CLIP: Cheap Lipschitz Training of Neural Networks*

Leon Bungert[1], René Raab[2], Tim Roith[1], Leo Schwinn[2], and Daniel Tenbrinck[1]

[1] Department Mathematics, Friedrich-Alexander University Erlangen-Nürnberg,
Cauerstraße 11, 91058 Erlangen
{leon.bungert,tim.roith,daniel.tenbrinck}@fau.de

[2] Machine Learning and Data Analytics Lab, Friedrich-Alexander University
Erlangen-Nürnberg, Carl-Thiersch-Straße 2b, 91052 Erlangen
{rene.raab,leo.schwinn}@fau.de

Abstract. Despite the large success of deep neural networks (DNN) in recent years, most neural networks still lack mathematical guarantees in terms of stability. For instance, DNNs are vulnerable to small or even imperceptible input perturbations, so called adversarial examples, that can cause false predictions. This instability can have severe consequences in applications which influence the health and safety of humans, e.g., biomedical imaging or autonomous driving. While bounding the Lipschitz constant of a neural network improves stability, most methods rely on restricting the Lipschitz constants of each layer which gives a poor bound for the actual Lipschitz constant.
In this paper we investigate a variational regularization method named *CLIP* for controlling the Lipschitz constant of a neural network, which can easily be integrated into the training procedure. We mathematically analyze the proposed model, in particular discussing the impact of the chosen regularization parameter on the output of the network. Finally, we numerically evaluate our method on both a nonlinear regression problem and the MNIST and Fashion-MNIST classification databases, and compare our results with a weight regularization approach.

Keywords: Deep neural network · Machine learning · Lipschitz constant · Variational regularization · Stability · Adversarial attack.

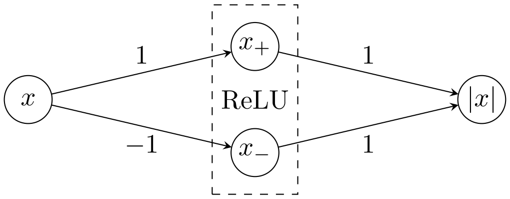
1 Introduction

Deep neural networks (DNNs) have led to astonishing results in various fields, such as computer vision [13] and language processing [18]. Despite its large success, deep learning and the resulting neural network architectures also bear certain drawbacks. On the one hand, their behaviour is mathematically not yet fully

* This work was supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 777826 (NoMADS) and by the German Ministry of Science and Technology (BMBF) under grant agreement No. 05M2020 (DELETO).

understood and there exist not many rigorous analytical results. On the other hand, most trained networks are vulnerable to adversarial attacks [9]. In image processing, adversarial examples are small, typically imperceptible perturbations to the input that cause misclassifications. In domains like autonomous driving or healthcare this can potentially have fatal consequences. To mitigate these weaknesses, many methods were proposed to make neural networks more robust and reliable. One straight-forward approach is to regularize the norms of the weight matrices [10]. Another idea is adversarial training [16,23,24] which uses adversarial examples generated from the training data to increase robustness locally around the training samples. In this context, also the Lipschitz constant of neural networks has attracted a lot of attention (e.g., [7,8,26,29]) since it constitutes a worst-case bound for their stability, and Lipschitz-regular networks are reported to have superior generalization properties [17]. In [27] Lipschitz regularization around training samples has been related to adversarial training.

Unfortunately, the Lipschitz constant of a neural network is NP-hard to compute [22], hence several methods in the literature aim to achieve more stability of neural networks by bounding the Lipschitz constant of each individual layer [1,10,12,20] or the activation functions [3]. However, this strategy is a very imprecise approximation to the real Lipschitz constant and hence the trained networks may suffer from inferior expressivity and can even be less robust [15]. The following example demonstrates a representation of the absolute value function, which clearly has Lipschitz constant 1, using a neural network whose individual layers have larger Lipschitz constants (cf. [11] for details).



(1)

In this paper we propose a new variational regularization method for training neural networks, while simultaneously controlling their Lipschitz constant. Since the additional computations can easily be embedded in the standard training process and are fully parallelizable, we name our method *Cheap Lipschitz Training of Neural Networks (CLIP)*. Instead of bounding the Lipschitz constant of each layer individually as in prior approaches, we aim to minimize the global Lipschitz constant of the neural network via an additional regularization term.

