

Regularized Gauss-Newton for Optimizing Overparameterized Neural Networks

Adeyemi D. Adeoye ^{1*}, Philipp Petersen ², Alberto Bemporad ¹

¹IMT School for Advanced Studies Lucca, Italy

²Faculty of Mathematics, University of Vienna, Austria

Abstract

The generalized Gauss-Newton (GGN) optimization method incorporates curvature estimates into its solution steps, and provides a good approximation to the Newton method for large-scale optimization problems. GGN has been found particularly interesting for practical training of deep neural networks, not only for its impressive convergence speed, but also for its close relation with neural tangent kernel regression, which is central to recent studies that aim to understand the optimization and generalization properties of neural networks. This work studies a GGN method for optimizing a two-layer neural network with explicit regularization. In particular, we consider a class of generalized self-concordant (GSC) functions that provide smooth approximations to commonly-used penalty terms in the objective function of the optimization problem. This approach provides an adaptive learning rate selection technique that requires little to no tuning for optimal performance. We study the convergence of the two-layer neural network, considered to be overparameterized, in the optimization loop of the resulting GGN method for a given scaling of the network parameters. Our numerical experiments highlight specific aspects of GSC regularization that help to improve generalization of the optimized neural network. The code to reproduce the experimental results is available at <https://github.com/adeyemiadeoye/ggn-score-nn>.

1. Introduction

Despite their superior convergence rates compared to first-order methods, (approximate) second-order methods are still rarely used — and as such, underexplored — for training large-scale machine learning and neural network (NN) models. This is due to their highly prohibitive computations and memory footprints at each iteration. Some past and recent works have, however, made efforts to reduce this overhead by proposing different approximations to the Hessian of the loss function, which the methods ultimately exploit to achieve their impressive convergence properties (see e.g., [1, 2, 3, 4, 5, 6, 7, 8, 9]).

One of the most appealing approximations to the Hessian matrix within the context of practical deep learning and nonlinear optimization in general is the generalized Gauss-Newton (GGN) approximation of [10], which uses a positive semi-definite (PSD) matrix to model the curvature about an arbitrary convex loss function. In fact, the Fisher information matrix (FIM) — a curvature approximating matrix which most other approximate second-order methods seek to estimate — is shown to have direct connections with the GGN matrix in many practical cases [4, 11]. Despite its close connection with the GGN matrix, the FIM, unlike the GGN matrix, potentially leads to over-approximating the second-order terms in more general loss functions, throwing away relevant curvature information [10]. In addition to the desirable property of maintaining positive-definiteness throughout the training procedure, other nice properties of the GGN matrix, in comparison with the Hessian matrix, are discussed in [12, Section 8.1]; see also [13] for discussions in the context of nonlinear least-squares estimation and [14] for efficient training of (deep) recurrent neural networks with a GGN approach.

*Email address: adeyemi.adeoye@imtlucca.it

Towards understanding the theoretical working of deep neural networks, a line of work [15, 16, 17, 18, 19, 20, 21, 22, 23, 24] attributes their optimization and generalization success in many applications to their immense overparameterization, that is, the property of having way more parameters than the number of data points they are being trained on. These generalization properties of the NN are known to have connections with the implicit regularization of the overparameterized NN by the gradient descent (GD) method [25, 26, 27, 28, 29, 30, 31, 32, 33]. For the (generalized) Gauss-Newton and its related FIM (or *natural gradient*), some recent works [8, 34, 35, 36, 37, 38] have shown similar approximation properties and global convergence in the overparameterized regime, also mostly attributing generalization to the implicit regularization effect of the Gauss-Newton via the NTK [8, 34, 35, 36, 37] and the mean-field [38].

In many of the works showing the implicit regularization effect of gradient-based optimizers, it is suggested that explicit regularizers are not needed at all in order to see the impressive generalization results. However, recent works such as [39, 40, 41] argue that explicit regularization of the network indeed matters and should at least be given as much attention, both from a generalization and an optimization point of view. In particular, [39] proved an approximation bound (in number of samples), via the lens of *margin theory*, for an infinite-width one-hidden-layer NN *weakly-regularized* by the ℓ_2 -norm, which significantly improves upon other results that rely on the NTK and/or implicit regularization formalism. In addition, they proved a global polynomial convergence rate for the noisy gradient descent, an improvement over related works that similarly study NN optimization in the infinite-width limit. In [8], the interpretation of the GGN updates as an explicit solution of the NTK regression is used to prove a global linear convergence in the mini-batch setting. Apart from the explicit addition of a regularization term to the objective function, explicit regularization is also induced in other forms [42, 43, 44, 45, 40, 41].

In this paper, we study the optimization of a one-hidden-layer NN by the GGN method, and by drawing inspiration from their performance in convex optimization, we consider explicit self-concordant regularization of the GGN. To the best of our knowledge, our convergence result is the first in this kind of setting: optimization of an explicitly regularized NN by the GGN method in the overparameterized regime. The structure of the class of regularization functions considered not only helps to control the local rate of change of their second derivatives [46], but is also used in the selection of adaptive learning rates. Unlike [39], we do not assume an arbitrarily *weak* regularization under our setting; instead the smoothing framework covered by our study allows to choose a regularization strength which may depend only on the initialization of the NN, and is characterized by a smoothing parameter. However, for a proper choice of the regularization strength, the final trained NN model can be made to be reasonably small and “simple” in spite of the overparameterization, and we can have a tradeoff between test error and training error.

1.1. Notation

The standard Euclidean norm is denoted by $\|\cdot\|$ or $\|\cdot\|_2$ and the 1-norm by $\|\cdot\|_1$. We denote the standard inner product between two vectors by $\langle \cdot, \cdot \rangle$, i.e., $\langle x, y \rangle \triangleq x^\top y$ for $x, y \in \mathbb{R}^p$. For a positive integer m , we define $[m] \triangleq \{1, 2, \dots, m\}$. We let $\mathbb{R}_{\geq 0}$ and $\mathbb{R}_{> 0}$ denote the set of nonnegative and positive real numbers, respectively. For an extended real-valued function $g: \mathbb{R}^p \rightarrow \mathbb{R} \cup \{+\infty\}$, we denote by $\text{dom } g \triangleq \{x \in \mathbb{R}^p \mid g(x) < +\infty\}$ the (*effective*) *domain* of g . $\Gamma_0(\mathcal{X})$ denotes the set of proper convex lower-semicontinuous (lsc) functions from $\mathcal{X} \subseteq \mathbb{R}^p$ to $\mathbb{R} \cup \{+\infty\}$. We denote by $\mathcal{C}^k(\mathbb{R}^p)$, the class of k -times continuously-differentiable functions on \mathbb{R}^p , $k \in \mathbb{R}_{\geq 0}$. For $g \in \mathcal{C}^3(\text{dom } g)$, we let $g'(t)$, $g''(t)$ and $g'''(t)$ denote the first, second, and third derivatives of g , at $t \in \mathbb{R}$, respectively. The gradient,

Hessian, and third-order derivative tensor of g at $x \in \mathbb{R}^p$ are respectively written as $\nabla_x g(x)$, $\nabla_x^2 g(x)$, and $\nabla_x^3 g(x)$. We omit the subscripts if the variables with respect to which the derivatives are taken are clear from the context. For a symmetric matrix $H \in \mathbb{R}^{p \times p}$, we write $H \succ 0$ (resp. $H \succeq 0$) to say H is positive definite (resp., positive semidefinite). We let $\lambda_1(H)$ denote the maximum eigenvalue of a matrix $H \in \mathbb{R}^{p \times p}$, and $\lambda_p(H)$ its minimum eigenvalue; $\text{tr}(H)$ denotes the trace of H . The scalars $\sigma_{\max}(A)$ and $\sigma_{\min}(A)$ respectively denote the maximum and minimum singular values of an $m \times p$ matrix A . Given that $\nabla^2 g(x) \succ 0$, the *local* norm $\|\cdot\|_x$ with respect to g at x is the weighted norm induced by $\nabla^2 g(x)$, i.e., $\|d\|_x \triangleq \langle \nabla^2 g(x)d, d \rangle^{1/2}$. The dual norm is $\|v\|_x^* \triangleq \langle \nabla^2 g(x)^{-1}v, v \rangle^{1/2}$. We also define the notations $\|x\|_H \triangleq \langle Hx, x \rangle^{1/2}$, $\|x\|_H^* \triangleq \langle H^{-1}x, x \rangle^{1/2}$, for $H \succ 0, x \in \mathbb{R}^p$. An Euclidean ball of radius r centered at \bar{x} is denoted by $\mathcal{B}_r(\bar{x}) \triangleq \{x \in \mathbb{R}^p \mid \|x - \bar{x}\| \leq r\}$. The (Dikin) ellipsoid of radius r centered at \bar{x} is defined by $\mathcal{E}_r(\bar{x}) \triangleq \{x \in \mathbb{R}^p \mid \|x - \bar{x}\|_H < r\}$, for $H \succ 0$. We define set convergence in the sense of Painlevé-Kuratowski [47, Chapter 4]. Given $\{g_t\}_{t \in \mathbb{R}_{\geq 0}}$ with $g_t: \mathbb{R}^p \rightarrow \mathbb{R} \cup \{-\infty, +\infty\}$, $\text{e-lim } g_t = g$ denotes the epigraphic convergence (*epi-convergence*) of $\{g_t\}_{t \in \mathbb{R}_{\geq 0}}$ to a function $g: \mathbb{R}^p \rightarrow \mathbb{R} \cup \{-\infty, +\infty\}$.

2. GGN for Learning Neural Networks

Given the sequence of data points $S \triangleq \{(x_i, y_i)\}_{i \in [m]}$ with $x_i \in \mathbb{R}^{n_0}, y_i \in \mathbb{R}^{n_L}$, an L -layer fully-connected feedforward NN is defined as follows. Starting with an input $z_0 \in \mathbb{R}^{n_0 \times m}$, and for $l = 1, \dots, L$,

$$a_l = W^{(l)} z_{l-1} + b^{(l)}, \quad z_l = \varrho_l(a_l), \quad (1)$$

where $W^{(l)} \in \mathbb{R}^{n_l \times n_{l-1}}$ and $b^{(l)} \in \mathbb{R}^{n_l}$ are the l -th layer weights and biases of the network, respectively; each $\varrho_l: \mathbb{R} \rightarrow \mathbb{R}$ is an element-wise activation function. Let $\Phi(\cdot; \theta) =: z_L$ be the output of the NN, where $\theta = [\theta_1, \theta_2, \dots, \theta_L]^\top \in \mathbb{R}^p$ with $\theta_l \triangleq \text{vec}([W^{(l)} \ b^{(l)}])$, the stacked vectorization of $W^{(l)}$ and $b^{(l)}$. In the *supervised learning* task, we look for the parameter vector θ minimizing the regularized empirical risk

$$\min_{\theta \in \mathbb{R}^p} \mathcal{L}(\theta) \triangleq \hat{R}_s(\Phi) + g(\theta), \quad \hat{R}_s(\Phi) \triangleq \frac{1}{m} \sum_{i=1}^m \ell(\Phi(x_i; \theta), y_i), \quad (2)$$

where $\hat{R}_s(\Phi)$ is the empirical risk associated with the NN learning task, $\ell: \mathbb{R}^{n_L} \times \mathbb{R}^{n_L} \rightarrow \mathbb{R}$ is a loss function, and $g: \mathbb{R}^p \rightarrow \mathbb{R}$ is a regularization function. We denote by Φ^* an output function that best interpolates the data set S .

Let $n_1 \equiv n$ (number of hidden neurons), $W^{(1)} \equiv u = [u_1, u_2, \dots, u_n]^\top \in \mathbb{R}^{n \times n_0}$ and $W^{(2)} \equiv v = [v_1, v_2, \dots, v_n] \in \mathbb{R}^n$. Without loss of generality, we consider a *biasless* one-hidden layer NN:

$$\mathbb{R}^{n_0} \ni x \mapsto \Phi(x; \theta) \triangleq \kappa(n) \sum_{i=1}^n v_i \varrho(u_i x), \quad (3)$$

where $\kappa(n)$ is some scaling that depends on n , e.g., $\kappa(n) = 1/\sqrt{n}$ as in [17]. We remark that for an $(L-1)$ -hidden layer NN written in the biasless form, the bias vectors can always be recovered by redefining

$$x \leftarrow \begin{bmatrix} x \\ 1 \end{bmatrix}, \quad \theta_l \leftarrow \text{vec} \left(\begin{bmatrix} W^{(l)} & b^{(l)} \\ 0 & 1 \end{bmatrix} \right),$$

for $l \in [L - 1]$, and $\theta_L \leftarrow \text{vec}([W^{(L)} \ b^{(L)}])$. We assume the following about the activation function ϱ , which is satisfied by most activation functions but piecewise linear ones.

A.1 The activation function ϱ is twice differentiable, Lipschitz, and smooth.

Below, we briefly describe the NTK regression and its connection with GGN for overparameterized networks by first considering the case $g = 0$ in (2).

