


Supervised Robustness-preserving Data-free Neural Network Pruning

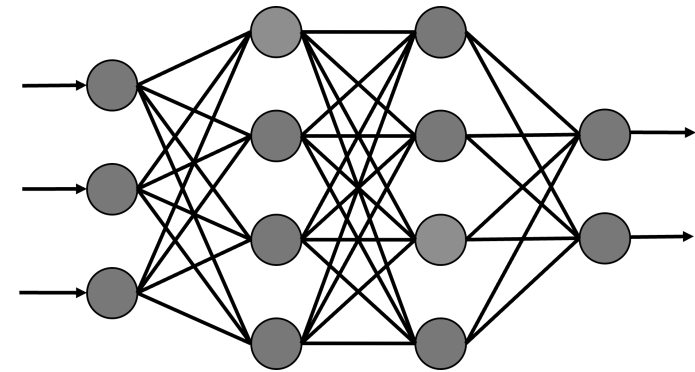
 **Mark Huasong Meng** (*National University of Singapore / Institute for Infocomm Research (I2R), A*STAR, Singapore*)
Guangdong Bai (*The University of Queensland, Australia*)
Sin G. Teo (*Institute for Infocomm Research (I2R), A*STAR, Singapore*)
Jin Song Dong (*National University of Singapore*)



Background

Neural Networks

- Deep learning is usually realized by a neural network model that is trained with a large amount of data.
- Neural networks aim to precisely approximate an arbitrary non-linear computation and are good at handling more complicated tasks, such as financial services, cybersecurity and sports analytic.
- It is a common practice to design a neural network model that is large and deep in size.

