Lipschitz Regularisation for Neural Networks in High Dimension

Raphaël BERNAS

ENSTA Paris - Institut Polytechnique de Paris

15^{th} August, 2024

Neural Networks : What does it look like ?

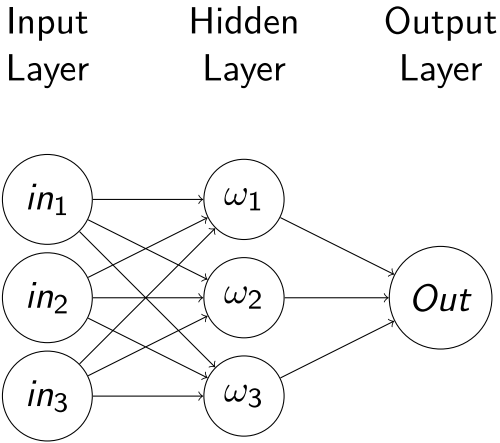


Figure. 1: Simplified neural network (one hidden layer).

Neural Networks : How to train them ?

We denote f_θ our model where $\theta = (\omega_1, \cdots, \omega_w)$.
Denote Γ our dataset containing couple $(x, y) \in \Gamma$:
$x \longrightarrow$ input (from a set noted $\mathcal{X}$)
$y \longrightarrow$ output (from a set noted $\mathcal{Y}$)

Neural Networks : How to train them ?

We denote f_θ our model where $\theta = (\omega_1, \cdots, \omega_w)$.
Denote Γ our dataset containing couple $(x, y) \in \Gamma$:
$x \longrightarrow$ input (from a set noted $\mathcal{X}$)
$y \longrightarrow$ output (from a set noted $\mathcal{Y}$)
Problem :

$$\min_\theta \sum_{(x,y)\in\Gamma} \text{loss}(f_\theta(x), y)$$

for example, *loss* could be : $\text{loss}(f_\theta(x), y) = \|f_\theta(x) - y\|^2$

Neural Networks : Is it correct ? Perfect ?

Perfect ? Obviously, not.

Overfitting :

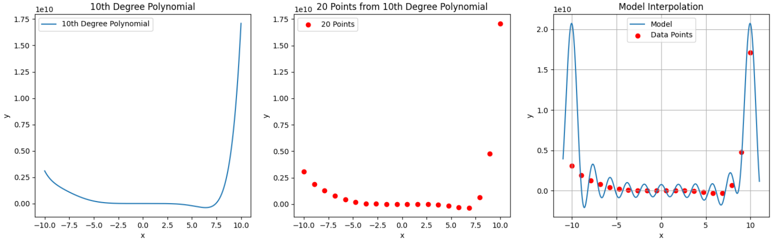


Figure. 2: Model interpolating a 10^{th} degrees polynomial function : Overfitting example.

Neural Networks : Is it correct ? Perfect ?

Perfect ? Obviously, not.

Robustness :

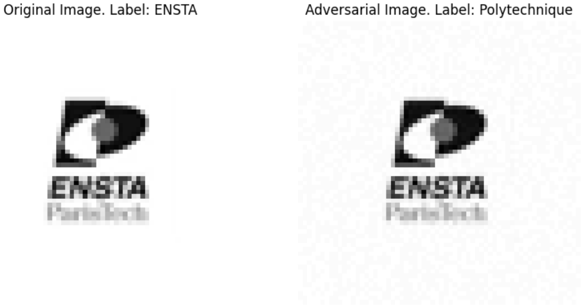


Figure. 3: Adversarial attacks on a model leading to misclassification (Find code on https://github.com/Raphael-Bernas/).

CLIP : Regularization

To regularize is to do an "intelligent training".

We introduce a new term *reg* to our theoretical training formula :

$$\min_{\theta} \sum_{(x,y)\in\Gamma} \text{loss}(f_\theta(x), y) + \lambda reg(f_\theta, \mathcal{X})$$

CLIP : Regularization

To regularize is to do an "intelligent training".
We introduce a new term *reg* to our theoretical training formula :

$$\min_{\theta} \sum_{(x,y)\in\Gamma} \text{loss}(f_\theta(x), y) + \lambda reg(f_\theta, \mathcal{X})$$

$\lambda \leftarrow$ weight of our regularizer
$\lambda >> 1$: then the model tends to be strongly regularized
$\lambda << 1$: then the regularizer has almost no effect

CLIP : Regularization

To regularize is to do an "intelligent training".
We introduce a new term *reg* to our theoretical training formula :

$$\min_{\theta} \sum_{(x,y)\in\Gamma} \text{loss}(f_\theta(x), y) + \lambda reg(f_\theta, \mathcal{X})$$

$\lambda \leftarrow$ weight of our regularizer
$\lambda >> 1$: then the model tends to be strongly regularized
$\lambda << 1$: then the regularizer has almost no effect
<u>ex :</u>

$$\min_{\theta} \left(\sum_{(x,y)\in\Gamma} \text{loss}(f_\theta(x), y) + 100 * \max_{x\in\mathcal{X}} \| f_\theta(x) - 8\|^2 \right)$$

CLIP : Lipschitz Constant

Lipschitz constant of f_θ : $Lip(f_\theta) = \sup_{(x,x') \in \mathcal{X}} \frac{\|f_\theta(x) - f_\theta(x')\|}{\|x - x'\|}$

CLIP : Lipschitz Constant

Lipschitz constant of f_θ : $Lip(f_\theta) = \sup_{(x,x')\in\mathcal{X}} \frac{\|f_\theta(x)-f_\theta(x')\|}{\|x-x'\|}$

f_θ is Lipschitz continuous if and only if $Lip(f_\theta) < +\infty$

$\hookrightarrow$ f_θ is continuous

$\hookrightarrow$ f_θ is almost everywhere differentiable (Rademacher Theorem)

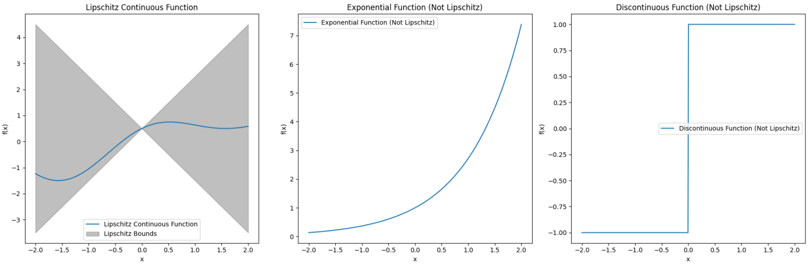


Figure. 4: Visualization of Lipschitz continuity.