The NTK and Gradient Descent. In the infinite-width limit, there is an established [21] relation between the steps obtained via a gradient-based method for NNs and the so-called *kernel gradient descent* in function space. In particular, it is shown that, as $n \rightarrow \infty$, $\forall i, j \in [m]$, $\langle \nabla_{\theta} \Phi(x_i, \theta_0), \nabla_{\theta} \Phi(x_j, \theta_0) \rangle$ converges to some positive definite deterministic kernel $k(x_i, x_j) = k(x_i, x_j)^{\top} \in \mathbb{R}^{n_L \times n_L}$ (the limiting NTK), and remains unchanged during training. Consider the case $g = 0$ in (2). In the infinite-width limit, the gradient descent for solving the resulting problem reduces to the kernel gradient descent:

$$\Phi_{t+1} = \Phi_t - \alpha_t G_t \nabla_{\Phi_t} \hat{R}_s(\Phi_t), \quad (4)$$

where $\Phi_t \triangleq (\Phi(x_i; \theta_t))_{i \in [m]} \in \mathbb{R}^m$ denotes the network outputs on x_i 's at iteration t , $\alpha_t \in \mathbb{R}_{>0}$ is a step-size (or *learning rate*), and G_t is an $m \times m$ matrix whose (i, j) -th entry is given by $\langle \nabla_{\theta} \Phi(x_i, \theta_t), \nabla_{\theta} \Phi(x_j, \theta_t) \rangle$; see, e.g., [48, Lemma 3.1] which considers the continuous-time evaluation of Φ_t , $t \in \mathbb{R}_{\geq 0}$.

GGN and the NTK regression. An important feature of GGN for infinite-width NNs is its direct relation with the NTK regression solution in the overparameterized regime. We introduce the notations $Q_t \equiv \nabla_{\Phi_t}^2 \hat{R}_s(\Phi_t)$, $e_t \equiv \nabla_{\Phi_t} \hat{R}_s(\Phi_t)$, and consider again the case $g = 0$ in problem (2). The GGN for the resulting problem is given by the following iterative process:

$$\theta_{t+1} = \theta_t - \alpha_t (J_t^{\top} Q_t J_t)^{-1} J_t^{\top} e_t, \quad (5)$$

where $J_t = (\nabla_{\theta} \Phi(x_1, \theta_t), \dots, \nabla_{\theta} \Phi(x_m, \theta_t))^{\top} \in \mathbb{R}^{m \times p}$ is the Jacobian matrix (of features) at iteration t . For overparameterized networks, if ℓ is the squared loss, Q_t becomes the identity matrix and we can conveniently rewrite the GGN updates with respect to the NTK matrix G_t as

$$\theta_{t+1} = \theta_t - \alpha_t J_t^{\top} G_t^{-1} e_t. \quad (6)$$

On the other hand, corresponding to (4), the updates to the parameters θ via θ_t can be obtained by solving the regression problem (cf. [8])

$$\theta_{t+1} = \underset{\theta}{\operatorname{argmin}} \frac{1}{2} \left\| \langle J_t, \theta - \theta_t \rangle + \nabla_{\Phi_t} \hat{R}_s(\Phi_t) \right\|^2, \quad (7)$$

which results from the linearization of Φ around θ_t . For an overparameterized NN, this linearization provides a good approximation to Φ , and hence the Hessian (with respect to θ) of the resulting empirical risk by replacing Φ by its linearization, which gives the GGN approximation (see, e.g., [49]), is expected to provide a good approximation of the actual Hessian. In terms of kernel “ridgeless” regression solution to problems of the form (2) (with $g = 0$) and ℓ taken as the squared loss, the update (6) provides a *minimum-norm* interpolating solution in the so-called Reproducing Kernel Hilbert Space (RKHS) [50, 51]. Hence, the GGN, in this case, provides a closed-form solution to the NTK regression, which efficiently replaces gradient descent in the NTK formalism.

2.1. Regularized GGN for Overparameterized Neural Networks

If $g \neq 0$, the relation between gradient descent and NTK will probably break [39]. Apart from the NTK parameterization, a commonly studied parameterization in the context of overparameterized NNs is the random feature (RF) model [52, 53] which, due to its close connection with a one-hidden layer NN, often provides a prototype for studying realistic NNs. Whether the NTK or the RF parameterization is used, one of the key properties we desire about the dynamics of the optimizer is *stability* which, when established, can help to still benefit a lot from overparameterization in the optimization scope. To this end, we first present the definition of generalized self-concordant (GSC) functions on \mathbb{R}^p from [54] as follows.

Definition 1. A convex function $g \in \mathcal{C}^3(\text{dom } g)$, with $\text{dom } g$ open, is said to be (M_g, ν) -GSC of the order $\nu \in \mathbb{R}_{>0}$, with $M_g \in \mathbb{R}_{\geq 0}$, if $\forall x \in \text{dom } g, \forall u, v \in \mathbb{R}^p$, $\left| \langle \nabla^3 g(x)[v]u, u \rangle \right| \leq M_g \|u\|_x^2 \|v\|_x^{\nu-2} \|v\|^{3-\nu}$, where $\nabla^3 g(x)[v] \triangleq \lim_{t \rightarrow 0} \left\{ (\nabla^2 g(x + tv) - \nabla^2 g(v)) / t \right\}$.

We now assume the following about g :

G.1 The regularization function g is convex and (M_g, ν) -GSC.

The class of regularization functions satisfying condition **G.1** includes the self-concordant smoothing functions for commonly used regularizers such as the ℓ_1 - and ℓ_2 -norms (see Definition 2 below). The resulting smooth approximation has the key property that it *epi-converges* to the original regularizer, providing useful features that can be exploited on the epigraph of g for optimization.

Definition 2 ([55]). The parameterized function $g: \mathbb{R}^p \times \mathbb{R}_{>0} \rightarrow \mathbb{R}$ is said to be a self-concordant smoothing function for a function $\bar{g} \in \Gamma_0(\mathbb{R}^p)$ if $e\text{-}\lim_{\mu \downarrow 0} g = \bar{g}$ and $g(\cdot; \mu)$ is (M_g, ν) -GSC, where $\mu \in \mathbb{R}_{>0}$ is a smoothing parameter.

For the regularized problem (2) (with g satisfying **G.1**), the corresponding GGN update is obtained by augmenting the terms Q_t , e_t and J_t , respectively by 0, 1 and $\nabla g(\theta_t)$ in the appropriate dimensions [9]. Let us denote these augmented counterparts by \hat{Q}_t , \hat{e}_t and \hat{J}_t . We then write for the GGN

$$\theta_{t+1} = \theta_t - \alpha_t (\hat{J}_t^\top \hat{Q}_t \hat{J}_t + H_t)^{-1} \hat{J}_t^\top \hat{e}_t, \quad (8)$$

or in its convenient form for overparameterized models as [9]

$$\theta_{t+1} = \theta_t - \alpha_t H_t^{-1} \hat{J}_t^\top (I + \hat{Q}_t \hat{J}_t H_t^{-1} \hat{J}_t^\top)^{-1} \hat{e}_t, \quad (9)$$

where $H_t \equiv \nabla^2 g(\theta_t)$. Relative to the minimal assumptions required to control the dynamics of the network outputs for the unregularized case, e.g., positive definiteness of G_t (which indeed holds in the overparameterized regime), we need the following standard regularity assumptions on \hat{R}_s , \hat{Q}_t and \hat{e}_t (see Appendix B for details on the regularity terms):

R.1 \hat{R}_s is γ_R -strongly convex, and has upper-bounded gradients and Hessian; g , \hat{Q}_t and \hat{e}_t are locally bounded.

An important consequence of condition **A.1** is that, in addition to g admitting a Lipschitz continuous gradient (see Lemma 2 in Appendix A.2), we get that J_t is (locally) Lipschitz continuous (see

Appendix A.1). Then, together with R.1 and the stability of H_t , we can control the key terms $H_t^{-1}\hat{J}_t^\top(I + \hat{Q}_t\hat{J}_tH_t^{-1}\hat{J}_t^\top)^{-1}$ and \hat{e}_t appearing in (9).

Corresponding to (4), the overparameterized NN trained according to (9) evolves in discrete-time as

$$\Phi_{t+1} = \Phi_t - \alpha_t \hat{G}_t \hat{e}_t, \quad (10)$$

where $\hat{G}_t \triangleq J_t H_t^{-1} \hat{J}_t^\top (I + \hat{Q}_t \hat{J}_t H_t^{-1} \hat{J}_t^\top)^{-1} \in \mathbb{R}^{m \times (m+1)}$. One major observation about the behaviour of the dynamics of G_t in (4) in the overparameterized setting is its stability throughout the training process, which characterizes the optimizer's global optimality [18]. In the analysis of gradient descent, most stability and convergence results in the literature heavily rely on the (strictly positive) minimum eigenvalue of G_t . These kinds of results are not immediate with \hat{G}_t or, in general, with explicit regularization. However, the self-concordant condition on g ensures that its Hessian is at least locally stable¹, and hence for an appropriate parameterization of the NN, we can ensure the stability of the dynamics of \hat{G}_t . In addition to these, also noteworthy is an immediate deduction from the Lipschitzness of ϱ and ∇g : the boundedness of the singular values of \hat{J}_t away from zero.

3. Theoretical Result

We study the convergence of self-concordant-regularized GGN for the one-hidden layer network. In line with the settings of Section 2, the learning rate selection rule we consider throughout is

$$\alpha_t = \frac{\bar{\alpha}_t}{1 + M_g \eta_t}, \quad (11)$$

where $0 < \bar{\alpha}_t \leq 1$ and $\eta_t = \|\nabla g(\theta_t)\|_{\theta_t}^*$. Without any emphasis on the particular choice of the target function Φ^* , we assume it is given by any *universally consistent*² algorithm as a minimum requirement. An example, in the case $\ell(\Phi, y) \triangleq \frac{1}{2}(\Phi - y)^2$ in (2), is the regularized least squares algorithm which, for a kernel prediction function $\Phi_{\text{RF}} \equiv \Phi^*$ is defined by

$$\mathbb{R}^{n_0} \ni x \mapsto \Phi_{\text{RF}}(x; (u^\infty, v^*)) \triangleq \sum_{i=1}^n v_i^* \varrho(u_i^\infty x). \quad (12)$$

This yields the estimator $v^* = [v_1^*, v_2^*, \dots, v_n^*] \in \mathbb{R}^n$, the unique minimizer of the ℓ_2 -regularized empirical loss $\frac{1}{2m} \sum_{i=1}^m (\Phi_{\text{RF}}(x_i; v) - y_i)^2 + \frac{\lambda}{2} \|v\|^2$, for some $\lambda \in \mathbb{R}_{\geq 0}$, where the entries of $u^\infty = [u_1^\infty, u_2^\infty, \dots, u_n^\infty]^\top \in \mathbb{R}^{n \times n_0}$ remain fixed iid random variables.

We state our main result of this section in Theorem 1 below. The detailed proof is given in Appendix B. We let $\tilde{\Phi}_t \in \mathbb{R}^{m+1}$ denote the vector obtained by augmenting Φ_t by 1. This $\tilde{\Phi}_t$ corresponds to augmenting the rows of J_t in the definition of \hat{G}_t by the vector whose entries are all zeros except the last entry which has the value $\tilde{\phi}_t = 1/\phi_{t-1}^{m+1} \in \mathbb{R}$, where ϕ_t^{m+1} denotes the last entry of $H_t^{-1} \hat{J}_t^\top (I +$

¹As noted in the introduction, self-concordance helps to control the rate at which the Hessian of g changes locally, and this property has been recently formalized and studied for the notion of local and global *Hessian stability* in convex optimization (see, e.g., [56, 57, 58]).

²Informally speaking, a learning algorithm is said to be universally consistent if the error of its estimate tends to zero as the sample size tends to infinity, for all distributions of the sample space such that the second moment of the output variable is finite. For a more precise context, see, e.g., [59] and the references therein.

$\hat{Q}_t \hat{J}_t H_t^{-1} \hat{J}_t^\top)^{-1} \hat{e}_t \in \mathbb{R}^{n \times 1}$ at a time t . Let this augmented version of J_t be denoted by \tilde{J}_t . Then, we define $\tilde{G}_t \triangleq \tilde{J}_t H_t^{-1} \hat{J}_t^\top (I + \hat{Q}_t \hat{J}_t H_t^{-1} \hat{J}_t^\top)^{-1} \in \mathbb{R}^{(m+1) \times (m+1)}$, and

$$\tilde{\Phi}_{t+1} = \tilde{\Phi}_t - \alpha_t \tilde{G}_t \hat{e}_t. \quad (13)$$

We also let $\tilde{\Phi}^* \in \mathbb{R}^{m+1}$ denote the vector obtained by augmenting Φ_t^* by 0. Additional regularity terms are explicitly defined in Appendix B.

Theorem 1. *Suppose that conditions A.1, G.1 and R.1 hold for problem (2). Let Φ^* be a universally consistent target function that best interpolates the training data set S . Let $B_R, B_\Phi, B_g, d_g, d_q, \beta, D_g$ and D_R be the regularity terms given by condition R.1, and denote $\hat{\beta}_m \triangleq \sigma_{\min}(J) \equiv \sigma_{\min}(J^\top)$. Then, in the ellipsoid $\mathcal{E}_r(\theta_0)$ for some $r \in \mathbb{R}_{>0}$ and an initialization θ_0 , the regularized GGN for problem (2) satisfies the following properties:*

P.1 Fix $0 < \bar{\alpha}_t \equiv \bar{\alpha} < 1$, and choose $T \triangleq \frac{1}{\bar{\alpha}} \log(\|\tilde{\Phi}_0 - \tilde{\Phi}^*\|^2/\epsilon)$ for any $\epsilon \in (0, 1)$. It holds that $\|\tilde{\Phi}_T - \tilde{\Phi}^*\|^2 \leq \epsilon + 1$ after T iterations, if $1 + M_g \eta_t \leq \|\tilde{G}_t\|_F$ and $|\tilde{G}_{22}| \geq |\langle \tilde{G}_{21}^\top + \tilde{G}_{12}, \tilde{v} \rangle|$ for some \tilde{v} depending on t when $t \leq T$ where, given a 2×2 block partitioning of \tilde{G}_t , $\tilde{G}_{22} \in \mathbb{R}^{1 \times 1}$, $\tilde{G}_{21} \in \mathbb{R}^{1 \times (m+1)}$, and $\tilde{G}_{12} \in \mathbb{R}^{(m+1) \times 1}$ respectively denote the lower right, lower left and upper right blocks of \tilde{G}_t ,

P.2 $\mathcal{L}(\theta_{t+1}) \leq \mathcal{L}(\theta_t) - \left[\vartheta L_{D_t}^2 (1 + D_g \varpi_t) - \xi L_{D_t} \right]$, for $t \in \mathbb{R}_{\geq 0}$, where $L_{D_t} \triangleq \frac{\alpha_t \beta \hat{\beta}_1 D_g}{d_g (D_g + d_q \hat{\beta}_m^2)}$, $\xi \triangleq B_R B_\Phi + B_g$, $\vartheta \triangleq B_\Phi^2 (\gamma_R - D_R)$, $\varpi_t \triangleq \omega_\nu(d_\nu(\theta_t, \theta_{t+1})) - \omega_\nu(-d_\nu(\theta_t, \theta_{t+1}))$, ω_ν is an increasing univariate function, d_ν is a scaled metric term associated with the self-concordance of g , and we assume $d_\nu(\theta_t, \theta_{t+1}) < 1$.

