(Almost) Smooth Sailing: Towards Numerical Stability of Neural Networks Through Differentiable Regularization of the Condition Number

Rossen Nenov [* 1 2]   Daniel Haider [* 1]   Peter Balazs [1]

Abstract

Maintaining numerical stability in machine learning models is crucial for their reliability and performance. One approach to maintain stability of a network layer is to integrate the condition number of the weight matrix as a regularizing term into the optimization algorithm. However, due to its discontinuous nature and lack of differentiability the condition number is not suitable for a gradient descent approach. This paper introduces a novel regularizer that is provably differentiable almost everywhere and promotes matrices with low condition numbers. In particular, we derive a formula for the gradient of this regularizer which can be easily implemented and integrated into existing optimization algorithms. We show the advantages of this approach for noisy classification and denoising of MNIST images.

1. Introduction

Numerical stability in neural networks refers to the sensitivity of model predictions and training dynamics to small perturbations in the input data, model parameters, or other computational operations. Good stability properties offer significant benefits, such as consistency enhancement and robustness in predictions, thereby improving generalization and interpretability [1]. Instabilities can arise due to several factors, such as the choice of activation functions, initialization, optimization hyperparameters, and quantization effects during training and inference. Several regularization methods have been developed to intercept this, e.g. dropout, lasso, randomness, etc. [2]–[4]. Differentiability is essential

---

*Equal contribution [1]Acoustics Research Institute, Austrian Academy of Sciences, Vienna, Austria [2]Department of Mathematics, University of Vienna, Vienna, Austria. Correspondence to: Rossen Nenov <rossen.nenov@oeaw.ac.at>, Daniel Haider <daniel.haider@oeaw.ac.at>.
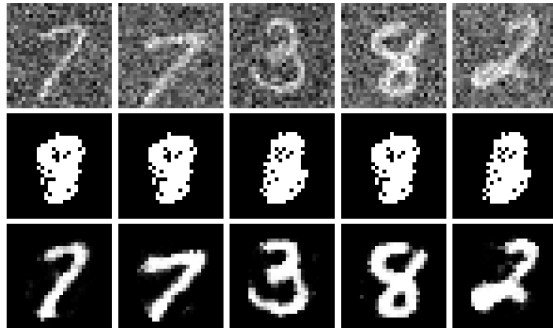
Figure 1. Results of MNIST denoising with autoencoders. Top: MNIST images with added Gaussian noise. Mid: No regularization. Bottom: Proposed regularization. While the vanilla autoencoder struggles significantly, the regularized one performs well.

here as it enables the incorporation of such methods into gradient-based optimization algorithms, hence allowing to gradually regularize the numerical stability during training.

In this paper, we focus on the numerical stability of a neural network by means of their weight matrices, and how to maintain it through regularization. In numerical linear algebra, a way to measure the numerical stability of a matrix S is via its condition number. For invertible matrices S it is defined as $\|S\|_2 \|S^{-1}\|_2$, where $\|\cdot\|_2$ is the matrix operator norm induced by the Euclidean norm [5]. If S is non-invertible or non-square, alternative definitions of a condition number have been proposed. In the context of least squares minimization a particularly common one is

$$\kappa(S) := \|S\|_2 \|S^{\dagger}\|_2, \tag{1.1}$$

where $S^{\dagger}$ is the Moore-Penrose inverse, or pseudoinverse [6]. In this sense, a matrix is considered optimally conditioned if $\kappa(S) = 1$. Existing approaches to promote this kind of stability are based on adaptive optimization techniques [7], iterative projections [8], or adding κ directly to the learning objective as a regularizing term [9]. In the latter approach, minimizing κ via gradient-based optimization has been shown to work well. However, it should be noted that the function $\kappa : \mathbb{R}^{n \times m} \to \mathbb{R}$ is discontinuous and, therefore, not differentiable.

To address this issue, this paper introduces an alternative quantity as regularizer that achieves an optimal condition number, guarantees full rank, and is differentiable almost everywhere.

This manuscript is structured as follows. After this introduction we introduce the proposed regularizer and prove that its minimization is equivalent to minimizing the condition number κ. Subsequently, we prove that the regularizer is differentiable almost everywhere and derive the (sub-)differential of it in Section 3. Finally, in Section 4 we demonstrate the benefits in numerical experiments.

2. Matrix Regularization

Let $S \in \mathbb{R}^{n \times m}$ be a matrix and let $\nu = \min\{n, m\}$. Define $\sigma(S) : \mathbb{R}^{n \times m} \to \mathbb{R}^{\nu}$ as the function that maps S to its singular values in decreasing order. Let $\sigma_{\max}(S) := \sigma_1(S)$ and $\sigma_{\min>0}(S) := \min_{i \in \{1,...,\nu\}}\{\sigma_i(S) \mid \sigma_i(S) > 0\}$ denote the largest and smallest non-zero singular values of S, respectively. The rank of a matrix is the number of its non-zero singular values.

We will denote the singular value decomposition of a matrix $S \in \mathbb{R}^{n \times m}$ as $S = U(\text{Diag}(\sigma(S)))V^T$, where $U \in \mathbb{R}^{n \times n}$ and $V \in \mathbb{R}^{m \times m}$ are unitary matrices and $\text{Diag}(\sigma(S)) \in \mathbb{R}^{n \times m}$ denotes a rectangular diagonal matrix with $\text{Diag}(\sigma(S))_{i,i} = (\sigma(S))_i$.

