COMP6248 REPRODUCIBILITY CHALLENGE: SPIKE-BASED CAUSAL INFERENCE FOR WEIGHT ALIGNMENT

Michael Lawrence
mkl1g19@soton.ac.uk

Esther Barlow-Smith
eabs1u19@soton.ac.uk

Aaron Wing
aw2g19@soton.ac.uk

Implementation can be found in COMP6248-Reproducability-Challenge/SpikeBasedCausalInference

ABSTRACT

In artificial neural networks the process of gradient descent with backpropagation has a feedforward and a feedback stage. Both stages require access to the weights that effect the input stimuli from node to node. Biological neurons behave differently from artificial neurons and may not have the architecture to allow backpropagation. To simulate these gradients in the brain the data must be propagated separately. This paper addresses regression discontinuity design as a solution to resolve the resulting weight transport problem, extending the reach of work done by J. Guerguiev et al.

1 INTRODUCTION

Our reproducibility report is based on the "Spike-based causal inference for weight alignment" paper presented in the ICLR 2020 conference by J. Guerguiev et al. This paper studies the link between artificial and biological neurons utilising gradient descent (Guerguiev et al., 2019). Gradient descent is critical in the two stages of backpropagation in artificial neural networks (ANNs) to converge towards a solution. Biological neurons may not be capable of calculating gradients and therefore are propagated separately. This results in the "weight transport problem", where two sets of weights that ideally should be symmetric cannot be guaranteed to be so. For the brain to approximate gradients, information must be propagated separately, so one set of synaptic weights is used for processing and another is used for backward passes.

Learning to form symmetric weights is more biologically feasible and provides less bias in the gradient estimator, which causes better training outcomes. Learning symmetric weights is a causal inference issue (Lansdell & Kording, 2019) and therefore regression discontinuity design (RDD) can be implemented to estimate causality. RDD applies the discontinuity produced from a threshold to determine the causal effects (Imbens & Lemieux, 2007). This is ideal for applying to spiking neural networks, which interact via discontinuous spikes (Guerguiev et al., 2019). Implementation of a method for learning feedback synaptic weights, demonstrates computational methods more accurately simulating biological neurons.

2 REPRODUCING ORIGINAL RESULTS

A GitHub repository by Guerguiev et al. was publicly available, the code of which was designed for an environment that used a redundant version of multi-threading. This meant the code would not work correctly on our machines, therefore major changes were made, and additional acceleration methods were implemented. A minor change was inspired after directly contacting J. Guerguiev; the construction of the network sizes were made dynamic depending on the dimensions of the data. This was made passable as a parameter in the command line and enabled the ability to run multiple datasets concurrently. This allowed for increased parallelisation and decreased the total run time dramatically. The initial project code also does not organise or plot any data. We have developed code to allow the data to be extracted from files and then preprocessed in order to get a usable graph as is shown above.

The reproduced results demonstrated in Figure 1 are shown to be very comparable to the original paper's results for the CIFAR 10 dataset. The most notable difference for all figures is that the spread of feedback weights may look slightly different, due to differently scaled axes and randomly initialised models and data. It should be noted, that the Akrout algorithm was not implemented, as the main objective of the paper was to show that using RDD was superior.

3 EXPERIMENTATION AND EXTENSION

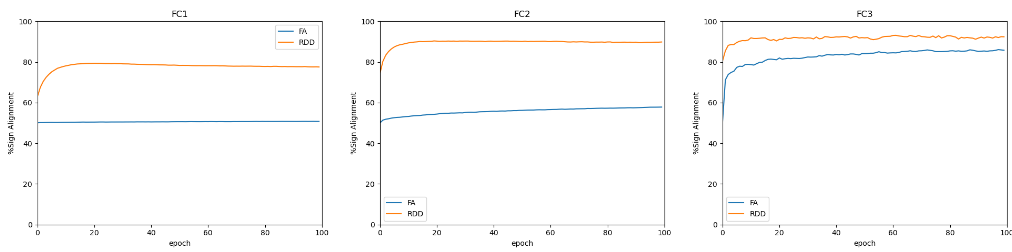
For the first stage of experimentation we wanted to get a large range of data sets to examine the scope of this solution. The data sets used were CIFAR-10, MNIST, MNIST-FASHION, KMNIST, and the USPS data set. There is a large amount of variation on these image datasets, ranging from structured sets of characters to pictures of clothes.

We also wanted to further understand what effect the RDD training time had on the performance of the model so we doubled the RDD pre-training. This increased the training time detrimentally as this was the slowest part of the program and could not be parallelised.