Remark 1. The condition that $1 + M_g \eta_t \leq \|\tilde{G}_t\|_F$ in P.1 highlights an important aspect of the regularization. We often want to control the regularization strength via a parameter $\tau \in \mathbb{R}_{>0}$. Consider this general case in which g takes the form $g(\theta) = \tau \bar{g}(\theta)$. Here, we only require that \bar{g} is GSC so that g satisfies G.1. Observe that $\|\tilde{G}_t\|_F \leq \sqrt{(m+1) \lambda_1(\tilde{G}_t^\top \tilde{G}_t)}$ for all t . Then since $M_g \eta_t$ can become arbitrarily small, it is only reasonable to choose τ satisfying $1 + \tau M_g \eta_0 \leq \sqrt{(m+1) \lambda_1(\tilde{G}_0^\top \tilde{G}_0)}$ in order to have the theoretical guarantee. In this setting, we essentially rely on the local stability of \tilde{G}_t via overparameterization and the self-concordance of g .

Since the direct relation between GGN and NTK probably breaks with an explicit regularization, our main proof step in Theorem 1 involves analyzing a partitioning of the matrix \hat{G}_t . In this way, we determine what conditions on the separate blocks help to combine certain spectral properties of \tilde{G}_t with our regularity conditions and the self-concordance of g . The second result becomes almost immediate in the optimization scope under the regularity conditions.

Under the strong convexity assumption on \hat{R}_s , the global convergence of GGN can be guaranteed in the case of no regularization, for example, by training only the last layer of the NN given the property of *no blow-up* of the GGN dynamics [38, Proposition 1, Proposition 3]. Consider the matrix G^∞ whose (i, j) -th entry is given by $\langle \nabla_v \Phi(x_i; (v, u^\infty)), \nabla_v \Phi(x_j; (v, u^\infty)) \rangle$, where u^∞ is as defined in (12). The main observation here is that the function $v \mapsto r(v) = \hat{R}_s(\Phi(\cdot; (v, u^\infty)))$ can be shown to satisfy a certain Polyak-Lojasiewicz (PL) inequality, that is [38, Proposition 4] $\frac{1}{2} \|\nabla_v l(v)\|^2 \geq \gamma_R \sigma_{\infty}^2 (l(v) - \hat{R}_s(\Phi^*))$, where σ_{∞} is the minimum singular value of G^∞ . With a self-concordant

regularization function g , this kind of global property is retained, provided that g and \hat{R}_s do not conflict. Consider, for example, the sublevel set $\mathcal{S}_{g(\theta)}(g) \triangleq \{\bar{\theta} \in \text{dom } g \mid g(\bar{\theta}) \leq g(\theta)\}$ of g . Then, following [54, Theorem 4], we get that $\mathcal{S}_{g(\theta)}(g)$ is bounded for $\nu \in [2, 3]$, and hence g attains its minimum.

4. Experiments

We present numerical results from experiments performed on GGN with self-concordant regularization (GGN-SCORE) for overparameterized NNs on synthetic datasets as well as on the MNIST dataset. Results of additional experiments on the FashionMNIST and three UCI datasets are reported in Appendix C. The code to reproduce the experimental results is available at <https://github.com/adeyemiadeoye/ggn-score-nn>.

Experimental setup. We consider the *teacher-student* setting in which Φ defined by (3) with $\kappa(n) = 1/\sqrt{n}$ is the student NN, while the teacher NN is the target function Φ^* , a one-hidden layer NN given as

$$\mathbb{R}^{n_0} \ni x \mapsto \Phi^*(x; \theta^*) \triangleq \sum_{i=1}^{n^*} v_i^* \varrho(u_i^* x), \quad (14)$$

where $\theta^* \equiv (u^*, v^*)$. In both teacher and student networks, we use the SiLU activation function [60] $\varrho(x) \triangleq x/(1 + \exp(-x))$. In each experiment, we generate m training data points $(x_i, y_i)_{i \in [m]}$, where the inputs x_i are uniformly sampled on the unit sphere $\mathbb{S}^{n_0-1} \triangleq \{x \mid \|x\| = 1\}$ and the corresponding target outputs are given by $y_i = \Phi^*(x_i; \theta^*)$. The weights of the teacher NN are randomly generated as in [23]: they are normalized random weights satisfying $\|v_i^* u_i^*\| = 1$ for $i = 1, \dots, n^*$. The student NN is initialized with randomly generated weights from the Gaussian distribution. In all the experiments, we fix $n = 500$ and $n^* = 5$. The student NN is trained by minimizing the regularized empirical risk in (2) with the squared loss ℓ (the empirical risk is unregularized for GD), and we consider regularization of the form $g(\theta) = \tau \bar{g}(\theta)$, where $\tau \in \mathbb{R}_{>0}$ and \bar{g} is given by [55, Example 1]: $\bar{g}(\theta) = (\|\cdot\|_1 \square h_\mu)(\theta) = \sum_{i=1}^n \frac{\mu^2 - \mu \sqrt{\mu^2 + \theta_i^2} + \theta_i^2}{\sqrt{\mu^2 + \theta_i^2}}$, $h_\mu(\cdot) \triangleq \mu h(\cdot/\mu)$, $h(\theta) = \sum_{i=1}^n ((1 + |\theta_i|^2)^{1/2} - 1)$, which gives the (M_g, ν) -GSC function

$$g(\theta) = \tau \sum_{i=1}^n \frac{\mu^2 - \mu \sqrt{\mu^2 + \theta_i^2} + \theta_i^2}{\sqrt{\mu^2 + \theta_i^2}}, \quad (15)$$

with $M_g = 2\mu^{-0.7}p^{0.2}$, $\nu = 2.6$ (see Lemma 2). We choose $\mu = 1/\kappa(n)$ and $\tau = 10^{-4}$, except for where we consider different values for comparison. We set $\bar{\alpha}_t \equiv \bar{\alpha} = 0.95$ in (11) for GGN and use a learning rate of 1 for GD. All experiments are performed on a laptop with $16 \times 2.30\text{GHz}$ Intel Core i7-11800H CPU and 32GB RAM.

4.1. Results and discussion

Test loss vs. smoothing parameter. We compare the performance of GGN-SCORE for different values of the regularization smoothing parameter μ evenly spaced in the range $[10^{-3}, 10]$, giving 41 different values in total. We use a reasonable amount of training samples, 500, which allows to

perform several independent runs for each value of μ considered. We use a test size of 1000 to measure generalization of the student NN for each μ . We perform 10 independent runs for each value of μ considered and took the average value of the results. These are shown in Figure 1. We observe that larger values of μ yields better performance in the optimization scope and also better generalization. This result is quite intuitive, since by definition of the regularization function, the size of μ should scale with the size of the variable θ in order to have an adequate smooth approximation of the original nonsmooth function. For this reason, it is recommended to choose $\mu = c/\kappa(n)$ for any $c > 0$ when scaling with $\kappa(n) = 1/\sqrt{n}$.

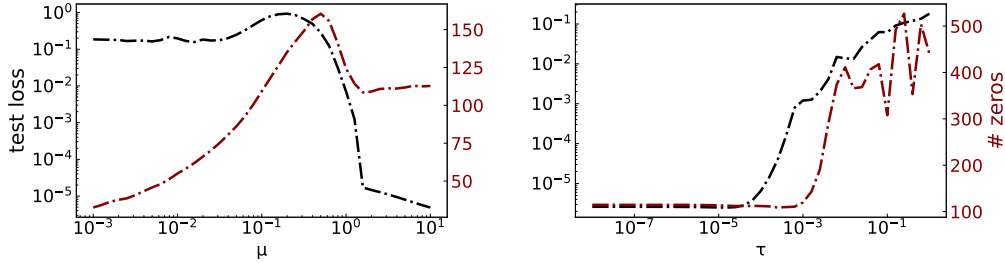


Fig. 1: Performance of GGN-SCORE with $g(\theta)$ as in (15). **Left:** Results for different values of μ , with $\tau = 10^{-4}$. **Right:** Results for different values of τ , with $\mu = 1/\sqrt{n}$. Results are averaged over 10 independent runs for each value of μ and τ , resp.; the total computation time is ~ 21 hours, 2 minutes on CPU.

Test loss vs. regularization strength. We study the influence of the regularization strength on the evolution of the test loss within the optimization loop of GGN-SCORE. We consider different values of τ evenly spaced in the range $[10^{-8}, 1]$, giving 41 different values in total, and set $\mu = 1/\sqrt{n}$. Here, we also use a training size of 500 and a test size 1000 which allows to perform several independent runs for each value of τ considered. We perform 10 independent runs for each value of τ and compute the average value of the results. These average values are shown in Figure 1. We observe that smaller values of τ yield smaller training and test errors for the overparameterized NN. This observation corroborates with the analysis of [39] for GD. However, contrarily to [39], what we observe for the GGN-SCORE is not an arbitrarily small regularization strength to achieve a good generalization performance. In fact, any value of τ slightly smaller than 10^{-4} in our experiment gives a similar generalization error as the choice $\tau = 10^{-6}$ (and smaller). Figure 1 also displays the average number of zero entries in the value of θ at the end of training. As observed, larger values of τ yields a sparser/simpler model. In principle, a desirable value of τ is one which helps to avoid overfitting of the NN model such that a simpler model implies better generalization.

Performance comparison in the optimization loop. We generate training and test datasets of sizes 1000 and 2000, respectively, and compare the training and test losses per iteration and time in seconds between GD and GGN-SCORE for training the student NN. The results are displayed in Figure 2. The dimension of the input data in this experiment is 20, and the number of hidden neurons for the student network is 500. Choosing a much smaller number of hidden neurons $n^* = 5$ for the teacher network keeps the optimization loop in the overparameterized regime. For the GD, we use a learning rate of 1 which yields a much better performance than smaller values. While larger learning rates could yield faster learning at the beginning of training, we notice traces of divergence

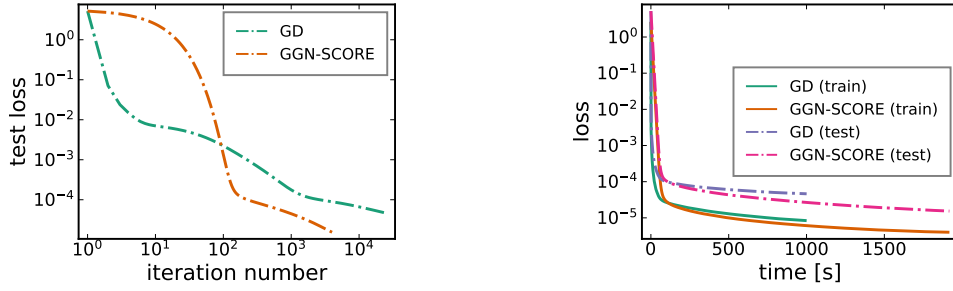


Fig. 2: Performance of GD and GGN-SCORE per iteration number (**left**) and time in seconds (**right**) with $g(\theta)$ as in (15) for GGN-SCORE, $\tau = 10^{-4}$, $\mu = 1/\sqrt{n}$.

later on; a learning rate of 1 gives a reasonably good descent and good performance of GD. We run GD for a total of 10000 steps and GGN-SCORE for a total of 4000 steps. GGN is well-known for its faster convergence in terms of number iterations, while sometimes, we may have to train for a longer time. Results here show that we do not trade total training time for better performance with our GGN-SCORE setup.

4.2. Experiments on real datasets

The computations involved in full-batch GD and/or GGN are intractable on real-world datasets. We study the performance of GGN-SCORE in the mini-batch setting on the standard MNIST dataset [61] with $n_0 = 784$, $m = 60000 : 10000$ (training:test splits). Experimental results in the teacher-student setup according to [38, Appendix C.4] with the SiLU activation function are reported in Appendix C. Additional experiments on three UCI datasets, as well as those on the FashionMNIST dataset, are also considered in the appendix. Here, we consider a NN of the form (3) with a hidden size $n = 512$, a scaling $\kappa(n) = 1/\sqrt{n}$ and the ReLU [62] activation function. The NN is initialized with randomly generated weights from the Gaussian distribution, and is trained with the squared loss. The regularization function g used in GGN-SCORE is given by (15). All results shown for GGN-SCORE are for a training batch size of 16 (i.e., 3750 training steps) and a single epoch. In addition to the test loss and prediction accuracy of the trained model, we adopt a time-invariance “T-I” measure, representing the average proportion (in percentage) of the entries of the *pre-activation* $a_l \in \mathbb{R}^{n \times n_0}$ that satisfy $\text{sgn}(a_{ij}^{\text{start}}) = \text{sgn}(a_{ij}^{\text{final}})$, where sgn is the *signum function*, a_{ij}^{start} are positional entries of a_l at initialization and a_{ij}^{final} are its entries at the end of training. This metric was used in [23] to measure the “stability of activations” where high values indicate an effective linearization of the NN model. See additional details and remark in Appendix C.1.