Throughout this paper we will use the notion of the condition number $\kappa(S)$ of a matrix S as defined in (1.1). This quantity determines the numerical stability of S by indicating how much the output Sx can change in response to small changes in the input vector x. It is known that $\|S\|_2 = \sigma_{\max}(S)$ and that $\|S^\dagger\|_2 = \sigma_{\min>0}(S)^{-1}$ [5]. This means minimizing $\kappa(S)$ corresponds to minimizing the ratio $\frac{\sigma_{\max}(S)}{\sigma_{\min>0}(S)} \geq 1$. While this quantity is perfectly well-defined for arbitrary non-zero matrices, it has drawbacks.

Clearly, $\kappa(S) = 1$ is equivalent to the situation where all non-zero singular values of S are equal. However, there is no guarantee or control over the amount of non-zero singular values. For example, matrices with only one non-zero singular value trivially attain the minimal condition number κ. Such matrices are not very useful as one is usually interested in full-rank solutions. Furthermore, the mapping $\kappa : \mathbb{R}^{n \times m} \to \mathbb{R}$ is discontinuous whenever a singular value approaches zero (Appendix A), and therefore does not allow a proper definition of a gradient. To circumvent both mentioned issues, we propose a different quantity instead:

$$r(S) := \frac{1}{2}\|S\|_2^2 - \frac{1}{2\nu}\|S\|_F^2, \quad (2.1)$$

where $\|\cdot\|_F$ denotes the Frobenius-norm for matrices. In the following theorem we show that matrices that minimize r also minimize κ and additionally have full rank, both desirable features for numerically stable matrices.

Theorem 2.1. *For any $S \in \mathbb{R}^{n \times m}$ the regularizer $r(S)$ defined in Eq. 2.1 is non-negative. If $S \neq 0$, then $r(S) = 0$ if and only if S has full rank and $\kappa(S) = 1$.*

Proof. Since $\|S\|_F^2 = \sum_{i=1}^{\nu} \sigma_i^2(S)$ [5], we observe that

$$\|S\|_2^2 - \frac{1}{\nu}\|S\|_F^2 = \sigma_{\max}^2(S) - \frac{1}{\nu}\sum_{i=1}^{\nu}\sigma_i^2(S) \geq 0, \quad (2.2)$$

as $\sigma_{\max}(S)$ is the largest singular value and, hence, the difference between $\sigma_{\max}^2(S)$ and the average value of the squared singular values of S is always non-negative, thus $r(S) \geq 0$. It is straightforward to see that (2.2) attains 0 if and only if S has ν singular values that are all equal. Since for full rank matrices S all ν singular values are non-zero, $\kappa(S) = 1$ is equivalent to $\sigma_{\max}(S) \equiv \sigma_i(S) > 0$ for all $i \in \{1,...,\nu\}$, which concludes the result. $\square$

As noticed before, the condition number is discontinuous and approaches $+\infty$ if a singular value is approaching zero. Thus, finding a tight connection between the regularizer r and the condition number κ becomes challenging. The following theorem provides a relationship between the regularizer and the condition number, capturing the divergent behavior of the condition number, whenever $\sigma_{\min>0}(S)$ vanishes. A proof can be found in Appendix C.

Theorem 2.2. *For $S \in \mathbb{R}^{n \times m}$ it holds that*

$$\kappa(S) \leq e^{\nu \sigma_{\min>0}(S)^{-2} r(S)}.$$

This theorem shows that for small $r(S)$, the condition number remains small as long as $\sigma_{\min>0}(S)$ is bounded away from zero. This is important as in practical scenarios, achieving the absolute minimum of regularizers is rarely possible.

We note the close relation to the well-known Tikhonov regularization [10], also known as L_2-regularization or Ridge regression [11], where $\|S\|_F^2$ is used as a regularizer. While it is effective in improving stability of the problem formulation [12], it does not directly address the issue of maintaining a low condition number of the solution, which is crucial for numerical stability.

3. Differential Calculus

In this chapter we derive the (sub)-differential properties and formula of the proposed regularizer r in Eq. (2.1).

First, note that the function $r : \mathbb{R}^{n \times m} \to \mathbb{R}$ is the difference of two convex functions and therefore non-convex. We will show that it is differentiable *almost everywhere* and its subdifferential exists everywhere. For a function $f : \mathbb{R}^n \to \mathbb{R}$ the subdifferential at a point $\bar{x} \in \mathbb{R}^n$ is simply denoted by $\partial f(\bar{x})$. In this work, we use the well-known Mordukhovich subdifferential ∂_M, also known as the limiting subdifferential (Appendix B).

Intuitively, this subdifferential generalizes the concept of subgradients to non-convex non-smooth functions, capturing the behavior at points where they are not differentiable by considering limits of gradients of nearby smooth points. The Mordukhovich subdifferential serves as a unifying framework for subdifferentials, accommodating both smooth and non-smooth, convex and non-convex functions. For convex functions it coincides with the convex subdifferential, and for smooth functions it contains only the gradient (Theorems B.6 and B.7). For this reason and to enhance clarity throughout the main part of the paper, we will use the notation ∂ for all subdifferentials at points, where the gradient is not unique.

In the following theorems we derive the (sub-)differential for the regularizer r, using classical results from convex analysis and the work of A.S. Lewis on the differentiability of univariate matrix functions [13]. The detailed definitions and theorems used are listed in the Appendix B.

Theorem 3.1. *Let $S \in \mathbb{R}^{n \times m}$ with a singular value decomposition given by $S = U(Diag(\sigma(S))V^T$. Let u_i and v_i denote the respective column vectors of U and V.*

If $\sigma_1(S) > \sigma_i(S)$ for $i > 1$, then r is differentiable at S:

$$\nabla r(S) = \sigma_{\max}(S)u_1 v_1^T - \frac{1}{\nu}S.$$

Otherwise, the subdifferential of r at S is given by:

$$\partial r(S) = conv\{\sigma_i(S)u_i v_i^T \mid i : \sigma_i(S) = \sigma_{\max}(S)\} - \frac{1}{\nu}S.$$

The following results underscores the practicality of our regularizer. It shows that applying gradient descent steps with respect to our regularizer decreases the condition number of the updated matrix. A proof is found in Appendix C.

