Enforcing Idempotency in Neural Networks

Nikolaj Banke Jensen [1] **Jamie Vicary** [2]

[1]University of Oxford [2]University of Cambridge

Introduction

We propose a new architecture-agnostic method for training idempotent neural networks. An operation $f_{\boldsymbol{\theta}} : X \to X$ is idempotent if it can be applied multiple times with no effect beyond the first application.

$$f_{\boldsymbol{\theta}}(\mathbf{x}) = f_{\boldsymbol{\theta}}(f_{\boldsymbol{\theta}}(\mathbf{x}))$$

Some neural networks used in data transformation tasks, such as image generation and augmentation, can represent non-linear idempotent projections. Training for idempotency using *e.g.* MSE-loss can lead to poor performance (Figure 5). Using methods from Perturbation Theory, we derive a training scheme that does not rely on gradients of a loss function to operate whilst yielding better reduction in idempotent error than the MSE baseline.

$$\mathcal{L}_{\text{idem}}(\mathbf{x}) = \frac{1}{m}\sum_{i=1}^{m}\left(f_{\boldsymbol{\theta}}(f_{\boldsymbol{\theta}}(\mathbf{x})) - f_{\boldsymbol{\theta}}(\mathbf{x})\right)^2$$

An Idea from Perturbation Theory

Near-idempotent to order n**.** *Let the matrix* $\mathbf{P} \in \mathbb{R}^{m\times m}$ *satisfy* $\mathbf{P} = \mathbf{P}^2$*. Let* $\mathbf{D} \in \mathbb{R}^{m\times m}$ *be arbitrary (e.g., noise) where there exists some* $n \in \mathbb{N}$ *such that* $\mathbf{D}^{n+1}$ *has coefficients with absolute value below* $\epsilon \ll 1$*. We say that* $\mathbf{K} = \mathbf{P} + \mathbf{D}$ *is* ***near-idempotent to order*** n*.*

If $\mathbf{K}$ is near idempotent to order 1, then up to some power j we define the ansatz

$$\mathbf{K}' = \alpha_1\mathbf{K} + \alpha_2\mathbf{K}^2 + \cdots + \alpha_j\mathbf{K}^j$$

If we demand $\mathbf{K}'$ satisfies $(\mathbf{K}')^2 - \mathbf{K}' = \mathbf{0}$ (*i.e.*, it is idempotent), then we get a set of polynomial equations. There is often no exact solution to these, but we find approximate solutions by recursively applying the following assumptions for all $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ matrices:

$$\mathbf{D}^2 \approx \mathbf{0}, \quad \mathbf{P}^2 = \mathbf{P}, \quad \mathbf{XDYDZ} \approx \mathbf{0}.$$

This reduces the number of terms by an exponential factor. For low j the system can now be tractably solved for α_i, parameterizing a mapping g such that $\mathbf{K}' = g(\mathbf{K})$. In other words, g maps near idempotent matrices to perfectly idempotent matrices.

The Solution and Its Properties

For $j \leq 2$ there are no solutions to the above problem. For $j = 3$ there is a single solution,

$$g(\mathbf{K}) = 3\mathbf{K}^2 - 2\mathbf{K}^3,$$

whilst for $j > 3$ there are families of solutions. We consider the solution g when $j = 3$, and show that taking γ-sized steps in direction of $g(\mathbf{K})$ for $0 < \gamma \leq 1$, gives a recurrence relation with nice properties:

- All idempotent matrices are solutions,
- *Only* idempotent matrices are attracting points,
- Wide area of attraction around idempotent points.

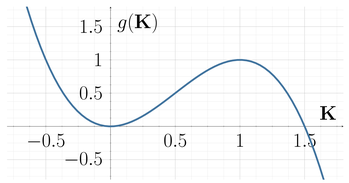

Figure 1. Plot of $g(\mathbf{K}) = 3\mathbf{K}^2 - 2\mathbf{K}^3$ in the case $\mathbf{K}$ is scalar.

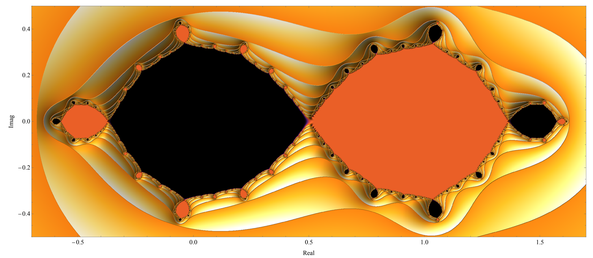

Figure 2. 10-time recursive application of $h(\lambda) = 3\lambda^2 - 2\lambda^3$ for each point on the complex plane. Black areas denote points converging onto 0, while orange areas denote points converging onto 1.