Influence of the regularization parameters. We investigate the performance of the GGN-SCORE-trained model for different values of the regularization smoothing parameter μ and the regularization strength τ on MNIST dataset. First, we fix $\mu = 1/\sqrt{n}$ and measure the performance of the trained model for different values of τ . Similarly, we fix $\tau = 10^{-4}$ and measure the trained model’s performance for varying values of μ . In each of the two cases, we use 7 different values of τ and μ as is respectively shown in Figure 3 and Figure 4. As in the case with synthetic datasets, optimal choices for τ and μ are seen to necessarily yield good generalization of the model, and are such that give a relatively simple model and stable dynamics (as indicated by the number of zeros in the parameters of the final optimized model and the T-I measure). The total computation time to generate the results

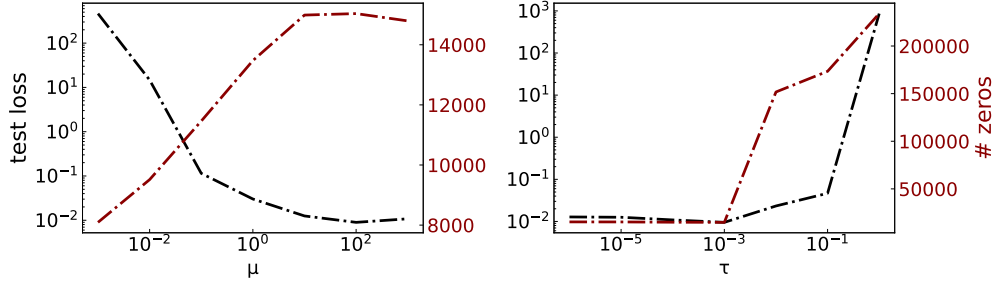


Fig. 3: Test loss evaluation of the GGN-SCORE-trained NN on MNIST dataset for different values of the regularization smoothing parameter μ fixing $\tau = 10^{-4}$ (**left**) and different values of the regularization strength τ fixing $\mu = 1/\sqrt{n}$ (**right**). The regularization function $g(\theta)$ is given by (15).

in Figure 3 and Figure 4 is ~ 23 hours, 5 minutes on CPU.

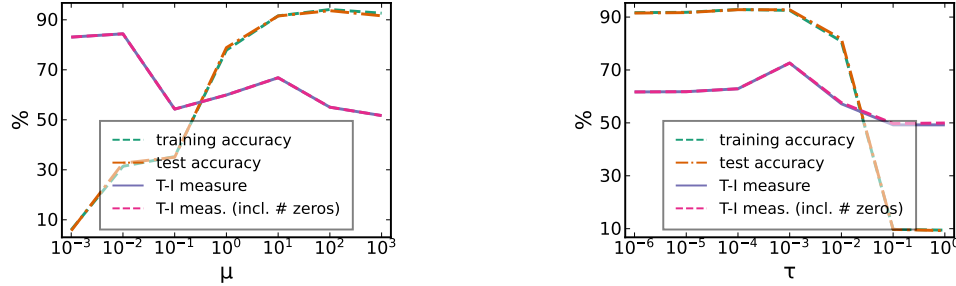


Fig. 4: Evaluation of GGN-SCORE on MNIST dataset for different values of the regularization smoothing parameter μ , fixing $\tau = 10^{-4}$ (**left**), and different values of the regularization strength τ , fixing $\mu = 1/\sqrt{n}$ (**right**). The regularization function $g(\theta)$ is given by (15).

5. Conclusion

We studied a Generalized Gauss-Newton method for optimizing a two-layer neural network with *explicit* regularization and for a specific type of two-layer neural network. We considered the class of generalized self-concordant regularization under which we proved convergence of the neural network predictions to the outputs of a given target function, and have quantified the decay of the problem’s objective function throughout the training iterations. Our experimental findings revealed that good generalization of the optimized neural network model can be achieved with the regularization framework. In future research, we will further investigate the regularization framework with GPU-supported experiments for wider and deeper neural networks and higher dimensional datasets.

References

- [1] Nicolas Roux, Pierre-Antoine Manzagol, and Yoshua Bengio. Topmoumoute online natural gradient algorithm. *Advances in Neural Information Processing Systems*, 20, 2007.

- [2] James Martens et al. Deep learning via Hessian-free optimization. In *ICML*, volume 27, pages 735–742, 2010.
- [3] Oriol Vinyals and Daniel Povey. Krylov subspace descent for deep learning. In *Artificial Intelligence and Statistics*, pages 1261–1268. PMLR, 2012.
- [4] James Martens and Roger Grosse. Optimizing neural networks with Kronecker-factored approximate curvature. In *International Conference on Machine Learning*, pages 2408–2417. PMLR, 2015.
- [5] Aleksandar Botev, Hippolyt Ritter, and David Barber. Practical gauss-newton optimisation for deep learning. In *International Conference on Machine Learning*, pages 557–565. PMLR, 2017.
- [6] Michael Arbel, Arthur Gretton, Wuchen Li, and Guido Montúfar. Kernelized wasserstein natural gradient. *arXiv preprint arXiv:1910.09652*, 2019.
- [7] Yi Ren and Donald Goldfarb. Efficient subsampled Gauss-Newton and natural gradient methods for training neural networks. *arXiv preprint arXiv:1906.02353*, 2019.
- [8] Tianle Cai, Ruiqi Gao, Jikai Hou, Siyu Chen, Dong Wang, Di He, Zhihua Zhang, and Liwei Wang. Gram-gauss-newton method: Learning overparameterized neural networks for regression problems. *arXiv preprint arXiv:1905.11675*, 2019.
- [9] Adeyemi D Adeoye and Alberto Bemporad. SCORE: approximating curvature information under self-concordant regularization. *Computational Optimization and Applications*, 86(2):599–626, 2023. ISSN 0926-6003. doi: 10.1007/s10589-023-00502-2.
- [10] Nicol N Schraudolph. Fast curvature matrix-vector products for second-order gradient descent. *Neural Computation*, 14(7):1723–1738, 2002.
- [11] Razvan Pascanu and Yoshua Bengio. Revisiting natural gradient for deep networks. *arXiv preprint arXiv:1301.3584*, 2013.
- [12] James Martens. New insights and perspectives on the natural gradient method. *The Journal of Machine Learning Research*, 21(1):5776–5851, 2020.
- [13] Pei Chen. Hessian matrix vs. Gauss–Newton hessian matrix. *SIAM Journal on Numerical Analysis*, 49(4):1417–1435, 2011.
- [14] Alberto Bemporad. Training recurrent neural networks by sequential least squares and the alternating direction method of multipliers. *Automatica*, 156:111183, 2023.
- [15] Amit Daniely. SGD learns the conjugate kernel class of the network. *Advances in Neural Information Processing Systems*, 30, 2017.
- [16] Yuanzhi Li and Yingyu Liang. Learning overparameterized neural networks via stochastic gradient descent on structured data. *Advances in Neural Information Processing Systems*, 31, 2018.
- [17] Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

- [18] Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, pages 1675–1685. PMLR, 2019.
- [19] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pages 242–252. PMLR, 2019.
- [20] Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Stochastic gradient descent optimizes over-parameterized deep relu networks. *arXiv preprint arXiv:1811.08888*, 2018.
- [21] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in Neural Information Processing Systems*, 31, 2018.
- [22] Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pages 322–332. PMLR, 2019.
- [23] Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in Neural Information Processing Systems*, 32, 2019.
- [24] Greg Yang. Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation. *arXiv preprint arXiv:1902.04760*, 2019.
- [25] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.
- [26] Yuanzhi Li, Tengyu Ma, and Hongyang Zhang. Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations. In *Conference On Learning Theory*, pages 2–47. PMLR, 2018.
- [27] Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. In *International Conference on Machine Learning*, pages 1832–1841. PMLR, 2018.
- [28] Ziwei Ji and Matus Telgarsky. The implicit bias of gradient descent on nonseparable data. In *Conference on Learning Theory*, pages 1772–1798. PMLR, 2019.
- [29] Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. *Advances in Neural Information Processing Systems*, 32, 2019.
- [30] Lenaic Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Conference on Learning Theory*, pages 1305–1338. PMLR, 2020.
- [31] Zhiyuan Li, Tianhao Wang, and Sanjeev Arora. What Happens after SGD Reaches Zero Loss?—A Mathematical Framework. *arXiv preprint arXiv:2110.06914*, 2021.

- [32] Salma Tarmoun, Guilherme Franca, Benjamin D Haeffele, and Rene Vidal. Understanding the dynamics of gradient flow in overparameterized linear models. In *International Conference on Machine Learning*, pages 10153–10161. PMLR, 2021.
- [33] Julius Berner, Philipp Grohs, Gitta Kutyniok, and Philipp Petersen. The modern mathematics of deep learning. *arXiv preprint arXiv:2105.04026*, pages 86–114, 2021.
- [34] Guodong Zhang, James Martens, and Roger B Grosse. Fast convergence of natural gradient descent for over-parameterized neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.
- [35] Ryo Karakida and Kazuki Osawa. Understanding approximate fisher information for fast convergence of natural gradient descent in wide neural networks. *Advances in Neural Information Processing Systems*, 33:10891–10901, 2020.
- [36] Anna Kerekes, Anna Mészáros, and Ferenc Huszár. Depth Without the Magic: Inductive Bias of Natural Gradient Descent. *arXiv preprint arXiv:2111.11542*, 2021.
- [37] Jezabel R Garcia, Federica Freddi, Stathi Fotiadis, Maolin Li, Sattar Vakili, Alberto Bernacchia, and Guillaume Hennequin. Fisher-Legendre (FishLeg) optimization of deep neural networks. In *The Eleventh International Conference on Learning Representations*, 2022.
- [38] Michael Arbel. Rethinking Gauss-Newton for learning over-parameterized models. *arXiv preprint arXiv:2302.02904*, 2023.
- [39] Colin Wei, Jason D Lee, Qiang Liu, and Tengyu Ma. Regularization matters: Generalization and optimization of neural nets vs their induced kernel. *Advances in Neural Information Processing Systems*, 32, 2019.
- [40] Anant Raj and Francis Bach. Explicit regularization of stochastic gradient methods through duality. In *International Conference on Artificial Intelligence and Statistics*, pages 1882–1890. PMLR, 2021.
- [41] Antonio Orvieto, Anant Raj, Hans Kersting, and Francis Bach. Explicit regularization in over-parametrized models via noise injection. In *International Conference on Artificial Intelligence and Statistics*, pages 7265–7287. PMLR, 2023.
- [42] Stephen Hanson and Lorien Pratt. Comparing biases for minimal network construction with back-propagation. *Advances in Neural Information Processing Systems*, 1, 1988.
- [43] Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26:289–315, 2007.
- [44] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [45] Alessandro Rudi, Raffaello Camoriano, and Lorenzo Rosasco. Less is more: Nyström computational regularization. *Advances in Neural Information Processing Systems*, 28, 2015.

- [46] Art B Owen. Self-concordance for empirical likelihood. *Canadian Journal of Statistics*, 41(3): 387–397, 2013.
- [47] R Tyrrell Rockafellar and Roger J-B Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.
- [48] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *Advances in Neural Information Processing Systems*, 32, 2019.
- [49] James Martens and Ilya Sutskever. Learning recurrent neural networks with hessian-free optimization. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1033–1040, 2011.
- [50] Tengyuan Liang and Alexander Rakhlin. Just interpolate: Kernel “Ridgeless” regression can generalize. *The Annals of Statistics*, 48(3):1329 – 1347, 2020. doi: 10.1214/19-AOS1849. URL <https://doi.org/10.1214/19-AOS1849>.
- [51] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.
- [52] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in Neural Information Processing Systems*, 20, 2007.
- [53] Maria-Florina Balcan, Avrim Blum, and Santosh Vempala. Kernels as features: On kernels, margins, and low-dimensional mappings. *Machine Learning*, 65:79–94, 2006.
- [54] Tianxiao Sun and Quoc Tran-Dinh. Generalized self-concordant functions: a recipe for Newton-type methods. *Mathematical Programming*, 178(1-2):145–213, 2019.
- [55] Adeyemi D Adeoye and Alberto Bemporad. Self-concordant smoothing for large-scale convex composite optimization. *arXiv preprint arXiv:2309.01781*, 2024.
- [56] Sai Praneeth Karimireddy, Sebastian U Stich, and Martin Jaggi. Global linear convergence of Newton’s method without strong-convexity or Lipschitz gradients. *arXiv preprint arXiv:1806.00413*, 2018.
- [57] Robert Gower, Dmitry Kovalev, Felix Lieder, and Peter Richtárik. RSN: randomized subspace Newton. *Advances in Neural Information Processing Systems*, 32, 2019.
- [58] Yair Carmon, Arun Jambulapati, Qijia Jiang, Yujia Jin, Yin Tat Lee, Aaron Sidford, and Kevin Tian. Acceleration with a ball optimization oracle. *Advances in Neural Information Processing Systems*, 33:19052–19063, 2020.
- [59] Andrea Caponnetto and Ernesto De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7:331–368, 2007.
- [60] Stefan Elfving, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018.
- [61] Yann LeCun, Corinna Cortes, and CJ Burges. MNIST handwritten digit database. *ATT Labs [Online]*. Available: <http://yann.lecun.com/exdb/mnist>, 2, 2010.