Theorem 3.2. *Assume that for $S \in \mathbb{R}^{n \times m}$ the largest singular value is unique, i.e. $\sigma_1(S) > \sigma_i(S)$ for $i \in \{2, \ldots, \nu\}$. Then there exists a $L \in (0,1]$, s.t. for $S' = S - \lambda\nabla r(S)$ with $\lambda \in (0, L]$ it holds*

$$\kappa(S') < \kappa(S).$$

Remark 3.1. For a weight matrix at random initialization, r is differentiable almost surely. This follows from the fact that the space of real symmetric matrices with at least one repeated eigenvalue has codimension 2 in the space of all real symmetric matrices [14]. Consequently, for a matrix S whose entries are i.i.d. samples from a typical continuous distribution, such as Gaussian or uniform, the matrix $S^T S$ has distinct eigenvalues with probability one. Therefore, S almost surely has distinct singular values.

4. Numerical Experiments

We demonstrate the functionality and the benefits of the proposed regularizer through a series of numerical experiments,
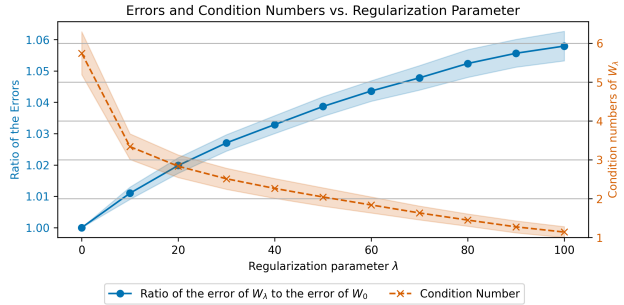

Figure 2. Results of least-squares minimization of (4.1) after 10^5 iterations for different regularization parameter λ values

which are intentionally kept basic to illustrate the core concepts. A comparative analysis with Tikhonov regularization of the results for the experiments is provided in Appendix D. A Python implementation can be accessed via the following link: github.com/danedane-haider/Almost-Smooth-Sailing.

4.1. Basic Functionality

Introducing regularization inevitably leads to a trade-off between the main optimization goal and achieving the desired regularization. To illustrate the impact of our regularizer, let us consider the matrix least-squares problem

$$\min_{W \in \mathbb{R}^{n \times m}} \|WX - Y\|_F^2 + \lambda \cdot r(W), \qquad (4.1)$$

where $X \in \mathbb{R}^{m \times d}$ and $Y \in \mathbb{R}^{n \times d}$ are fixed. The parameter $\lambda \geq 0$ is our regularization parameter, which controls the trade-off between the objective and the regularization.

For the experiment we choose $n = 20$, $m = 50$ and $d = 100$, and let X and Y be Gaussian random matrices. We employ gradient descent with the gradient formulas provided in Theorem 3.1 for 10^5 steps, compare the results for different values of λ and denote the resulting matrices by W_λ. We repeat this experiment 10 times for each $\lambda \in \{0, 10, \ldots, 100\}$ value.

In Figure 2 we plot the ratio of the (approximation) errors $\frac{\|W_\lambda X - Y\|_F}{\|W_0 X - Y\|_F}$ and the condition number $\kappa(W_\lambda)$. We observe that increasing λ leads to an slight increase in the error and a sharp decrease in the condition number, as expected. For large values of λ the condition number becomes nearly optimal, while the error increases by less than 6% compared to the non-regularized case.

In the following experiments we integrate the regularizer into the training of a neural network on MNIST [15], demonstrating the benefits of a well-conditioned model in the presence of noise.

λ	$\kappa(W_1)$	SNR ∞	SNR 1	SNR 0.5
0	43.12	**98.42 %**	**93.80 %**	71.91 %
10^{-3}	9.43	**98.38 %**	91.72 %	62.51 %
10^{-2}	4.62	98.11 %	91.95 %	63.61 %
10^{-1}	**1.45**	96.77 %	**93.68 %**	**74.27 %**
1	**1.53**	96.50 %	92.61 %	**84.25 %**

Table 1. The table presents the condition numbers of the first network layer ($\kappa(W_1)$) and the classification accuracy on the test set with different SNRs: ∞ (no noise), 1, and 0.5. These results are compared for different values of the hyperparameter λ. The more noise is present (low SNR), the more effective the regularizer is.

4.2. Noisy MNIST Classification

For a proof-of-concept approach, we choose the classifier model Φ to be a small neural network with two dense layers, the first one with sigmoid activation, the second one with soft-max. To stabilize the weight matrix of the first layer, $W_1 \in \mathbb{R}^{2048 \times 784}$, we perform empirical risk minimization (ERM) using a regularized cross-entropy loss

$$\mathcal{L}(x; \Phi) = -\sum_{i=1}^{10} y[i] - \log(\Phi(x)[i]) + \lambda \cdot r(W_1), \quad (4.2)$$

where $y[i]$ and $\Phi(x)[i]$ are i-th components of the target and predicted label vector, respectively. Optimization is done for 50 epochs using Adam with a learning rate of 0.0001.

Table 1 shows the performances of the same model, trained with different values of λ, and tested on unseen data with added Gaussian noise for different signal-to-noise ratios: $\infty, 1$, and 0.5. In the absence of noise (SNR ∞) one can clearly see the influence of the regularizer on the classification performance in terms of the mentioned trade-off. When increasing the noise level the better conditioned models start outperforming the others due to the induced robustness properties. At initialization the condition number is 4.28.

