SSSmoothRazor:

SynthesiS of Smooth parameters using Ockham's Razor

Laurent Fribourg

CNRS & ENS Paris-Saclay
(Laboratoire Méthodes Formelles)

SYNCOP Workshop – Luxemburg (April 06-07, 2024)

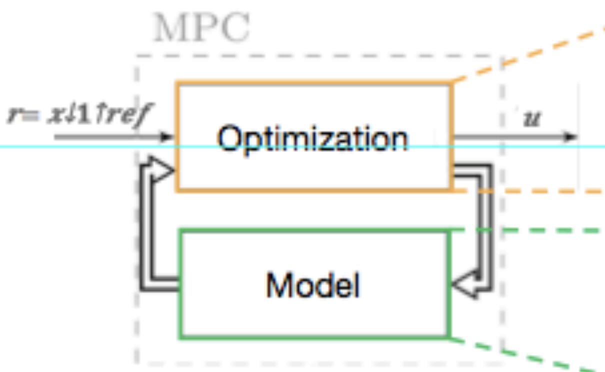
PLAN

1. Data-Driven Control
2. 1-hidden layer Neural Network
3. Gradient Descent
4. Training Error
5. Generalization Error
6. Early Stopping

MODEL PREDICTIVE CONTROL (MPC)

- **MPC simulation:** carried out offline with the goal of following a reference trajectory

x_1^{ref}.
  - ✓ Prediction Horizon: $N=5$
  - ✓ Initial condition: $x_0=[1,0]$
  - ✓ Time step: $T_s=0.5$s
  - ✓ Scale factor: $\alpha=0.1$



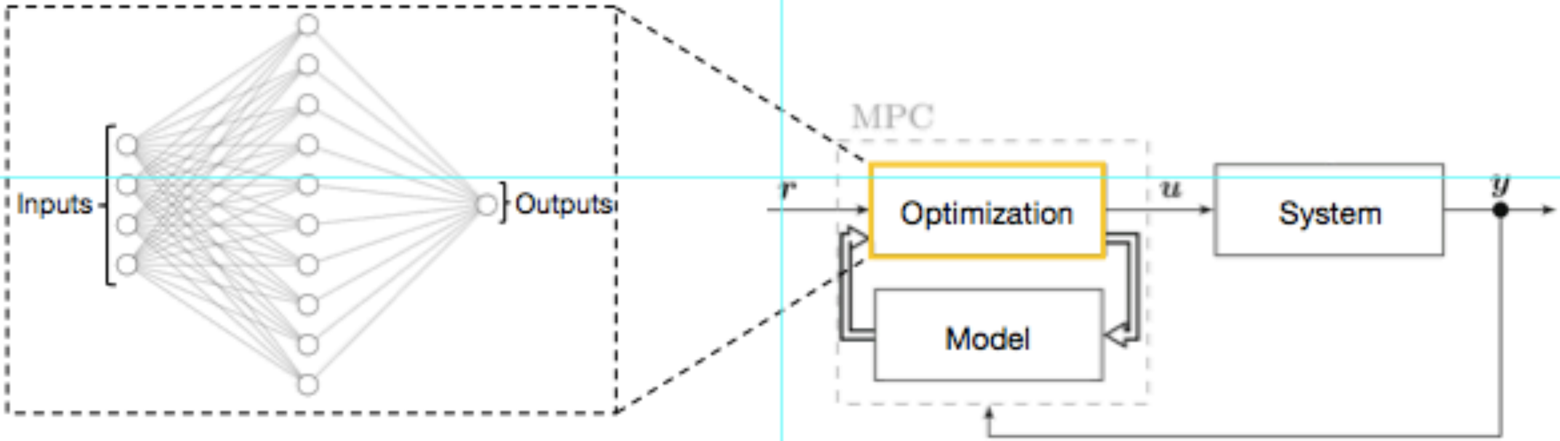
$$u=\arg\min_{u} \quad J(x,u)=\sum_{\ell=0}^{N-1} \|x_1^{ref}(k+\ell)-x_1(k+\ell)\|+\alpha\|\Delta u(k+\ell)\|$$

$$\begin{cases} \dot{x}_1=x_2 \\ \dot{x}_2=(1-x_1^2)x_2-x_1-u \end{cases}$$

- **Synthetic data:** ✓ Training: $S_{train}=6000$ ✓ Test: $S_{test}=2000$

Simulation of MPC with a Neural Network

- Simulation of an MPC controller to **quickly** compute the command from data obtained **offline** (**Chen et al. 2018**).
    - **Method:** use a **neural network** to mimic the behavior of an MPC controller.



- **Supervised learning:**
    - **Data:** obtained from an **MPC simulation.**

Example: Control of Van der Pol Oscillator

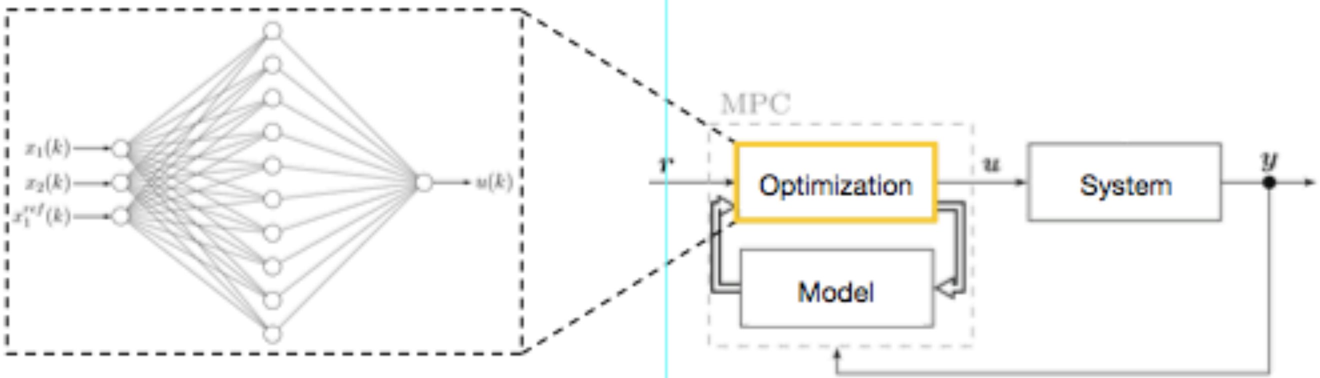
- **Van der Pol oscillator:** models the oscillations of triodes in electrical circuits.
  - **Model:**