The **main contributions** of this paper are as follows. First, we motivate and introduce the proposed variational regularization method for Lipschitz training of neural networks, which leads to a min-max problem. For optimizing the proposed model we formulate a stochastic gradient descent-ascent method, the CLIP algorithm. Subsequently, we perform mathematical analysis of the proposed variational regularization method by discussing existence of solutions. Here we use techniques, developed for the analysis of variational regularization methods of inverse problems [4,6]. We analyse the two limit cases, namely the regularization parameter tending to zero and infinity, where we prove convergence to a

Lipschitz minimal fit of the data and a generalized barycenter, respectively. Finally, we evaluate the proposed approach by performing numerical experiments on a regression example and the MNIST and Fashion-MNIST classification databases [14,28]. We show that the introduced variational regularization term leads to improved stability compared to networks without additional regularization or with layerwise Lipschitz regularization. For this we investigate the impact of noise and adversarial attacks on the trained neural networks.

2 Model and Algorithms

Given a finite training set $\mathcal{T} \subset \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ denote the input and output space, we propose to determine parameters $\theta \in \Theta$ of a neural network $f_\theta : \mathcal{X} \to \mathcal{Y}$ by solving the empirical risk minimization problem with an additional Lipschitz regularization term

$$\theta_\lambda \in \arg\min_{\theta \in \Theta} \frac{1}{|\mathcal{T}|} \sum_{(x,y)\in\mathcal{T}} \ell(f_\theta(x), y) + \lambda \operatorname{Lip}(f_\theta). \tag{2}$$

Here, $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a loss function, $\lambda > 0$ is a regularization parameter and the Lipschitz constant of the network $f_\theta : \mathcal{X} \to \mathcal{Y}$ is defined as

$$\operatorname{Lip}(f_\theta) := \sup_{x,x' \in \mathcal{X}} \frac{\|f_\theta(x) - f_\theta(x')\|}{\|x - x'\|}. \tag{3}$$

The norms involved in this definition can be chosen freely, however, for our CLIP algorithm we assume differentiability. The fundamental difference of the proposed model (2) compared to existing approaches in the literature [1,10,12,20] is that we use the actual Lipschitz constant (3) as regularizer and do not rely on the layer-based upper bound

$$\operatorname{Lip}(f_\theta) \;\leq\; \prod_{l=1}^{L} \operatorname{Lip}(\Phi_l) \tag{4}$$

for neural networks of the form $f_\theta = \Phi_L \circ \dots \Phi_1$. By plugging (3) into (2) this becomes a min-max problem, which we solve numerically by using a modification of the stochastic batch gradient descent algorithm with momentum $\mathrm{SGDM}_{\eta,\gamma}$ with learning rate $\eta > 0$ and momentum $\gamma \geq 0$, see, e.g. [21].

Since evaluating the Lipschitz constant of a neural network is NP-hard [22], we approximate it on a finite subset $\mathcal{X}_{\mathrm{Lip}} \subset \mathcal{X} \times \mathcal{X}$ by setting

$$\operatorname{Lip}(f_\theta, \mathcal{X}_{\mathrm{Lip}}) := \max_{(x,x')\in\mathcal{X}_{\mathrm{Lip}}} \frac{\|f_\theta(x) - f_\theta(x')\|}{\|x - x'\|}. \tag{5}$$

To make sure that the tuples in $\mathcal{X}_{\mathrm{Lip}}$ correspond to points with a high Lipschitz constant during the whole training process, we update $\mathcal{X}_{\mathrm{Lip}}$ using gradient ascent of the difference quotient in (5) with adaptive step size, see Algorithm 1. We call

Algorithm 1: AdversarialUpdate$_\tau$ (with step size $\tau > 0$)

```
$L(x, x') := \|f_\theta(x) - f_\theta(x')\| \,/\, \|x - x'\|, \quad x, x' \in \mathcal{X}$
for $(x, x') \in \mathcal{X}_{\text{Lip}}$ do
    $x \leftarrow x + \tau\, L(x, x') \nabla_x L(x, x')$
    $x' \leftarrow x' + \tau\, L(x, x') \nabla_{x'} L(x, x')$
```

Algorithm 2: CLIP (Cheap Lipschitz Training)

```
for epoch $e = 1$ to $E$ do
    for minibatch $B \subset \mathcal{T}$ do
        $\mathcal{X}_{\text{Lip}} \leftarrow \text{AdversarialUpdate}_\tau(\mathcal{X}_{\text{Lip}})$
        $\theta \leftarrow \text{SGDM}_{\eta,\gamma}\left(\frac{1}{|B|}\sum_{(x,y)\in B} \ell(f_\theta(x), y) + \lambda\, \text{Lip}(f_\theta, \mathcal{X}_{\text{Lip}})\right)$
        if $\mathcal{A}(f_\theta; \mathcal{T}) > \alpha$ then
            $\lambda \leftarrow \lambda + d\lambda$
        else
            $\lambda \leftarrow \lambda - d\lambda$
```

this *adversarial update*, since the tuples in $\mathcal{X}_{\text{Lip}}$ approach points which have a small distance to each other while still getting classified differently by the network.