- [62] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.
- [63] Sheng-De Wang, Te-Son Kuo, and Chen-Fa Hsu. Trace bounds on the solution of the algebraic matrix Riccati and Lyapunov equation. *IEEE Transactions on Automatic Control*, 31(7):654–656, 1986.
- [64] S.S. Haykin. *Neural Networks and Learning Machines*. Pearson International Edition. Pearson, 2009. ISBN 9780131293762. URL <https://books.google.it/books?id=KCwW0AAACAAJ>.
- [65] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *CoRR*, abs/1708.07747, 2017. URL <http://arxiv.org/abs/1708.07747>.

A. Preliminary results

A.1. Lipschitz continuity of J

Here, we look at the Lipschitz property of the Jacobian matrix J . For this, we need the following additional standard assumptions:

J.1 $\|x\| = \|x^\top\| \leq 1$.

J.2 $\exists L_\varrho$ such that $\|\varrho(\bar{x}) - \varrho(\tilde{x})\| \leq L_\varrho \|\bar{x} - \tilde{x}\|$ and $\|\varrho'(\bar{x}) - \varrho'(\tilde{x})\| \leq L_\varrho \|\bar{x} - \tilde{x}\|$ for all $\bar{x}, \tilde{x} \in \mathbb{R}$.

J.3 $\exists L_v$ such that $\|v\| \leq L_v$ at all time t until the training is stopped.

Condition **J.2** simply restates the Lipschitzness and smoothness assumptions in **A.1** explicitly. This kind of condition has been used, for example in [18, Condition 3.1], to show the stability of the training process of NNs via gradient descent. A consequence of the condition is that it also provides an upper bound on the gradients of ϱ , that is, $\|\varrho'(\bar{x})\| \leq L_\varrho$ for all $\bar{x} \in \mathbb{R}$ (see, e.g., Lemma 3 below).

Proposition 1 (Lipschitz constant of J ; training both layers). *Under assumptions **J.1**, **J.2** and **J.3**, J is L_J -Lipschitz continuous, where $L_J \triangleq m\kappa(n)(1 + L_v)L_\varrho\sqrt{2}$.*

Proof. Let $(\bar{u}, \bar{v}) \equiv \bar{\theta}$, $(\tilde{u}, \tilde{v}) \equiv \tilde{\theta}$ for any $\bar{\theta}, \tilde{\theta} \in \mathbb{R}^p$. We have

$$\begin{aligned}
 \|J(\bar{\theta}) - J(\tilde{\theta})\| &\leq \kappa(n) \left(\sum_{i=1}^n \left| \varrho'(\bar{u}_i x) \bar{v}_i x^\top - \varrho'(\tilde{u}_i x) \tilde{v}_i x^\top \right| + \sum_{i=1}^n \left| \varrho(\bar{u}_i x) - \varrho(\tilde{u}_i x) \right| \right) \\
 &\leq \kappa(n) \left(\sum_{i=1}^n \left(\left| \varrho'(\bar{u}_i x) \bar{v}_i x^\top - \varrho'(\tilde{u}_i x) \bar{v}_i x^\top \right| + \left| \varrho'(\tilde{u}_i x) \bar{v}_i x^\top - \varrho'(\tilde{u}_i x) \tilde{v}_i x^\top \right| \right) \right. \\
 &\quad \left. + \sum_{i=1}^n \left| \varrho(\bar{u}_i x) - \varrho(\tilde{u}_i x) \right| \right)
 \end{aligned}$$

$$\begin{aligned}
 &\leq \kappa(n) \left(\sum_{i=1}^n (|\varrho'(\bar{u}_i x) - \varrho'(\tilde{u}_i x)| |\bar{v}_i| \|x\| + |\bar{v}_i - \tilde{v}_i| |\varrho'(\tilde{u}_i x)| \|x\|) + \sum_{i=1}^n |\varrho(\bar{u}_i x) - \varrho(\tilde{u}_i x)| \right) \\
 &\leq m\kappa(n) ((1 + L_v)L_\varrho \|\bar{u} - \tilde{u}\| + L_\varrho \|\bar{v} - \tilde{v}\|) \\
 &\leq m\kappa(n)(1 + L_v)L_\varrho (\|\bar{u} - \tilde{u}\| + \|\bar{v} - \tilde{v}\|) \\
 &= m\kappa(n)(1 + L_v)L_\varrho (\|(\bar{u}, 0) - (\tilde{u}, 0)\| + \|(0, \bar{v}) - (0, \tilde{v})\|). \tag{16}
 \end{aligned}$$

Next, we recall Peter-Paul inequality for two quantities $a, b \in \mathbb{R}_{\geq 0}$ which reads $2ab \leq a^2 + b^2$, from which we obtain

$$(a + b)^2 = a^2 + 2ab + b^2 \leq a^2 + a^2 + b^2 + b^2 = 2(a^2 + b^2). \tag{17}$$

We also derive the expression

$$\begin{aligned}
 \|(\bar{u}, 0) - (\tilde{u}, 0)\|^2 + \|(0, \bar{v}) - (0, \tilde{v})\|^2 &= \|(\bar{u}, 0)\|^2 - 2\langle(\bar{u}, 0), (\tilde{u}, 0)\rangle + \|(\tilde{u}, 0)\|^2 \\
 &\quad + \|(0, \bar{v})\|^2 - 2\langle(0, \bar{v}), (0, \tilde{v})\rangle + \|(0, \tilde{v})\|^2 \\
 &= \|(\bar{u}, \bar{v})\|^2 - 2\langle(\bar{u}, \bar{v}), (\tilde{u}, \tilde{v})\rangle + \|(\tilde{u}, \tilde{v})\|^2 \\
 &= \|(\bar{u}, \bar{v}) - (\tilde{u}, \tilde{v})\|^2. \tag{18}
 \end{aligned}$$

Now, using (17) and (18) in (16) with $a = \|(\bar{u}, 0) - (\tilde{u}, 0)\|$ and $b = \|(0, \bar{v}) - (0, \tilde{v})\|$, we get

$$\|J(\bar{\theta}) - J(\tilde{\theta})\| \leq m\kappa(n)(1 + L_v)L_\varrho\sqrt{2}\|(\bar{u}, \bar{v}) - (\tilde{u}, \tilde{v})\|,$$

which proves the result. \square

A.2. Useful results on the generalized self-concordance of g

We define the following metric term for the regularization function g (under condition G.1). As is customary, our results are restricted to the case $\nu \in [2, 3]$.

$$d_\nu(\bar{\theta}, \tilde{\theta}) \triangleq \begin{cases} M_g \|\tilde{\theta} - \bar{\theta}\| & \text{if } \nu = 2, \\ \left(\frac{\nu}{2} - 1\right) M_g \|\tilde{\theta} - \bar{\theta}\|_2^{3-\nu} \|\tilde{\theta} - \bar{\theta}\|_{\bar{\theta}}^{\nu-2} & \text{if } \nu > 2. \end{cases} \tag{19}$$

Lemma 1. [54, Proposition 10] Under condition G.1, we have for any $\bar{\theta}, \tilde{\theta} \in \text{dom } g$

$$\omega_\nu(-d_\nu(\bar{\theta}, \tilde{\theta})) \|\tilde{\theta} - \bar{\theta}\|_{\bar{\theta}}^2 \leq g(\tilde{\theta}) - g(\bar{\theta}) - \langle \nabla g(\bar{\theta}), \tilde{\theta} - \bar{\theta} \rangle \leq \omega_\nu(d_\nu(\bar{\theta}, \tilde{\theta})) \|\tilde{\theta} - \bar{\theta}\|_{\bar{\theta}}^2, \tag{20}$$

in which, if $\nu > 2$, the right-hand side inequality holds if $d_\nu(\bar{\theta}, \tilde{\theta}) < 1$, and

$$\omega_\nu(r) \triangleq \begin{cases} \frac{\exp(r)-r-1}{r^2} & \text{if } \nu = 2, \\ \frac{-r-\ln(1-r)}{r^2} & \text{if } \nu = 3, \\ \frac{(1-r)\ln(1-r)+r}{r^2} & \text{if } \nu = 4, \\ \left(\frac{\nu-2}{4-\nu}\right) \frac{1}{r} \left[\frac{\nu-2}{2(3-\nu)r} \left((1-r)^{\frac{2(3-\nu)}{2-\nu}} - 1 \right) - 1 \right] & \text{otherwise.} \end{cases} \tag{21}$$

In case g is nonsmooth and hence does not satisfy the condition G.1 but is closed, proper and convex, the following result from [55, Proposition 2] shows some properties of a self-concordant smoothing function for g constructed in the sense of Definition 2.

Lemma 2. *Let \bar{g}, h be two functions in $\Gamma_0(\mathbb{R}^p)$. Suppose that h is (M_h, ν) -GSC and supercoercive, and define $g \triangleq \bar{g} \square h_\mu$ for all $\mu \in \mathbb{R}_{>0}$, where $h_\mu(\cdot) \triangleq \mu h\left(\frac{\cdot}{\mu}\right)$ and $\bar{g} \square h_\mu$ denotes the infimal convolution of \bar{g} and h_μ defined by*

$$(\bar{g} \square h_\mu)(\bar{\theta}) \triangleq \inf_{\tilde{\theta} \in \mathbb{R}^p} \left\{ \bar{g}(\tilde{\theta}) + h_\mu(\bar{\theta} - \tilde{\theta}) \right\}. \quad (22)$$

Then,

(i) $g \in \Gamma_0(\mathbb{R}^p)$ and is exact.

(ii) g is (M_g, ν) -GSC with

$$M_g = \begin{cases} p^{\frac{3-\nu}{2}} \mu^{\frac{\nu}{2}-2} M_h, & \text{if } \nu \in (0, 3], \\ \mu^{4-\frac{3\nu}{2}} M_h, & \text{if } \nu > 3. \end{cases}$$

(iii) g is locally Lipschitz continuous.

Lemma 3. *Let g be a convex and (locally) L -Lipschitz function. Then,*

$$\|\nabla g(\bar{\theta})\| \leq L,$$

for some $\bar{\theta}$ in a set $\mathcal{X} \subset \mathbb{R}^p$.

Proof. Take some $\tilde{\theta} = \bar{\theta} + \alpha \nabla g(\bar{\theta})$ for $\alpha \in \mathbb{R}_{>0}$ small enough. By the convexity and Lipschitzness of g , we have

$$\begin{aligned} \|\alpha \nabla g(\bar{\theta})\|^2 &= \alpha^2 |\langle \nabla g(\bar{\theta}), \nabla g(\bar{\theta}) \rangle| \\ &= \alpha |\langle \tilde{\theta} - \bar{\theta}, \nabla g(\bar{\theta}) \rangle| \\ &\leq \alpha |g(\tilde{\theta}) - g(\bar{\theta})| \\ &\leq \alpha L |\tilde{\theta} - \bar{\theta}| \\ &= \alpha^2 L \|\nabla g(\bar{\theta})\|, \end{aligned}$$

which completes the proof. \square

B. Proof of the main result

Detailed regularity assumptions in R.1. We detail missing regularity terms in condition R.1 as follows. For this, we define the ball $\mathcal{B}_{r_0}(\theta_0) \subset \mathcal{E}_r(\theta_0)$ for some initialization θ_0 . Note that, corresponding to the ellipsoid $\mathcal{E}_r(\theta_0)$, we compute local norms with respect to g .

RR.1 $\hat{R}_s(\Phi(\cdot, \bar{\theta})) \geq \hat{R}_s(\Phi(\cdot, \tilde{\theta})) + \langle \nabla_\Phi \hat{R}_s(\Phi(\cdot, \tilde{\theta})), \Phi(\cdot, \bar{\theta}) - \Phi(\cdot, \tilde{\theta}) \rangle + \frac{\gamma_R}{2} \|\Phi(\cdot, \bar{\theta}) - \Phi(\cdot, \tilde{\theta})\|^2 \forall \bar{\theta}, \tilde{\theta} \in \mathbb{R}^p$,
and $\exists B_R, D_R$ such that $\|\nabla_\Phi \hat{R}_s(\Phi(\cdot, \bar{\theta}))\| \leq B_R, \|\nabla_\Phi^2 \hat{R}_s(\Phi(\cdot, \bar{\theta}))\|_{op} \leq D_R \forall \bar{\theta} \in \mathcal{B}_{r_0}(\theta_0)$.

RR.2 $d_g I \leq H_t \leq D_g I$ with $D_g \geq d_g > 0$, $d_q I \leq \hat{Q}_t \leq D_q I$ with $D_q \geq d_q \geq 0$, and $\|\hat{e}_t\| \leq \beta$ $\forall \theta_t \in \mathcal{B}_{r_0}(\theta_0)$.

Using the Lipschitzness of ϱ , one can easily find some B_Φ satisfying $\|\Phi(\cdot, \bar{\theta}) - \Phi(\cdot, \tilde{\theta})\| \leq B_\Phi \|\bar{\theta} - \tilde{\theta}\|$ for some $\bar{\theta}, \tilde{\theta} \in \mathbb{R}^p$ at least near the initialization. Hence, we do not impose this regularity property as an additional assumption. Subsequently, we recall the following notations: $L_{D_t} \triangleq \frac{\alpha_t \beta \hat{\beta}_1 D_g}{d_g (D_g + d_q \hat{\beta}_m^2)}$, $\xi \triangleq B_R B_\Phi + B_g$, $\vartheta \triangleq B_\Phi^2 (\gamma_R - D_R)$, $\varpi_t \triangleq \omega_\nu(d_\nu(\theta_t, \theta_{t+1})) - \omega_\nu(-d_\nu(\theta_t, \theta_{t+1}))$, $\hat{\beta}_m \triangleq \sigma_{\min}(\hat{J}_t)$. We also introduce the notations $\hat{\beta}_1 \triangleq \sigma_{\max}(\hat{J}_t)$ and $\delta_t \triangleq \theta_{t+1} - \theta_t$.