$$\begin{cases} x_1 = x_2 \\ x_2 = (1 - x_1^2)x_2 - x_1 - u \end{cases}$$

where x_1 is the **position**, x_1^{ref} the **reference**, x_2 the **speed** and u the **command**.

  - **Constraints :** $u \in [-1, 1]$ and x_1 , $x_2 \in [-3, 3]$ (**Antonelo et al. 2022**).

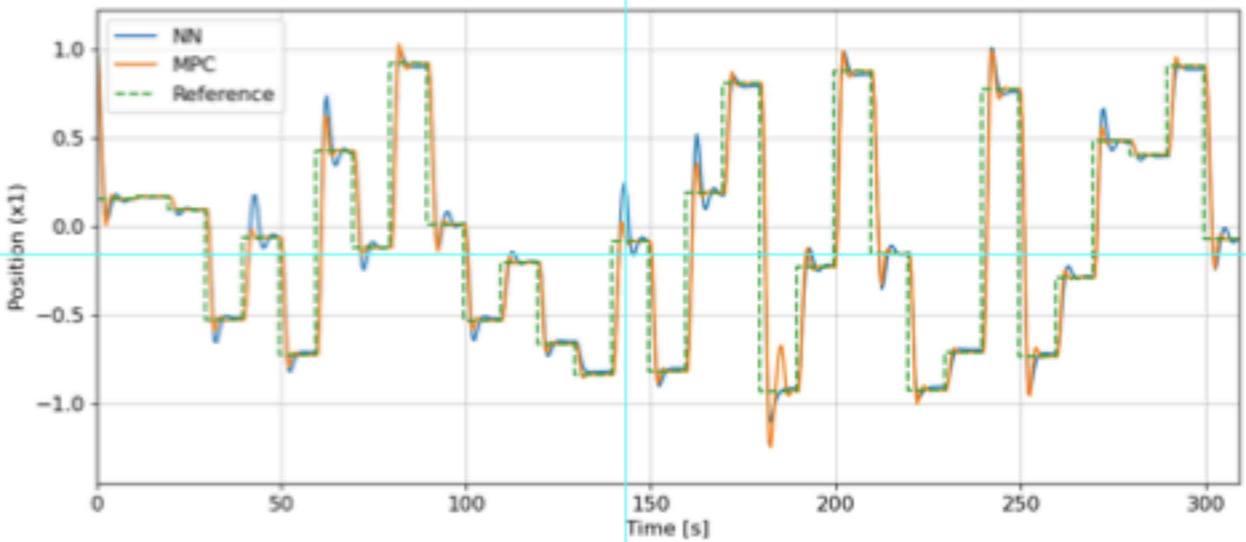
- **Goal:** make the system converge towards a reference trajectory x_1^{ref}.



  - **Data:** obtained from an **MPC simulation.**

Comparison between **MPC** and **NN**

- **MPC vs Neural Network (supervised):** closed-loop simulation using a reference trajectory.



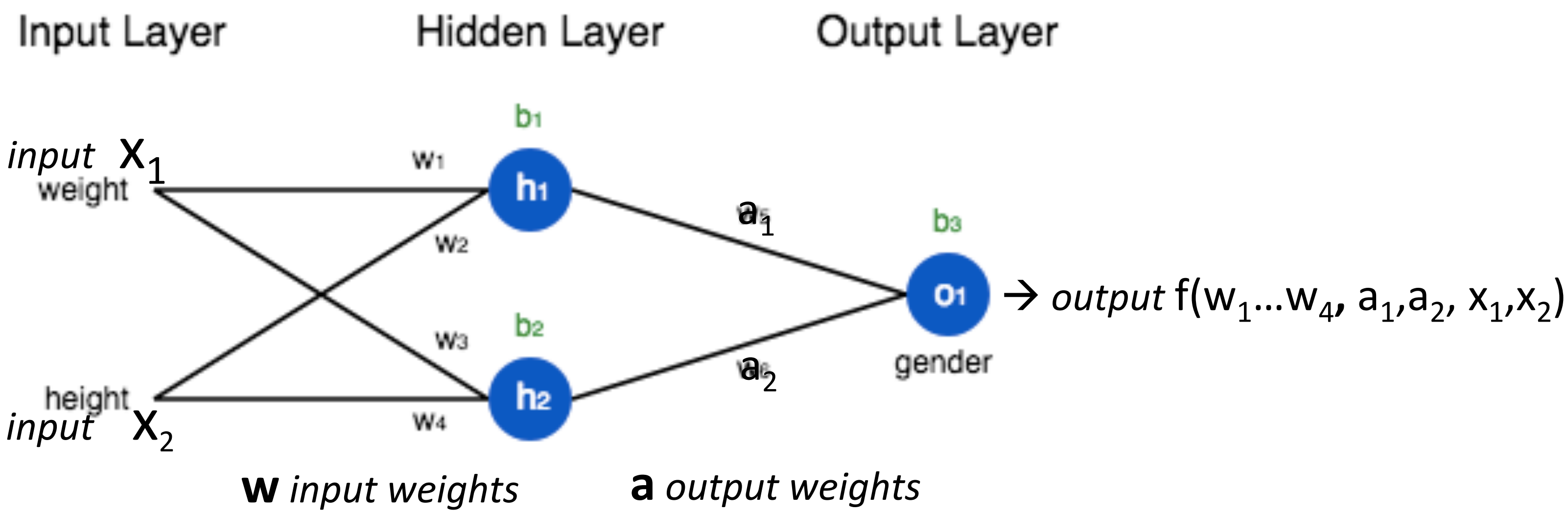
Mean squared error:

