Enforcing idempotency in neural networks

Nikolaj Banke Jensen
Department of Computer Science
University of Oxford
nikolaj.jensen@cs.ox.ac.uk

Jamie Vicary
Department of Computer Science and Technology
University of Cambridge
jamie.vicary@cl.cam.ac.uk

Abstract

In this work we propose a new architecture-agnostic method for training idempotent neural networks. An idempotent operator satisfies $f(x) = f(f(x))$, meaning it can be applied iteratively with no effect beyond the first application. Neural networks used in image generation, sorting algorithms, data compression, and many other areas represent non-linear idempotent projections. Using methods from perturbation theory and linear algebra, we derive the recurrence relation $w' \leftarrow 3w^2 + 2w^3$ and show that in the linear case it has 1) idempotent fixed points, and 2) is attracting around idempotent points. Additionally, we provide experimental results for the non-linear case showing more than 10x idempotent error improvement over the canonical gradient-based approach. Finally, we demonstrate scalability as we train a generative network successfully using only a simple reconstruction loss paired with our method. As a gradient-free optimisation strategy for a non-convex optimisation problem we believe the ideas presented here could be applied more broadly beyond idempotent neural networks.

1 Introduction

Introduce the idea of idempotent neural networks; give the definition. Justification for usefulness in applications where solutions can be both idempotent and not, and where the idempotent solution may be beneficial. Give a brief overview of the rest of the paper.

2 Method

Outline the perturbation analysis which yielded the update rule. Jordan Normal form justification for looking at eigenvalues and for fixed-points. Stability analysis (and nice Julia heatmap) to argue that specific fixed-points are reached. Compare ordinary and modified autodiff by looking at derivative on a single-layer, non-bias network.

3 Experimental Results

On experimental networks we give line graphs comparing absolute error at varying learning rates. We also give line graphs for LRs over many epochs. Takeaway message is that for deeper/wider networks we outperform ordinary autodiff by a lot.

We replicate the results of IGN on MNIST and CelebA. Latent space analysis. Takeaway is that we show application to a U-net GAN architecture based on Conv layers.

Preprint. Under review.

4 Related Work

Review Idempotent Generative Networks, contrasting our work on a gradient-free approach. Have others applied perturbation theory to ML? We suffer same problems as GANs: mode collapse.

5 Conclusion

Give a conclusion of central idea: the use of perturbation analysis to find an iterator which we have successfully applied to a range of toy examples and GAN scenarios.