Lemma 4. *Under assumption RR.2, we have*

$$\|\delta_t\| \leq L_{D_t}, \quad \|\delta_t\|_{\theta_t} \leq \sqrt{D_g} L_{D_t}.$$

Proof. We obtain the following estimate

$$\left\| H_t^{-1} \hat{J}_t^\top \right\| \leq \left\| H_t^{-1} \right\| \left\| \hat{J}_t^\top \right\| \leq \frac{\beta_1}{d_g}. \quad (23)$$

We have also

$$\left\| I + \hat{Q}_t \hat{J}_t H_t^{-1} \hat{J}_t^\top \right\| \geq 1 + \frac{d_q \hat{\beta}_m^2}{D_g}. \quad (24)$$

From (9), we have

$$\|\delta_t\| \leq \alpha_t \left\| H_t^{-1} \hat{J}_t^\top (I + \hat{Q}_t \hat{J}_t H_t^{-1} \hat{J}_t^\top)^{-1} \hat{e}_t \right\|. \quad (25)$$

Using (23) and (24) in (25), we obtain

$$\|\delta_t\| \leq L_{D_t}.$$

The result follows, noting that $\|\delta_t\|_{\theta_t} = \left\| H_t^{1/2} \delta_t \right\|$ by definition. \square

We obtain the following slightly loose estimate of the Lipschitz constant of the objective function \mathcal{L} in problem (2).

Lemma 5. *Let g be constructed under the settings of Lemma 2 such that condition G.1 holds. Then, under the additional condition RR.1, the objective function \mathcal{L} is $(B_R B_\Phi + B_g)$ -Lipschitz continuous, where B_g is the local Lipschitz constant of g in Lemma 2(iii).*

Proof. By the convexity of \hat{R}_s and g , we have

$$\hat{R}_s(\Phi(\cdot, \tilde{\theta})) \geq \hat{R}_s(\Phi(\cdot, \bar{\theta})) + \langle \nabla \hat{R}_s(\Phi(\cdot, \tilde{\theta})), \Phi(\cdot, \tilde{\theta}) - \Phi(\cdot, \bar{\theta}) \rangle, \quad g(\tilde{\theta}) \geq g(\bar{\theta}) + \langle \nabla g(\bar{\theta}), \tilde{\theta} - \bar{\theta} \rangle,$$

for some $\tilde{\theta}, \bar{\theta}$ in the vicinity of θ_0 . Then, using $\mathcal{L} \triangleq \hat{R}_s + g$ and the Cauchy-Schwarz inequality, we get

$$\left| \mathcal{L}(\bar{\theta}) - \mathcal{L}(\tilde{\theta}) \right| \leq \left\| \nabla_\Phi \hat{R}_s(\Phi(\cdot, \bar{\theta})) \right\| \left\| \Phi(\cdot, \bar{\theta}) - \Phi(\cdot, \tilde{\theta}) \right\| + \left\| \nabla g(\bar{\theta}) \right\| \left\| \bar{\theta} - \tilde{\theta} \right\|.$$

By assumption [RR.1](#), we have $\left\| \nabla_{\Phi} \hat{R}_s(\Phi(\cdot, \bar{\theta})) \right\| \leq B_R$. By Lemma [2\(iii\)](#), g is locally Lipschitz, and hence we have $\left\| \nabla g(\bar{\theta}) \right\| \leq B_g$ for some B_g , according to Lemma [3](#). Then, using the local Lipschitz property of Φ , we obtain

$$\begin{aligned} \left| \mathcal{L}(\bar{\theta}) - \mathcal{L}(\tilde{\theta}) \right| &\leq B_R B_{\Phi} \left\| \bar{\theta} - \tilde{\theta} \right\| + B_g \left\| \bar{\theta} - \tilde{\theta} \right\| \\ &= (B_R B_{\Phi} + B_g) \left\| \bar{\theta} - \tilde{\theta} \right\|. \end{aligned}$$

□

The following result from [[63](#), Lemma 1] provides a useful inequality for the trace of the product of two symmetric matrices, one of which is positive semidefinite.

Lemma 6. *Let $P, Q \in \mathbb{R}^{n \times n}$. If $P = P^{\top} \succeq 0$ and Q is symmetric, then*

$$\text{tr}(P) \lambda_n(Q) \leq \text{tr}(PQ) \leq \text{tr}(P) \lambda_1(Q).$$

The next result concerns the 2×2 block partitioning of \tilde{G}_t , and characterizes the positive-definiteness of its leading principal blocks. For this, we require that the function g is such that $H_t \succ 0$. This, indeed, is a property of many functions constructed from the ℓ_1 -norm in the sense of Definition [2](#). An example is the pseudo-Huber function or the function \bar{g} considered in Section [4](#).

Lemma 7. *Consider a 2×2 block partitioning of \tilde{G}_t , and let $\tilde{G}_{11} \in \mathbb{R}^{m \times m}$, $\tilde{G}_{22} \in \mathbb{R}^{1 \times 1}$ respectively denote the upper left and lower right blocks. If $H_t \succ 0$, then it holds that $\tilde{G}_{22} \in \mathbb{R}_{>0}$ and $\tilde{G}_{11} \succ 0$.*

Proof. By the definition of \tilde{G}_t and using [\(8\)](#), we have $\tilde{G}_t = \tilde{J}_t(\hat{J}_t^{\top} \hat{Q}_t \hat{J}_t + H_t)^{-1} \hat{J}_t^{\top}$. We note that for the squared loss that we consider, Q_t is the identity matrix and that we can write $\tilde{G}_t = \tilde{J}_t(J_t^{\top} Q_t J_t + H_t)^{-1} \hat{J}_t^{\top} = \tilde{J}_t(J_t^{\top} J_t + H_t)^{-1} \hat{J}_t^{\top}$. Notice the removal of the augmentations, as the last diagonal entry of \hat{Q}_t is zero. We have $\langle \hat{v}, J_t^{\top} J_t \hat{v} \rangle = \|J_t \hat{v}\|^2 \geq 0$ for all non-zero $\hat{v} \in \mathbb{R}^n$, and hence $B_t \triangleq J_t^{\top} J_t + H_t \succ 0$. Next, observe that \tilde{G}_{11} results from removing the augmentations on \tilde{J} and \hat{Q}_t in \tilde{G}_t , that is, $\tilde{G}_{11} \equiv J_t B_t^{-1} J_t^{\top}$. Let $\hat{u} \triangleq B_t \hat{v}$; we have $\langle \hat{u}, B_t^{-1} \hat{u} \rangle = \langle B_t \hat{v}, B_t^{-1} B_t \hat{v} \rangle = \langle \hat{v}, B_t^{\top} \hat{v} \rangle \succ 0$. Then, in a similar way, if J_t does not have all its entries equal to zero, we get that $\tilde{G}_{11} \succ 0$.

To show $\tilde{G}_{22} \in \mathbb{R}_{>0}$, we note that since $B_t \succ 0$, it has a non-zero determinant, and hence by Sylvester's criterion, we have $\frac{1}{\det(B_t)}((-1)^{2n} M_{n,n}) > 0$, where $\det(B_t)$ denotes the determinant of B_t and $M_{n,n}$ denotes the (n, n) -th minor of B_t . Then, $\tilde{G}_{22} \in \mathbb{R}_{>0}$ follows from the definition of \tilde{J} . □

We are now ready to prove our main result.

Proof of Theorem [1](#).

Proof. Consider the time evolution of the regularized NN given by [\(10\)](#). Using the augmentation specified by [\(13\)](#), we have

$$\begin{aligned} \left\| \tilde{\Phi}_{t+1} - \tilde{\Phi}^* \right\|^2 &= \left\| \tilde{\Phi}_t - \alpha_t \tilde{G}_t \hat{e}_t - \tilde{\Phi}^* \right\|^2 \\ &= \left\| \tilde{\Phi}_t - \tilde{\Phi}^* \right\|^2 - 2\alpha_t \langle \tilde{\Phi}_t - \tilde{\Phi}^*, \tilde{G}_t(\tilde{\Phi}_t - \tilde{\Phi}^*) \rangle + \alpha_t^2 \left\| \tilde{G}_t(\tilde{\Phi}_t - \tilde{\Phi}^*) \right\|^2. \end{aligned} \quad (26)$$

Let us partition $\tilde{\Phi}_t - \tilde{\Phi}^*$ and \tilde{G}_t as follows (omitting dependence on t in the blocks for brevity):

$$\tilde{\Phi}_t - \tilde{\Phi}^* \equiv \begin{bmatrix} \tilde{\Phi}_1 \\ \tilde{\Phi}_2 \end{bmatrix}, \quad \tilde{G}_t \equiv \begin{bmatrix} \tilde{G}_{11} & \tilde{G}_{12} \\ \tilde{G}_{21} & \tilde{G}_{22} \end{bmatrix}, \quad (27)$$

where $\tilde{\Phi}_1 = \Phi_t - \Phi^* \in \mathbb{R}^m$, $\tilde{\Phi}_2 = 1$ and hence $\tilde{G}_{11} \in \mathbb{R}^{m \times m}$. Then, we have

$$\begin{aligned} \langle \tilde{\Phi}_t - \tilde{\Phi}^*, \tilde{G}_t(\tilde{\Phi}_t - \tilde{\Phi}^*) \rangle &= \langle \tilde{\Phi}_1, \tilde{G}_{11}\tilde{\Phi}_1 \rangle + \langle \tilde{\Phi}_2, \tilde{G}_{21}\tilde{\Phi}_1 \rangle + \langle \tilde{\Phi}_1, \tilde{G}_{12}\tilde{\Phi}_2 \rangle + \langle \tilde{\Phi}_2, \tilde{G}_{22}\tilde{\Phi}_2 \rangle \\ &= \langle \tilde{\Phi}_1, \tilde{G}_{11}\tilde{\Phi}_1 \rangle + \langle \tilde{G}_{21} + \tilde{G}_{12}^\top, \tilde{\Phi}_1 \rangle + \tilde{G}_{22}, \end{aligned} \quad (28)$$

where we have used $\tilde{\Phi}_2 = 1$. Recall that by Lemma 7, we get $\tilde{G}_{22} \in \mathbb{R}_{>0}$ and $\tilde{G}_{11} \succ 0$.

Using the block partitioning of \tilde{G}_t in (27), the product $\tilde{G}_t^\top \tilde{G}_t$ gives the following block structure

$$\tilde{G}_t^\top \tilde{G}_t = \begin{bmatrix} (\tilde{G}_t^\top \tilde{G}_t)_{11} & (\tilde{G}_t^\top \tilde{G}_t)_{12} \\ (\tilde{G}_t^\top \tilde{G}_t)_{21} & (\tilde{G}_t^\top \tilde{G}_t)_{22} \end{bmatrix} \triangleq \begin{bmatrix} \tilde{G}_{11}^\top \tilde{G}_{11} + \tilde{G}_{21}^\top \tilde{G}_{21} & \tilde{G}_{11}^\top \tilde{G}_{12} + \tilde{G}_{21}^\top \tilde{G}_{22} \\ \tilde{G}_{12}^\top \tilde{G}_{11} + \tilde{G}_{22}^\top \tilde{G}_{21} & \tilde{G}_{12}^\top \tilde{G}_{12} + \tilde{G}_{22}^\top \tilde{G}_{22} \end{bmatrix}.$$

Consider the congruence

$$\begin{aligned} &\begin{bmatrix} (\tilde{G}_t^\top \tilde{G}_t)_{11} & (\tilde{G}_t^\top \tilde{G}_t)_{12} \\ (\tilde{G}_t^\top \tilde{G}_t)_{21} & (\tilde{G}_t^\top \tilde{G}_t)_{22} \end{bmatrix} \sim \\ &\begin{bmatrix} (\tilde{G}_t^\top \tilde{G}_t)_{11}^{-1/2} & 0 \\ 0 & (\tilde{G}_t^\top \tilde{G}_t)_{22}^{-1/2} \end{bmatrix} \begin{bmatrix} (\tilde{G}_t^\top \tilde{G}_t)_{11} & (\tilde{G}_t^\top \tilde{G}_t)_{12} \\ (\tilde{G}_t^\top \tilde{G}_t)_{21} & (\tilde{G}_t^\top \tilde{G}_t)_{22} \end{bmatrix} \begin{bmatrix} (\tilde{G}_t^\top \tilde{G}_t)_{11}^{-1/2} & 0 \\ 0 & (\tilde{G}_t^\top \tilde{G}_t)_{22}^{-1/2} \end{bmatrix} \\ &= \begin{bmatrix} I & (\tilde{G}_t^\top \tilde{G}_t)_{11}^{-1/2}(\tilde{G}_t^\top \tilde{G}_t)_{12}(\tilde{G}_t^\top \tilde{G}_t)_{22}^{-1/2} \\ (\tilde{G}_t^\top \tilde{G}_t)_{22}^{-1/2}(\tilde{G}_t^\top \tilde{G}_t)_{21}(\tilde{G}_t^\top \tilde{G}_t)_{11}^{-1/2} & I \end{bmatrix}. \end{aligned}$$

