How these robustness characteristics scale

Florian Suess

***Abstract*—With the main work done so far[1] showcasing evidence of some generally favourable robustness characteristics of our ParaLIF activation within LeNet style architectures, it is important to stress that these results are drawn from behaviour on simpler single colour channel datasets MNIST, KMNIST, SVHN and fashionMNIST. In the world of VisionTransformers and the "tried and true" ResNet architectures, we'd like to also enquire how ParaLIF adoption in these more relatable architectures perform within the context of more difficult three channel image datasets like CIFAR-10/100 to get a sense of how these characteristics scale, and at what cost.**

***Index Terms*—ResNet, ViT, ParaLIF, LIF, CIFAR-10, CIFAR-100, Advaserial Robustness, DeepFool Attack, Fast Gradient Sign Method Attack, Square Attack**

I. Introduction

As we have seen, direct training methodologies for S-NN's remain in itś infancy. With common approaches typically requiring backpropogation through time methods that unroll potentially an entire network linearly to the number of steps you use for a spike encoding of an input, this leads us to exceedingly longer training times per dataset. ParaLIF, our focus and variant of the leaky integrate and fire neuron (LIF), addresses this shortcoming of S-NN's by selling us the idea that we can backpropogate through time *in parallel* - but as we've seen this hasn't prevented the difficulty associated to training these networks even on small models (60k parameters).

One could speculate that this issue broadly steps parallel to issues faced by the training of traditional recurrent networks, albeit in the case of S-NN's, we face these issues without the mechanisms such as the LSTM and GRU gradient preserving gates. In spirit of our original research objective, we pivoted early on to *hybrid* models, that is, where typical continuous network features like convolutions could be adopted in front of ParaLIF networks. This is considered a slight concession as pure S-NN models have been shown to have significant potential to exceed in terms of energy efficiency and speed[2]. In writing that, hybrid networks still remain valuable to enquire into though as the potential gains of adversarial robustness of networks remains a separate and important objective of any network.

We have previously in our journey to break our performance boundaries found ANN-SNN conversions effective[3]. We specifically adapted methods published in the literature specific to ResNets. This involved priming techniques (concretely, that is, swapping all ReLU's with "quantised clip-floor shifted" ReLU) such that a recursive activation swap with LIF had a zero conversion error loss. However we deemed this method a bit too restrictive to continue with - predominately because although mapping to LIF based networks was clear, a mapping to ParaLIF activations remained unclear. With the little time we had, we picked out what worked well.

II. Adoption of ParaLIF in ResNet Models

The key idea here is to use a ResNets as an encoder of our input up to the point of exit from the final residual layer, decoding this feature map using a single ParaLIF or LIF feed forward linear layer. We batch min max normalise the feature map values into the range $p \in [0, 1]$. We use this to "rate encode" this over 20 steps using the same Bernoulli distribution method used throughout this research project. Each step assuming 1 with probability p and 0 with probability 1-p. We semi-arbitrarily choose the ResNet models 18 and 50 due to their ubiquitous names as benchmarks.

A. Transfer Learning

To boost learning time we adopt what we learnt previously and adopt conversion tactics. We fine-tune these ResNets onto CIFAR-10/100. Leveraging the speed up boost by transferring pre-trained weights from heavily trained ResNets on the ImageNet. At the time of this enquiry, we had some readily available (and verifiable) performance characteristics on ImageNet of the following;

Model	Accuracy %	Parameters (M)
ResNet-18	69.76	11.7
ResNet-50	80.86	25.6

[1] https://github.com/suessflorian/biological-neurons/

[2] cite

[3] Available here; https://github.com/suessflorian/qcfs/

Fine-tuning via training the full network is prone to destructive learning, especially to those low level feature layers - and so we experimented with freezing early residual layers, found that freezing all parameters pre layer 3 was effective.

We didn't have too much time to experiment much with all the various available meta learning hyper parameters but we found the classic SGD optimization method was sufficient, with a low $\eta = 0.001$, basic L2 weight regularisation with some training speed increases too with introduced momentum during descent (0.9).

We found more training speed gains by stabilising the 4th residual layer using the default ResNet configuration before transferring to ParaLIF and LIF activations - and so we have indeed demonstrated some sort of *double* learning transfer approach despite our issue of very limited compute.

III. Test Performance Measure

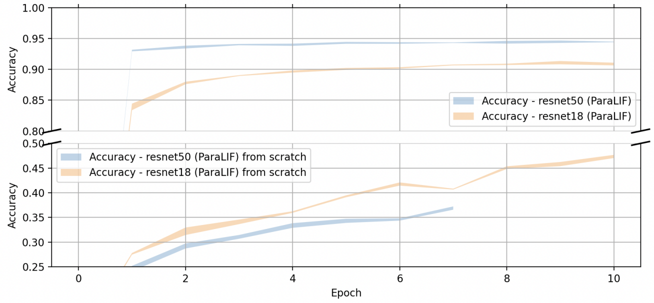

Fig. 1. The double transfer method test accuracy climbing quickly relative to scratch training

With all of this careful planning, on execution - we quickly surpass previous model performance by significant margin. Allowing for deeper enquiry into CIFAR-10 with a comfortable dataset beat allowing us to start paying attention to CIFAR-100. Although both LIF and ParaLIF contain a stochastic element in their inference (the Bernoulli distribution rate encoding method), we found notable variance in inference outcomes given a fixed image on different random seeds - hence we accompany test results with standard deviation.