Clearly, the final goal is to achieve both, best performance for all noise levels and optimal stability. However, it needs further research to determine if and when this is possible.

4.3. Denoising MNIST

For denoising we use a basic autoencoder architecture with two dense layers in the encoder and decoder, respectively. Let the weight matrices in order of application be denoted by E_1, E_2, D_2, D_1. We set $E_1, D_1^\top \in \mathbb{R}^{256 \times 784}$ and $E_2, D_2^\top \in \mathbb{R}^{32 \times 256}$ and use ReLU activation for all except the last layer, which uses a sigmoid activation. For training, we perform ERM with respect to the regularized ℓ_2 loss

$$\mathcal{L}(\hat{x}; \Phi) = \|x - \Phi(\hat{x})\|_2 + \lambda_1 \cdot (r(E_1) + r(D_1)) \\ + \lambda_2 \cdot (r(E_2) + r(E_2)), \quad (4.3)$$

λ_1	λ_2	$\kappa(E_1)$	$\kappa(E_2)$	$\kappa(D_2)$	$\kappa(D_1)$
0	0	604.43	59.58	30.76	102.82
0.1	0.005	23.39	1.08	1.07	12.35

Table 2. Condition numbers of all weight matrices of the trained autoencoder for image denoising, with and without regularization. It is noticeable that the weight matrices of the outer layers (E_1, D_1) are absurdly high if not regularized.

where $\hat{x}$ are noisy versions of x with a SNR of 1. The optimizer is Adam with a learning rate of 0.05 for 50 epochs.

Table 2 shows the condition numbers of all weight matrices of the naively trained autoencoder ($\lambda_1 = \lambda_2 = 0$) and the regularizer one ($\lambda_1 = 0.05, \lambda_2 = 0.01$). Upon initial observation, it becomes immediately apparent that the condition numbers of the outer weight matrices are absurdly high (≈ 600!), indicating severe numerical instabilities. Figure 1 illustrates the denoising performance, qualitatively. It is noteworthy that the naively trained model appears to encounter difficulties in reconstructing the images at all, whereas the regularized model demonstrates remarkable performance, even in the presence of substantial noise.

We are aware that there exist more optimal architectures for both classification [16] and denoising [17], however, the objective of this experiments is to illustrate the advantages of the proposed regularizer in a fundamental setting.

5. Conclusion

Traditional approaches to regularization, such as Tikhonov regularization, often fail to directly address the condition number, which is crucial for maintaining numerical stability. This paper introduces a novel almost everywhere differentiable regularizer that enhances numerical stability in neural networks by promoting low condition numbers in their weight matrices. Through a theoretical analysis and a series of numerical experiments, we proved and demonstrated the properties of this regularizer. In a noisy classification and a denoising task using the MNIST dataset, models that are trained with the proposed regularizer exhibit significantly lower condition numbers of the weight matrices and show robustness against noise successfully compared to models that are trained without regularization.

Acknowledgment

References

[1] C. Santos and J. Papa, "Avoiding overfitting: A survey on regularization methods for convolutional neural networks," *ACM Computing Surveys*, vol. 54, 2022.

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[3] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 67, no. 2, pp. 301–320, 2005.

[4] P. Hagemann and S. Neumayer, "Stabilizing invertible neural networks using mixture models," *Inverse Problems*, vol. 37, 2020.

[5] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Third. The Johns Hopkins University Press, 1996.

[6] L. N. Trefethen and D. Bau III, *Numerical Linear Algebra*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1997.

[7] M. Hasannasab, J. Hertrich, S. Neumayer, G. Plonka, S. Setzer, and G. Steidl, "Parseval proximal neural networks," *Journal of Fourier Analysis and Applications*, vol. 26, pp. 1–31, 2020.

[8] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier, "Parseval networks: Improving robustness to adversarial examples," *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[9] P. Balazs, D. Haider, F. Perfler, and V. Lostanlen, "Trainable signal encoders that are robust against noise," in *Proceedings of the Internoise Conference*, 2024.

[10] A. Tikhonov, "Solution of incorrectly formulated problems and the regularization method," *Soviet Math. Dokl.*, vol. 4, no. 2, pp. 1035–1038, 1963.

[11] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 42, no. 1, pp. 80–86, 2000.

[12] C. Budd, M. Freitag, and N. Nichols, "Regularization techniques for ill-posed inverse problems in data assimilation," *Computers & Fluids*, vol. 46, pp. 168–173, 2011.

[13] A. Lewis, "The convex analysis of unitarily invariant matrix functions," *Journal of Convex Analysis*, vol. 2, 1995.

[14] B. Schlein, "Terence Tao: "Topics in random matrix theory"," *Jahresbericht der Deutschen Mathematiker-Vereinigung*, vol. 115, pp. 57–59, 2013.

[15] Y. LeCun, C. Cortes, and C. Burges, "MNIST handwritten digit database," *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, vol. 2, 2010.

[16] W. Hua, C. Li, and X. Wang, "Review of convolutional neural network models and image classification," *Academic Journal of Science and Technology*, vol. 10, pp. 178–184, 2024.

[17] Q. Zhang, J. Xiao, C. Tian, J. Lin, and S. Zhang, "A robust deformed convolutional neural network (CNN) for image denoising," *CAAI Transactions on Intelligence Technology*, vol. 8, 2022.

[18] V. Rakočević, "On continuity of the Moore-Penrose and Drazin inverses.," *Matematichki Vesnik*, vol. 49, no. 3-4, pp. 163–172, 1997.

[19] R. Rockafellar, *Convex Analysis* (Princeton Landmarks in Mathematics and Physics). Princeton University Press, 1997.

