificial Neural Network topology. This technique proves to remove a vast number of weights and achieve higher accuracy than its non-evolutionary or densely connected counterparts, although the connection addition and removal follows a relatively simple algorithm. Inspired by the synaptic pruning in the human brain, we propose an advanced approach for weight evolution in the Sparse Evolutionary Training algorithm. We suggest gradually removing connections during the training phase as the accuracy increases. We show that the number of parameters can be significantly reduced with almost no loss in accuracy and negligible additional computational complexity. We demonstrate the performance of the algorithm on the Multilayer Perceptron trained on benchmark image and tabular datasets. This research contributes to the understanding of Sparse Artificial Neural Networks and makes a step towards more efficient models.

Keywords

Artificial Neural Network, Sparse Neural Network, Sparse Evolutionary Training, Weight Optimisation, Multilayer Perceptron, Adaptive training.

1. INTRODUCTION

Inspired by the human brain, Deep Learning has found applications in various tasks such as speech recognition, computer vision, autonomous vehicles, natural language processing, robotics and many others. Although ANNs have proven to be notably successful and have the potential for advancing further, they still encounter many limitations.

According to Herculano-Houzel et al. [13], in the human brain, the number of connections between neurons decreases as the grey matter gains neurons. Yet, most of the Artificial Neural Networks have fully-connected neuron layers. Consequently, the number of connections grows quadratically with the increase in the number of neurons. However, a high redundancy in those connections has been observed [5, 16], as, after the training, a large number of weights are close to 0 [8]. In other words, models contain many connections that do not have a significant influence on ANN accuracy but dramatically increase the computational complexity.

This problem has been addressed by introducing sparse neural networks, where some of the parameters are set to 0. Sparse ANNs have shown to improve generalisation, reduce the memory footprint, and increase the training speed in comparison with the dense networks [11, 9, 16]. Some research done in this field is based on the predefined sparsity [18, 17]. Contrary, biological neural networks evolve sparsity over time; for instance, up to 40% of neuron synapses in the human brain are replaced with new ones every day [12]. Therefore, to make Artificial Neural Networks more similar to the biological ones, it is crucial to incorporate evolutionary neuron connectivity optimisation.

Mocanu et al. have suggested the Sparse Evolutionary Training (SET) to address this problem [20]. This method is simple and efficient: after each training epoch a fixed number of smallest weights are removed, and the same number of new connections are grown randomly. Researchers have observed a high decrease in the number of parameters with no loss in accuracy. During the training, SET keeps the number of connections constant. Contrary, in the biological brain, overproduction of synapses is followed by the gradual reduction.

Within this research, we propose a novel approach to neuron connectivity evolution in the SET procedure, Accuracy based Sparse Evolutionary Training (AccSET). Taking inspiration from the synaptic pruning, we suggest gradually cutting down a number of connections as accuracy increases during the training procedure. This method reduces the risk of overfitting, increases generalisation and requires negligible extra computational cost compared to SET. We test our algorithm on CIFAR10[15], FashionMNIST[26] and HIGGS[1] datasets based on the Multilayer Perceptron (MLP).

AccSET allows ANNs having a much higher number of neurons. At the same time, because of small number of non-zero weights, trained ANN model takes less memory than its fully-connected counterpart, which may allow saving trained models even on smartphones. Additionally, this research contributes to the understanding of the sparsity of Artificial Neural Networks.

The Background section of this paper introduces concepts on which this research is based. Next, the Related Work talks about researches provided in the similar direction. In the Proposed Method, the details of implementation of the AccSET is presented. Results section illustrate how the experiments were performed. Finally, the Discussion and Future Work summarises the paper and suggest the direction of the future work.

32^{th} *Twente Student Conference on IT* Jan. 31^{st}, 2020, Enschede, The Netherlands.

2. BACKGROUND

2.1 Artificial Neural Networks

Artificial Neural Network (ANN) is a mathematical model inspired by the animal brain. Like its biological counterpart, ANN consists of neurons and neuron connections. The signal transmission by synapses in the biological brain is modelled as weights of the ANN. Initially, the goal of the ANNs was to perform similarly to a human brain. However, currently, they are mostly used for classification, prediction clustering and association.

There are plenty types of the ANNs, but, for the purpose of this research we discuss only Multilayer Perceptron(MLP). MLP is the classical neural network, that consists of the input layer, one or more hidden layers and the output layer of neurons. MLP is trained using two steps: forward propagation and backward propagation. Forward propagation sequentially calculates and stores intermediate variables within the compute graph defined by the neural network. It proceeds from the input to the output layer. Backpropagation sequentially calculates and stores the gradients of intermediate variables and parameters within the neural network in the reversed order. When training deep learning models, forward propagation and backpropagation are interdependent.