- ✓ Neural Network = 0.067
- ✓ MPC = 0.066

Computational cost:

- ✓ Neural Network = 0.17 ms
- ✓ MPC = 2.34 ms

Neural Network with **1 hidden layer**



$$f(w_1,\ldots,w_4,\ a_1,a_2,\ x_1,x_2) = \sigma(w_1x_1 + w_2x_2) \times a_1 + \sigma(w_3x_1 + w_4x_2) \times a_2$$

Compact form: $f(\mathbf{w}, \mathbf{a}, \mathbf{x}) = \sum_{r=1,2} \sigma(\mathbf{w}_r^T \mathbf{x}) \times a_r$ with $\mathbf{w}_1 = (w_1, w_2)^T$, $\mathbf{w}_2 = (w_3, w_4)^T$

Neural Network (2)

We consider NN with **1 hidden layer**:

output: $f(\mathbf{w}, \mathbf{x}) = 1/\sqrt{m} \;\; \sum_{r=1..m} \; a_r \, \sigma(\mathbf{w}_r^T \, \mathbf{x})$

where:

- $\mathbf{x}$ in R^d *input data*
- $\mathbf{w} = (\mathbf{w}_1, \ldots, \mathbf{w}_m)^T$ with $\mathbf{w}_r$ in R^d *input weights*
- $\mathbf{a} = (a_1, \ldots, a_m)^T$ with a_r in R *output weights*
- $\sigma(\cdot)$ **nonlinear** *activation* function (e.g.: $\sigma(z)=\max(z,0)$ for ReLU)

Besides output weight **a** assumed **fixed** (**a** = *unif* $\{\pm 1\}$)

Training error minimization

Problem:

Given the training data set $S = \{ (\mathbf{x}_i, y_i) \}_{i=1,\ldots,n}$

minimize the *quadratic loss*:

$$L_S(\mathbf{w}) = \tfrac{1}{2} \sum_{i=1 \ldots n} (f(\mathbf{w}, \mathbf{x}_i) - y_i)^2 = \tfrac{1}{2} \sum_i |v_i|^2 = \tfrac{1}{2} |\mathbf{v}|^2$$

with $\mathbf{v} := (v_1, \ldots, v_n)^T$ *training error vector*

and $v_i := f(\mathbf{w}, \mathbf{x}_i) - y_i$ $\quad i = 1, \ldots, n.$

GRADIENT DESCENT

Apply **GD** on **w**. In continuous time with r =1, … , m:

$$\frac{d\mathbf{w}_r(t)}{dt} = -\frac{\partial \mathcal{L}_S(\mathbf{w})}{\partial \mathbf{w}_r}$$

$$= -\sum_{i=1}^{n} v_i \frac{\partial f(\mathbf{w}, \mathbf{x}_i)}{\partial \mathbf{w}_r}.$$

For example for ReLU:

$$\frac{d\mathbf{w}_r(t)}{dt} = -\frac{a_r}{\sqrt{m}} \sum_{i=1}^{n} v_i \mathbf{x}_i \mathbb{I}\{\mathbf{w}_r^\top \mathbf{x}_i \geqslant 0\}.$$

with $\mathbb{I}$ indicator event $\mathbf{w}_r^\top \mathbf{x}_i \geqslant 0$ happens.

GRADIENT DESCENT (2)

$$\frac{d\mathbf{w}_r(t)}{dt} = -\sum_{i=1}^{n} v_i \frac{\partial f(\mathbf{w}, \mathbf{x}_i)}{\partial \mathbf{w}_r}$$

$$= -\sum_{i=1}^{n} v_i \frac{1}{\sqrt{m}} a_r \sigma'(\mathbf{x}_i^\top \mathbf{w}_r) \frac{\partial \mathbf{x}_i^\top \mathbf{w}_r}{\partial \mathbf{w}_r}$$

$$= -\frac{a_r}{\sqrt{m}} \sum_{i=1}^{n} v_i \sigma'(\mathbf{x}_i^\top \mathbf{w}_r) \mathbf{x}_i.$$

It follows:

$$\frac{d}{dt}\|\mathbf{w}_r(t)\| \leqslant \frac{1}{\sqrt{m}} \sum_{i=1}^{n} \|v_i(t)\| \quad \leq \sqrt{n}/\sqrt{m} \;\; |\mathbf{v}(t)|$$

because: $|a_r| = 1$, $\sigma'(z) \leq 1$, $|\mathbf{x}_i| = 1$ (normalized input data)

Training error dynamics

Consider $\mathbf{v}(t) = (v_1, \dots, v_n)^T$ with $v_i = f(\mathbf{w}, \mathbf{x}_i) - y_i$