Using this relation, one can show that $\tilde{G}_t^\top \tilde{G}_t \succ 0$; since $(\tilde{G}_t^\top \tilde{G}_t)_{21} = (\tilde{G}_t^\top \tilde{G}_t)_{12}^\top$, we only require that $\|(\tilde{G}_t^\top \tilde{G}_t)_{11}^{-1/2}(\tilde{G}_t^\top \tilde{G}_t)_{12}(\tilde{G}_t^\top \tilde{G}_t)_{22}^{-1/2}\| \leq 1$. We assert that this holds with a high probability by our assumptions, for example, by overparameterization and the condition that $|\tilde{G}_{22}| \geq |\langle \tilde{G}_{21} + \tilde{G}_{12}, \tilde{v} \rangle|$ for any $\tilde{v} \in \mathbb{R}^{m+1}$. As a result, we invoke Lemma 6 and obtain

$$\begin{aligned} \|\tilde{G}_t(\tilde{\Phi}_t - \tilde{\Phi}^*)\|^2 &= \text{tr}(\tilde{G}_t^\top \tilde{G}_t(\tilde{\Phi}_t - \tilde{\Phi}^*)(\tilde{\Phi}_t - \tilde{\Phi}^*)^\top) \\ &\leq \text{tr}(\tilde{G}_t^\top \tilde{G}_t) \lambda_1((\tilde{\Phi}_t - \tilde{\Phi}^*)(\tilde{\Phi}_t - \tilde{\Phi}^*)^\top) \\ &= \text{tr}(\tilde{G}_t^\top \tilde{G}_t) \|\tilde{\Phi}_t - \tilde{\Phi}^*\|^2. \end{aligned} \quad (29)$$

Using (28) and (29) in (26), we have

$$\begin{aligned} \|\tilde{\Phi}_{t+1} - \tilde{\Phi}^*\|^2 &\leq \|\tilde{\Phi}_t - \tilde{\Phi}^*\|^2 - 2\alpha_t \|\tilde{G}_{11}^{1/2} \tilde{\Phi}_1\|^2 - 2\alpha_t \langle \tilde{G}_{21}^\top + \tilde{G}_{12}, \tilde{\Phi}_1 \rangle - 2\alpha_t \tilde{G}_{22} \\ &\quad + \alpha_t^2 \text{tr}(\tilde{G}_t^\top \tilde{G}_t) \|\tilde{\Phi}_t - \tilde{\Phi}^*\|^2. \end{aligned}$$

Now, using the inequality $-\left| \langle \tilde{G}_{21}^\top + \tilde{G}_{12}, \tilde{\Phi}_1 \rangle \right| \leq \langle \tilde{G}_{21}^\top + \tilde{G}_{12}, \tilde{\Phi}_1 \rangle \leq \left| \langle \tilde{G}_{21}^\top + \tilde{G}_{12}, \tilde{\Phi}_1 \rangle \right|$, we get that

$$-\langle \tilde{G}_{21}^\top + \tilde{G}_{12}, \tilde{\Phi}_1 \rangle \leq \left| \langle \tilde{G}_{21}^\top + \tilde{G}_{12}, \tilde{\Phi}_1 \rangle \right|,$$

and then,

$$\begin{aligned} \left\| \tilde{\Phi}_{t+1} - \tilde{\Phi}^* \right\|^2 &\leq \left\| \tilde{\Phi}_t - \tilde{\Phi}^* \right\|^2 - 2\alpha_t \left\| \tilde{G}_{11}^{1/2} \tilde{\Phi}_1 \right\|^2 + 2\alpha_t \left| \langle \tilde{G}_{21}^\top + \tilde{G}_{12}, \tilde{\Phi}_1 \rangle \right| - 2\alpha_t \tilde{G}_{22} \\ &\quad + \alpha_t^2 \text{tr}(\tilde{G}_t^\top \tilde{G}_t) \left\| \tilde{\Phi}_t - \tilde{\Phi}^* \right\|^2. \end{aligned}$$

Setting $\tilde{v} = \tilde{\Phi}_1$ in the condition $|\tilde{G}_{22}| \geq |\langle \tilde{G}_{21} + \tilde{G}_{12}^\top, \tilde{v} \rangle|$, it holds that $|\tilde{G}_{22}| > |\langle \tilde{G}_{21} + \tilde{G}_{12}^\top, \tilde{\Phi}_1 \rangle| - C_1$ for any arbitrary constant $C_1 > 0$. Set $C_1 = \frac{1}{\alpha_t C_2}$ for some constant $C_2 > 0$, noting that $\alpha_t > 0$ for all t , then we get

$$-\tilde{G}_{22} \leq -\left| \langle \tilde{G}_{21} + \tilde{G}_{12}^\top, \tilde{\Phi}_1 \rangle \right| + \frac{1}{\alpha_t C_2}.$$

Consequently,

$$\left\| \tilde{\Phi}_{t+1} - \tilde{\Phi}^* \right\|^2 \leq \left\| \tilde{\Phi}_t - \tilde{\Phi}^* \right\|^2 - 2\alpha_t \left\| \tilde{G}_{11}^{1/2} \tilde{\Phi}_1 \right\|^2 + \frac{2}{C_2} + \alpha_t^2 \text{tr}(\tilde{G}_t^\top \tilde{G}_t) \left\| \tilde{\Phi}_t - \tilde{\Phi}^* \right\|^2.$$

Now, if the condition $1 + M_g \eta_t \leq \|\tilde{G}_t\|_F$ is such that

$$\frac{\left\| \tilde{G}_{11}^{1/2} \tilde{\Phi}_1 \right\|^2}{\text{tr}(\tilde{G}_t^\top \tilde{G}_t) \left\| \tilde{\Phi}_t - \tilde{\Phi}^* \right\|^2} \geq \alpha_t \triangleq \frac{\bar{\alpha}_t}{1 + M_g \eta_t} \geq \frac{\bar{\alpha}_t}{\|\tilde{G}_t\|_F} \equiv \frac{\bar{\alpha}_t}{\sqrt{\text{tr}(\tilde{G}_t^\top \tilde{G}_t)}},$$

by fixing $0 < \bar{\alpha}_t \equiv \bar{\alpha} < 1$, then

$$\left\| \tilde{\Phi}_{t+1} - \tilde{\Phi}^* \right\|^2 \leq (1 - \bar{\alpha}) \left\| \tilde{\Phi}_t - \tilde{\Phi}^* \right\|^2 + \frac{2}{C_2}. \quad (30)$$

The recurrence in (30) can be expanded as follows:

$$\begin{aligned} \left\| \tilde{\Phi}_{t+1} - \tilde{\Phi}^* \right\|^2 &\leq (1 - \bar{\alpha}) \left\| \tilde{\Phi}_t - \tilde{\Phi}^* \right\|^2 + \frac{2}{C_2} \\ &\leq (1 - \bar{\alpha}) \left((1 - \bar{\alpha}) \left\| \tilde{\Phi}_{t-1} - \tilde{\Phi}^* \right\|^2 + \frac{2}{C_2} \right) + \frac{2}{C_2} \\ &\leq (1 - \bar{\alpha}) \left((1 - \bar{\alpha}) \left((1 - \bar{\alpha}) \left\| \tilde{\Phi}_{t-2} - \tilde{\Phi}^* \right\|^2 + \frac{2}{C_2} \right) + \frac{2}{C_2} \right) + \frac{2}{C_2} \\ &= (1 - \bar{\alpha})^3 \left\| \tilde{\Phi}_{t-2} - \tilde{\Phi}^* \right\|^2 + (1 - \bar{\alpha})^2 \frac{2}{C_2} + (1 - \bar{\alpha}) \frac{2}{C_2} + \frac{2}{C_2}, \end{aligned}$$

and so on. This gives, for any $T \geq 1$,

$$\left\| \tilde{\Phi}_T - \tilde{\Phi}^* \right\|^2 \leq (1 - \bar{\alpha})^T \left\| \tilde{\Phi}_0 - \tilde{\Phi}^* \right\|^2 + \frac{2}{C_2} \sum_{j=0}^{T-1} (1 - \bar{\alpha})^{T-j-1}. \quad (31)$$

Since $C_2 > 0$ is arbitrary, we set $C_2 = 2 \sum_{j=0}^{T-1} (1 - \bar{\alpha})^{T-j-1}$. We also have that since $\bar{\alpha} > 0$, it satisfies the inequality $1 - \bar{\alpha} \leq \exp(-\bar{\alpha})$. Then (31) gives

$$\left\| \tilde{\Phi}_T - \tilde{\Phi}^* \right\|^2 \leq \exp(-\bar{\alpha}T) \left\| \tilde{\Phi}_0 - \tilde{\Phi}^* \right\|^2 + 1. \quad (32)$$

Substituting our choice of T into (32) gives

$$\left\| \tilde{\Phi}_T - \tilde{\Phi}^* \right\|^2 \leq \epsilon + 1,$$

which is result P.1.

To prove P.2, we first notice that the local condition $\left\| \nabla_{\Phi}^2 \hat{R}_s(\Phi) \right\|_{op} \leq D_R$ in RR.1 implies local D_R -Lipschitz continuity of $\nabla_{\Phi} \hat{R}_s(\Phi)$ with respect to Φ , that is, for $\bar{\theta}, \tilde{\theta}$ around the initialization, we have

$$\left\| \nabla \hat{R}_s(\Phi(\cdot, \bar{\theta})) - \nabla \hat{R}_s(\Phi(\cdot, \tilde{\theta})) \right\| \leq D_R \left\| \Phi(\cdot, \bar{\theta}) - \Phi(\cdot, \tilde{\theta}) \right\|,$$

or equivalently,

$$\hat{R}_s(\Phi(\cdot, \bar{\theta})) \leq \hat{R}_s(\Phi(\cdot, \tilde{\theta})) + \langle \nabla \hat{R}_s(\Phi(\cdot, \tilde{\theta})), \Phi(\cdot, \bar{\theta}) - \Phi(\cdot, \tilde{\theta}) \rangle + \frac{D_R}{2} \left\| \Phi(\cdot, \bar{\theta}) - \Phi(\cdot, \tilde{\theta}) \right\|^2. \quad (33)$$

We recall the notation $\Phi_t \triangleq \Phi(\cdot, \theta_t)$ for all $t \in \mathbb{R}_{\geq 0}$. Then, using $\mathcal{L} \triangleq \hat{R}_s + g$, Lemma 1, and (33), we get

$$\begin{aligned} \mathcal{L}(\theta_{t+1}) &\leq \mathcal{L}(\theta_t) + \langle \nabla \hat{R}_s(\Phi_t), \Phi_{t+1} - \Phi_t \rangle + \frac{D_R}{2} \left\| \Phi_{t+1} - \Phi_t \right\|^2 + \langle \nabla g(\theta_t), \theta_{t+1} - \theta_t \rangle \\ &\quad + \omega_{\nu}(d_{\nu}(\theta_t, \theta_{t+1})) \left\| \theta_{t+1} - \theta_t \right\|_{\theta_t}^2 \\ &\leq \mathcal{L}(\theta_t) - \frac{\gamma_R}{2} \left\| \Phi_{t+1} - \Phi_t \right\|^2 - \mathcal{L}(\theta_t) + \mathcal{L}(\theta_{t+1}) - \omega_{\nu}(-d_{\nu}(\theta_t, \theta_{t+1})) \left\| \theta_{t+1} - \theta_t \right\|_{\theta_t}^2 \\ &\quad + \frac{D_R}{2} \left\| \Phi_{t+1} - \Phi_t \right\|^2 + \omega_{\nu}(d_{\nu}(\theta_t, \theta_{t+1})) \left\| \theta_{t+1} - \theta_t \right\|_{\theta_t}^2. \end{aligned}$$

Using the γ_R -strong convexity assumption on \hat{R} in RR.1 and the Lipschitz property of \mathcal{L} in Lemma 5, this gives

$$\begin{aligned} \mathcal{L}(\theta_{t+1}) &\leq \mathcal{L}(\theta_t) + (B_R B_{\Phi} + B_g) \left\| \theta_{t+1} - \theta_t \right\| + \frac{D_R - \gamma_R}{2} \left\| \Phi_{t+1} - \Phi_t \right\|^2 \\ &\quad + (\omega_{\nu}(d_{\nu}(\theta_t, \theta_{t+1})) - \omega_{\nu}(-d_{\nu}(\theta_t, \theta_{t+1}))) \left\| \theta_{t+1} - \theta_t \right\|_{\theta_t}^2 \\ &\leq \mathcal{L}(\theta_t) + (B_R B_{\Phi} + B_g) \left\| \theta_{t+1} - \theta_t \right\| + \frac{B_{\Phi}^2 (D_R - \gamma_R)}{2} \left\| \theta_{t+1} - \theta_t \right\|^2 \\ &\quad + (\omega_{\nu}(d_{\nu}(\theta_t, \theta_{t+1})) - \omega_{\nu}(-d_{\nu}(\theta_t, \theta_{t+1}))) \left\| \theta_{t+1} - \theta_t \right\|_{\theta_t}^2. \end{aligned} \quad (34)$$

Recalling the notation $\delta_t \triangleq \theta_{t+1} - \theta_t$ and substituting the estimates on $\|\delta_t\|$ and $\|\delta_t\|_{\theta_t}$ from Lemma 4 into (34) yields result P.2. \square

C. Additional experimental details and results

C.1. Remark on the T-I measure

The time-invariance measure provides a way to measure stability of the optimizer's dynamics from initialization. However, since the signum function does not account for indices (i, j) of a_l with $a_{ij} = 0$, i.e.,

$$\text{sgn}(a_{ij}) \triangleq \begin{cases} +1 & \text{if } a_{ij} > 0, \\ -1 & \text{if } a_{ij} < 0, \end{cases}$$

and, as we have seen, the GGN-SCORE framework potentially produces many of this instance (with $a_{ij} = 0$) to reduce the model’s complexity and/or improve generalization, a natural question is what state should be assumed for neuron a_{ij} when it is exactly zero. For this, we follow the standard convention that if $a_{ij} = 0$, then the (i, j) -th neuron remains unchanged from its initial state [64, Section 13.7]. Under this convention, the proportion of the indices (i, j) of a_l^{final} satisfying $\text{sgn}(a_{ij}^{\text{start}}) \neq \text{sgn}(a_{ij}^{\text{final}})$ with $a_{ij}^{\text{final}} = 0$ contribute to the stability of activations, and hence should be accounted for in the T-I measure. However, this contribution appear to be insignificant for the values of τ and μ that give the best test accuracies. From what we observe in Figure 3 and Figure 4, proper choices of μ and τ reliably produces stable dynamics of the optimizer as well as a good generalization of the final trained model.