CLIP : Lipschitz computation

Lipschitz regularization problem :

$$\min_{\theta} \sum_{(x,y)\in\Gamma} \text{loss}(f_\theta(x), y) + \lambda Lip(f_\theta)$$

CLIP : Lipschitz computation

Lipschitz regularization problem :

$$\min_{\theta} \sum_{(x,y)\in\Gamma} \text{loss}(f_\theta(x), y) + \lambda Lip(f_\theta)$$

But computing a Lipschitz constant is NP-hard ! [Sca18]

CLIP : Lipschitz computation

Lipschitz regularization problem :

$$\min_{\theta} \sum_{(x,y)\in\Gamma} \text{loss}(f_\theta(x), y) + \lambda Lip(f_\theta)$$

But computing a Lipschitz constant is NP-hard ! [Sca18]

Solution : Estimating a Lipschitz constant with cheap computational method [Bun22]

CLIP : Cheap Lipschitz

CLIP problem :

$$\min_{\theta} \sum_{(x,y)\in\Gamma} \text{loss}(f_\theta(x), y) + \lambda \max_{(x,x')\in\mathcal{X}_{lip}} \frac{\|f_\theta(x) - f_\theta(x')\|}{\|x - x'\|}$$

CLIP : Cheap Lipschitz

CLIP problem :

$$\min_{\theta} \sum_{(x,y)\in\Gamma} \text{loss}(f_\theta(x), y) + \lambda \max_{(x,x')\in\mathcal{X}_{lip}} \frac{\|f_\theta(x) - f_\theta(x')\|}{\|x - x'\|}$$

Where $\mathcal{X}_{lip}$ verify :

$$\mathcal{X}_{lip}^{k+1} \leftarrow AdversarialUpdate(f_\theta, \mathcal{X}_{lip}^{k})$$

AdversarialUpdate : For all $({}^k x, {}^k x') \in \mathcal{X}_{lip}^{k}$

$${}^{k+1}x \leftarrow {}^k x + \tau \nabla_x (Lip({}^k x, {}^k x'))^2$$

$${}^{k+1}x' \leftarrow {}^k x' + \tau \nabla_{x'} (Lip({}^k x, {}^k x'))^2$$

CLIP : Adversarial Update, how does it works ?

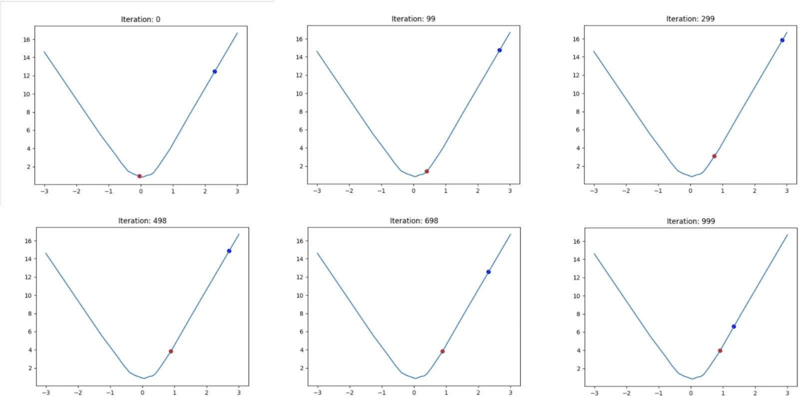


Figure. 5: Adversarial Update Evolution of $\mathcal{X}_{lip}$ (Click here to see this evolution in video).

CLIP : At what cost ?

Estimation precision : local extrema problem

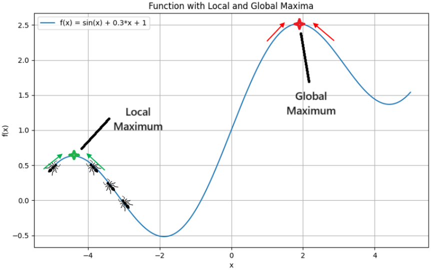


Figure. 6: Example of a function with global and local maxima, where the ant mistakenly reaches the local maximum.

CLIP : At what cost ?

Estimation precision : local extrema problem

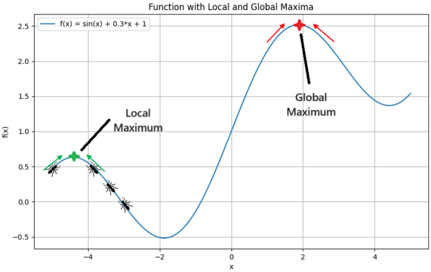


Figure. 6: Example of a function with global and local maxima, where the ant mistakenly reaches the local maximum.

Computation time : computing $\mathcal{X}_{lip}$ is still long

Test : Polynomial approximation

We define the following polynomial function :

$$P = \epsilon(0.5 + 0.01X + X^6)$$

Where ϵ is a scaling value.