ANNs have a risk of overfitting, which is often prevented by the use of dropout, randomly eliminating neurons from the network during training. The loss function is used to calculate the loss of the training, the difference between the expected and actual output. Parameters of the model are updated by the optimiser, most commonly gradient descent, in the direction that lowers the loss. The step taken along the gradient is determined by the learning rate, a number between 0 and 1. The activation function is a mathematical equation which defines whether the neuron should be activated or not.

Training requires significant memory and storage. The reason behind this is that most of the ANNs have fully connected (FC) layers, meaning that each neuron from layer n is connected to each neuron from the subsequent layer $n+1$. The layer is called sparsely connected (SP) if at least one of the connections is missing.

2.2 Synaptic pruning in the human-brain

According to E.Santos and C.A.Noggle, synaptic pruning is a process which eliminates the synaptic connections in order to increase the efficiency of neural transmissions [25]. At birth, the human brain generates considerably more synapses than functionally needed. Synaptic pruning refines the neural circuit and increases the efficiency by pruning away unnecessary synapses: the least used connections are removed and more frequently used connections are strengthened. Until approximately 10 years of age, about 50% of synapses that were present at the age of 2 are eliminated. However, that process also continues during the later years. This way, the brain reaches optimal learning performance.

The synaptic pruning primarily inspires the AccSET method proposed in this research.

3. RELATED WORK

The topic of the sparsity of artificial neural networks has recently gotten serious attention. Some researchers suggest first training fully connected neural network, afterwards pruning unimportant connections and retraining the network again to tune the weights of the remaining connections [10, 16, 11] Another approach is to prune connections prior to the training[18, 7]. These works have shown that pruning can remove a large number of parameters with a little cost in accuracy. The main disadvantage of these methods is that the sparsity is fixed and not adapted through the training. Contrary, the human brain continually removes and grows new synapses.

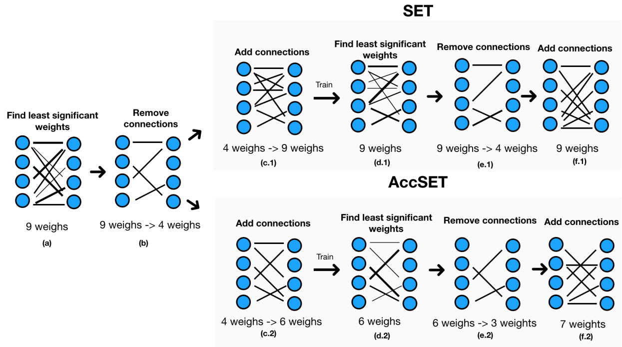


Figure 1: *An illustration of difference between the SET and AccSET procedure. For each sparsely connected layer SC (a), the fraction of connections is removed (b). SET grows the same number of connections (c.1), while AccSET calcultates the number of connections to grow based on the Equation 1 (c.2). The process continues for the finite number of training epochs. If accuracy after a training epoch $e+1$ is lower than accuracy after the training epoch e, AccSET will grow more connection after $e+1$ (f.2). SET keeps number of connection constant during the training.*

3.1 Evolutionary Sparsity

Addressing this problem, in 2017 Dai et. al [4] have introduced a Neural Network Synthesis Tool (NeST). The method starts with randomly initialising sparse Deep Neural Network (DNN) architecture. During the training, neuron connections are grown based on the gradient information, and finally, insignificant connections are pruned away based on their magnitude. Narang et al. have proposed gradually pruning connections using a monotonically increasing threshold [23].

Another approach, Deep Rewiring (DeepR) suggests updating both weights and the connectivity graph during training [2].

Advancing previous techniques, in 2019, the dynamic reallocation of weights was suggested[21]. This method uses weight magnitude for pruning connections and dynamically reallocates weights between layers at the end of each training epoch. A similar approach, Sparse Momentum [6] redistributes pruned weights across layers according to the mean momentum magnitude of each layer.

Mocanu et al. have introduced Sparse Evolutionary Training (SET). First, the fully-connected (FC) layers of ANN are replaced with the sparsely connected (SP) layers. Then, after each training epoch, for each SC layer, fraction of weights closest to 0 is removed and the same number of connections is added randomly. The addition step is omitted in the last training epoch. This algorithms has outperformed its fully connected counterparts and quadratically reduced the number of parameters of neural networks layers. The illustration of this method can be found in Figure 1 and Algorithm 1 [1].