The continuous dynamics of $\mathbf{v}(t)$ is given by:

$$\frac{d}{dt}\mathbf{v}(t) = -\mathbf{H}[\mathbf{w}(t)]\mathbf{v}(t), \quad \mathbf{v}(0) = \mathbf{v}_0$$

with for i, j in {1,…,n}:

$$\mathbf{H}_{i,j} := \sum_{r=1}^{m} \left\langle \frac{\partial f(\mathbf{w}, \mathbf{x}_i)}{\partial \mathbf{w}_r}, \frac{\partial f(\mathbf{w}, \mathbf{x}_j)}{\partial \mathbf{w}_r} \right\rangle$$

and $\mathbf{w} = (\mathbf{w}_1, \dots, \mathbf{w}_m)^T$

Proof. For $i \in [n]$:

$$\begin{aligned}
\frac{d}{dt}v_i &= \frac{d}{dt}(f(\mathbf{w}, \mathbf{x}_i) - y_i) \\
&= \langle \frac{\partial f(\mathbf{w}, \mathbf{x}_i)}{\partial \mathbf{w}}, \frac{d\mathbf{w}}{dt} \rangle \\
&= \sum_{=r1}^{m} \langle \frac{\partial f(\mathbf{w}, \mathbf{x}_i)}{\partial \mathbf{w}_r}, \frac{d\mathbf{w}_r}{dt} \rangle \\
&= -\sum_{r=1}^{m} \langle \frac{\partial f(\mathbf{w}, \mathbf{x}_i)}{\partial \mathbf{w}_r}, \frac{\partial \mathcal{L}_S(\mathbf{w})}{\partial \mathbf{w}_r} \rangle \\
&= -\sum_{r=1}^{m} \langle \frac{\partial f(\mathbf{w}, \mathbf{x}_i)}{\partial \mathbf{w}_r}, \sum_{j=1}^{n} v_j \frac{\partial f(\mathbf{w}, \mathbf{x}_j)}{\partial \mathbf{w}_r} \rangle \\
&= -\sum_{j=1}^{n} \left[\sum_{r=1}^{m} \langle \frac{\partial f(\mathbf{w}_r, \mathbf{x}_i)}{\partial \mathbf{w}_r}, \frac{\partial f(\mathbf{w}, \mathbf{x}_j)}{\partial \mathbf{w}_r} \rangle \right] v_j.
\end{aligned}$$

Hence:

$$\frac{d}{dt}v_i = -\sum_{j=1}^{n} \mathbf{H}_{i,j} v_j.$$

□

Training error dynamics (2)

H[**w**] symmetric **Gram** time-varying matrix called:

Neural Tangent Kernel *(NTK)* or *Input Data Covariance* matrix

For ReLU: **H[w]** n×n matrix with (*i, j*)-th entry:

$$\mathbf{H}_{ij} = \frac{1}{m}\mathbf{x}_i^\top \mathbf{x}_j \sum_{r=1}^{m} \mathbb{I}\{\mathbf{x}_i^\top \mathbf{w}_r \geqslant 0, \mathbf{x}_j^\top \mathbf{w}_r \geqslant 0\}.$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are *i*-th and *j*-th elements of input data set *S*

Convergence of $|\mathbf{v}(t)|$ (case $\boldsymbol{\lambda_n > 0}$)

Let

- $U_1(t), \ldots, U_n(t)$ **eigenvectors** of the NTK $\mathbf{H}[\mathbf{w}(t)]$ at time t,
- $\lambda_1(t), \ldots, \lambda_n(t)$ **eigenvalues** (they are all ≥ 0),
- λ_i **lower bound** of $\lambda_i(t)$ for $t \geq 0$ ($i = 1, \ldots, n$):

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0.$$

Particular case (holding if $m >> n$, i.e. *overparameterized* NN) : $\lambda_n > 0$.

Then *[Jaqot et al. 2019]* : $|\mathbf{v}(t)| \leq |\mathbf{v_0}| \exp(-\boldsymbol{\lambda_n} t)$.

$|\mathbf{v}(t)|$ converges **linearly** to **0** as $t \to \infty$, *whatever* initial weight $\mathbf{w}(0)$

→ All the valleys of the *loss landscape* of **overparameterized** NNs are connected (all the local minima are « global »)