Test : Polynomial approximation

We define the following polynomial function :

$$P = \epsilon(0.5 + 0.01X + X^6)$$

Where ϵ is a scaling value.
Then we define :

$$\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_N) \sim (\mathbb{U}([-3, -2] \cup [2, 3]))^N$$

Then the data set :

$$\Gamma = \{(\mathbf{x}_i, P(\mathbf{x}_i)) \ : \ i \in [|1, N|]\}$$

Test : Polynomial result

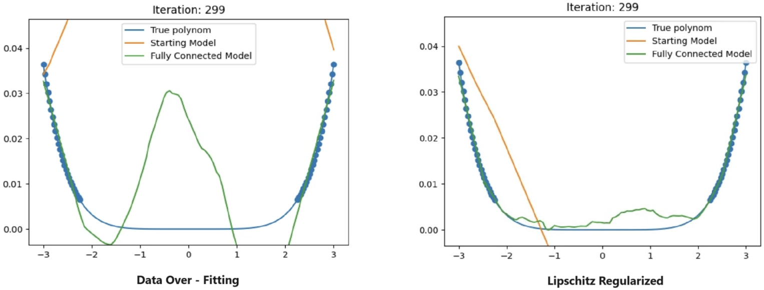


Figure. 7: DNN fitting polynomial function - Over-fitting VS lipschitz regularization (training video : with Lipschitz and without Lipschitz)

Test : Polynomial result remark

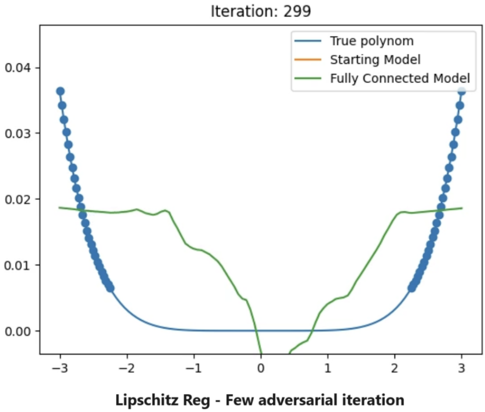


Figure. 8: DNN fitting polynomial function - lipschitz regularization with few X_{lip} update (training video : here)

Test : How does CLIP evolve with λ variations ?

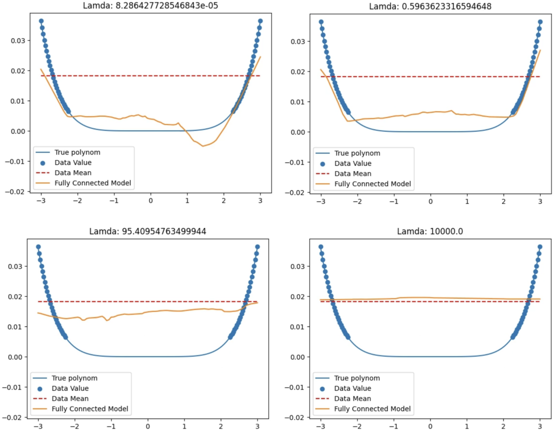


Figure. 9: Model evolution for different values of λ (Full evolution video : here)

Test : λ must be chosen carefully

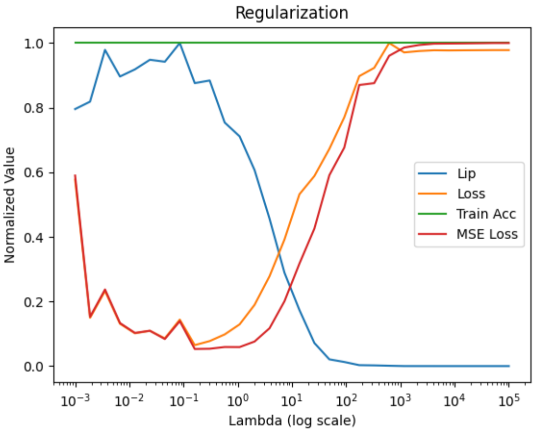


Figure. 10: Normalized Lipschitz constant, MSE Loss and train accuracy evolution for different values of λ

Improvement : Better & Faster

SGDM : Stochastic Gradient Descent Method

Basic CLIP :

$$\min_{\theta} \sum_{(x,y)\in\Gamma} \text{loss}(f_\theta(x), y) + \lambda CLip(f_\theta, \mathcal{X}_{lip}) \longleftarrow SGDM$$

Improvement : Better & Faster

SGDM : Stochastic Gradient Descent Method

Basic CLIP :

$$\min_{\theta} \sum_{(x,y)\in\Gamma} \mathrm{loss}(f_\theta(x), y) + \lambda CLip(f_\theta, \mathcal{X}_{lip}) \longleftarrow SGDM$$

Other possible gradient method :

NAGM : Nesterov Accelerated Gradient Method

Improvement : Better & Faster

SGDM : Stochastic Gradient Descent Method

Basic CLIP :

$$\min_{\theta} \sum_{(x,y)\in\Gamma} \text{loss}(f_\theta(x), y) + \lambda CLip(f_\theta, \mathcal{X}_{lip}) \longleftarrow SGDM$$

Other possible gradient method :

NAGM : Nesterov Accelerated Gradient Method

ADAM : Adaptive Moment Method ([Kin15])

Improvement : Optimizer test comparison

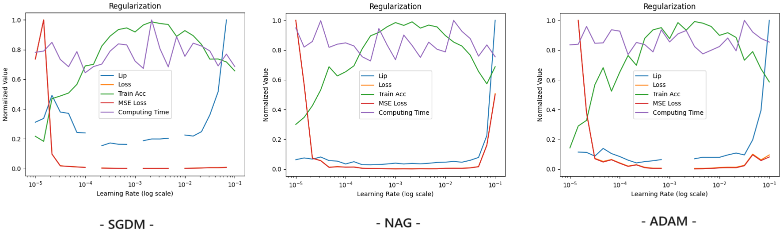


Figure. 11: Comparison of SGDM, NAG, ADAM results for log scaled learning rate - SGDM $time^{computation}_{maximal} = 0.0185\,s$ - NAG $time^{computation}_{maximal} = 0.0173\,s$ - ADAM $time^{computation}_{maximal} = 0.0176\,s$ -

Improvement : Stronger

Warm-up Phase :

To improve CLIP efficiency, we propose adding a "warm-up" phase to the CLIP algorithm.