It has been shown that, compared to SET, Sparse momentum[6] has a higher accuracy with a Wide Resnet model WRN-28- 2 on the CIFAR10 dataset, nevertheless the improvement is not significant. According to other researches, SET has significantly higher accuracy compared to the DeepR and performs slightly worse than the Dynamic Sparse[6, 21] and Sparse Momentum [6].

Recently, a more sophisticated variation of SET which uses the cosine similarity for growing new connections was introduced [24].The approach was tested on 8 dataset and proved to be significantly more effective for Madelon dataset (13.17% more accurate than SET- MLP). Overall, the most efficient way for connection evolution was to use the cosine similarity for

[1] For more details see Section 4.

Table 1: *Definition of parameters*

Parameter	Definition	Parameter	Definition
ϵ	sparsity level	λ_e^l	number of connections at layer l after training epoch e after removing zeta smallest connections
ζ	fraction of connections to add after each training epoch e	γ^l	number of connections at layer l before the training
acc_e	accuracy after the training epoch e	θ_e^l	fraction of connections to add at layer l after training epoch e

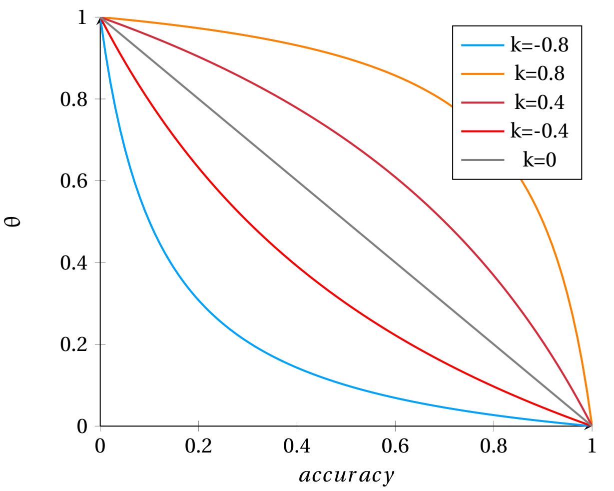


Figure 2: *Graph of the function* $\theta = 1 - \dfrac{(acc_e - acc_e \times k)}{k - |acc_e| \times 2 \times k + 1}$ *for different k*

both removal and addition of weights. Nevertheless, the pure SET-MLP worked better on the COIL-100 dataset.
This research is mainly based on the SET. We propose a novel approach for growing new connections in the ANN model.

4. ACCURACY BASED SPARSE EVOLUTIONARY TRAINING

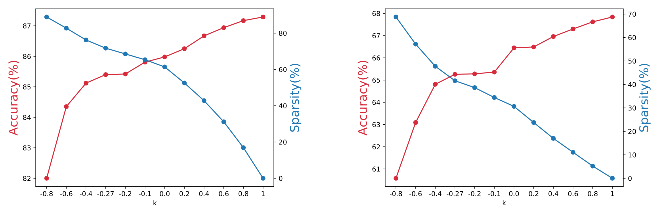
Within this research, we introduce Accuracy based Sparse Evolutionary Training (AccSET) method inspired by SET and the synaptic pruning of the biological brain. The general idea is to start with an already sparse ANN and update the connectivity graph during the training, reducing the number of connections. Note that contrary to synaptic pruning, within the AccSET, we start with the high number of connections, instead of producing them gradually at the beginning of the training. Such an approach is motivated by memory efficiency and the speed of the training. After each training epoch, the number of connections is reduced based the on accuracy. The main rule is with the increase in accuracy, connections are removed, but if accuracy drops, more connections are added. This way, the ANN learns the optimal number of connections.
We start with initialising the SC ANN model with the sparsity level ϵ, as it is done in SET. Then, after each training epoch, for each layer, the fraction ζ weights closest to zero are set to 0. Afterwards, the number of weights to be added is calculated as follows

$$(\gamma^l - \lambda_e^l) \times (1 - \frac{(acc_e - acc_e \times k)}{k - |acc_e| \times 2 \times k + 1}) \qquad (1)$$

where γ^l corresponds to the initial number of connections before training at the l^{th} layer, λ_e^l is the number of connections at the e^{th} epoch and l^{th} layer; and acc_e is the corresponding accuracy, k is tunable parameter. Finally, $\theta_e^l \times \lambda_e^l$ connections are added randomly to the layer l of the ANN. This step is omitted at the last training epoch.