[20] B. Mordukhovich, *Variational Analysis and Applications* (Springer Monographs in Mathematics). Springer International Publishing, 2018.

[21] D. Calvetti and L. Reichel, "Tikhonov regularization of large linear problems," *BIT Numerical Mathematics*, vol. 43, no. 2, pp. 263–283, 2003.

[22] M. P. Deisenroth, A. A. Faisal, and C. S. Ong, *Mathematics for Machine Learning*. Cambridge University Press, 2020.

Appendix

A. On the Discontinuity of the Condition Number

Example A.1. This example shows that the condition number $\kappa(S) := \|S\|_2 \left\|S^\dagger\right\|_2$ is not a continuous mapping [18]. Let

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \quad \text{and} \quad E = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

For each $1 > \varepsilon > 0$ we have

$$(A + \varepsilon E)^\dagger = \begin{pmatrix} 1 & 0 \\ 0 & \varepsilon \end{pmatrix}^\dagger = \begin{pmatrix} 1 & 0 \\ 0 & \varepsilon^{-1} \end{pmatrix}.$$

Therefore $\kappa(A + \varepsilon E) = \|A + \varepsilon E\|_2 \left\|(A + \varepsilon E)^\dagger\right\|_2 = \varepsilon^{-1}$. Hence $A + \varepsilon E \to A$ as $\varepsilon \to 0$, but $\lim_{\varepsilon \to 0} \kappa(A + \varepsilon E)$ does not exist, even though $\kappa(A) = 1$.

B. Essentials from Subdifferential Calculus

In this part of the appendix, we include all definitions and results used in the paper to make the document self-contained and the derivation of the Theorems precise and unambiguous.

B.1. Convex Subdifferential [19]

Definition B.1 (Proper function). An extended value function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is called proper, if its domain $\text{dom}(f) := \{x \in \mathbb{R}^n : f(x) < +\infty\}$ is not empty.

Definition B.2 (Convex Subdifferential). Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a proper convex function. The *(convex) subdifferential* of f at $x \in \text{dom}(f)$ is defined as

$$\partial f(x) = \{v \in \mathbb{R}^n \mid f(y) \geq f(x) + \langle v, y - x \rangle \ \forall y \in \mathbb{R}^n\}.$$

The elements of $\partial f(x)$ are called *subgradients* of f at x.

Remark B.3. If $x \notin \text{dom}(f)$, then $\partial f(x) = \emptyset$.

B.2. Mordukhovich Subdifferential [20]

Definition B.4 (Mordukhovich Subdifferential). Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a lower semicontinuous function. The *Mordukhovich (or limiting) subdifferential* of f at $x \in \text{dom}(f)$ is defined as

$$\partial_M f(x) = \left\{v \in \mathbb{R}^n \mid \exists x^k \to x, \ v^k \to v \text{ with } v^k \in \hat\partial f(x^k) \text{ and } f(x^k) \to f(x)\right\},$$

where $\hat\partial f(x)$ denotes the *Fréchet subdifferential* of f at x, defined by

$$\hat\partial f(x) = \left\{v \in \mathbb{R}^n \mid \liminf_{y \to x} \frac{f(y) - f(x) - \langle v, y - x \rangle}{\|y - x\|} \geq 0\right\}.$$

Remark B.5. The Mordukhovich subdifferential generalizes the concept of subgradients to non-convex functions and is particularly useful in variational analysis and optimization.

B.2.1. COINCIDENCE OF SUBDIFFERENTIALS [20]

Theorem B.6. *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a proper, lower semicontinuous, and convex function. Then, for any $x \in \mathbb{R}^n$,*

$$\partial f(x) = \partial_M f(x).$$

Theorem B.7. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function at $x \in \mathbb{R}^n$. Then,*

$$\partial f(x) = \partial_M f(x) = \{\nabla f(x)\}.$$

Theorem B.8. *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a proper, lower semicontinuous, and convex function. If the convex subdifferential $\partial f(x)$ at $x \in \mathbb{R}^n$ is a singleton, say $\partial f(x) = \{v\}$, then f is differentiable at x and $\nabla f(x) = v$.*

B.2.2. MORDUKHOVICH SUBDIFFERENTIAL OF THE SUM OF FUNCTIONS [20]

Theorem B.9. *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a proper, lower semicontinuous, and convex function, and let $g : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function. Then, for any $x \in \mathrm{dom}\, f(x)$:*

$$\partial_M(f + g)(x) = \partial f(x) + \nabla g(x).$$

B.3. Rules of Differentiation [20]

Definition B.10 (Convex Hull). Let S be a subset of $\mathbb{R}^n$. The *convex hull* of S, denoted by $\mathrm{conv}(S)$, is the smallest convex set containing S. It can be defined as:

$$\mathrm{conv}(S) = \left\{ \sum_{i=1}^k \lambda_i x_i \,\middle|\, x_i \in S,\ \lambda_i \geq 0,\ \sum_{i=1}^k \lambda_i = 1,\ k \in \mathbb{N} \right\}.$$

In other words, the convex hull of S is the set of all convex combinations of points in S.

Theorem B.11. *Let $\{f_i\}_{i \in I}$ be a finite family of proper, lower semicontinuous, convex, and differentiable functions from $\mathbb{R}^n \to \mathbb{R}$. Define $f : \mathbb{R}^n \to \mathbb{R}$ by*

$$f(x) = \max_{i \in I} f_i(x).$$

Then, for any $x \in \mathbb{R}^n$,

$$\partial_M f(x) = conv\left\{ \nabla f_i(x) \mid i \in I_x \right\},$$

where $I_x = \{i \in I \mid f_i(x) = \max_{i \in I} f_i(x)\}$ denotes the set of indices, for which f_i attains the largest value at x.