- All default variants are trained for 10 epochs.
- ParaLIF/LIF variants are trained for 10 epochs, with transferred weights of a fine-tuned 4th residual layer from the above.

FashionMNIST

Variant	ResNet-18	ResNet-50
Default	93.2	93.2
LIF	93.7	93.4
ParaLIF	91.5% ± 0.08%	92.5% ± 0.14%

CIFAR-10

Variant	ResNet-18%	ResNet-50
Default	92.4	94.7
LIF	92.8	94.8
ParaLIF	90.9% ± 0.15%	94.4% ± 0.034%

CIFAR-100

Variant	ResNet-18 %	ResNet-50
Default	73.3	77.9
LIF	74.3	79.4
ParaLIF	34.9% ± 0.08%	52.9% ± 0.14%

IV. Attacks

We proceed from here to start attacking these models with same attacks our earlier endeavours, particularly measuring contrastive performance wrt. each dataset and model.

Fast Gradient Sign Method Attack

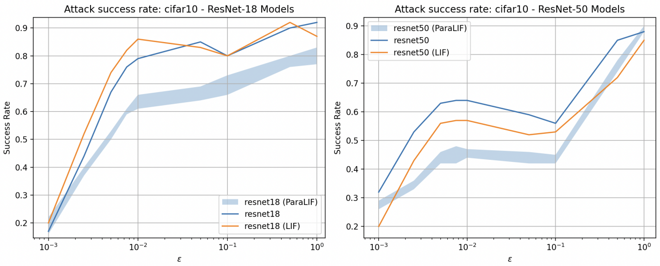

Fig. 2. FGSM attack at various epsilons

DeepFool Attack

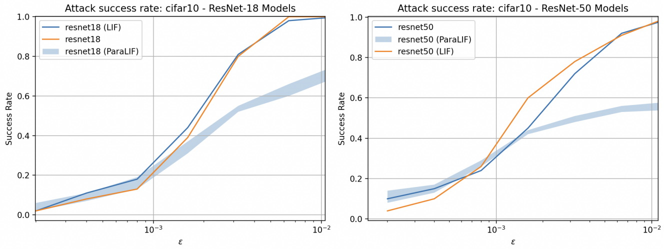

Fig. 3. Deepfool attack at a fixed max iterations of 100 at various epsilons

Square Attack

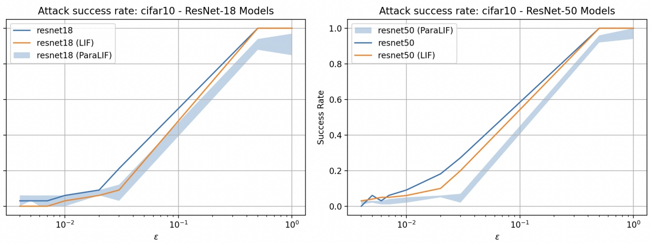

Fig. 4. Square attack at a fixed max iterations of 100 at various epsilons

V. Adoption of ParaLIF in Vision Transformer Models

ResNets as an encoder of our input up to the point of exit from the final residual layer, decoding this feature map using a single ParaLIF or LIF feed forward linear layer. We batch min max normalise the feature map values into the range $p \in [0, 1]$. We use this to ”rate encode” this over 20 steps using the same Bernoulli distribution method used throughout this research project. Each step assuming 1 with probability p and 0 with probability 1-p. We semi-arbitrarily choose the ResNet models 18 and 50 due to their ubiquitous names as benchmarks.

VI. TradeOff

Although we see various robustness characteristics in ParaLIF based networks.... TODO

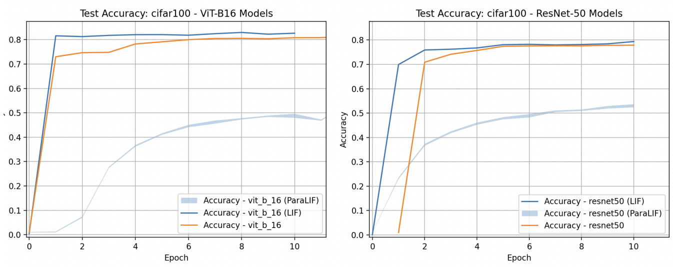

Fig. 5. Transformer and ResNet ParaLIF models struggling to learn on CIFAR-100