To illustrate the effect of adversarial updates, Figure 1 depicts a pair of images from the Fashion-MNIST database before and after applying Algorithm 1, using a neural network trained with CLIP (see Section 4.2 for details). The second pair realizes a large Lipschitz constant and hence lies close to the decision boundary of the neural network.

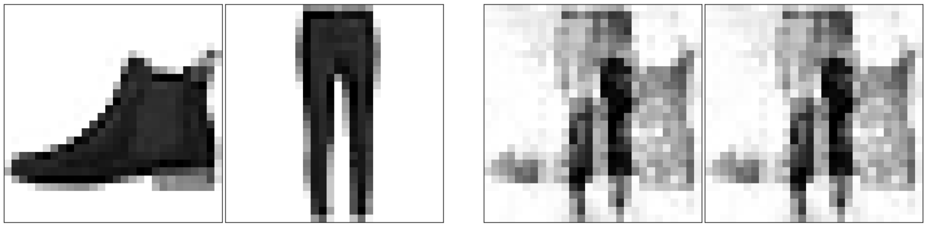


Fig. 1: Effect of Algorithm 1. The initial tuple gets classified correctly with high confidence ($> 90\%$). After adversarial updates the pair gets misclassified as the respective other class with low confidence ($< 30\%$).

The regularization parameter λ in (2) is chosen by a discrepancy principle from inverse problems [2,5]: We compare an accuracy measure $\mathcal{A}(f_\theta;\mathcal{T})$ for the neural network f_θ, evaluated on the training data, to a target accuracy α. Combining the adversarial update from Algorithm 1 with stochastic gradient descent with momentum, we propose the CLIP algorithm, which is given in Algorithm 2. Similarly to [24] one can reuse computations of the backward pass in the gradient step of the Lipschitz regularizer with respect to θ in Algorithm 2 to compute the gradient with respect to the input x, needed in Algorithm 1, which indeed makes CLIP a 'cheap' extension to conventional training. We would like to emphasize that the set $\mathcal{X}_{\mathrm{Lip}}$, on which we approximate the Lipschitz constant of the neural network, is not fixed but is updated in an optimal way in every minibatch. Hence, it plays the role of an adaptive 'training set' for the Lipschitz constant. Future analysis will investigate the approximation quality of this approach in the framework of stochastic gradient descent methods.

3 Analysis

In this section we prove analytical results on the model (2), which are inspired by analogous statements for variational regularization methods for inverse problems (see, e.g., [4,6]). While the CLIP Algorithm 1 is stochastic in nature, our analysis focuses on the deterministic model (2), whose solutions are approximated by CLIP. For our results we have to pose some mild assumptions on the loss function ℓ and the neural network f_θ which are fulfilled for most loss functions and network architectures, such as mean squared error or cross entropy loss and architectures like feed-forward, convolutional, or residual networks with continuous activation functions. Furthermore, we assume that $\mathcal{X},\mathcal{Y}$, and Θ are finite-dimensional spaces, which is the case in most applications and lets us state our theory more compactly than using Banach space topologies.

Assumption 1. We assume that the loss function $\ell:\mathcal{Y}\times\mathcal{Y}\to\mathbb{R}\cup\{\infty\}$ satisfies:

(a) $\ell(y,y')\geq 0$ for all $y,y'\in\mathcal{Y}$,

(b) $y\mapsto\ell(y,y')$ is lower semi-continuous for all $y'\in\mathcal{Y}$.

Assumption 2. We assume that the map $\theta\mapsto f_\theta(x)$ is continuous for all $x\in\mathcal{X}$.

Assumption 3. We assume that there exists $\theta\in\Theta$ such that

$$\frac{1}{|\mathcal{T}|}\sum_{(x,y)\in\mathcal{T}}\ell(f_\theta(x),y)+\lambda\,\mathrm{Lip}(f_\theta)<\infty.$$

We will also need the following lemma, which states that the Lipschitz constant of f_θ is lower semi-continuous with respect to the parameters θ.