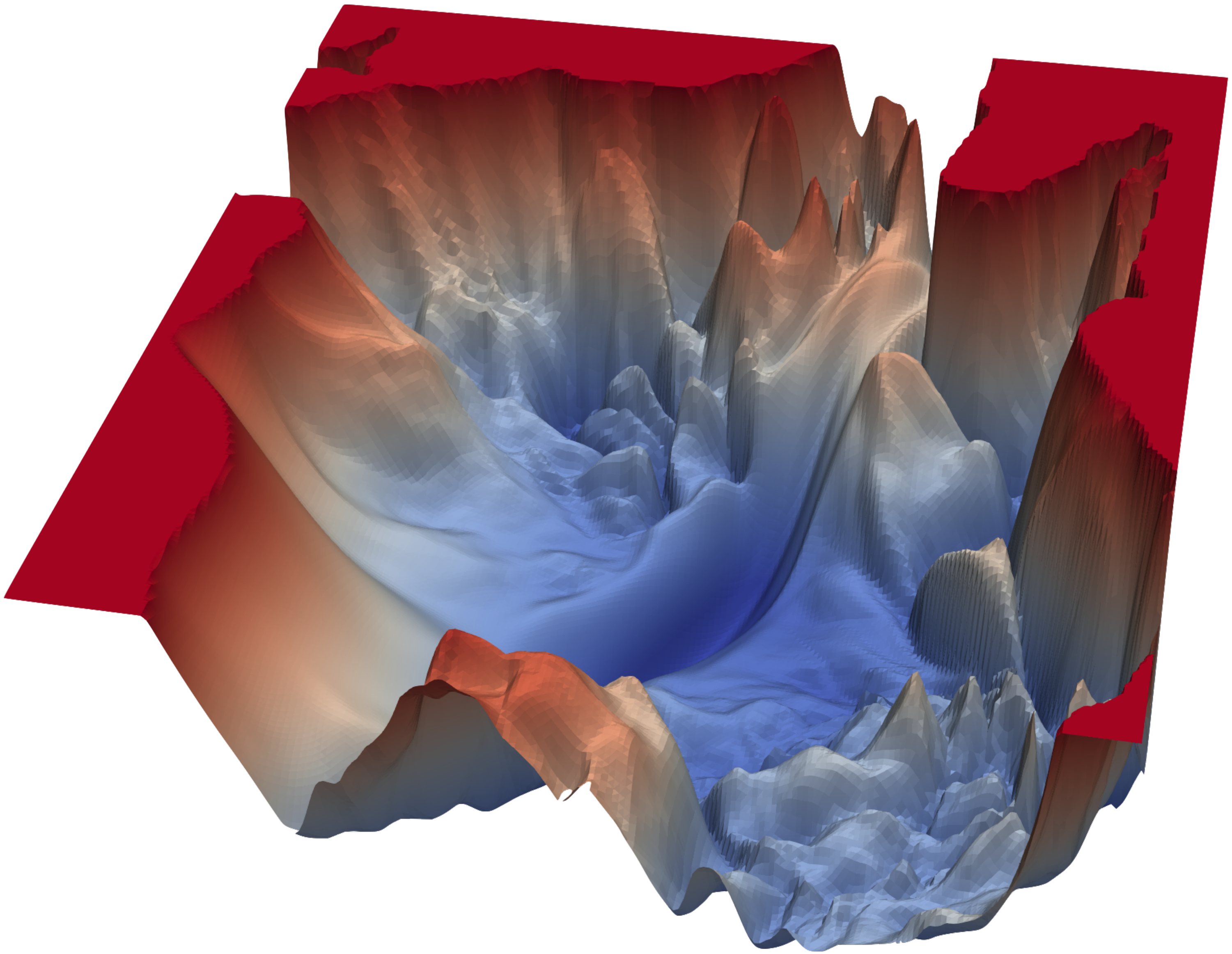
Proof

$$\tfrac{1}{2}\,(d|\mathbf{v}|^2 / dt) = \mathbf{v}^T (d\mathbf{v} / dt) = -\mathbf{v}^T (\mathbf{Hv}) \leq -\lambda_n |\mathbf{v}|^2$$

$$d|\mathbf{v}|^2 / |\mathbf{v}|^2 \leq -2\lambda_n\, dt$$

$$|\mathbf{v}|^2 \leq |\mathbf{v_0}|^2 \exp(-2\lambda_n t)$$

$$|\mathbf{v}| \leq |\mathbf{v_0}| \exp(-\lambda_n t) \qquad \blacksquare$$



https://www.cs.umd.edu/~tomg/projects/landscapes/

➔ All the valleys of the *loss landscape* of *overparameterized* NNs are connected
(all the local minima are « global »)

Convergence of $|\mathbf{v}(t)|$ (case $\boldsymbol{\lambda_n = 0}$)

Let

- $\lambda_{\mathbf{K}}$ be the **last** > 0 eigenvalue (i.e.: $\lambda_{\mathbf{K}+1} = \dots = \lambda_n = 0$)

- $u_k(t)$ *projection* of $\mathbf{v}(t)$ on k-th eigenvector $\boldsymbol{U}_k(t)$ of eigenvalue $\lambda_k(t)$

<u>Theorem 1</u> (upper bound on $|\mathbf{v}|$) *[Martin-Chamoin-F 2023]*

$$|\mathbf{v}(t)|^2 = \Sigma_{k=1,..,\mathbf{K}} |u_k(t)|^2 + \Sigma_{k=\mathbf{K}+1,..,n} |u_k(t)|^2$$

$$\leq \Sigma_{k=1,..,\mathbf{K}} \boldsymbol{\mu}_k(t) + \Sigma_{k=\mathbf{K}+1,..,n} |u_k(0)|^2 =: \boldsymbol{\mu}(t)$$