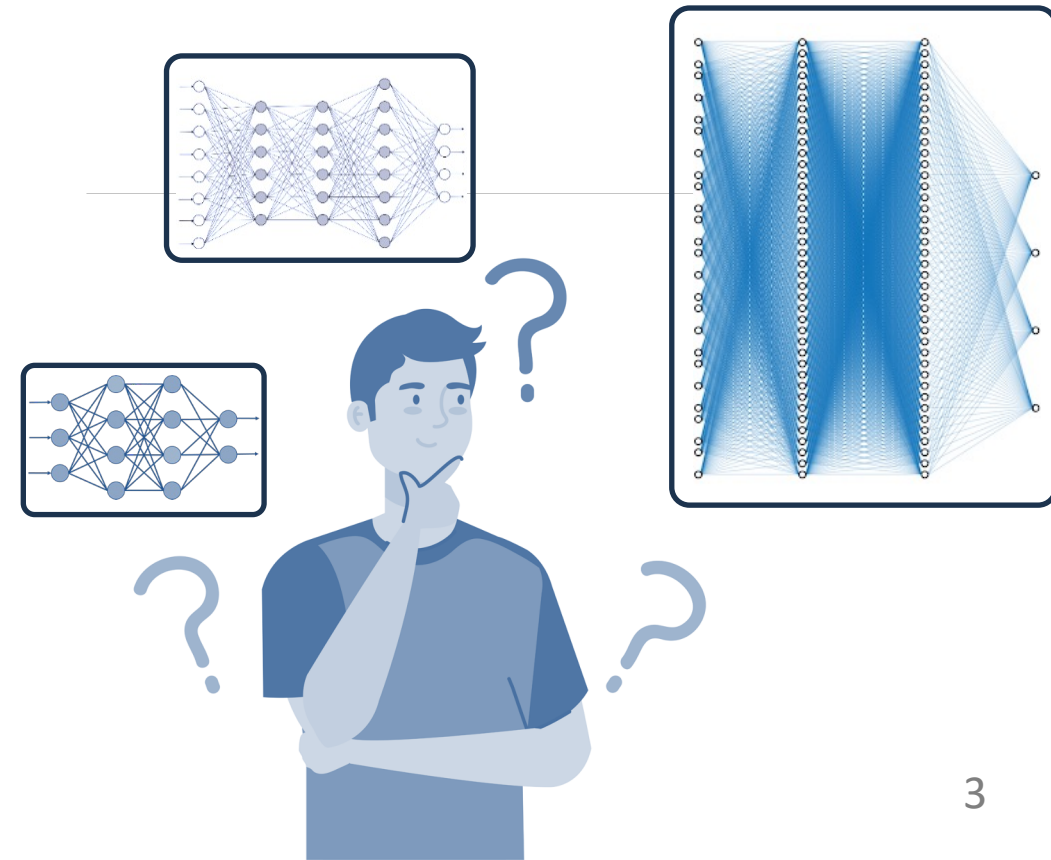


A sample neural network model

Background

Deep Neural Networks – Effective But Obese

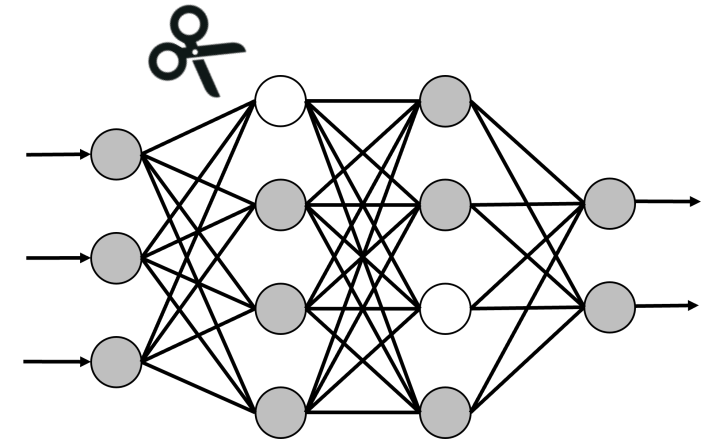
- Deep neural networks achieve favorable accuracy and generalization, however, at the cost of being over-parameterized and computationally intensive.
- Many neural networks for image classification or NLP tasks contain millions or even billions of trainable parameters.
- Deploying neural network models on resource-constraint platforms, e.g., IoT or mobile devices, is challenging.



Background

Pruning

- Neural network pruning technique is extensively used to remove parameters that are redundant or useless (*Han et al., 2015, Hu et al., 2016, Li et al., 2017, Lee et al., 2018, Molchanov et al., 2017, etc*)
- Pruning aims to reduce the model size as well as the demand for computational resources.
- The common practice of existing pruning techniques
 - Performed by the model owner
 - Require post-pruning retraining and/or fine-tuning.
 - Tend to adopt an aggressive and coarse-grained one-shot strategy.
 - Given a chance to fix any mis-pruned neurons.