Lemma 1. *Under Assumption 2 the functional $\theta\mapsto\mathrm{Lip}(f_\theta)$ is lower semi-continuous.*

Proof. Let $(\theta_i)_{i\in\mathbb{N}} \subset \Theta$ converge to $\theta \in \Theta$. Using the continuity of $\theta \mapsto f_\theta(x)$ and the lower semi-continuity of the norm $\|\cdot\|$, one can compute

$$\begin{aligned}\operatorname{Lip}(f_\theta) &= \sup_{x,x'\in\mathcal{X}} \frac{\|f_\theta(x)-f_\theta(x')\|}{\|x-x'\|} \leq \sup_{x,x'\in\mathcal{X}} \liminf_{i\to\infty} \frac{\|f_{\theta_i}(x)-f_{\theta_i}(x')\|}{\|x-x'\|} \\ &\leq \liminf_{i\to\infty} \sup_{x,x'\in\mathcal{X}} \frac{\|f_{\theta_i}(x)-f_{\theta_i}(x')\|}{\|x-x'\|} = \liminf_{i\to\infty} \operatorname{Lip}(f_{\theta_i}).\end{aligned}$$

3.1 Existence of Solutions

We start with an existence statement for a slight modification of (2), which also regularizes the network parameters and guarantees coercivity of the objective.

Proposition 1. *Under Assumptions 1–3 the problem*

$$\min_{\theta\in\Theta} \frac{1}{|\mathcal{T}|} \sum_{(x,y)\in\mathcal{T}} \ell(f_\theta(x),y) + \lambda \operatorname{Lip}(f_\theta) + \mu \|\theta\|_\Theta \tag{6}$$

has a solution for all values $\lambda, \mu > 0$. *Here,* $\|\cdot\|_\Theta$ *denotes a norm on* Θ.

Proof. We let $(\theta_i)_{i\in\mathbb{N}} \subset \Theta$ be a minimizing sequence, whose existence is assured by Assumption 3. Since $\mu > 0$ in (6), the norms $\|\theta_i\|_\Theta$ for $i \in \mathbb{N}$ are uniformly bounded and hence, up to a subsequence which we do not relabel, $\theta_i \to \theta^* \in \Theta$ as $i \to \infty$. The lower semi-continuity of ℓ, Lip, and $\|\cdot\|_\Theta$ together with the continuity of $\theta \mapsto f_\theta(x)$ then shows that θ^* solves (6).

Remark 1. If Θ is compact or even finite, existence for (2) is assured since any minimizing sequence in Θ is compact. The reason that in the general case we can only show existence for the regularized problem (6) is that it assures boundedness of minimizing sequences. Even for the unregularized empirical risk minimization problem existence is typically assumed in the *realizability assumption* [25].

3.2 Dependency on the Regularization Parameter

We start with the statement that the Lipschitz constant in (2) decreases and the empirical risk increases as the regularization parameter λ grows.

Proposition 2. *Let* θ_λ *solve* (2) *for* $\lambda \geq 0$. *Then it holds*

$$\lambda \longmapsto \frac{1}{|\mathcal{T}|} \sum_{(x,y)\in\mathcal{T}} \ell(f_{\theta_\lambda}(x),y) \quad \textit{is non-decreasing,} \tag{7}$$

$$\lambda \longmapsto \operatorname{Lip}(f_{\theta_\lambda}) \quad \textit{is non-increasing.} \tag{8}$$

Proof. The proof works precisely as in [6].

Now we will study the limit cases $\lambda \searrow 0$ and $\lambda \to \infty$ in (2). As in Section 3.1, because of lack of coercivity, we have to assume that the corresponding sequences of optimal network parameters converge.

Our first statement deals with the behavior as the regularization parameter λ tends to zero. We show that in this case the learned neural networks converge to one which fits the training data with the smallest Lipschitz constant.

Proposition 3. *Let Assumptions 1–3 and the realizability assumption [25]*

$$\min_{\theta\in\Theta} \frac{1}{|\mathcal{T}|} \sum_{(x,y)\in\mathcal{T}} \ell(f_\theta(x), y) = 0 \tag{9}$$

be satisfied. Let θ_λ be a solution of (2) for $\lambda > 0$. If $\theta_\lambda \to \theta^\dagger \in \Theta$ as $\lambda \searrow 0$, then

$$\theta^\dagger \in \arg\min \left\{ \mathrm{Lip}(f_\theta) \,:\, \theta \in \Theta,\, \frac{1}{|\mathcal{T}|} \sum_{(x,y)\in\mathcal{T}} \ell(f_\theta(x), y) = 0 \right\} \tag{10}$$

if this problem admits a solution with $\mathrm{Lip}(f_\theta) < \infty$.