C.2. MNIST teacher-student setting

In order to evaluate GGN-SCORE on the MNIST dataset such that we are close to the theoretical framework, we consider a teacher-student setup for the MNIST dataset in a similar way as [38, Appendix C.4]:

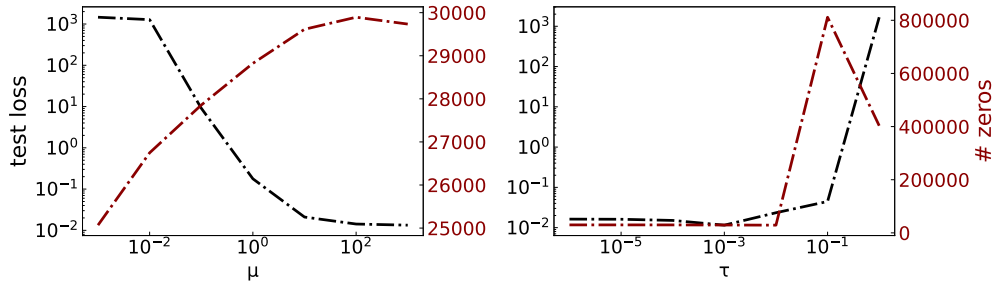


Fig. 5: Test loss of the GGN-SCORE-trained NN on MNIST dataset (teacher-student) with $g(\theta)$ given by (15). **Left:** Results for different values of the regularization smoothing parameter μ with $\tau = 10^{-4}$ fixed. **Right:** Results for different values of the regularization strength τ with $\mu = 1/\sqrt{n}$ fixed.

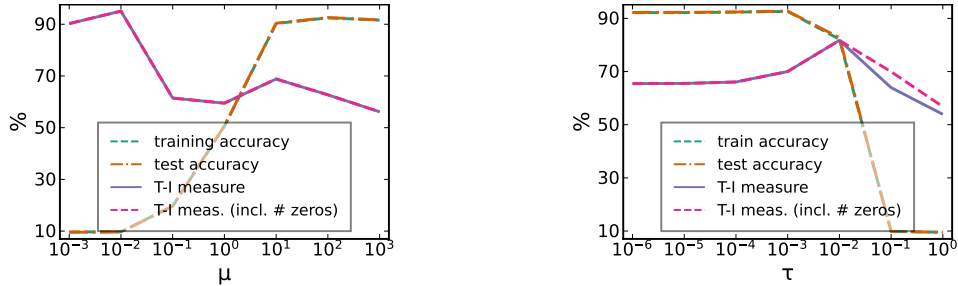


Fig. 6: Accuracy and T-I measure of the GGN-SCORE-trained NN on MNIST dataset (teacher-student) for different values of μ (left) and different values of τ (right), with the regularization function $g(\theta)$ given by (15). In the left figure, $\tau = 10^{-4}$ is used. In the right figure, $\mu = 1/\sqrt{n}$ is used.

- We create a custom training dataset by combining the original MNIST test dataset (containing 10000 sample points) and a balanced subset of the original training dataset. This balanced subset

is created by “undersampling” the first 3000 samples of the original training dataset to give 2610 sample points. In total, the custom training dataset contains 12610 sample points.

- We then train a teacher NN Φ^* of the form (14) and hidden size $n^* = 16$ on this training dataset with the cross-entropy loss function and the SiLU activation function.
- A training “target” dataset is created from Φ^* (with the softmax function applied on each output of Φ^*).
- The student NN of the form (3) with the SiLU activation and hidden size $n = 1024$ is then trained on the custom training input samples and their corresponding target samples constructed from Φ^* . The trained student NN is tested on the original MNIST test dataset.

The training and test results are displayed Figure 5 and Figure 6. We follow a similar evaluation procedure as in Section 4.2, i.e., the results are evaluated on the basis of the test loss, training and test accuracy, and T-I measure of the trained student NN. Interestingly, similar observations as in Section 4.2 are made from the displayed results. The total computation time to generate the results in Figure 5 and Figure 6 is ~ 8 hours, 39 minutes on CPU.

C.3. FashionMNIST experiments

We perform experiments on the FashionMNIST dataset [65] under the same setting as the MNIST experiments in Section 4.2. While the FashionMNIST classification tends to be a harder task than the MNIST, results shown in Figure 7 and Figure 8 indicate similar behaviours as those described in Section 4.2 regarding the influence of the regularization parameters.

C.4. Comparison with GD

We now compare GGN-SCORE with GD on three UCI benchmark datasets³: pendigits, letter, and avila, summarized in Table 1. As in Section 4, we use a learning rate of 1 for GD, and set the hidden size $n = 128$ in all the experiments for a NN of the form (3), and a scaling $\kappa(n) = 1/\sqrt{n}$. The function g in GGN-SCORE is given by (15) with $\tau = 10^{-4}$. The results are shown Figure 9 and Table 2. We observe faster convergence and better generalization in most cases for GGN-SCORE, and as in the case for the full-batch deterministic setting in Section 4 on synthetic datasets, we achieve this performance in faster time compared to GD. Note that much of the computational burden associated with the regularized GGN is greatly reduced by using the stylized expression (9), since the mini-batch size is typically much smaller than p , the size of the optimization variable θ .

Table 1: Summary of UCI datasets used for comparison.

Dataset	Num. of samples		Input dim.	Num. of classes
	Training	Test		
pendigits	7494	3498	16	10
letter	10500	5000	16	26
avila	10430	10437	11	12

³<https://archive.ics.uci.edu>.

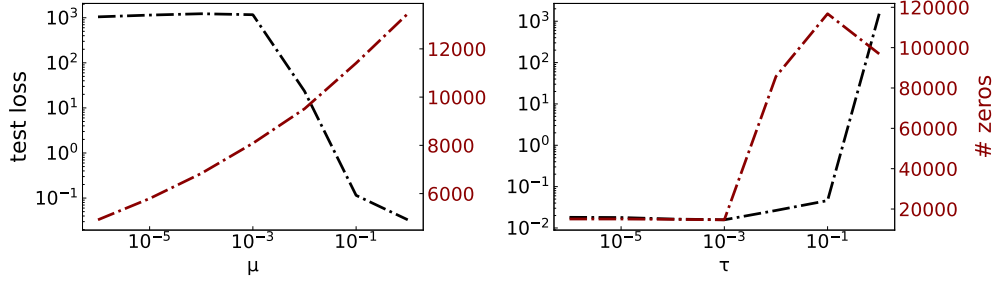


Fig. 7: Test loss evaluation of the GGN-SCORE-trained NN on FashionMNIST dataset for different values of the regularization smoothing parameter μ fixing $\tau = 10^{-4}$ (**left**) and different values of the regularization strength τ fixing $\mu = 1/\sqrt{n}$ (**right**), where the regularization function $g(\theta)$ is given by (15).

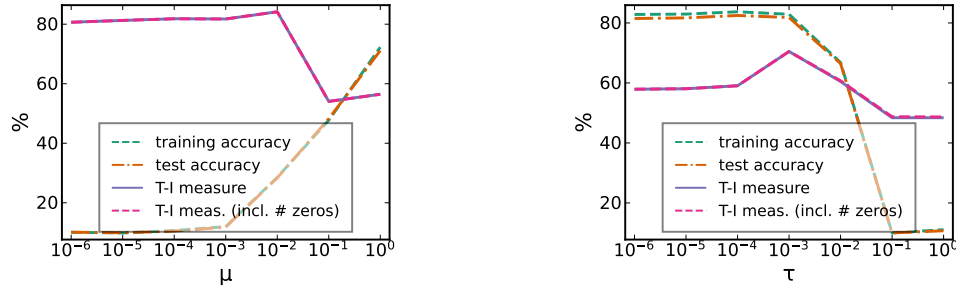


Fig. 8: Evaluation of GGN-SCORE on FashionMNIST dataset for different values of the regularization strength μ (**left**) and different values of the regularization smoothing parameter τ (**right**). The regularization function $g(\theta)$ is given by (15). In the left figure, $\mu = 1/\sqrt{n}$ is used. In the right figure, $\tau = 10^{-4}$ is used.

Table 2: Stability of activation measure.

Dataset	Batch-size	μ	T-I measure (%)		T-I meas. incl. $a_{ij} = 0$ (%)	
			GD	GGN-SCORE	GD	GGN-SCORE
pendigits	8	$0.001/\sqrt{n}$	50.066	52.7331	50.1041	52.7331
letter	64	$10/\sqrt{n}$	55.9237	55.2016	55.925	55.2112
avila	64	$10/\sqrt{n}$	73.318	71.4815	73.3189	71.4833

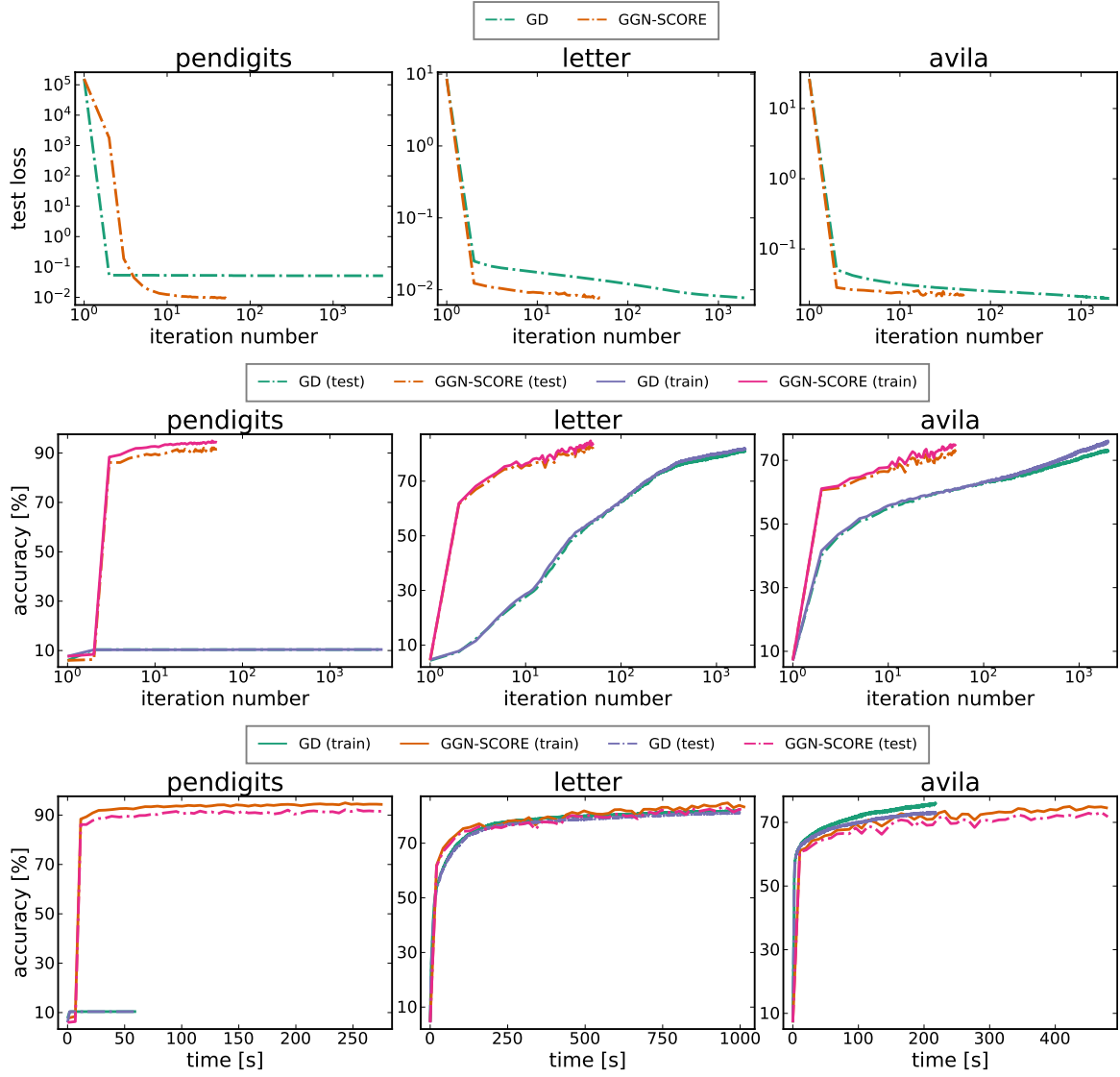


Fig. 9: Test loss and accuracy evolution for GD and GGN-SCORE on pendigits, letter and avila datasets. The regularization function $g(\theta)$ in GGN-SCORE is given by (15) with $\tau = 10^{-4}$ and μ given in Table 2.