Algorithm 1 Accuracy SET pseudocode

set ϵ and ζ
initialise ANN model
for each fully-connected (FC) layer of the ANN **do**
  replace FC with a Sparse Connected (SC) layer
end for
for each training epoch e **do**
  perform standard training procedure
  for each layer **do**
    remove ζ weights closest to zero
    λ_e^l := current number of connections
    $\Delta_e^l \leftarrow \gamma_e^l - \lambda_e^l$
    if e is not the last training epoch **then**
      $\theta_e^l \leftarrow 1 - \dfrac{(acc^l - acc^l \times k)}{k - |acc^l| \times 2 \times k + 1}$
      add $\theta_e^l \times \Delta_e^l$ new random connections
    end if
  end for
end for



(a) *FashionMNIST* **(b)** *CIFAR10*

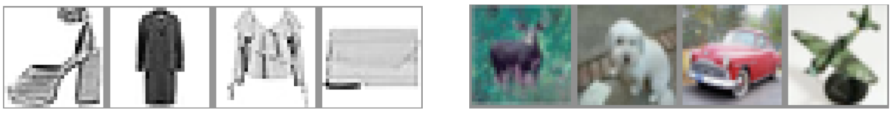
Figure 3: *The relation between the Sparsity and Accuracy of the MLP. Overall, higher sparsity is associated with lower accuracy.*

Figure 2 illustrates the graph of the function based on which the number of connections to be added after each training epoch is calculated. Note that if $k = 0$ the function becomes linear $\theta_e^l = (\gamma^l - \lambda_e^l) \times acc_e$. This function was chosen in an attempt to mimic the gradual reduction of synapses in the human brain. Moreover, with changing parameter k, a different degree of sparsity can be achieved, which can be relevant for performing different tasks.
The pseudocode can be found in the Algorithm 1. [2]. Note that if $k = 1$, AccSET transforms to SET, as number of connections that will be added after the training epoch e for layer l is $\theta \times \lambda$. All parameters and their definitions used in this paper can be found in Table 1.
The essence of AccSET lays in the modified number of connections added after each training epoch. While SET keeps the number of connections constant during the entire training, AccSET gradually removes connections as accuracy increases. The difference is illustrated in Figure1.

[2]For more implementation details, e.g. how the SET model is initialised, please refer to the original paper [20]

Table 2: *Dataset statistics*

Dataset	Domain	Data Type	Classes	Train samples	Test samples
FashionMNIST	Image	Grayscale	10	60 000	10 000
CIFAR10	Image	RGB colors	10	50 000	10 000
HIGGS	Particle physics	Real values	2	10 500 000	500 000



(a) *FashionMNIST* **(b)** *CIFAR10*

Figure 4: *Examples of datasets images*

Table 3: *Summarisation of all experiments on image datasets*

		FashionMNIST			*CIFAR10*		
Model	**k**	**Accuracy[%]**	**Loss**	**Connections[#]**	**Accuracy[%]**	**Loss**	**Connections[#]**
AccSET-MLP	-0,8	82,00	0,51382	2668	60,58	1,12056	86942
	-0,6	84,35	0,45237	4151	63,09	1,04248	119122
	-0,4	85,12	0,42821	5714	64,81	0,99992	145436
	-0,27	85,40	0,41268	6802	65,26	0,99251	162595
	-0,2	85,42	0,41164	7568	65,28	0,98274	170728
	-0,1	85,81	0,40097	8337	65,36	0,98311	182433
	0	85,98	0,39284	9290	66,45	0,95943	192994
	0,2	86,25	0,38742	11417	66,49	0,95341	212103
	0,4	86,67	0,37544	13760	66,96	0,9413	230939
	0,6	86,94	0,36764	16560	67,3	0,92904	247406
	0,8	87,17	0,36019	19991	67,62	0,92915	263786
SET-MLP		87,29	0,34977	24052	67,84	0,91545	278264
MLP		88,22	0,46067	247272	65,78	0,9971	20328000

5. RESULTS

5.1 Implementations details

The method was implemented with Keras, using the weight mask to set weights to 0 in order to obtain the sparse structure. Hyperparameters used in the training can be found in Table 4. Perhaps the most used activation functions nowadays are Rectified linear unit (ReLU) [22], and their variations. We have used LeakyReLU [19], while Mocanu et al. have chosen SReLU [14] in their research [20]. We believe that this is the reason why SET results slightly differ between our experiments and the ones performed by Mocanu et al.

5.2 Dataset characteristics

Benchmark datasets, FashionMNIST[26], CIFAR10[15], and HIGGS[1], are helping us to asses the AccSET performance on the multi-class classification problem. Characteristics of all three datasets can be found in Table 2.

Figure 4 illustrates sample images of image datasets. Both FashionMNIST and CIFAR10 datasets classify images into 10 categories. CIFAR10 is a more challenging dataset, as it consists of 3 colour channel images, while FashionMNIST images are grayscale with a lower resolution. Additionally, contrary to CIFAR10, FashionMNIST images don't contain any background or noises, making it easier to achieve better performance.