Proof. We fix $\theta \in \Theta$ which satisfies (9) and $\mathrm{Lip}(f_\theta) < \infty$. Using the optimality of θ_λ we infer

$$\liminf_{\lambda\searrow 0} \frac{1}{\lambda\,|\mathcal{T}|} \sum_{(x,y)\in\mathcal{T}} \ell(f_{\theta_\lambda}(x), y) + \mathrm{Lip}(f_{\theta_\lambda}) \;\le\; \mathrm{Lip}(f_\theta). \tag{11}$$

Using that ℓ is non-negative we conclude that $\mathrm{Lip}(f_{\theta_\lambda}) \le \mathrm{Lip}(f_\theta)$ and using Lemma 1 we get

$$\mathrm{Lip}(f_{\theta^\dagger}) \;\le\; \liminf_{\lambda\searrow 0} \mathrm{Lip}(f_{\theta_\lambda}) \;\le\; \mathrm{Lip}(f_\theta).$$

The lower semi-continuity of the loss ℓ and the continuity of the map $\theta \mapsto f_\theta(x)$ for all $x \in \mathcal{X}$ imply

$$\ell(f_{\theta^\dagger}(x), y) \le \liminf_{\lambda\searrow 0} \ell(f_{\theta_\lambda}(x), y), \quad \forall (x,y) \in \mathcal{T}.$$

If we assume that $\ell(f_{\theta^\dagger}(x), y) > 0$ for some $(x,y) \in \mathcal{T}$, then we obtain

$$\liminf_{\lambda\searrow 0} \frac{1}{\lambda\,|\mathcal{T}|} \sum_{(x,y)\in\mathcal{T}} \ell(f_{\theta_\lambda}(x), y) \ge \infty$$

which is a contradiction to (11) and implies that $\theta^\dagger$ satisfies (10).

Note that if the realizability assumption (9) is not satisfied, e.g., for noisy data or small network architectures, the previous statement has to be refined, which is subject to future work. The next proposition deals with the case in which the parameter λ approaches infinity. In this case the learned neural networks approach a constant map, coinciding with a generalized barycenter of the data. We denote by $\mathcal{T}_\mathcal{Y} = \{y : (x,y) \in \mathcal{T}\}$ the set of the data in the output space.

Proposition 4. *Let Assumptions 1–3 be satisfied and assume that*

$$\mathcal{M} := \{y \in \mathcal{Y} \,:\, \exists \theta \in \Theta,\ f_\theta(x) = y,\ \forall x \in \mathcal{X}\} \neq \emptyset. \tag{12}$$

Let θ_λ denote a solution of (2) for $\lambda > 0$. If $\theta_\lambda \to \theta_\infty \in \Theta$ as $\lambda \to \infty$, then $f_{\theta_\infty}(x) = \hat{y}$ for all $x \in \mathcal{X}$ where

$$\hat{y} \in \arg\min \left\{ \frac{1}{|\mathcal{T}|} \sum_{y \in \mathcal{T}_\mathcal{Y}} \ell(y', y) \,:\, y' \in \mathcal{M} \right\}. \tag{13}$$

Proof. From the optimality of θ_λ we deduce

$$\sum_{(x,y) \in \mathcal{T}} \ell(f_{\theta_\lambda}(x), y) + \lambda \operatorname{Lip}(f_{\theta_\lambda}) \ \leq \ \sum_{y \in \mathcal{T}_\mathcal{Y}} \ell(y', y), \quad \forall y' \in \mathcal{M}.$$

Hence, by letting $\lambda \to \infty$ we obtain $\lim_{\lambda \to \infty} \operatorname{Lip}(f_{\theta_\lambda}) = 0$. If we now assume that $\theta_\lambda \to \theta_\infty$ as $\lambda \to \infty$, we can use the lower semi-continuity of the Lipschitz constant to obtain

$$\operatorname{Lip}(f_{\theta_\infty}) \leq \liminf_{\lambda \to \infty} \operatorname{Lip}(f_{\theta_\lambda}) = 0.$$

Hence, $f_{\theta_\infty} \equiv \hat{y}$ for some element $\hat{y} \in \mathcal{Y}$. Using the lower semi-continuity of the loss function, we can conclude the proof with

$$\begin{aligned}\sum_{y \in \mathcal{T}_\mathcal{Y}} \ell(\hat{y}, y) \ &= \sum_{(x,y) \in \mathcal{T}} \ell(f_{\theta_\infty}(x), y) \ \leq \ \liminf_{\lambda \to \infty} \sum_{(x,y) \in \mathcal{T}} \ell(f_{\theta_\lambda}(x), y) \\ &\leq \sum_{y \in \mathcal{T}_\mathcal{Y}} \ell(y', y), \quad \forall y' \in \mathcal{M}.\end{aligned}$$