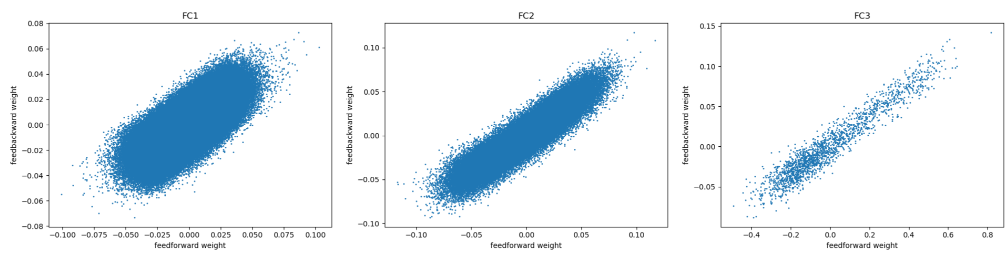
The final part of our experimentation method was to implement a Fuzzy RDD (Bertanha & Imbens, 2019) method to replace the standard RDD. The difference between Sharp RDD (which was originally tested by Guerguiev et al.) and Fuzzy RDD is best explained by the situations in which they are applied. Sharp RDD is used when the treatment (the conditional output) is deterministic (Angrist & Pischke, 2008). This is displayed by the left hand equation below.

$$D_i = \begin{cases} 1, & \text{if } x_i \geq x_0 \\ 0, & \text{if } x_i < x_0 \end{cases} \qquad \mathbf{P}\left[D_i = 1 | x_i\right] = \begin{cases} g_1\left(x_i\right), & \text{if } x_i \geq x_0 \\ g_0\left(x_i\right), & \text{if } x_i < x_0 \end{cases} \tag{1}$$

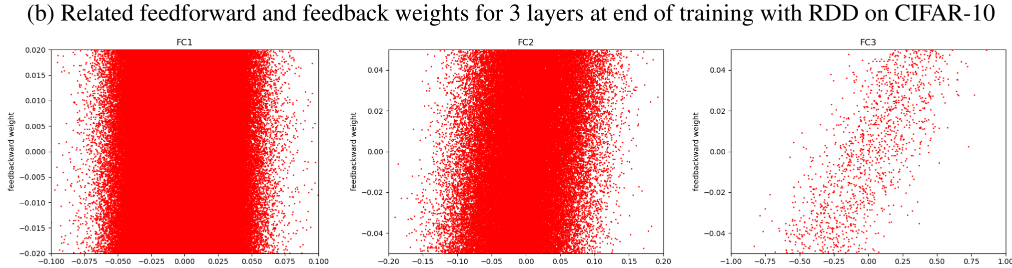
Here D_i is the treatment for input x_i and x_0 is a threshold value. Clearly this design is useful in spiking neural networks as the spike itself is a threshold, and the discontinuity can be interpreted as where neurons just spiked and almost spiked, as resembled in the update RDD estimate function in the code. Fuzzy RDD on the other hand handles the probability of a treatment, as shown



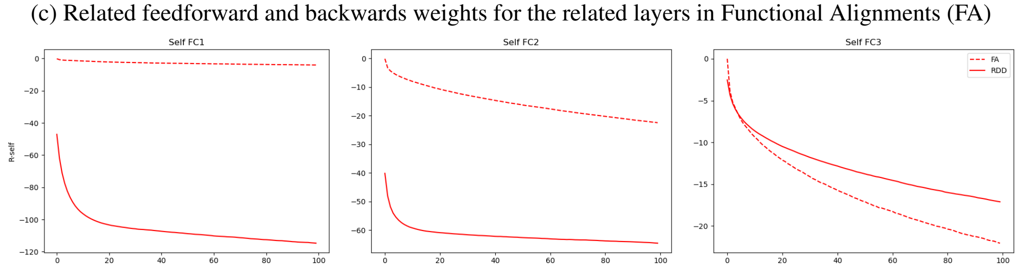
(a) Sign alignment percentage over epoch for 3 layers of the network on CIFAR-10



(b) Related feedforward and feedback weights for 3 layers at end of training with RDD on CIFAR-10



(c) Related feedforward and backwards weights for the related layers in Functional Alignments (FA)



(d) R_{self} per epoch for each of the three fully-connected layers

Figure 1: Reproducing original results

in the right hand equation above. What this means is that instead of defining the output of the treatment as 1 or 0 as RDD was doing, it defines the output as a function, given the probability that the neuron would fire. Bayesian probabilities are already used to calculate the β values that are used to calculate the feedback weights.