B.4. Results on unitarily invariant matrix functions [13]

Definition B.12 (Absolutely Symmetric Function). A function $f : \mathbb{R}^q \to \mathbb{R}$ is said to be *absolutely symmetric* if $f(\gamma) = f(\gamma^s)$ for any permutation s of the components of γ and for any $\gamma \in \mathbb{R}^q$. Equivalently, f is absolutely symmetric if

$$f(Q\gamma) = f(\gamma) \quad \text{for all } \gamma \in \mathbb{R}^q \text{ and } Q \in \Lambda_q,$$

where Λ_q denotes the set of generalized permutation matrices (matrices with exactly one non-zero entry in each row and each column, that entry being ± 1).

This means that the function value at x of absolutely symmetric functions is independent of the ordering of the entries of x. Recall that σ maps a matrix onto its singular values in nonincreasing order.

Theorem B.13 (Characterization of Convexity). *Suppose that the function $f : \mathbb{R}^q \to (-\infty, +\infty]$ is absolutely symmetric. Then the corresponding unitarily invariant function $f \circ \sigma$ is convex and lower semicontinuous on $\mathbb{C}^{m \times n}$ if and only if f is convex and lower semicontinuous.*

Theorem B.14 (Characterization of Subgradients). *Suppose that the function $f : \mathbb{R}^q \to (-\infty, +\infty]$ is absolutely symmetric, and that the $m \times n$ matrix X has $\sigma(X)$ in $\mathrm{dom}(f)$. Then*

$$\partial(f \circ \sigma)(X) = \left\{ U(Diag\,\mu)V^T \mid \mu \in \partial f(\sigma(X)),\ X = U(Diag\,\sigma(X))V^T \right\}.$$

Theorem B.15 (Gradient Formula). *If a function $f : \mathbb{R}^q \to (-\infty, +\infty]$ is convex and absolutely symmetric then the corresponding convex, unitarily invariant function $f \circ \sigma$ is differentiable at the $m \times n$ matrix X if and only if f is differentiable at $\sigma(X)$. In this case,*

$$\nabla(f \circ \sigma)(X) = U(Diag\,\nabla f(\sigma(X)))V^T,$$

for $X = U(Diag\,\sigma(X))V^T$.

C. Proofs

In this section we proof Theorems 3.1, 2.2, and 3.2 formulated in the main body of the paper.

Proof of Theorem 3.1

Proof. We define $f : \mathbb{R}^\nu \to \mathbb{R}$ and $g : \mathbb{R}^\nu \to \mathbb{R}$ as

$$f(x) = \max_{i \in 1 \ldots \nu} \frac{1}{2} x_i^2, \quad g(x) = \frac{1}{2\nu} \sum_{i=1}^\nu x_i^2.$$

One can see that by Definition B.12 both f and g are absolutely symmetric and $r(S) = f(\sigma(S)) + g(\sigma(S))$. The function f is convex by the rule for pointwise maxima Theorem B.11 and its subdifferential is given by

$$\partial f(x) = \text{conv}\{x_i e_i \mid i : x_i^2 = x_{\max}^2\},$$

where e_i denotes the $i - th$ standard unit vector and $\max$ denotes the index, for which the largest value is attained. By the results of Theorem B.13 we deduce that $f(\sigma(S))$ is convex and by Theorem B.14

$$\partial(f \circ \sigma)(S) = \text{conv}\{\sigma_i(S) u_i v_i^T \mid i : \sigma_i(S) = \sigma_{max}(S)\}.$$

The function g is differentiable everywhere and its gradient is given by $\nabla g(x) = \frac{1}{\nu} x$. By Theorem B.15 we deduce

$$\nabla(g \circ \sigma)(S) = U(\text{Diag}(\nabla g(\sigma(S)))) V^T$$
$$= -\frac{1}{\nu} U(\text{Diag}(\sigma(S)) V^T = -\frac{1}{\nu} S.$$

Thus, by Theorem B.9, we derive

$$\partial r(S) = \text{conv}\{\sigma_i(S) u_i v_i^T \mid i : \sigma_i(S) = \sigma_{\max}(S)\} - \frac{1}{\nu} S.$$

If the largest singular value of S is unique, $\partial r(S)$ is a singleton and by Theorem B.8 therefore differentiable with

$$\nabla r(S) = \sigma_{\max}(S) u_1 v_1^T - \frac{1}{N} S.$$

$\square$

Proof of Theorem 2.2

Proof. Using the Mean Value Theorem for the logarithm, we derive for $c \in (\sigma_k(S)^2, \sigma_1(S)^2)$

$$2 \ln(\kappa(S)) = \ln(\kappa(S)^2)$$
$$= \ln(\sigma_1(S)^2) - \ln(\sigma_k(S)^2),$$
$$= \frac{1}{c} \left| \sigma_1(S)^2 - \sigma_k(S)^2 \right|,$$
$$\leq \frac{1}{\sigma_k(S)^2} (\sigma_1(S)^2 - \sigma_k(S)^2).$$

Furthermore estimating the regularizer yields

$$2r(S) = \sigma_1(S)^2 - \frac{1}{\nu} \sum_{i=1}^k \sigma_i(S)^2,$$
$$\geq \sigma_1(S)^2 - \frac{1}{\nu} \left((k-1)\sigma_1(S)^2 + \sigma_k(S)^2\right),$$
$$\geq \sigma_1(S)^2 - \frac{1}{\nu} \left((\nu-1)\sigma_1(S)^2 + \sigma_k(S)^2\right),$$
$$= \frac{1}{\nu} (\sigma_1(S)^2 - \sigma_k(S)^2).$$