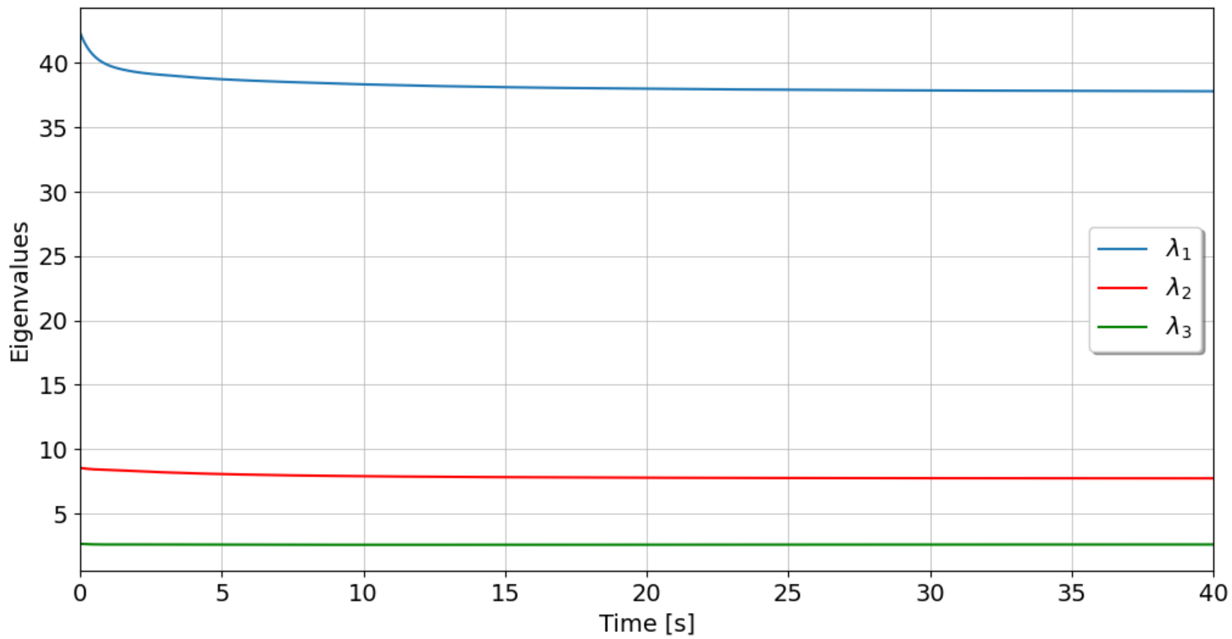
where

$$\boldsymbol{\mu}_k(t) := \boldsymbol{\alpha}_k^{\mathbf{2}} + \boldsymbol{\beta}_k^{\mathbf{2}} \exp(\mathbf{-}\,\boldsymbol{\lambda}_k\, t)$$

with $\boldsymbol{\alpha}_k^{\mathbf{2}} + \boldsymbol{\beta}_k^{\mathbf{2}} = |\mathbf{u}_k(\mathbf{0})|^{\mathbf{2}}$ for $k = 1, \dots, \mathbf{K}$

Besides:

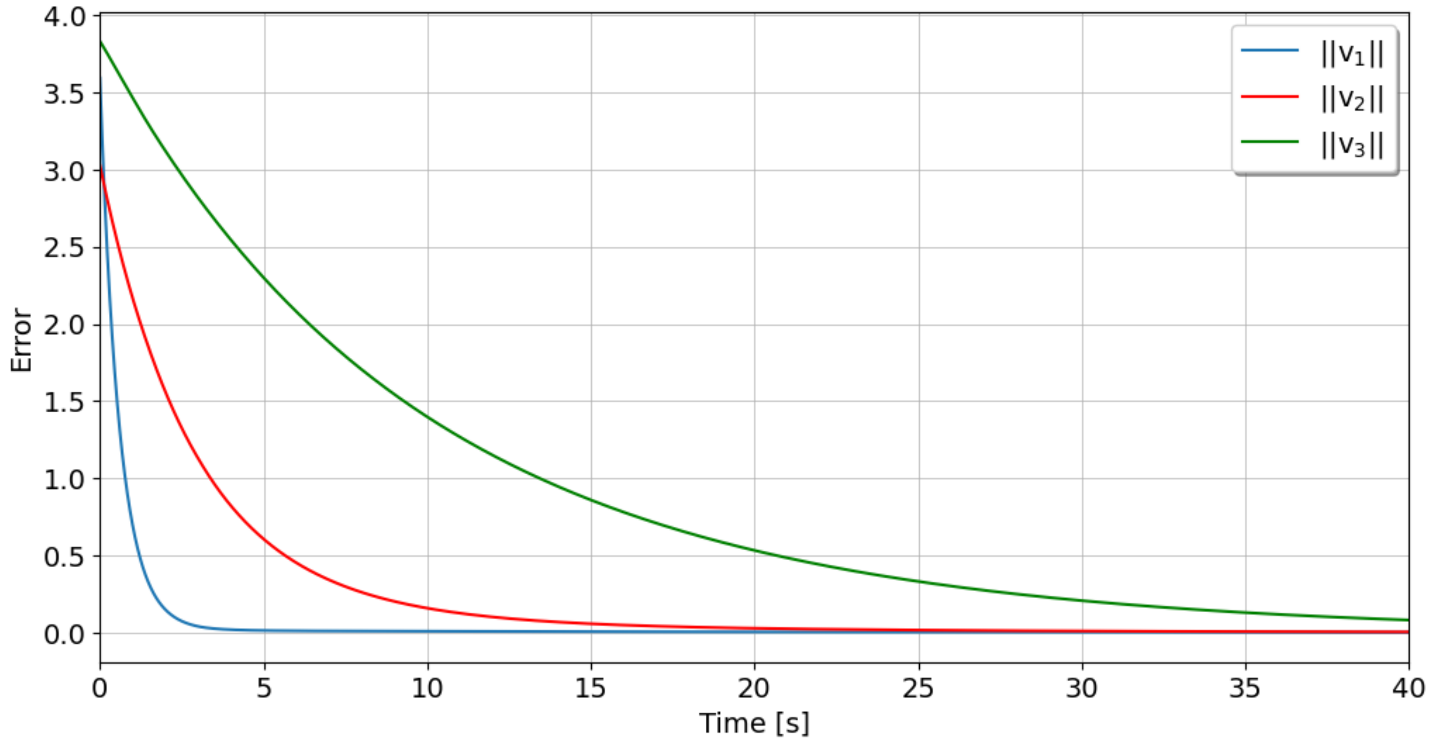
- $|u_k(t)|^2 \leq \boldsymbol{\mu}_k(t)$ for $k = 1, \dots, \mathbf{K}$
- $|u_k(t)|^2 \leq |u_k(0)|^2$ for $k = \mathbf{K}+1, \dots, n$



eigenvalues of **H**:

$\lambda_1 = 38 \geq \lambda_2 = 8 \geq \lambda_3 = 3$

$\lambda_4 = \ldots = \lambda_{50} = 0$



norm of projections

u_1, u_2, u_3

of error $\mathbf{v}(t)$

on $\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3$

Spectral bias:

eigenvectors

corresponding to

large eigenvalues

are learned **quicker**.