Remark 2. For a Euclidean loss function, i.e., $\ell(y', y) = \|y' - y\|^2$, the quantity (13) coincides with a projection of the barycenter of the training samples in $\mathcal{T}_\mathcal{Y}$ onto the manifold of constant networks $\mathcal{M}$, given by (12). This can be seen as follows: Let $b := \frac{1}{|\mathcal{T}|} \sum_{y \in \mathcal{T}_\mathcal{Y}} y$ be the barycenter of the training samples. Then

$$\begin{aligned}\frac{1}{|\mathcal{T}|} \sum_{y \in \mathcal{T}_\mathcal{Y}} \|y' - y\|^2 &= \|y'\|^2 - 2\,\langle y', b \rangle + \frac{1}{|\mathcal{T}|} \sum_{y \in \mathcal{T}_\mathcal{Y}} \|y\|^2 \\ &= \|y' - b\|^2 - \|b\|^2 + \frac{1}{|\mathcal{T}|} \sum_{y \in \mathcal{T}_\mathcal{Y}} \|y\|^2, \qquad \forall y' \in \mathcal{Y}.\end{aligned}$$

Using this equality for general $y' \in \mathcal{M}$ and for $y' = \hat{y}$ given by (13) we obtain

$$0 \ \leq \ \frac{1}{|\mathcal{T}|} \sum_{y \in \mathcal{T}_\mathcal{Y}} \|y' - y\|^2 - \frac{1}{|\mathcal{T}|} \sum_{y \in \mathcal{T}_\mathcal{Y}} \|\hat{y} - y\|^2 \ = \ \|y' - b\|^2 - \|\hat{y} - b\|^2.$$

This in particular implies that $\|\hat{y} - b\| \leq \|y' - b\|$ for all $y' \in \mathcal{M}$ as desired.

4 Experiments

After having studied theoretical properties of the proposed variational Lipschitz regularization method (2), we will apply the CLIP algorithm to regression and classification tasks in this section. In a first experiment we train a neural network to approximate a nonlinear function given by noisy samples, where we illustrate that CLIP prevents strong oscillations, which appear in unregularized networks. In a second experiment we use CLIP for training two different classification networks on the MNIST and Fashion-MNIST databases. Here, we quantitatively show that CLIP yields superior robustness to noise and adversarial attacks compared to unregularized and weight-regularized neural networks. In all experiments we used Euclidean norms for the Lipschitz constant (3).

Our implementation of CLIP is available on `github`.[3]

4.1 Nonlinear Regression

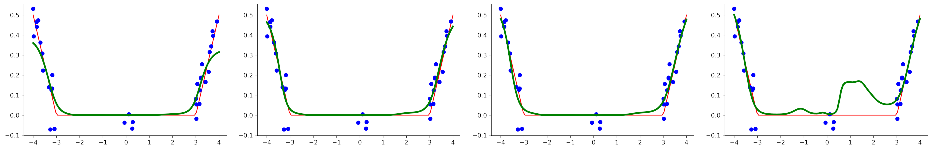
In this experiment we train a neural network to approximate the nonlinear function $f(x) = \frac{1}{2}\max(|x| - 3, 0)$ with three hidden layers consisting of 500, 200 and 100 neurons using the *sigmoid* activation function. We construct 100 noisy training pairs by sampling values of f on the set $[-4, -3] \cup [-0.3, 0.3] \cup [3, 4]$. The set of Lipschitz training samples is randomly chosen in every epoch and consists of tuples (x, x') where $x \in [-4, 4]$ and x' is a noise perturbation of x. Figure 2 shows the learned networks after applying the CLIP Algorithm 2 with different values of the regularization parameter λ. Note that for this experiment we do not use a target accuracy but choose a fixed value of λ. For $\lambda = 10$ the learned network is very smooth, which yields a poor approximation of the ground truth especially at the boundary of the domain. For decreasing values of λ the networks approach the ground truth solution steadily, showing no instabilities. In contrast, the unregularized network with $\lambda = 0$ shows strong oscillations in regions of missing data, which implies a poor generalization capability. Both for $\lambda = 10^{-10}$ and $\lambda = 0$ the trained networks do not fit the noisy data. A reason for this is that some of the data points are negative and we use non-negative sigmoid activation in the last layer. Also, using stochastic gradient descent implicitly regularizes the problem and avoids convergence to spurious critical points of the loss.

Note that we warmstart the computation for $\lambda = 10$ with the pretrained unregularized network. The computations for $\lambda = 1$ and $\lambda = 10^{-10}$ are warmstarted with the respective larger value of λ. The latter successive reduction of the regularization parameter is necessary in order to "select" the desired Lipschitz-minimal solution as $\lambda \to 0$, which resembles Bregman iterations [19].

4.2 Classification on MNIST and Fashion-MNIST

We evaluate the robustness of neural networks trained with the proposed variational CLIP algorithm on the popular MNIST [14] and Fashion-MNIST [28]

[3] `https://github.com/TimRoith/CLIP`



(a) $\lambda = 10$ (b) $\lambda = 1$ (c) $\lambda = 10^{-10}$ (d) $\lambda = 0$

Fig. 2: Lipschitz regularized networks for different values of λ. **Red:** ground truth. **Blue:** noisy ground truth samples. **Green:** trained neural network.