A brief demonstration of neural network model pruning

Motivations

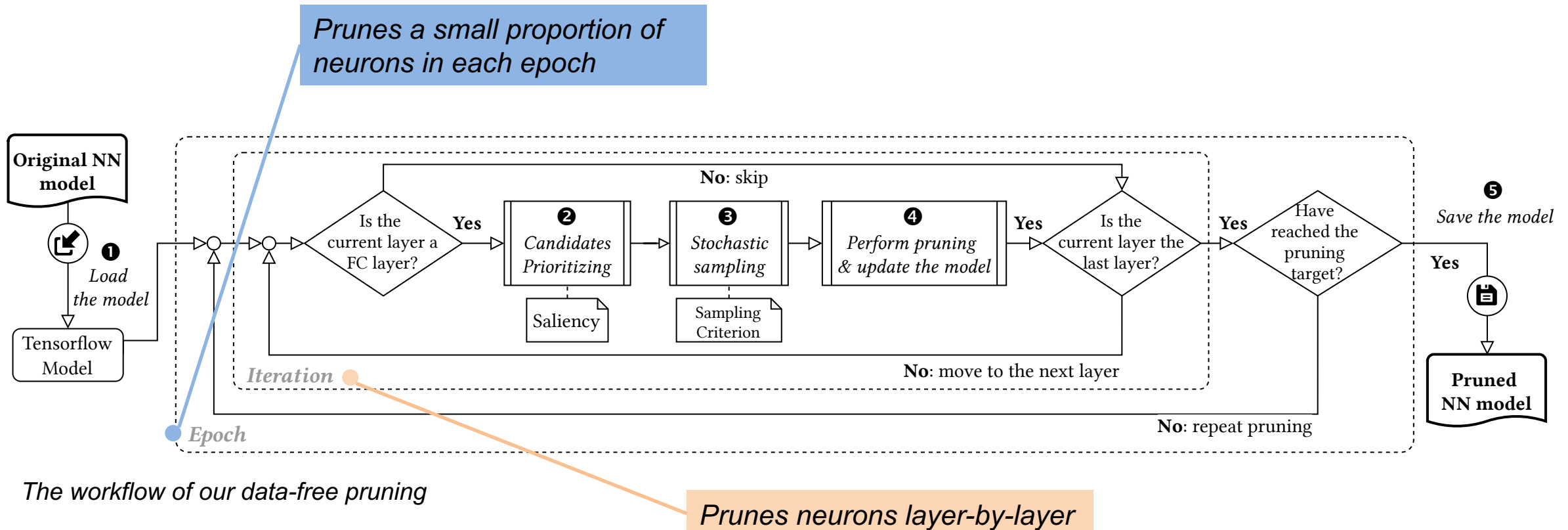
Data-free pruning

- Model pruning can be performed by not only the model owner, but also the model users.
- Model users usually have no access to the original training dataset, especially after the introduction of US CCPA and European GDPR.
- One real-world example is Facebook's DeepFace model*, which is trained with a large number of face images. Users can run the model, but have no access to the training data.

We study the model pruning that ⁽¹⁾does not rely on the training data and therefore, does not require retraining or fine-tuning; and ⁽²⁾preserves the original model fidelity and robustness to the maximum extent.

Our Approach

Approach Overview



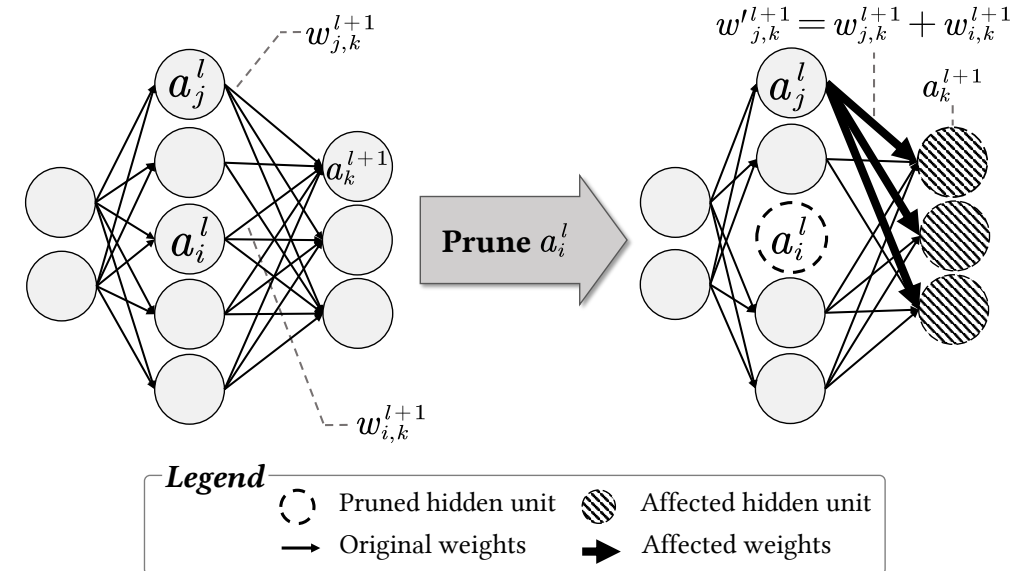
Our Approach

Primitive Pruning Operation

How does our pruning work?

The pair-wise primitive pruning operation:

- Each time we choose a pair of two neurons (aka. hidden units) called **candidates** (written as a_i^l and a_j^l) at the same layer
- One unit is the **nominee** (i.e., a_i^l), and the other unit is the **delegate** (i.e., a_j^l).
- We prune (remove) the nominee, then modify the parameters of the delegate to minimize the pruning impact.



Our Approach

Saliency of Pruning Candidates

Which neurons will be removed?

We use a metric called **saliency** to prioritize the pruning candidates (neurons to be removed).

The saliency describes the **importance** of a neuron (i.e., the nominee, written as a_i^l) when it is to be replaced by another (i.e., the delegate, written as a_j^l) in its layer.