In 2019, Bourgin et al. [3] showed, by analyzing three datasets, that SET can be a very good solution for tabular data. Thus, our third dataset was specially chosen to asses the AccSET capabilities on an even larger tabular dataset, HIGGS. This dataset is from particle physics and contains 2 labels "s" and "b," that stand for "signal" and "background" event of Higgs Boson decay.

Table 4: *Hyperparameters used in training of MLP*

Hyper-parameter	Value	Hyper-parameter	Value
Learning rate	0.01	ζ	0.3
Optimiser	SGD	Batch size	100
Momentum	0.9	ϵ	20
Activation Function	LeakyReLU	Loss Function	Categorical Cross-entropy
Dropout rate	0.3		

5.3 AccSET: the optimal trade-off between Sparsity and Accuracy

We have compared AccSET-MLP, SET-MLP and standard fully-connected MLP on all three datasets.

5.3.1 Image datasets

For both image datasets, all models were trained for 1000 epochs.

First, we have compared the performance of AccSET-MLP with different values of k as specified in Equation 1 to find the optimal value of k for a given dataset. Table 3 summarises all training results for image datasets and Figure 3 shows the

Table 5: *Summarisation of the best experiments with MLP variants. For each dataset we report the best validation accuracy for MLP, SET-MLP and AccSET-MLP. For the AccSET-MLP one best value of k was chosen achieving the best accuracy sparsity trade-off.*

Dataset	Training epochs [#]	Architecture	MLP		SET-MLP		AccSET-MLP		
			Accuracy [%]	Connections [#]	Accuracy [%]	Connections [#]	Accuracy [%]	Connections [#]	k
FashionMNIST	1000	784-256-128-100-10	88.22	247 272	87.29	24 052	85.81	8 337	-0.1
CIFAR10	1000	3072-4000-1000-4000-10	65.78	20 328 000	67.84	278 264	65.26	162 595	-0.27
HIGGS	100	28-1000-1000-1000-2	75.87	2 030 000	75.05	72 308	74.43	29 936	-0.27