	Training Type	Train	Test	Noise	PGD	μ, λ
MNIST	Standard	**95.6**	89.8	71.8	25.3	0
	Weight Reg.$_{95}$	93.1	89.0	71.4	25.4	0.0003
	Weight Reg.$_{90}$	89.8	89.7	71.9	24.8	0.0015
	Weight Reg.$_{85}$	84.8	87.2	72.3	24.9	0.0031
	CLIP$_{95}$	95.3	90.7	70.3	24.4	0.01
	CLIP$_{90}$	91.8	**91.6**	**78.0**	36.2	3.91
	CLIP$_{85}$	87.6	87.2	74.5	**37.7**	9.82
Fashion-MNIST	Standard	**98.5**	**92.0**	25.7	1.9	0
	Weight Reg.$_{95}$	94.8	91.2	26.9	3.5	0.0011
	Weight Reg.$_{90}$	89.8	89.6	28.1	6.0	0.0023
	Weight Reg.$_{85}$	85.3	85.4	**34.2**	10.2	0.0091
	CLIP$_{95}$	94.6	**91.9**	19.8	9.1	0.18
	CLIP$_{90}$	90.6	88.9	23.3	13.0	0.74
	CLIP$_{85}$	85.9	83.1	31.9	**14.9**	4.18

Table 1: Accuracies [%] for different training setups, tested on train, test, noisy, and adversarial **(Fashion-)MNIST** data. The target accuracies are given by subscript. The regularization parameters μ and λ after training are also shown.

databases, which contain 28×28 grayscale images of handwritten digits and fashion articles, respectively. They are split into $60,000$ samples for training and $10,000$ samples for evaluation. For MNIST we train a neural network with two hidden layers with sigmoid activation function containing 64 neurons each. For Fashion-MNIST we use a simple convolutional neural network from [16]. We train non-regularized networks, CLIP regularized networks, and networks where we perform a L_2 regularization of the weights. The latter corresponds to (6) with $\lambda = 0$ and $\mu > 0$, where we update μ with the same discrepancy principle as for CLIP. Note that the weight regularization implies a layerwise Lipschitz bound of the type (4). The quantity $\mathcal{A}(f_\theta; \mathcal{T})$ in Algorithm 2 is chosen as the percentage of correctly classified training samples. We set the target accuracies for L_2 regularized and CLIP regularized networks to 85%, 90%, and 95%. The Lipschitz training set consists of 6,000 images removed from the training set and their noisy versions. We compare the robustness of all trained networks to adversarial perturbations $x + \delta$ of a sample $x \in \mathcal{X}$ in the L_2 norm, such that $\|\delta\| \leq \varepsilon$ with

$\varepsilon = 2$, calculated with the Projected Gradient Descent (PGD) attack [16] as

$$x_{\text{adv}}^{t+1} = \Pi \left(x_{\text{adv}}^{t} + \alpha \operatorname{sign}(\nabla_x \ell(f_\theta(x_{\text{adv}}^{t}), y) \right), \tag{14}$$

where $\alpha > 0$ is the step size and x_{adv}^{t} describes the adversarial example at iteration t. Here, $\Pi : \mathcal{X} \to \mathcal{X}$ is a projection operator onto the L_2-ball with radius ε, and sign is the componentwise signum operator. We set the step size $\alpha = 0.25$ and the total number of iterations to 100. Furthermore, we use Gaussian noise with zero mean and unit variance for an additional robustness evaluation. We track the performance against the PGD attack of each model during training and use the model checkpoint with the highest robustness for testing.

Table 1 shows the networks' performance for the MNIST and Fashion-MNIST databases. The CLIP algorithm increases robustness with respect to Gaussian noise and PGD attacks depending on the target accuracy. One observes an inversely proportional relationship between the target accuracy and the magnitude of the regularization parameter λ as expected from Proposition 2. The results for MNIST indicate a higher robustness of the CLIP regularized neural networks with respect to noise perturbations and adversarial attacks in comparison to unregularized or weight-regularized neural networks with similar accuracy on unperturbed test sets. Similar observations can be made for the Fashion-MNIST database. While our method is also more robust under adversarial attacks, it is slightly more prone to noise than the weight-regularized neural networks.

5 Conclusion and Outlook

We have proposed a variational regularization method to limit the Lipschitz constant of a neural network during training and derived *CLIP*, a stochastic gradient descent-ascent training method. Furthermore, we have studied the dependency of the trained neural networks on the regularization parameter and investigated the limiting behavior as the parameter tends to zero and infinity. Here, we have proved convergence to Lipschitz-minimal data fits or constant networks, respectively. We have evaluated the CLIP algorithm on regression and classification tasks and showed that our method effectively increases the stability of the learned neural networks compared to weight regularization methods and unregularized neural networks. In future work we will analyze convergence and theoretical guarantees of CLIP using techniques from stochastic analysis.