Combining those two results we conclude:

$$\ln\left(\kappa(S)\right) \le \frac{\nu}{\sigma_k(S)^2} r(S),$$

$$\kappa(S) \le e^{\frac{\nu}{\sigma_k(S)^2} r(S)}.$$

$\square$

Proof of Theorem 3.2

Proof. Let $S = U\mathrm{Diag}(\sigma(S))V^T = \sum_{i=1}^{\nu} \sigma_i(S)u_i v_i^T$ be the singular value decomposition of S. Let k be the index of the smallest non-zero singular value, i.e. $k = \mathrm{argmin}_i\{\sigma_i(S) \mid \sigma_i(S) > 0 \text{ for } k \in \{1,\dots,\nu\}\}$. Then it holds that:

$$
\begin{aligned}
S' &= S - \lambda\nabla r(S) \\
&= \left(1 + \frac{\lambda}{\nu}\right)S - \lambda\sigma_1(S)u_1 v_1^T \\
&= \left(1 + \frac{\lambda}{\nu} - \lambda\right)\sigma_1(S)u_1 v_1^T + \left(1 + \frac{\lambda}{\nu}\right)\sum_{i=2}^{\nu}\sigma_i(S)u_i v_i^T \\
&= U\mathrm{Diag}(\sigma')V^T,
\end{aligned}
$$

where $\sigma' = \left(\left(1 + \frac{\lambda}{\nu} - \lambda\right)\sigma_1(S), \left(1 + \frac{\lambda}{\nu}\right)\sigma_2(S),\dots,\left(1 + \frac{\lambda}{\nu}\right)\sigma_\nu(S)\right)$. Given this decomposition of S', we notice that σ' are the singular values of S', but not necessarily in the right order. Therefore we are going to distinct two cases.

Since S is assumed to have a unique largest singular value, there exists an $\alpha > 1$ s.t. $\sigma_1(S) = \alpha\sigma_2(S)$. Notice that $\kappa(S) \ge \alpha$ holds. Choose $L \in (0,1]$, s.t. for all $\lambda \in (0,L] : 1 - \frac{\lambda}{1+\frac{\lambda}{\nu}} \ge \frac{1}{\kappa(S)}$.

1. Case: $\frac{1}{\alpha} < 1 - \frac{\lambda}{1+\frac{\lambda}{\nu}}$.

By the case distinction we see:

$$
\begin{aligned}
\left(1 + \frac{\lambda}{\nu} - \lambda\right)\sigma_1(S) &> \frac{1}{\alpha}\left(1 + \frac{\lambda}{\nu}\right)\sigma_1(S), \\
&= \left(1 + \frac{\lambda}{\nu}\right)\sigma_2(S),
\end{aligned}
$$

and thus $\sigma_1' > \sigma_2'$. Since $\sigma_i(S) \ge \sigma_j(S)$ for $i < j$, we deduce that σ' are exactly the singular values of S' in non-increasing order. Furthermore the amount of non-zero singular values stays the same. Therefore

$$
\begin{aligned}
\kappa(S') = \frac{\sigma_1'}{\sigma_k'} &= \frac{\left(1 + \frac{\lambda}{\nu} - \lambda\right)\sigma_1(S)}{\left(1 + \frac{\lambda}{\nu}\right)\sigma_k(S)} \\
&= \left(1 - \frac{\lambda}{1 + \frac{\lambda}{\nu}}\right)\kappa(S).
\end{aligned}
$$

2. Case: $\frac{1}{\alpha} \ge 1 - \frac{\lambda}{1+\frac{\lambda}{\nu}} \ge \frac{1}{\kappa(S)}$.

Similarly to the previous case, simple arithmetics show that in this case σ_2' becomes the largest singular value of S' and σ_k' remains the smallest. Therefore:

$$
\begin{aligned}
\kappa(S') = \frac{\sigma_2'}{\sigma_k'} &= \frac{\sigma_2(S)}{\sigma_k(S)} = \frac{1}{\alpha}\frac{\sigma_1(S)}{\sigma_k(S)} \\
&= \frac{1}{\alpha}\kappa(S).
\end{aligned}
$$

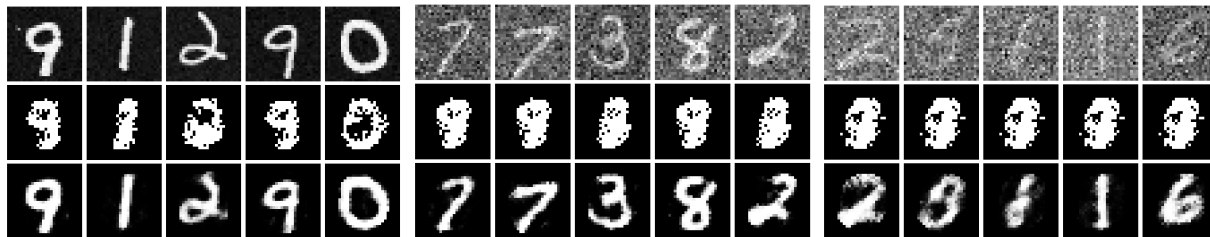In both cases we see that $\kappa(S') < \kappa(S)$, since $\lambda \in (0,L]$ and $\alpha > 1$. $\square$