We adopt the saliency measurement from the existing literature (Srinivas & Babu, 2015), which is calculated based on:

- The weight parameters are connected from the previous layer, i.e., $W_{*,i}^l$ and $W_{*,j}^l$.
- The weight parameters connect to the next layer, i.e., $W_{i,*}^{l+1}$ and $W_{j,*}^{l+1}$.
- The bias parameters, i.e., b_i^l and b_j^l .

$$S(a_i^l, a_j^l) = \frac{\sum W_{i,*}^{l+1}}{|W_{i,*}^{l+1}|} \left(\|W_{*,i}^l - W_{*,j}^l\|_2 + \frac{|b_i^l - b_j^l|}{|b_i^l + b_j^l|} \right),$$

where $W_{i,*}^l = \{w_{i,m}^l | a_m^l \in a^{[l]}\},$
 $W_{*,i}^l = \{w_{n,i}^l | a_n^l \in a^{[l-1]}\}$

We remark that merely taking saliency into account is insufficient because it ignores the propagation of pruning impact.

Our Approach

Candidates Prioritizing – Estimating Pruning Impact

Any pruning operation alters the computation of the hidden units in subsequent layers and will eventually propagate to the output layer.

We resort to interval arithmetic to approximate the value range of each neuron in the hidden layer, given legitimate inputs are in bounded values, e.g., $[0,1]$.

Suppose we are pruning a_i^l and a_j^l , its impact of an arbitrary neuron at the next layer, i.e., a_k^{l+1} , can be evaluated as below:

$$\begin{aligned}\Delta a_k^{l+1} &= \left(w_{j,k}^{l+1} + w_{i,k}^{l+1}\right) a_j^l - \left(w_{i,k}^{l+1} a_i^l + w_{j,k}^{l+1} a_j^l\right) \\ &= w_{i,k}^{l+1} (a_j^l - a_i^l)\end{aligned}$$

Accordingly, we manage to estimate the propagated impact of the output layer, written as $\Delta a^{[out]}$, for each pruning candidate.

Our Approach

Candidates Prioritizing – Defining Sampling Criterion

We design our sampling criterion based on an insight that **a small and uniformly distributed impact is less possible to drive the pruned model to misclassify.**

Our proposed criterion is composed of two metrics*:

1) The **scale** of the pruning impact

- L1-norm of the propagated impact at the output layer
- A greater scale means the current pruning operation generates a larger magnitude of impact on the output layer.

2) The **entropy** of the pruning impact

- Adopts Shannon's information entropy to assess the similarity of the propagated impact on each output unit*.
- A greater entropy implies the pruning impact on each output unit shows a lower similarity, making the pruned model more likely to misclassify.

$$NORM \left(\Delta a^{[out]} \right) = \sum_{(u^-, u^+) \in \Delta a^{[out]}} |u^+ - u^-|$$

$$ENT \left(\Delta a^{[out]} \right) = - \sum_{u_i \in \Delta a^{[out]}} \rho_{\phi} (u_i) \cdot \log \rho_{\phi} (u_i)$$

* We use u to indicate the interval of the propagated impact of a specific unit at the output layer.

+ Please refer to our manuscript for more references and technical details.

Our Approach

Guided Stochastic Sampling

We normalize the scale and entropy, and then introduce a pair of parameters (α, β) to integrate the two metrics into a single criterion, namely **energy**^{*}.

Our pruning determines which pruning candidate to be accepted (or otherwise discarded) through a **stochastic sampling** process, guided by the energy.

The motivation of applying a stochastic process is to avoid our sampling being stuck at a local optimal.

$$\begin{aligned} \text{Energy}_{(\alpha, \beta)} \left(\Delta a^{[out]} \right) &= \alpha \cdot \sigma \left(\text{NORM} \left(\Delta a^{[out]} \right) \right) \\ &\quad + \beta \cdot \sigma \left(\text{ENT} \left(\Delta a^{[out]} \right) \right), \\ &\text{subject to } \alpha \geq 0, \beta \geq 0, \beta = 1 - \alpha \end{aligned}$$

^{*} We set $\alpha = 0.75$ and $\beta = 0.25$ while calculating energy for the stochastic sampling in evaluation.