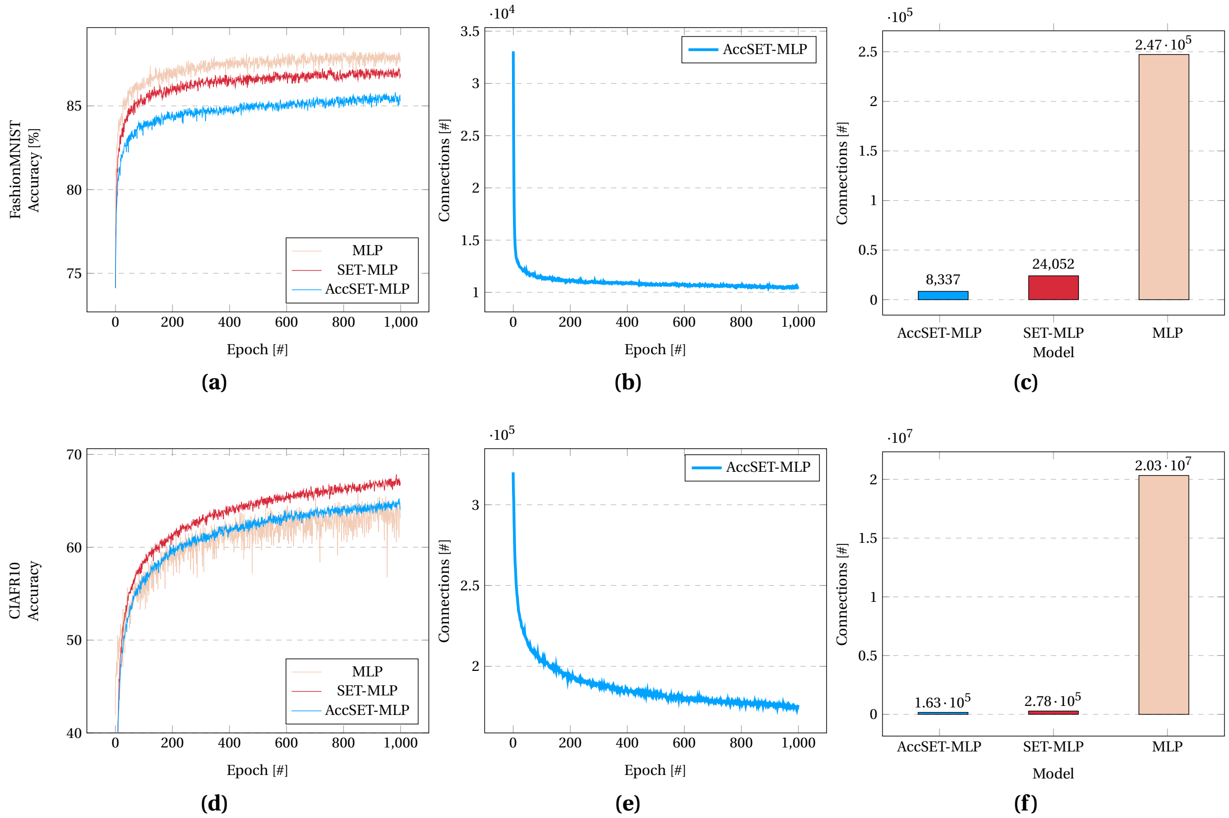


Figure 5: *Experiments on MLP using FashionMNIST (a) (b) (c) and CIFAR10 (d) (e) (f) datasets. Figures (a) and (d) depict accuracy over training; figures (b) and (e)illustrate the number of connection during the training except for the last epoch for AccSET-MLP. Number of connections for SET-MLP and MLP remains constant during the training. Figures (c) (f) and illustrate the final number of connections at the end of the training.*

relation between sparsity and accuracy based on k. Note that the sparsity in Figure 3 is compared to SET-MLP, not to dense MLP. More precisely, the sparsity is calculated as $1 - n_{AccSET}/n_{SET}$ where n_{AccSET} and n_{SET} represent the number of connections in AccSET-MLP and SET-MLP respectively. The main reason behind this is the visibility of numbers, as the sparsity of AccSET-MLP compared to the dense MLP reaches at least 98% for most values of k on CIFAR10 and 90% on FashionMNIST.

Figure 3 illustrates that the higher sparsity is generally associated with lower accuracy and vice versa when training AccSET-MLP. As depicted in Table 3, the lowest accuracy is observed on models with $k = -0.8$ for both datasets, corresponding to the highest sparsity. On the FashionMNIST dataset, the lowest accuracy for AccSET-MLP is 82%, which is 5% lower than the accuracy of SET-MLP and 6% lower than MLP. However, this AccSET-MLP model is 88.9% more sparse than SET-MLP and 98.9% than MLP. On CIFAR10, a similar performance is observed. The lowest accuracy for AccSET-MLP is about 7% lower than SET-MLP and 5% lower than MLP, while AccSET-MLP sparsity is about 68.8% and 99.6% compared to SET-MLP and MLP respectively. For both datasets, accuracy is comparatively low for $k < -0.4$, but as $k \geqslant -0.4$, accuracy loss becomes not significant. The more k grows, the higher becomes accuracy, and the more drops the sparsity. The highest accuracy on FashionMNIST is observed with the value of $k = 1$, which is the same as SET-MLP. The best trade-off between sparsity and accuracy is reached with approximately $k = -0.1$ and $k = -0.27$ for FashionMNIST and CIFAR10 respectively as illustrated in Figure 3.

Table 5 and Figure 5 depict training results for the best values of k for AccSET-MLP as well as SET-MLP and MLP on FashionMNIST and CIFAR10 datasets. We observe a higher accuracy on AccSET-MLP and SET-MLP than MLP on the CIFAR10 dataset. It is often difficult to identify the optimal number of neurons for the MLP model. As in the case of the CIFAR10 dataset, a high number of parameters cause overfitting of standard dense MLP. On the other hand, as discussed earlier, the FashionMNIST dataset is simpler than CIFAR10, consequently, FashionMNIST requires a smaller number of parameters. Hence, we have chosen a different architecture for training models on FashionMNIST than on CIFAR10. We see that a more suitable architecture on FashionMNIST results