- **First Phase:** [Pre-training] Train the model on the dataset Γ using only the regular loss (without regularization).

Improvement : Stronger

Warm-up Phase :

To improve CLIP efficiency, we propose adding a "warm-up" phase to the CLIP algorithm.

- **First Phase:** [Pre-training] Train the model on the dataset Γ using only the regular loss (without regularization).
- **Second Phase:** Stop the warm-up phase when the model's training accuracy reaches a predefined threshold α.

Improvement : Stronger

Warm-up Phase :

To improve CLIP efficiency, we propose adding a "warm-up" phase to the CLIP algorithm.

- **First Phase:** [Pre-training] Train the model on the dataset Γ using only the regular loss (without regularization).
- **Second Phase:** Stop the warm-up phase when the model's training accuracy reaches a predefined threshold α.
- **Third Phase:** [Training] Begin training the pre-trained model using the regularizer.

FLIP : Max & Sum

We compute :

$$MaxLip(f_\theta) = \max_{(x,x')\in\mathcal{X}_{lip}} \frac{\|f_\theta(x) - f_\theta(x')\|}{\|x - x'\|}$$

But to do so, we need to compute $\forall\,(x, x') \in \mathcal{X}_{lip},\ \frac{\|f_\theta(x)-f_\theta(x')\|}{\|x-x'\|}$

$\hookrightarrow$ Information loss...

$\hookrightarrow$ Computationally expensive...

FLIP : Max & Sum

We compute :

$$MaxLip(f_\theta) = \max_{(x,x')\in\mathcal{X}_{lip}} \frac{\|f_\theta(x) - f_\theta(x')\|}{\|x - x'\|}$$

But to do so, we need to compute $\forall\,(x, x') \in \mathcal{X}_{lip},\ \frac{\|f_\theta(x)-f_\theta(x')\|}{\|x-x'\|}$

$\hookrightarrow$ Information loss...

$\hookrightarrow$ Computationally expensive...

We introduce :

$$SumLip(f_\theta) = \frac{1}{|\mathcal{X}_{lip}|} \sum_{(x,x')\in\mathcal{X}_{lip}} \frac{\|f_\theta(x) - f_\theta(x')\|}{\|x - x'\|}$$

FLIP : From Sum FLIP to Total Variation (TV)

Definition : (Local Lipschitz constant) For $\epsilon > 0$ and $x \in \mathcal{X}$, we define $Lip_{\epsilon}^{loc}(f_\theta, x) = \sup_{x_1, x_2 \in \mathcal{B}_\epsilon(x)} \frac{||f_\theta(x_1) - f_\theta(x_2)||}{||x_1 - x_2||}$ where : $\mathcal{B}_\epsilon(x) = \{x' \in \mathcal{X}, \|x' - x\| \leq \epsilon\}$.

FLIP : From Sum FLIP to Total Variation (TV)

Definition : (Local Lipschitz constant) For $\epsilon > 0$ and $x \in \mathcal{X}$, we define $Lip_{\epsilon}^{loc}(f_\theta, x) = \sup_{x_1, x_2 \in \mathcal{B}_\epsilon(x)} \frac{||f_\theta(x_1) - f_\theta(x_2)||}{||x_1 - x_2||}$ where : $\mathcal{B}_\epsilon(x) = \{x' \in \mathcal{X}, \|x' - x\| \leq \epsilon\}$.

Remark : $(x, x') \in \mathcal{X}_{lip}, \ \frac{\|f_\theta(x) - f_\theta(x')\|}{\|x - x'\|} \underset{\epsilon > \|x - x'\|}{\approx} Lip_{\epsilon}^{loc}(f_\theta, x)$

FLIP : From Sum FLIP to Total Variation (TV)

Definition : (Local Lipschitz constant) For $\epsilon > 0$ and $x \in \mathcal{X}$, we define $Lip_{\epsilon}^{loc}(f_\theta, x) = \sup_{x_1, x_2 \in \mathcal{B}_\epsilon(x)} \frac{||f_\theta(x_1) - f_\theta(x_2)||}{||x_1 - x_2||}$ where :
$\mathcal{B}_\epsilon(x) = \{x' \in \mathcal{X}, \|x' - x\| \leq \epsilon\}$.

Remark : $(x, x') \in \mathcal{X}_{lip}, \frac{\|f_\theta(x) - f_\theta(x')\|}{\|x - x'\|} \underset{\epsilon > \|x - x'\|}{\approx} Lip_{\epsilon}^{loc}(f_\theta, x)$

Property : (Local Lipschitz to gradient)
$a.e\, x \in \mathcal{X}, \lim_{\epsilon \to 0} Lip_{\epsilon}^{loc}(f, x) = \|\nabla f(x)\|_*$, where :
$\|\cdot\|_* = \sup_{h \in \mathcal{B}_1(0)} \| \cdot h\|$.

FLIP : From Sum to TV

$$\frac{1}{|\mathcal{X}_{lip}|}\sum_{(x,x')\in\mathcal{X}_{lip}}\frac{\|f_\theta(x)-f_\theta(x')\|}{\|x-x'\|} \rightsquigarrow \frac{1}{|\Gamma_{\mathcal{X}}|}\sum_{x\in\Gamma_{\mathcal{X}}} Lip_\epsilon^{loc}(f_\theta, x)$$

FLIP : From Sum to TV

$$\frac{1}{|\mathcal{X}_{lip}|}\sum_{(x,x')\in\mathcal{X}_{lip}}\frac{\|f_\theta(x)-f_\theta(x')\|}{\|x-x'\|} \rightsquigarrow \frac{1}{|\Gamma_{\mathcal{X}}|}\sum_{x\in\Gamma_{\mathcal{X}}} Lip^{loc}_{\epsilon}(f_\theta, x)$$

$$\underset{\epsilon\rightarrow 0}{\rightsquigarrow} \frac{1}{|\Gamma_{\mathcal{X}}|}\sum_{x\in\Gamma_{\mathcal{X}}}\|\nabla f_\theta(x)\|_*$$