Figure 3. MNIST denoising results with (bottom) and without regularization (mid) with three different SNRs, from left to right: $10, 1, 0.5$. Already with reconstructing the images from almost no noise, the non-regularized autoencoder struggles. Due to the high condition numbers in the network, the output is very sensitive to perturbations in the input, resulting in the network being unable to learn properly.
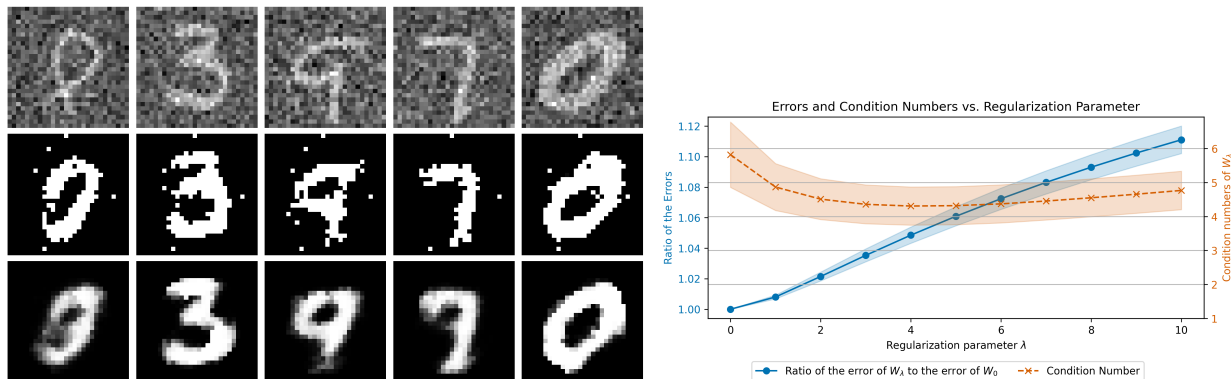

Figure 4. Left: Results of MNIST denoising with SNR 1 with Tikhonov regularization for two different sets of parameters. Mid: $\lambda_1 = 0.01, \lambda_2 = 0.0001$. Bottom: $\lambda_1 = 1, \lambda_2 = 0.01$. Right: Results of least-squares minimization of (4.1) after 10^5 iterations with Tikhonov Regularizer for different regularization parameter λ values

D. Discussion on Tikhonov Regularization

Tikhonov Regularization is a well-known method used to stabilize the solutions of ill-posed problems and to prevent overfitting in machine learning models [21], [22]. This method adds a regularization term to the loss function, which penalizes large coefficients in the solution, and helps in balancing between fitting the data and simplicity in the solution.

We include this discussion on Tikhonov regularization due to its widespread recognition and frequent application as a common regularizer, although its direct impact on the condition number of the weight matrices remains, to the best of the authors knowledge, ambiguous in the existing literature.

Mathematically, Tikhonov regularization adds to the standard least squares problem a penalty term proportional to the square of the norm of the coefficients, in our case:

$$\min_{W \in \mathbb{R}^{n \times m}} \|WX - Y\|_F^2 + \lambda \cdot \|W\|_F^2,$$

where λ is the regularization parameter controlling the trade-off between model fit and the magnitude of coefficients [10].

D.1. Impact on Condition Number

While Tikhonov regularization is effective in promoting stability in poorly behaved optimization problems and preventing overfitting, it does not directly address the condition number of the weight matrices. In the context of neural networks, using the Frobenius norm as a regularizer can help in reducing the overall magnitude of the weights but might not sufficiently control the condition number, as we will see in the following simulation.

We repeat the simulations from Section 4.1 for Tikhonov Regularization, where $r(W) = \|W\|_F^2$ in (4.1). The results are shown in Figure 4. We observe that Tikhonov regularization slightly reduces the condition number, but not significantly, before the (approximation) error $\|W_\lambda X - Y\|_F$ becomes too large compared to the non-regularized error $\|W_0 X - Y\|_F$.

λ	$\kappa(W_1)$	SNR ∞	SNR 1	SNR 0.5
0	**43.29**	**98.42 %**	93.80 %	71.91 %
10^{-3}	**56.43**	**98.31 %**	94.34 %	74.24 %
10^{-2}	242.39	98.25 %	94.83 %	76.94 %
10^{-1}	4044.34	98.15 %	**95.10 %**	**78.67 %**
1	4020.47	97.68 %	**95.02 %**	**82.79 %**

Table 3. Results when using Tikhonov regularization for MNIST classification.

λ_1	λ_2	$\kappa(E_1)$	$\kappa(E_2)$	$\kappa(D_2)$	$\kappa(D_1)$
1	0.01	12.80	1.43	2.65	6061.53
0.1	0.001	1277.21	31641.66	20358.70	5347.09

Table 4. Condition numbers of the weight matrices when using Tikhonov regularization for denoising in two parameter settings.

D.2. Numerical Experiments with Tikhonov Regularization

We conducted numerical experiments on the same problems as in Section 4 to compare the performance of Tikhonov regularization with the proposed method. The results are summarized below:

D.2.1. MNIST CLASSIFICATION

Table 3 shows the classification accuracy and condition numbers for different levels of noise (SNR) using Tikhonov regularization. As λ increases, the condition numbers of the weight matrices increase dramatically, indicating potential numerical instability. Yet, we see that it performs well at a medium noise level (SNR 1) and outperforms the baseline, as well as our proposed regularizer in all settings of λ. For the high noise level (SNR 0.5), however, it falls back again.

D.2.2. MNIST DENOISING

Figure 4 illustrates the denoising performance with low (mid) and high (bottom) influence by means of the values λ_1, λ_2. Table 4 shows the corresponding condition numbers. We see that Tikhonov regularization fails to maintain low condition numbers consistently across all layers, which explains the worse denoising performance compared to the proposed regularizer.

D.3. Conclusion

While Tikhonov regularization offers benefits in terms of regularizing the magnitude of weights and robustness to a medium level of noise, it does not effectively control the condition number of weight matrices, leading to potential numerical instability. The proposed regularizer in this paper addresses this limitation by specifically targeting the condition number, thus enhancing the robustness and stability of neural networks, especially in very noisy environments.