References

1. Anil, C., Lucas, J., Grosse, R.B.: Sorting Out Lipschitz Function Approximation. In: ICML. PMLR, vol. 97, pp. 291–301 (2019)
2. Anzengruber, S.W., Ramlau, R.: Morozov's discrepancy principle for Tikhonov-type functionals with nonlinear operators. Inverse Problems **26**(2), 025001 (2009)
3. Aziznejad, S., Gupta, H., Campos, J., Unser, M.: Deep neural networks with trainable activations and controlled Lipschitz constant. IEEE Transactions on Signal Processing **68**, 4688–4699 (2020)

4. Bungert, L., Burger, M.: Solution paths of variational regularization methods for inverse problems. Inverse Problems **35**(10), 105012 (2019)
5. Bungert, L., Burger, M., Korolev, Y., Schönlieb, C.B.: Variational regularisation for inverse problems with imperfect forward operators and general noise models. Inverse Problems **36**(12), 125014 (2020)
6. Burger, M., Osher, S.: A guide to the TV zoo. In: Level set and PDE based reconstruction methods in imaging, pp. 1–70. Springer (2013)
7. Combettes, P.L., Pesquet, J.C.: Lipschitz certificates for layered network structures driven by averaged activation operators. SIAM Journal on Mathematics of Data Science **2**(2), 529–557 (2020)
8. Fazlyab, M., Robey, A., Hassani, H., Morari, M., Pappas, G.: Efficient and Accurate Estimation of Lipschitz Constants for Deep Neural Networks. In: NIPS (2019)
9. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and Harnessing Adversarial Examples. In: ICLR (2015)
10. Gouk, H., Frank, E., Pfahringer, B., Cree, M.J.: Regularisation of neural networks by enforcing Lipschitz continuity. Machine Learning pp. 1–24 (2020)
11. Huster, T., Chiang, C.Y.J., Chadha, R.: Limitations of the Lipschitz constant as a defense against adversarial examples. In: ECML PKDD. pp. 16–29. Springer (2018)
12. Krishnan, V., Makdah, A.A.A., Pasqualetti, F.: Lipschitz Bounds and Provably Robust Training by Laplacian Smoothing. arXiv preprint arXiv:2006.03712 (2020)
13. Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech. rep. (2009)
14. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
15. Liang, Y., Huang, D.: Large Norms of CNN Layers Do Not Hurt Adversarial Robustness. arXiv preprint arXiv:2009.08435 (2020)
16. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards Deep Learning Models Resistant to Adversarial Attacks. In: ICLR (2018)
17. Oberman, A.M., Calder, J.: Lipschitz regularized deep neural networks converge and generalize. arXiv preprint arXiv:1808.09540 (2018)
18. van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A.W., Kavukcuoglu, K.: WaveNet: A Generative Model for Raw Audio. In: The 9th ISCA Speech Synthesis Workshop. p. 125 (2016)
19. Osher, S., Burger, M., Goldfarb, D., Xu, J., Yin, W.: An iterative regularization method for total variation-based image restoration. Multiscale Model Sim. **4**(2), 460–489 (2005)
20. Roth, K., Kilcher, Y., Hofmann, T.: Adversarial Training is a Form of Data-dependent Operator Norm Regularization. In: NeurIPS (2019)
21. Ruder, S.: An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747 (2016)
22. Scaman, K., Virmaux, A.: Lipschitz Regularity of Deep Neural Networks: Analysis and Efficient Estimation. In: NeurIPS (2018)
23. Schwinn, L., Raab, R., Eskofier, B.: Towards rapid and robust adversarial training with one-step attacks. arXiv preprint arXiv:2002.10097 (2020)
24. Shafahi, A., Najibi, M., Ghiasi, A., Xu, Z., Dickerson, J., Davis, L.S., Goldstein, T., Studer, C., Taylor, G.: Adversarial Training for Free! In: NeurIPS. pp. 3353–3364 (2019)
25. Shalev-Shwartz, S., Ben-David, S.: Understanding machine learning: From theory to algorithms. Cambridge university press (2014)

26. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. In: International Conference on Learning Representations (2014)
27. Terjék, D.: Adversarial Lipschitz regularization. arXiv preprint arXiv:1907.05681 (2019)
28. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms (2017)
29. Zou, D., Balan, R., Singh, M.: On Lipschitz bounds of general convolutional neural networks. IEEE Transactions on Information Theory **66**(3), 1738–1759 (2019)