Our Approach

Guided Stochastic Sampling

We adopt the **simulated annealing algorithm** in this work:

- Pruning candidates will be assessed in ascending order of its saliency.
- The pruning impact of each pruning candidate will be evaluated, and the corresponding energy value will be calculated.
 - If the energy is less than the current temperature, (unconditionally) accept the current candidate.
 - Otherwise, the acceptance will be stochastically determined.
- If the current pruning candidate is accepted, its energy value will be set as the temperature (refer to the simulated annealing algorithm) for the next round.
- Stop once it reaches the pruning target (e.g., 50% of neurons have been pruned).

$$P = \min \left(1, \exp \left(-\frac{energy' - energy_{t-1}}{T_{t-1}} \right) \right)$$

Evaluation

RQ1: Fidelity and Robustness Preservation.

How effective is our pruning method in terms of fidelity and robustness preservation? Does our method generalize on diverse neural network models?

Four models trained with different datasets, with number of parameters ranging from 6,145 to 753,866.

Model (num. of param / dataset)	Model Architecture
#1 (6,145 / Kaggle Credit Card)	4-layer MLP
#2 (245,655 / KDD Cup '99)	6-layer MLP
#3 (125,898 / MNIST)	5-layer MLP
#4 (753,866 / CIFAR-10)	13-layer CNN (w/ 3 FC layers)

Experimental Setup

- We adopt FGSM ($\epsilon = 0.01$ and $\epsilon = 0.05^*$) to evaluate the robustness against undesirable inputs (MNIST and CIFAR-10 models only).
- We evaluate the robustness and accuracy of each pruning epoch until 80% pruning.

* We refer to the literature (Goodfellow et al., 2015, Bastani et al., 2016) to find proper values of epsilon for robustness assessment.

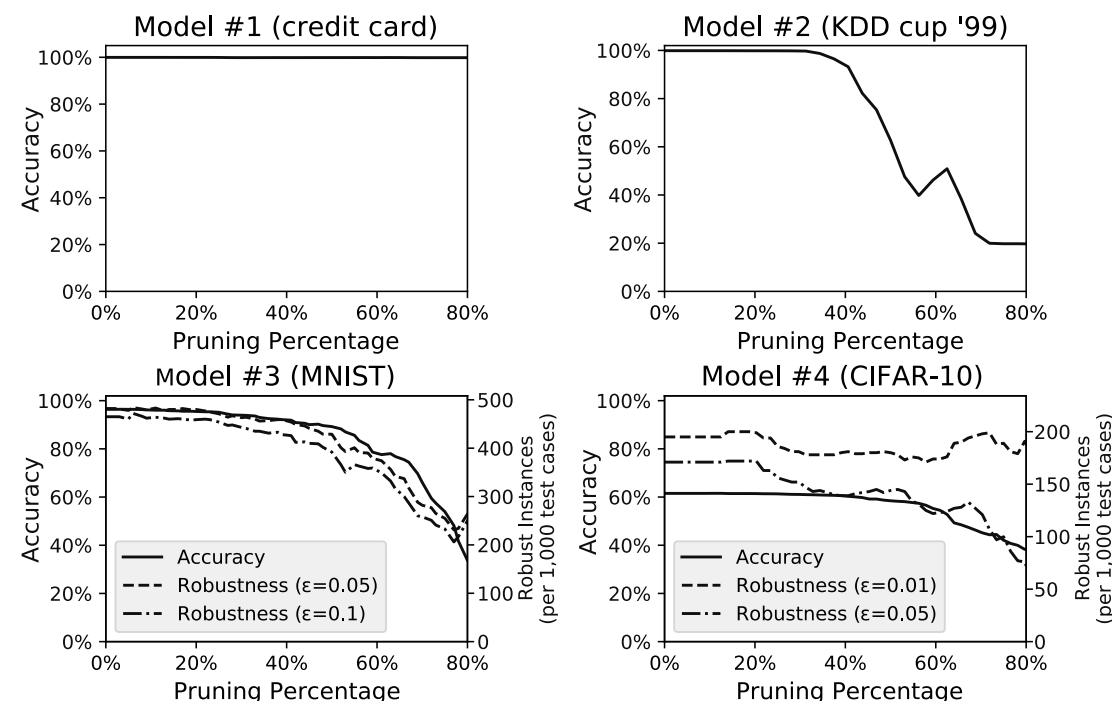
Evaluation

RQ1: Fidelity and Robustness Preservation.