FLIP : From Sum to TV

$$\frac{1}{|\mathcal{X}_{lip}|}\sum_{(x,x')\in\mathcal{X}_{lip}}\frac{\|f_\theta(x)-f_\theta(x')\|}{\|x-x'\|} \rightsquigarrow \frac{1}{|\Gamma_{\mathcal{X}}|}\sum_{x\in\Gamma_{\mathcal{X}}} Lip_\epsilon^{loc}(f_\theta, x)$$

$$\underset{\epsilon\to 0}{\rightsquigarrow} \frac{1}{|\Gamma_{\mathcal{X}}|}\sum_{x\in\Gamma_{\mathcal{X}}}\|\nabla f_\theta(x)\|_*$$

Introducing TV regularization [Rud92] :

$$\min_\theta \sum_{(x,y)\in\Gamma} \text{loss}(f_\theta(x), y) + \lambda\int_{\mathcal{X}}\|\nabla f_\theta(x)\|_2\, dx.$$

↪ Not used for NN (but for image processing)

↪ Norm used is Euclidean norm

FLIP : TV Regularization

With Norm Equivalence Theorem[1] & Setting for exemple $\mathcal{X}$ as :
$\mathcal{X} = \{x \in \mathbb{R}^m,\ \|x\| < 1 + \underset{x_\gamma \in \Gamma_{\mathcal{X}}}{sup} \|x_\gamma\|\}$

$$SumLip(f_\theta) \rightsquigarrow \int_{\mathcal{X}} \|\nabla f_\theta(x)\|_* \, dx$$

FLIP : TV Regularization

With Norm Equivalence Theorem[1] & Setting for exemple $\mathcal{X}$ as :
$\mathcal{X} = \{x \in \mathbb{R}^m,\ \|x\| < 1 + \underset{x_\gamma \in \Gamma_{\mathcal{X}}}{sup} \|x_\gamma\|\}$

$$SumLip(f_\theta) \rightsquigarrow \int_{\mathcal{X}} \|\nabla f_\theta(x)\|_* \, dx$$

$$\rightsquigarrow \int_{\mathcal{X}} \lim_{\epsilon \to 0} Lip_\epsilon^{loc}(f_\theta, x) \, dx$$

FLIP : TV Regularization

With Norm Equivalence Theorem[1] & Setting for exemple $\mathcal{X}$ as :
$\mathcal{X} = \{x \in \mathbb{R}^m, \; \|x\| < 1 + \underset{x_\gamma \in \Gamma_{\mathcal{X}}}{sup} \|x_\gamma\|\}$

$$SumLip(f_\theta) \rightsquigarrow \int_{\mathcal{X}} \|\nabla f_\theta(x)\|_* \, dx$$

$$\rightsquigarrow \int_{\mathcal{X}} \lim_{\epsilon \to 0} Lip_\epsilon^{loc}(f_\theta, x) \, dx$$

Under peculiar assumptions, Lebesgue Dominated Convergence Theorem :

$$\int_{\mathcal{X}} \underset{\epsilon \to 0}{lim} \, Lip_\epsilon^{loc}(f_\theta, x) \, dx = \underset{\epsilon \to 0}{lim} \int_{\mathcal{X}} Lip_\epsilon^{loc}(f_\theta, x) \, dx$$

FLIP : Cheap computation for TV

Our computed TV regularizer use cheap Lipschitz :

$$\forall B \subset \Gamma, \quad pTV_{reg}(f_\theta, B, \epsilon_k) = \sum_{(x,y)\in B} \max_{\substack{(x_1,x_2)\in\mathcal{X}_{\text{lip}} \\ s.t\ \|x-x_{1,2}\|\leq\epsilon_k}} \frac{||f_\theta(x_1) - f_\theta(x_2)||}{||x_1 - x_2||}$$

With $\epsilon_k \underset{k\to+\infty}{\longrightarrow} 0$ where k denotes each epochs of our training.

FLIP : Cheap computation for TV

Our computed TV regularizer use cheap Lipschitz :

$$\forall B \subset \Gamma, \quad pTV_{reg}(f_\theta, B, \epsilon_k) = \sum_{(x,y)\in B} \max_{\substack{(x_1,x_2)\in\mathcal{X}_{\text{lip}} \\ s.t\ \|x-x_{1,2}\|\leq\epsilon_k}} \frac{||f_\theta(x_1) - f_\theta(x_2)||}{||x_1 - x_2||}$$

With $\epsilon_k \underset{k\to+\infty}{\longrightarrow} 0$ where k denotes each epochs of our training.

Let see what it looks like...

FLIP : Projected Finite Lipschitz Total Variation

for *epoch* $e = 1$ **to** E **do**
  for *minibatch* $B \subset \Gamma$ **do**
    $\mathcal{X}_{lip} \leftarrow B$
    for $(x_1, x_2) \in \mathcal{X}_{lip}$ **do**
      $x \leftarrow x_1$
      $x_1 \leftarrow x_1 + \tau \nabla_{x_1} (Lip(x_1, x_2))^2$
      $x_2 \leftarrow x_2 + \tau \nabla_{x_2} (Lip(x_1, x_2))^2$
      $x_1 \, ; \, x_2 \leftarrow x + \epsilon \frac{(x_1 - x)}{\|x_1 - x\|} \, ; \, x + \epsilon \frac{(x_2 - x)}{\|x_2 - x\|}$
    end
    $\theta \leftarrow \mathrm{SGDM}_{\eta,\gamma} \left(\frac{1}{|B|} \sum_{(x,y) \in B} loss(f_\theta(x), y) + \lambda \sum_{(x_1,x_2) \in \mathcal{X}_{lip}} Lip(f_\theta, (x_1, x_2)) \right)$
    $accuracy \leftarrow accuracy + \frac{1}{|B|} \sum_{(x,y) \in B} \mathbf{1}(x, y)_{\{f_\theta(x) = y\}}$
    if $accuracy > \alpha$ **then**
      $\lambda \leftarrow \lambda + d\lambda$
    end
    else
      $\lambda \leftarrow \lambda - d\lambda$
    end
  end
  $\epsilon \leftarrow \epsilon \times 0.9$
end

Input :
Training data Γ,
learning rate η,
momentum γ,
regularization parameter λ,
threshold α,
update step $d\lambda$,
initial ball ray ϵ,
total epochs E.