So the principle behind implementing fuzzy RDD was that, currently, neurons would reach the threshold and fire. If they didn't reach this threshold they were not counted. This is a detriment to the neurons that did not get above the threshold but got close, their 'effort' would not have been counted. There are many reasons this can be in artificial networks such as incompatibility with a leaky-integrate-and-fire model or refractoriness, but there are many more reasons this could happen in biological neurons which we are currently trying to replicate. By implementing Fuzzy RDD neurons that get close to the threshold have a probability of being counted or not depending on their distance from the cutoff.

4 Results

Re-implementing the initial paper by Guerguiev et al. (2019) produced near identical results as shown in Figure 1. The inclusion of a GitHub repository was incredibly useful as it enabled us to rapidly validate their results. Without which we would have suffered with regards to time on extending the papers results and we wouldn't have been able to test their model as thoroughly.

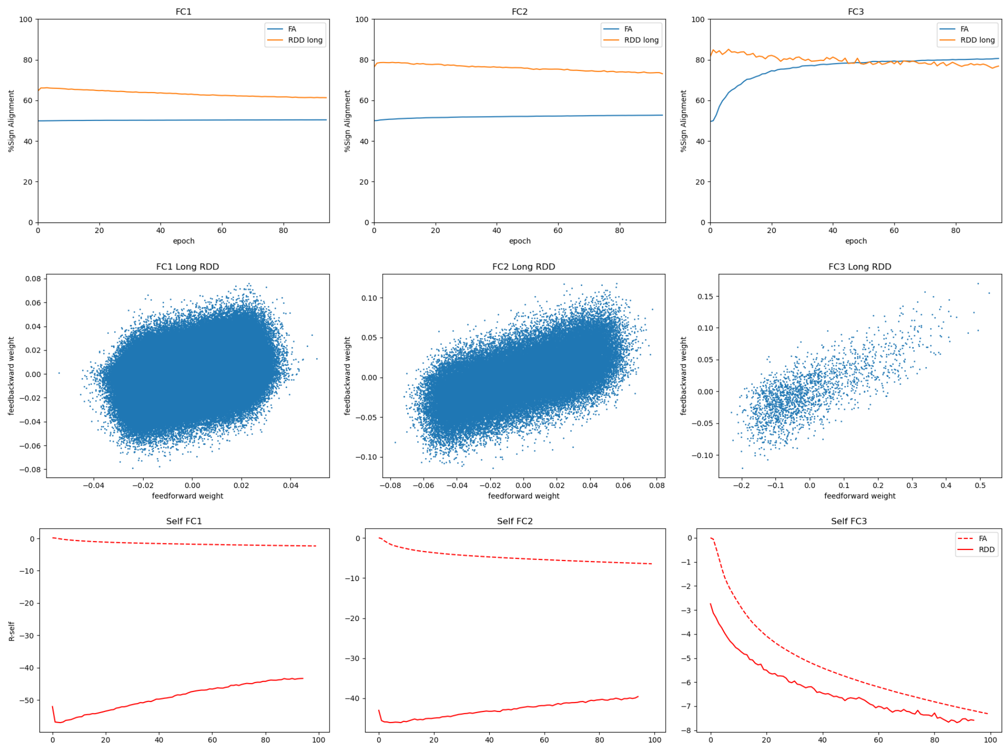


Figure 2: Results of extending RDD time, CIFAR 10

5 Discussion and Conclusion

Testing on multiple datasets showed that RDD successfully aligned weights more effectively than FA regardless of dimensionality or total dataset size. RDD with extended pre-training time was also tested on the CIFAR 10 dataset and the results were unexpected. The accuracy of sign alignment rapidly increased and performed better than FA, but the increased RDD time allowed for a much larger window and the accuracy gradually decreased. This negative effect can be clearly seen in the weight plots in Figure 2. Clearly the correlation is worse than in Figure 1. Interestingly, the overall test error did not perform as badly as expected indicating that, having exactly the same weights is not necessary for accurate computation.

Finally, the results of fuzzy RDD can be seen in Figure 3 clearly performed better. Though the sign alignment percentage only matches FA by the final epoch, the overall effect of fuzzy RDD is clear from the error plot. The model seems to initially perform worse than both standard RDD and the extended pre-training method, but then outperforms both of them with a much faster convergence speed. Fuzzy RDD assumes that distinct continuous functions are present on either side of a threshold, and that the output of marginal cases are decided based on probability. In spiking neural networks, due the fact that a neuron either spikes or does not (1 or 0), the output functions are two flat lines of differing magnitude. Effectively, fuzzy RDD behaves as a