Findings

- Our pruning shows favorable fidelity and robustness preservation against adversarial perturbations.
 - It imposes almost no impact on the binary model (#1).
 - Models trained for more complex classification tasks (#2-4) still preserve 50% of their original accuracy after 50% pruning, and 70% of their original robustness after 50% pruning.
 - Change of robustness generally shares the same trend as test accuracy.
- Our method can generalize on different types of models.

Accuracy and robustness decay of 4 models when applying our pruning method (median observed from 10 rounds of experiments).



Evaluation

RQ2: Pruning Efficiency

RQ2 aims to study if our method could complete the pruning within an acceptable time.

Experiment Setup

- We run it on the four models with a “worst-case setting”, i.e., prune the least proportion of neurons at each step.
- The actual adoption of our approach would be much faster than our experimental scenario.
- All experiments are performed on a PC environment (Intel i7, 64GB RAM).

Model (num. of param / dataset)	Batch size (per layer)	Elapsed time (seconds)
#1 (6,145 / Kaggle Credit Card)	3.13%	30.5
#2 (245,655 / KDD Cup '99)	3.13%	76.7
#3 (125,898 / MNIST)	1.56%	435.4
#4 (753,866 / CIFAR-10)	1.56%	265.3

Evaluation

RQ3: Benchmarking

RQ3 aims to explore whether our method can outperform existing one-shot data-free pruning methods.

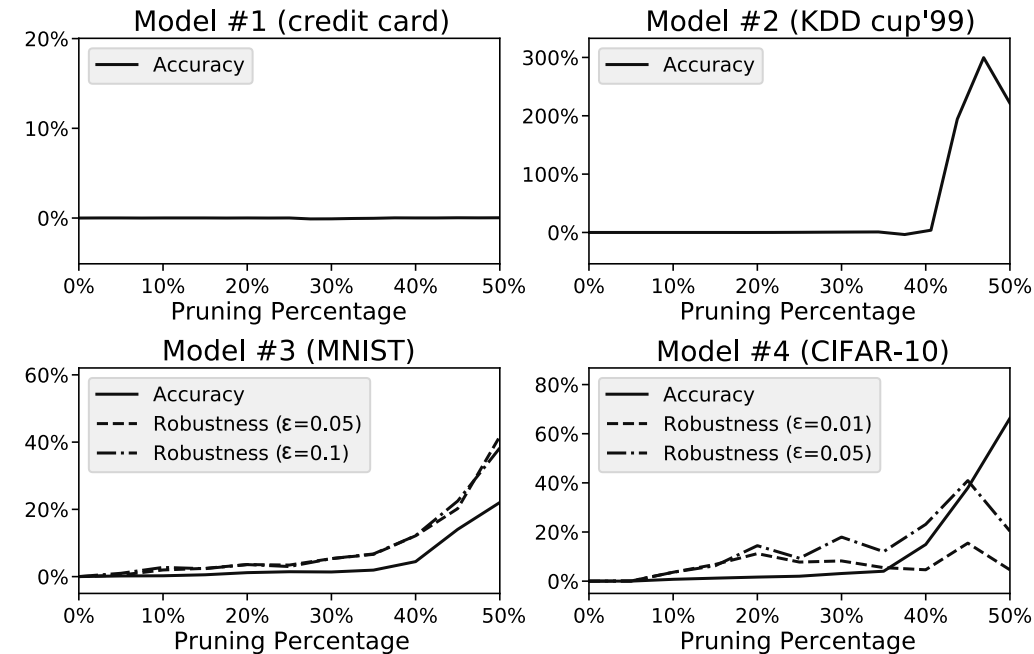
Experimental Setup

- Set the existing one-shot data-free approach as the baseline (*Srinivas and Babu, 2015*).
- Assess both the fidelity and robustness of the pruned models.