Output :
Trained model parameters θ.

TV : Assumptions

- **Assumption 1 :** Our *Loss* function is proper

TV : Assumptions

- **Assumption 1 :** Our *Loss* function is proper
- **Assumption 2 :** Our NN f_θ is Lipschitz continuous and continous upon θ

TV : Assumptions

- **Assumption 1 :** Our *Loss* function is proper
- **Assumption 2 :** Our NN f_θ is Lipschitz continuous and continous upon θ
- **Assumption 3 :** There exist θ_c such that f_{θ_c} is constant

TV : Assumptions

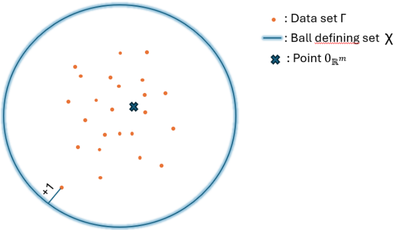
- **Assumption 1 :** Our *Loss* function is proper
- **Assumption 2 :** Our NN f_θ is Lipschitz continuous and continous upon θ
- **Assumption 3 :** There exist θ_c such that f_{θ_c} is constant
- **Assumption 4 :** The set $\mathcal{X}$ is bounded, open and connected (For example $\mathcal{X} = \{x \in \mathbb{R}^m, \ \|x\| < 1 + \sup_{x_\gamma \in \Gamma_\mathcal{X}} \|x_\gamma\|\}$).

TV : Assumptions

- **Assumption 1 :** Our *Loss* function is proper
- **Assumption 2 :** Our NN f_θ is Lipschitz continuous and continous upon θ
- **Assumption 3 :** There exist θ_c such that f_{θ_c} is constant
- **Assumption 4 :** The set $\mathcal{X}$ is bounded, open and connected (For example $\mathcal{X} = \{x \in \mathbb{R}^m, \; \|x\| < 1 + \sup_{x_\gamma \in \Gamma_\mathcal{X}} \|x_\gamma\|\}$).

TV : Assumptions

- **Assumption 1 :** Our *Loss* function is proper
- **Assumption 2 :** Our NN f_θ is Lipschitz continuous and continous upon θ
- **Assumption 3 :** There exist θ_c such that f_{θ_c} is constant
- **Assumption 4 :** The set $\mathcal{X}$ is bounded, open and connected (For example $\mathcal{X} = \{x \in \mathbb{R}^m,\ \|x\| < 1 + \sup_{x_\gamma \in \Gamma_\mathcal{X}} \|x_\gamma\|\}$).



TV : Solution existence

Under Assumption **1-4**.
Our problem admit a finite solution :

$$\exists \theta, \quad \sum_{(x,y)\in\Gamma} \mathrm{loss}(f_\theta(x), y) + \lambda \int_{\mathcal{X}} \lim_{\epsilon \to 0} Lip_\epsilon^{loc}(f_\theta, x)\, dx < +\infty.$$

TV : Solution existence

Under Assumption 1-4.
Our problem admit a finite solution :

$$\exists\theta, \quad \sum_{(x,y)\in\Gamma} \text{loss}(f_\theta(x), y) + \lambda \int_{\mathcal{X}} \lim_{\epsilon\to 0} Lip_\epsilon^{loc}(f_\theta, x)\, dx < +\infty.$$

And we have **Lemma** :

$$\theta \mapsto \int_{\mathcal{X}} \lim_{\epsilon\to 0} Lip_\epsilon^{loc}(f_\theta, x)\, dx \text{ is lower semi-continuous.}$$

TV : Solution existence

Under Assumption **1-4**.
Our problem admit a finite solution :

$$\exists \theta, \quad \sum_{(x,y)\in\Gamma} \mathrm{loss}(f_\theta(x), y) + \lambda \int_{\mathcal{X}} \lim_{\epsilon \to 0} Lip_\epsilon^{loc}(f_\theta, x)\, dx < +\infty.$$

And we have **Lemma** :

$$\theta \mapsto \int_{\mathcal{X}} \lim_{\epsilon \to 0} Lip_\epsilon^{loc}(f_\theta, x)\, dx \text{ is lower semi-continuous.}$$

Thus : **Existence Theorem :**

$$\min_\theta \sum_{(x,y)\in\Gamma} \mathrm{loss}(f_\theta(x), y) + \lambda \int_{\mathcal{X}} \lim_{\epsilon \to 0} Lip_\epsilon^{loc}(f_\theta, x)\, dx \quad \textit{admit a solution}$$

TV : Properties

Under Assumption **1-4** :

TV : Properties

Under Assumption **1-4** :

Property (A limit for infinite λ) : There exist θ^*_∞ a limit : $\theta^*_{\lambda_n} \underset{\lambda_n \to +\infty}{\longrightarrow} \theta^*_\infty$ such that TV-Problem with λ_n is solved by $f_{\theta^*_{\lambda_n}}$ and $f_{\theta^*_\infty}$ is constant.

TV : Properties

Under Assumption 1-4 :

Property (A limit for infinite λ) : There exist θ^*_∞ a limit : $\theta^*_{\lambda_n} \underset{\lambda_n \to +\infty}{\longrightarrow} \theta^*_\infty$ such that TV-Problem with λ_n is solved by $f_{\theta^*_{\lambda_n}}$ and $f_{\theta^*_\infty}$ is constant.