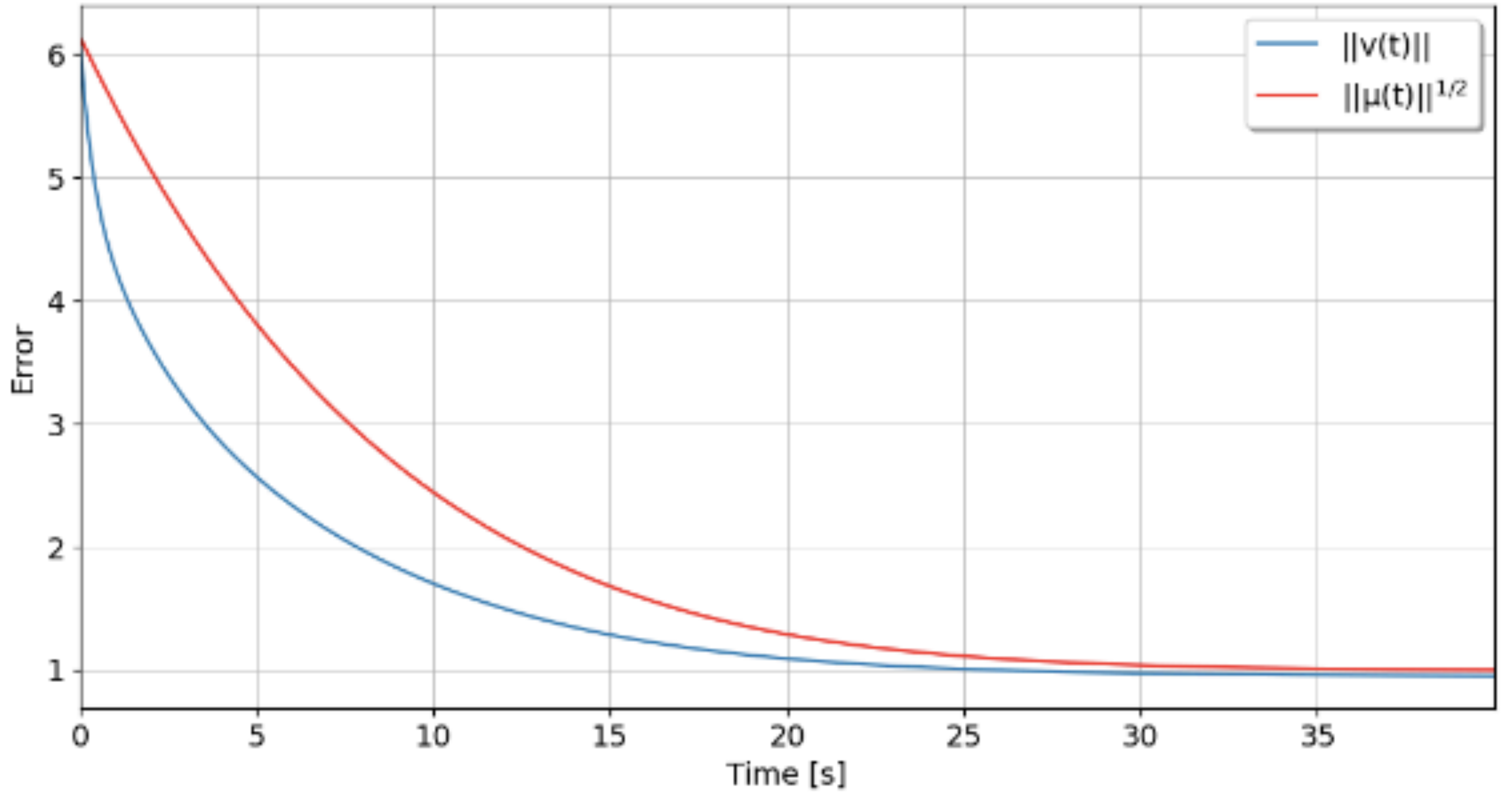
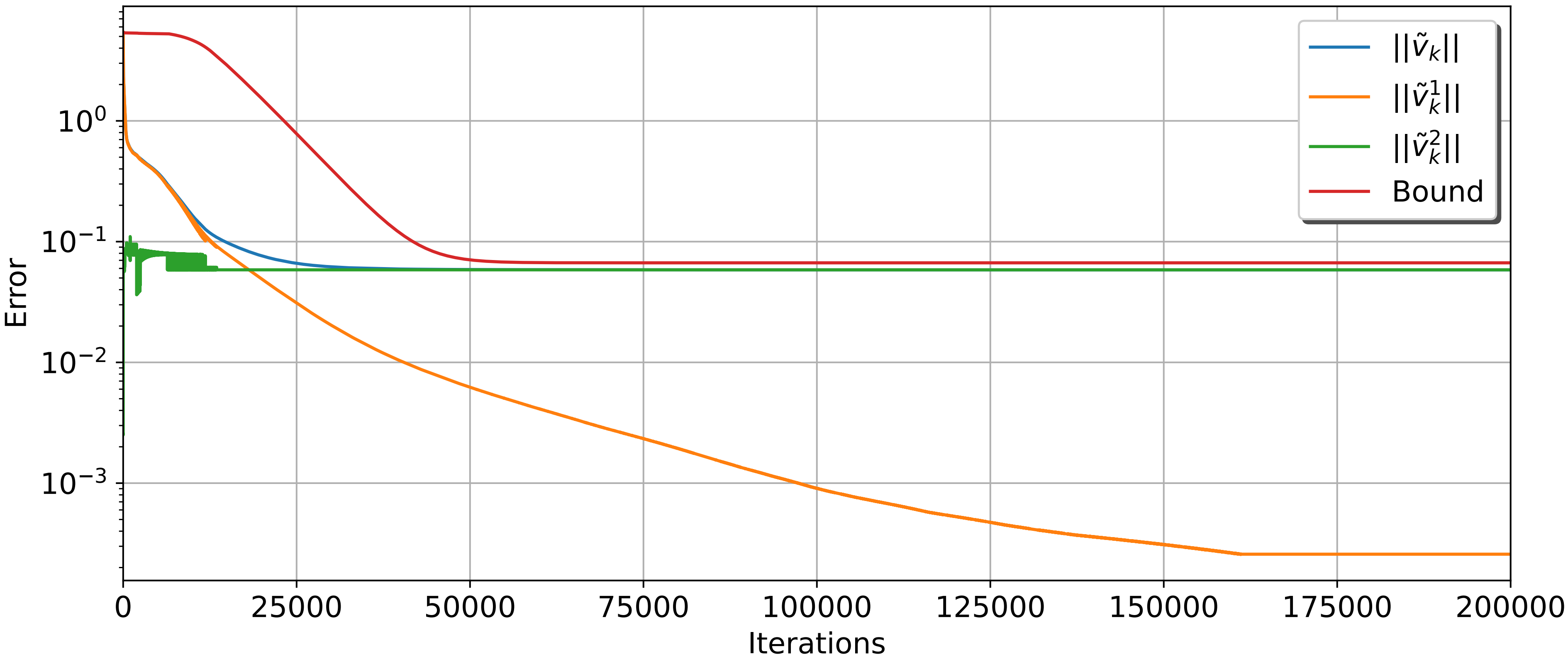