Findings

- Both our approach and baseline perform well on the binary model.
- Our approach outperforms the baseline in models #2-4.
 - Achieves ~3x improvement in accuracy in model #2 (observed after 46% pruning).
 - Up to 42% improvement in robustness preservation, and up to 66% improvement in accuracy in models #3-4.
- Overall, the fidelity and robustness preservation of a model is dependent on the utilization of its hidden units.

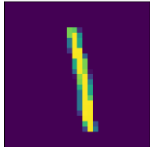
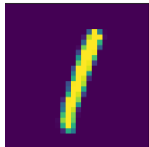


Improvement of our method against the baseline



Evaluation

RQ3: Benchmarking

Demonstration of the robustness of the original and pruned models (misclassification results are shown in bold and italic font)

Model & dataset configurations			MNIST (model #3, $\epsilon=0.05$, $\alpha=0.75$)		CIFAR-10 (model #4, $\epsilon=0.05$, $\alpha=0.75$)	
Test samples						
			#5	#31	#1	#43
Labels			1	1	8 (ship)	6 (frog)
Adversarial evaluation (with confidence)	Original		1 - 100%	1 - 100%	8 (ship) - 100%	6 (frog) - 100%
	Baseline	25% pruned	<i>4 - 100%</i>	<i>3 - 73.80%</i>	<i>1 (automobile) - 100%</i>	<i>3 (cat) - 99.41%</i>
		50% pruned	<i>4 - 100%</i>	1 - 100%	<i>0 (airplane) - 100%</i>	<i>4 (deer) - 98.8%</i>
	Our supervised pruning	25% pruned	1 - 100%	1 - 90.61%	8 (ship) - 100%	6 (frog) - 100%
		50% pruned	1 - 100%	1 - 87.73%	8 (ship) - 100%	<i>3 (cat) - 97.12%</i>

Threat to Validity & Future Work

- **Our method is primarily designed for fully connected components of a neural network model.**
 - Expand the coverage of model pruning (e.g., support of residual blocks)
- **A better accuracy preservation is desired.**
 - Use test set to guide model pruning
- **Our sampling performance heavily relies on interval arithmetic approximation**
 - A more precise interval approximation or refinement technique could be applied to optimize the pruning criteria.

Conclusion

- **A novel data-free neural network pruning**
 - Guided by the simulated pruning impact
 - Without relying on post-pruning retraining and/or fine-tuning
 - Requests no access to the original training data-set
- **An empirical study to explore the model fidelity and robustness preservation**
 - Our approach can substantially reduce model size without a significant compromise in model fidelity and robustness
 - Outperforms the baseline data-free pruning
 - We call for more future endeavors of model optimization to cope with the data-free challenge

Contact

Should you have any question, please feel free to contact us:

Mark Huasong Meng (huasong.meng@u.nus.edu)

Guangdong Bai (g.bai@uq.edu.au)

Sin Gee Teo (teo_sin_gee@i2r.a-star.edu.sg)

Jin Song Dong (dcstdjs@nus.edu.sg)



Reference

- O. Bastani, Y. Ioannou, L. Lampropoulos, D. Vytiniotis, A. Nori, and A. Criminisi, “Measuring neural net robustness with constraints,” in *Advances in Neural Information Processing Systems*, 2016.
- S. Han, J. Pool, J. Tran, and W. Dally, “Learning both weights and connections for efficient neural network,” in *Advances in Neural Information Processing Systems*, 2015, pp. 1135–1143.
- H. Hu, R. Peng, Y.-W. Tai, and C.-K. Tang, “Network trimming: A data-driven neuron pruning approach towards efficient deep architectures,” *arXiv preprint arXiv:1607.03250*, 2016.
- I. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in *International Conference on Learning Representations*, 2015.
- N. Lee, T. Ajanthan, and P. Torr, “SNIP: Single-shot network pruning based on connection sensitivity,” in *International Conference on Learning Representations*, 2018.
- H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, “Pruning filters for efficient convnets,” in *International Conference on Learning Representations*, 2017.
- P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, “Pruning convolutional neural networks for resource efficient inference,” 2017.

Please refer to our manuscript for more references and technical details.