Deriving a Training Scheme

Consider a general network $f_{\boldsymbol{\theta}}$ and its application to input $\mathbf{y} = f_{\boldsymbol{\theta}}(\mathbf{x})$, then our solution becomes:

$$\mathbf{y}' = 3f_{\boldsymbol{\theta}}(\mathbf{y}) - 2f_{\boldsymbol{\theta}}(f_{\boldsymbol{\theta}}(\mathbf{y}))$$

This describes a desired change in the output of the network which we denote $\Delta f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{y}' - \mathbf{y}$. In other words, $\Delta f_{\boldsymbol{\theta}}(\mathbf{x})$ describes the change in $\mathbf{y}$ which moves $\mathbf{y}$ towards an idempotent projection. We therefore define

$$\frac{\partial(-\mathcal{L}_{\text{idem}}(\mathbf{y}))}{\partial \mathbf{y}} \equiv \Delta f_{\boldsymbol{\theta}}(\mathbf{x})$$

throughout the computational graph. We call ordinary backpropagation with this change "**Modified Backpropagation**". Modified Backpropagation has the same asymptotic computational cost as Ordinary Backpropagation, but does not require a backwards pass.

Optimization Trajectories

Preliminary results show that Modified Backpropagation explores the loss landscape significantly differently compared to Ordinary Backpropagation.

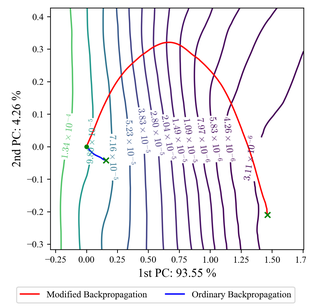

Figure 3. Optimizer trajectory of Modified Backpropagation deviates significantly from that of Ordinary Backpropagation.

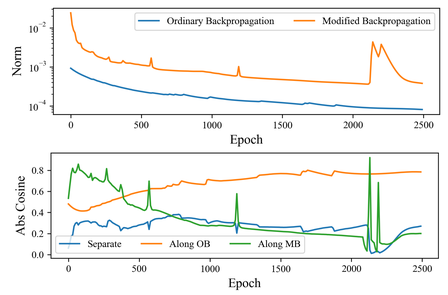

Figure 4. **Top**: Norm of gradients. Modified Backpropagation gives stronger gradient signal than Ordinary Backpropagation. **Bottom**: Absolute cosine similarity of gradients. "Along OB" trains with Ordinary Backpropagation and compares with suggested gradient from Modified Backpropagation. "Along MB" is similar. Gradients suggested by Modified Backpropagation remain significantly different from those suggested by Ordinary Backpropagation.

Improved Error Reduction

We train a variety of MLP architectures with both algorithms on random samples drawn from noise distributions. Across wide/narrow and deep/shallow configurations, Modified Backpropagation outperforms Ordinary Backpropagation at roughly the same computational cost.

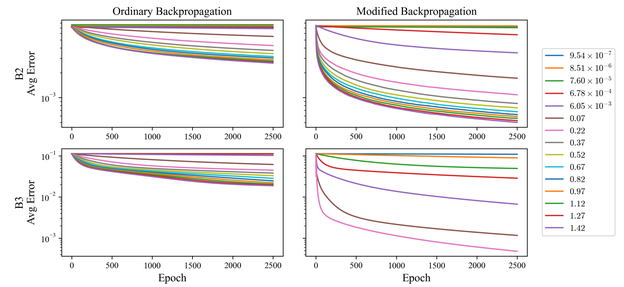

Figure 5. Average absolute idempotent error across learning rates for each algorithm. Modified Backpropagation achieves lower idempotent error at lower learning rates than Ordinary Backpropagation. The biggest relative improvement between algorithms occurs in the first ∼500 epochs.

Use in Generative Networks

We replicate the results of Shocher et al. 2023 and train a DCGAN in a U-net configuration ($G(D(\mathbf{x})) = \mathbf{y}$) on MNIST and CelebA datasets. The loss function has a reconstruction objective and an idempotent objective trained by Modified Backpropagation. We observe comparable behaviour in correction of visual artefacts from first to second application of the network and out-of-distribution mapping. Although results are inferior to SOTA, with more careful hyperparameter tuning we believe these could improve significantly.

Figure 6. Generations of the U-net style DCGAN model trained on MNIST and CelebA with Modified Backpropagation for optimizing idempotent and tightness losses.

References

Assaf Shocher, Amil Dravid, Yossi Gandelsman, Inbar Mosseri, Michael Rubinstein, and Alexei A. Efros. Idempotent Generative Network. In *The Twelfth International Conference on Learning Representations*, 2023. URL `https://arxiv.org/abs/2311.01462`.

nikolaj.jensen@cs.ox.ac.uk