Property (Infinite limit constant value) : We have that $f_{\theta^*_\infty} = \underset{y_\theta \in \mathcal{Y}_\Theta}{\arg\min} \left\{ \sum_{(x,y)\in\Gamma} \text{loss}(y_\theta, y) \right\}$, where $\mathcal{Y}_\Theta = \{y \in \mathcal{Y} \mid \exists \theta \in \Theta, \ \forall x \in \mathcal{X}, \ f_\theta(x) = y\}$.

TV : Illustrated property

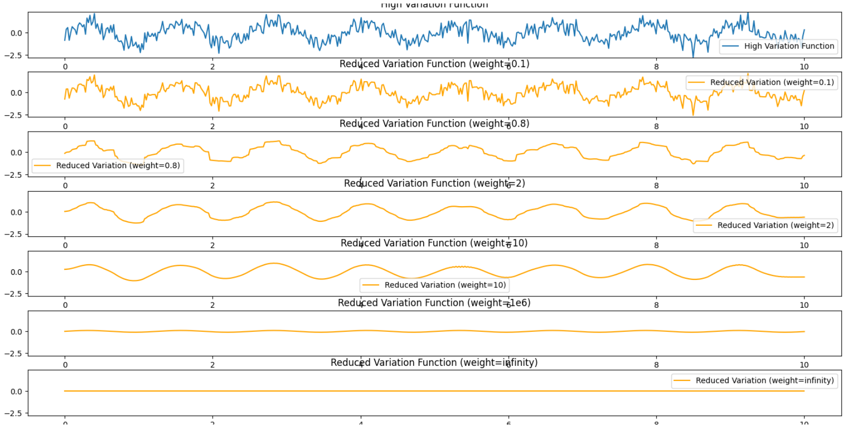


Figure. 12: Illustration of TV regularization with an increasing weight

TV : Algorithm Validity

With our algorithm we solve :

$$\min_{\theta} \sum_{(x,y)\in\Gamma} \mathrm{loss}(f_\theta(x), y) + \int_{\mathcal{X}} Lip_{\epsilon}^{loc}(f_\theta, x)\, dx$$

for different $\epsilon \longrightarrow 0$. But...

TV : Algorithm Validity

With our algorithm we solve :

$$\min_{\theta} \sum_{(x,y)\in\Gamma} \mathrm{loss}(f_\theta(x), y) + \int_{\mathcal{X}} Lip_{\epsilon}^{loc}(f_\theta, x)\, dx$$

for different $\epsilon \longrightarrow 0$. But...
Does it solve :

$$\min_{\theta} \sum_{(x,y)\in\Gamma} \mathrm{loss}(f_\theta(x), y) + \int_{\mathcal{X}} \lim_{\epsilon\to 0} Lip_{\epsilon}^{loc}(f_\theta, x)\, dx$$

TV : Gamma convergence

Using [Shi24] we can prove in $(1, p)$-Sobolev spaces that :

$$\int_{\mathcal{X}} (Lip_{\epsilon}^{loc}(f_{\theta}, x))^p \, dx \xrightarrow[\epsilon \to 0]{\gamma} \int_{\mathcal{X}} \lim_{\epsilon \to 0} (Lip_{\epsilon}^{loc}(f_{\theta}, x))^p \, dx$$

γ-convergence is a weaker convergence that permits to show minimizer convergence.

TV : Gamma convergence

Using [Shi24] we can prove in $(1, p)$-Sobolev spaces that :

$$\int_{\mathcal{X}} (Lip_{\epsilon}^{loc}(f_\theta, x))^p \, dx \xrightarrow[\epsilon \to 0]{\gamma} \int_{\mathcal{X}} \lim_{\epsilon \to 0} (Lip_{\epsilon}^{loc}(f_\theta, x))^p \, dx$$

γ-convergence is a weaker convergence that permits to show minimizer convergence.

But can we add the term : $\sum_{(x,y)\in\Gamma} \text{loss}(f_\theta(x), y)$?

Not for all $p \in \mathbb{N}$... Issue :

TV : Gamma convergence

Using [Shi24] we can prove in $(1, p)$-Sobolev spaces that :

$$\int_{\mathcal{X}} (Lip_{\epsilon}^{loc}(f_\theta, x))^p \, dx \xrightarrow[\epsilon \to 0]{\gamma} \int_{\mathcal{X}} \lim_{\epsilon \to 0} (Lip_{\epsilon}^{loc}(f_\theta, x))^p \, dx$$

γ-convergence is a weaker convergence that permits to show minimizer convergence.
But can we add the term : $\sum_{(x,y)\in\Gamma} \text{loss}(f_\theta(x), y)$?
Not for all $p \in \mathbb{N}$... Issue :
$f_\theta(x)$ impossible in Sobolev spaces for $p < n$.
$\hookrightarrow$ Need Sobolev Embedding Theorem in continuous function which is true for $p \geq n$.
(Reminder : we are set in $\mathbb{R}^n$)

TV : Gamma convergence

But for MNIST images $n = 784$ and for high quality images n easily explode...

TV : Gamma convergence

But for MNIST images $n = 784$ and for high quality images n easily explode...
For such n, if $p \geq n$ we are computationally near $p = +\infty$ which is basically computing a Lipschitz constant.

TV : Gamma convergence

But for MNIST images $n = 784$ and for high quality images n easily explode...
For such n, if $p \geq n$ we are computationally near $p = +\infty$ which is basically computing a Lipschitz constant.
But using Sobolev Embedding Theorem just for continuity here is **OVERKILL** !
Thus, we need to change our space !