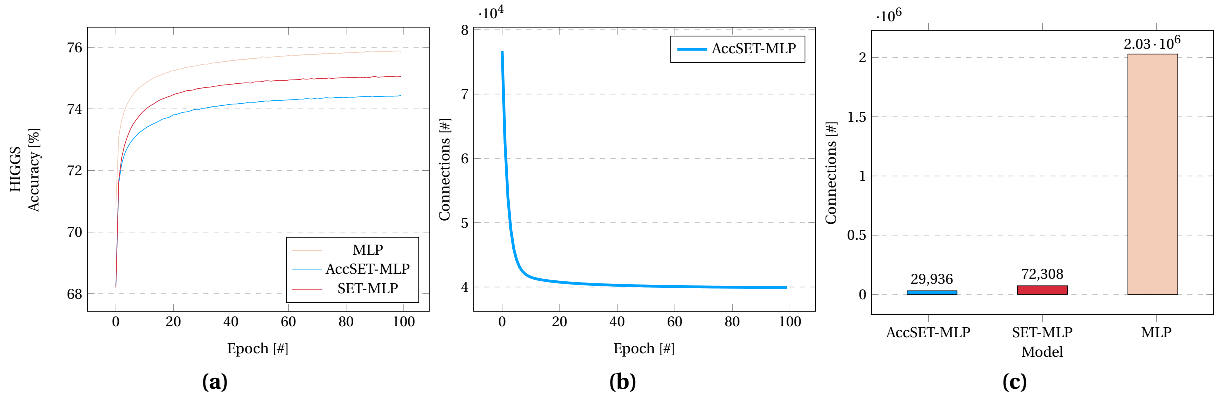


Figure 6: *Experiments on MLP using HIGGS dataset. Figure 6a illustrates accuracy for dense MLP, SET-MLP and AccSET-MLP during 100 epochs. Figure 6b illustrate the evolution of connections of AccSET-MLP for 100 epochs. Note that MLP and SET-MLP keep number of connections constant. Figure 6c show the number of connections during training for 100 epochs for dense MLP, SET-MLP and AccSET-MLP.*

in higher accuracy for standard dense MLP compared to both AccSET-MLP and SET-MLP. Nevertheless, with the AccSET method, we have managed to increase the sparsity by 99.6% compared to dense MLP resulting in only 2.5% lower accuracy. At the same time, AccSET-MLP reaches approximately the same accuracy as dense MLP while having up to 99.2% fewer connections on CIFAR10 dataset.
In general, AccSET-MLP reaches about 2% lower accuracy than SET-MLP, while having about twice fewer connections on CIFAR10 and 70% fewer connections on FashionMNIST.
Figures 5a and 5d compare accuracy during the training for AccSET-MLP, SET-MLP and standard dense MLP on Fashion-MNIST and CIFAR10 datasets respectively. Figures 5b and 5e illustrate the connection evolution in AccSET-MLP, excluding the final cutting of the connection after the training. As in the synaptic pruning in the human brain, the number of connections is significantly reduced at the beginning of the training, followed by the gradual reduction until the end of the training. Finally, figures 5c and 5f visually compare the number of connections after the training in AccSET-MLP, SET-MLP and standard dense MLP.

5.3.2 Particle Physics Dataset

Similar experiments were conducted on the HIGGS dataset; all models were trained for 100 training epochs to find the best value of k. From our experiments we have concluded that HIGGS dataset doesn't require a high number of training epochs to illustrate the performance of AccSET algorithm.
Interestingly, for the HIGGS dataset, the optimal trade-off between sparsity and accuracy was achieved for $k = -0.27$, the same value as for CIFAR10. The training results for dense MLP, SET-MLP and AccSET-MLP with $k = -0.27$ can be found in Table 5 and the illustration of those results is presented in Figure 6. Similarly to FashionMNIST, on the HIGGS dataset, we observe a higher accuracy for dense MLP than SET-MLP and AccSET-MLP. However, the difference in the accuracy is even smaller. AccSET-MLP achieves 1.5% lower accuracy than MLP, while having only 15% of MLP connections. At the same time, AccSET performs only 0.6% worse than the SET method, while having more than twice fewer connections.

Overall, AccSET method offers a flexible trade-off between sparsity and accuracy, which can be adjusted according to the needs.

6. CONCLUSION AND FUTURE WORK

This research proposes a novel method for sparsifying Artificial Neural Networks, Accuracy based Sparse Evolutionary Training (AccSET), taking the inspiration from synaptic pruning in the human brain. This method starts with already sparse ANN and gradually removes more connections during the training as the accuracy increases. On the example of MLP, we show that the number of connections can be reduced up to 99.2% with no trade-off in accuracy, compared to the densely connected MLP. AccSET models were trained on FashionMNIST, CIFAR10, and HIGGS and performed comparatively well on all three datasets. At this stage, these successful results can be considered as a proof-of-concept. Still, more experiments should be performed in order to prove the robustness of the AccSET training method over various types of data, such as video, audio or multi-modal data.
AccSET has shown exemplary performance on MLP; nonetheless, it will also be valuable to conduct experiments on different types of ANN models, such as, for instance, convolutional and recurrent neural networks.
AccSET approach does not add significant computational complexity and performs as fast as SET.
AccSET follows a relatively simple logic for growing new connections. Future research might look into more complex functions for calculating the fraction of connections to be added during the training. Random allocation of weights can also be replaced with a more sophisticated method. Another future research direction could be combining AccSET with neural pruning methods. AccSET training results in more lightweight ANNs than its fully-connected counterparts, because of the significantly lower number of connections. This allows including more neurons in the ANN, possibly reaching billion-scale models, consequently solving more significant real-world tasks. Additionally, such lightweight models have the potential to be stored on laptops or mobile devices.
Overall, the AccSET model has reached an exceptionally low number of connections while having a negligible lower accuracy compared to its densely connected counterparts.

7. ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisor Elena Mocanu for her valuable feedback and personal motivation during this research.

8. REFERENCES

[1] P. Baldi, P. Sadowski, and D. Whiteson. Searching for exotic particles in high-energy physics with deep learning. *Nature Communications*, 5(1), Jul 2014.

[2] G. Bellec, D. Kappel, W. Maass, and R. A. Legenstein. Deep rewiring: Training very sparse deep networks. *CoRR*, abs/1711.05136, 2017.

[3] D. D. Bourgin, J. C. Peterson, D. Reichman, S. J. Russell, and T. L. Griffiths. Cognitive model priors for predicting human decisions. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5133–5141, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[4] X. Dai, H. Yin, and N. K. Jha. Nest: A neural network synthesis tool based on a grow-and-prune paradigm. *CoRR*, abs/1711.02017, 2017.

[5] M. Denil, B. Shakibi, L. Dinh, M. Ranzato, and N. de Freitas. Predicting parameters in deep learning. *CoRR*, abs/1306.0543, 2013.

[6] T. Dettmers and L. Zettlemoyer. Sparse networks from scratch: Faster training without losing performance. *CoRR*, abs/1907.04840, 2019.

[7] S. Dey, Y. Shao, K. M. Chugg, and P. A. Beerel. Accelerating training of deep neural networks via sparse edge processing. *CoRR*, abs/1711.01343, 2017.

[8] S. Dieleman and B. Schrauwen. Accelerating sparse restricted boltzmann machine training using non-gaussianity measures. In Y. Bengio, J. Bergstra, and Q. Le, editors, *Deep Learning and Unsupervised Feature Learning, Proceedings*, page 9, 2012.

[9] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding, 2015.

[10] S. Han, J. Pool, J. Tran, and W. J. Dally. Learning both weights and connections for efficient neural networks. *CoRR*, abs/1506.02626, 2015.

[11] B. Hassibi, D. G. Stork, and G. J. Wolff. Optimal brain surgeon and general network pruning. In *IEEE International Conference on Neural Networks*, pages 293–299 vol.1, March 1993.

[12] J. Hawkins. Special report : Can we copy the brain? - what intelligent machines need to learn from the neocortex. *IEEE Spectrum*, 54(6):34–71, June 2017.

[13] S. Herculano-Houzel, B. Mota, P. Wong, and J. H. Kaas. Connectivity-driven white matter scaling and folding in primate cerebral cortex. *Proceedings of the National Academy of Sciences*, 107(44):19008–19013, 2010.

[14] X. Jin, C. Xu, J. Feng, Y. Wei, J. Xiong, and S. Yan. Deep learning with s-shaped rectified linear activation units, 2015.

[15] A. Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 05 2012.

[16] Y. LeCun, J. S. Denker, and S. A. Solla. Optimal brain damage. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems 2*, pages 598–605. Morgan-Kaufmann, 1990.

[17] N. Lee, T. Ajanthan, S. Gould, and P. H. S. Torr. A signal propagation perspective for pruning neural networks at initialization, 2019.

[18] N. Lee, T. Ajanthan, and P. H. S. Torr. SNIP: single-shot network pruning based on connection sensitivity. *CoRR*, abs/1810.02340, 2018.

[19] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

[20] D. Mocanu, E. Mocanu, P. Stone, P. Nguyen, M. Gibescu, and A. Liotta. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature Communications*, 9, 12 2018.

[21] H. Mostafa and X. Wang. Parameter efficient training of deep convolutional neural networks by dynamic sparse reparameterization. *CoRR*, abs/1902.05967, 2019.

[22] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, page 807–814, Madison, WI, USA, 2010. Omnipress.

[23] S. Narang, G. F. Diamos, S. Sengupta, and E. Elsen. Exploring sparsity in recurrent neural networks. *CoRR*, abs/1704.05119, 2017.

[24] J. Pieterse and D. C. Mocanu. Evolving and understanding sparse deep neural networks using cosine similarity. *CoRR*, abs/1903.07138, 2019.

[25] E. Santos and C. A. Noggle. *Synaptic Pruning*, pages 1464–1465. Springer US, Boston, MA, 2011.

[26] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.