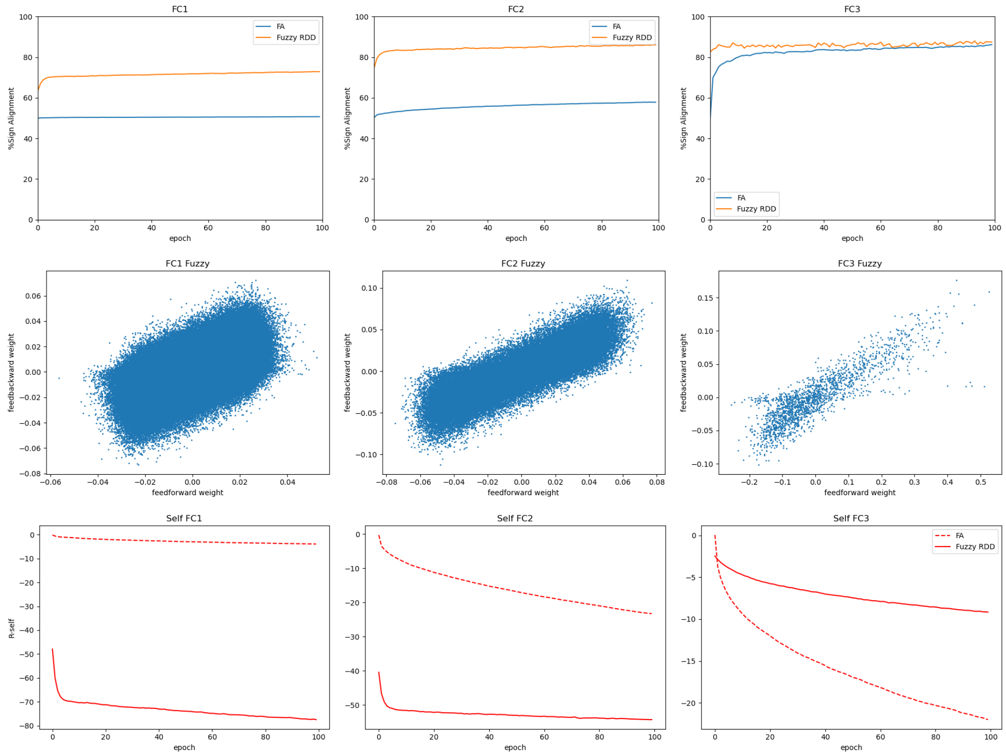


Figure 3: Results of implementing Fuzzy RDD, CIFAR 10

logistic function. One strange effect fuzzy RDD appears to have in the weight plots of the third fully connected layer (FC3) is that there are two clearly discernible lines of correlation. Both lines have positive gradients, suggesting that both are showing correctly aligned weights. As the less steep line has feedback weights around 0, it shows that fuzzy RDD treats marginal cases with more scrutiny to allow for higher accuracy.

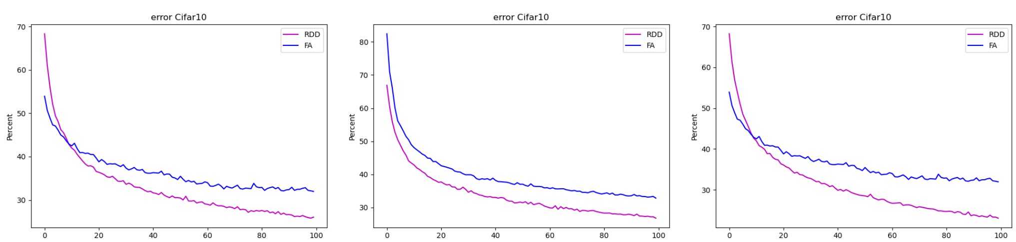


Figure 4: Error plots. **Left:** Reproducing Results from paper. **Middle:** Extending RDD time. **Right:** Implementing Fuzzy RDD

References

Joshua D. Angrist and Jörn-Steffen Pischke. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton University Press, December 2008. ISBN 0691120358.

Marinho Bertanha and Guido W. Imbens. External validity in fuzzy regression discontinuity designs. *Journal of Business & Economic Statistics*, 0(0):1–39, 2019. doi: 10.1080/07350015.2018.1546590. URL https://doi.org/10.1080/07350015.2018.1546590.

Jordan Guerguiev, Konrad P. Kording, and Blake A. Richards. Spike-based causal inference for weight alignment, 2019.

Guido Imbens and Thomas Lemieux. Regression discontinuity designs: A guide to practice. Working Paper 13039, National Bureau of Economic Research, April 2007. URL http://www.nber.org/papers/w13039.

Benjamin James Lansdell and Konrad Paul Kording. Spiking allows neurons to estimate their causal effect. *bioRxiv*, 2019. doi: 10.1101/253351. URL https://www.biorxiv.org/content/early/2019/02/06/253351.