The article direction in which we are going :

Using Barron spaces

$\hookrightarrow$ Neural Networks spaces

Results : Adversarial Attacks

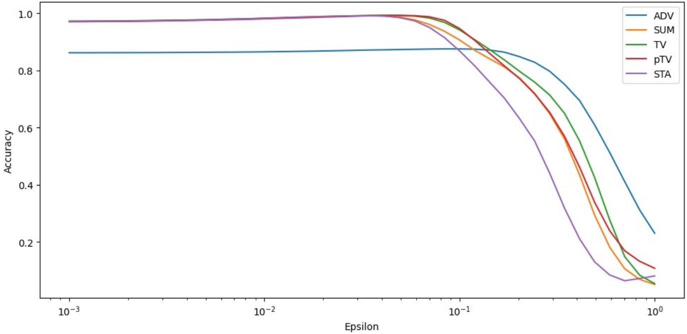


Figure. 13: Comparison of accuracy against **FGSM** adversarial attacks for different training methods.

Find the code for those experiments here : https://github.com/TimRoith/CLIP

Results : Adversarial Attacks

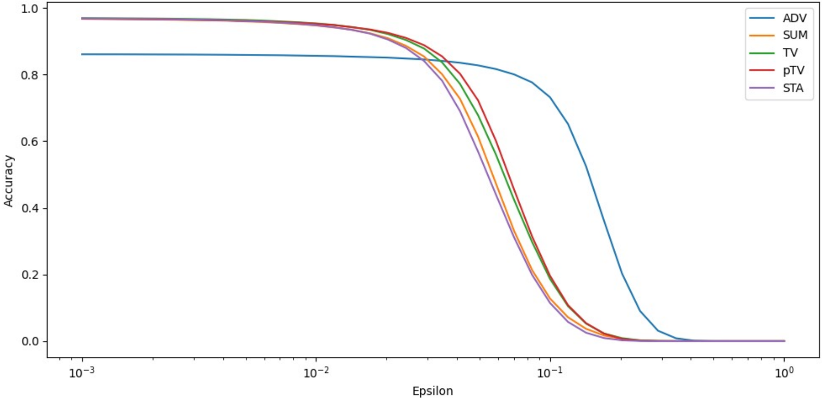


Figure. 14: Comparison of accuracy against **PGD** adversarial attacks for different training methods.

Results : Noise Attack

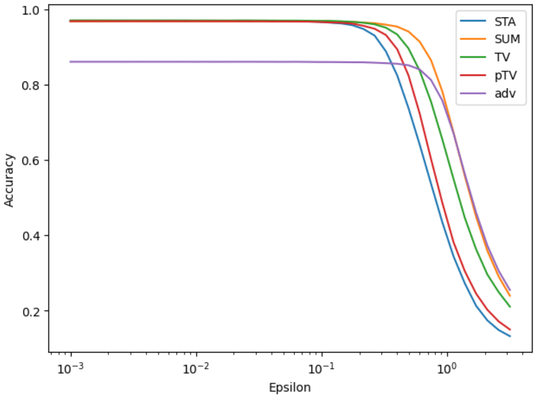


Figure. 15: Comparison of accuracy against Gaussian noise for different training methods.

Results : Training Computation Time

Estimator	Train Accuracy	Computation Time (s)
SUM FLIP	0.979	126
ADV FGSM	0.837	116
TV	0.981	95
pTV	0.978	94
STA	0.979	54

Table. 1: Performance comparison of different training methods (On DESY server hardware : AMD EPYC 7402).

Conclusion: Towards Robust Neural Network Training

Exploration of robust training methods - Key Findings:

Conclusion: Towards Robust Neural Network Training

Exploration of robust training methods - Key Findings:

- Validation of CLIP algorithm: Ensured stability and efficiency on 1D problems before scaling.

Conclusion: Towards Robust Neural Network Training

Exploration of robust training methods - Key Findings:

- Validation of CLIP algorithm: Ensured stability and efficiency on 1D problems before scaling.
- Novel Regularizer Development: Derived a TV-based regularizer, preserving edges and reducing noise, with theoretical and empirical support.

Conclusion: Towards Robust Neural Network Training

Exploration of robust training methods - Key Findings:

- Validation of CLIP algorithm: Ensured stability and efficiency on 1D problems before scaling.
- Novel Regularizer Development: Derived a TV-based regularizer, preserving edges and reducing noise, with theoretical and empirical support.
- Theoretical Advances: Studied theoretical behavior of TV. Utilized γ-convergence to verify correctness of our algorithm.

Conclusion: Towards Robust Neural Network Training

Exploration of robust training methods - Key Findings:

- Validation of CLIP algorithm: Ensured stability and efficiency on 1D problems before scaling.
- Novel Regularizer Development: Derived a TV-based regularizer, preserving edges and reducing noise, with theoretical and empirical support.
- Theoretical Advances: Studied theoretical behavior of TV. Utilized γ-convergence to verify correctness of our algorithm.
- Empirical Results: TV-based methods demonstrated superior robustness under FGSM/PGD attacks and Gaussian noise compared to SUM FLIP and even sometime compared to adversarial training.

Conclusion: Towards Robust Neural Network Training

Exploration of robust training methods - Future Directions:

- Refine algorithms for more accurate TV estimation.
- Extend theoretical analysis for FLIP regularizers.
- Apply to complex, real-world datasets to test scalability.

1. Introduction
2. CLIP : Test & Improvement
3. FLIP : Finite Lipschitz
4. Conclusion
5. References

[Bun22] Raab R. Roith T. Schwinn L. Tenbrinck D. Bungert, L.
Clip: Cheap lipschitz training of neural networks.
arXiv, preprint arXiv:2103.12531, (2022).

[Kin15] Lei Ba J. Kingma, D. P.
Adam: A method for stochastic optimization.
ICLR, (2015).

[Rud92] Osher S. Fatemi E. Rudin, L. I.
Nonlinear total variation based noise removal algorithms.
Physica, D. 60 (1–4): 259–268., (1992).

[Sca18] Virmaux A. Scaman, K.
Lipschitz regularity of deep neural networks: Analysis and efficient estimation.
In: NeurIPS, (2018).

[Shi24] Burger M. Shi, K.
Hypergraph p-laplacian regularization on point clouds for data interpolation.
arXiv, preprint arXiv:2405.01109, (2024).

Thanks!!!