Fig. 1. Evolution of the empirical error vector $\|v(t)\|$ (blue) in comparison to the upper bound $\mu(t)^{1/2}$ (red) derived in Theorem 1.

$|\mathbf{v}(0)| = 6.35, \quad |\mathbf{v}(\infty)| = \Sigma_k \ \alpha_k^2 = 1$



$||\tilde{v}_k||$
$||\tilde{v}_k^1||$
$||\tilde{v}_k^2||$
Bound
Error
10^0
10^{-1}
10^{-2}
10^{-3}
0
25000
50000
75000
100000
125000
150000
175000
200000
Iterations

1

Empirical Loss, Population Loss, Generalization Loss

- *Empirical Loss* $Ł_S$ over sample data set $S = \{(\mathbf{x}_i, y_i)\}_{i=1,..,n}$

$$Ł_S(f_{\mathbf{w}}) := 1/n \ \Sigma_{i=1,..,n} (f_{\mathbf{w}}(\mathbf{x}_i) - y_i)^2 = \mathbf{v}^2/n \leq \mu$$

- *Population Loss* $Ł_D$ over data distribution D

$$Ł_D(f_{\mathbf{w}}) := \mathbf{E}_{(\mathbf{x},y)\sim D} (f_{\mathbf{w}}(\mathbf{x}) - y)^2$$

- *Generalization Loss* $Ł_{gen}$ for S and D

$$Ł_{gen}(f_{\mathbf{w}}) := Ł_D(f_{\mathbf{w}}) - Ł_S(f_{\mathbf{w}})$$

→ *enhancement of standard GD* *(Ockham's razor):*
synthesize the **smallest possible w** (``*regularization*'')

Generalization Loss

Enhancement of Gradient Descent (Ockham's razor):

find strategy of GD which synthesizes **smallest** possible **w** (``*regularization*'')

We know via **Rademacher complexity** theory that for 1-hidden layer:

$$Ł_{gen} \leq 4\,(\sqrt{m}/\sqrt{n}) \times max_{r=1,..,m}|\mathbf{w}_r|$$

$$Ł_D \leq Ł_{gen} + Ł_S$$ [with high probability]

Hence (using **Theorem** 1)*:*

$$Ł_D \leq 4\,(\sqrt{m}/\sqrt{n}) \times max_{r=1,..,m}|\mathbf{w}_r| + \mu/n$$

Let us now find an *upper bound* on $|\mathbf{w}_r|$.

$$\|\frac{d}{dt}w^r(t)\| \leqslant \frac{\sqrt{n}}{\sqrt{m}}\|v(t)\| \leqslant \frac{\sqrt{n}}{\sqrt{m}}\sqrt{\sum_{k=1}^{n}\|u_k(t)\|^2}$$

$$\|\frac{d}{dt}w^r(t)\| \leqslant \frac{\sqrt{n}}{\sqrt{m}}\sqrt{\sum_{k=1}^{K}\mu_k(t) + \sum_{k=K+1}^{n}\|u_k(0)\|}.$$

Hence (assuming $\|w^r(0)\| \approx 0$)

$$\|w^r(t)\| \leqslant \frac{\sqrt{n}}{\sqrt{m}}\int_0^t \sqrt{\sum_{i=1}^{K}\mu_k(s) + \sum_{k=K+1}^{n}\|u_k(0)\|}ds$$

$$= \frac{\sqrt{n}}{\sqrt{m}}\int_0^t \sqrt{\sum_{k=1}^{K}\alpha_k^2 + \beta_k^2 e^{-\lambda^* s} + \sum_{k=K+1}^{n}\|u_k(0)\|}ds$$

$$= \frac{\sqrt{n}}{\sqrt{m}}\int_0^t \sqrt{A + Be^{-\lambda^* s}}ds$$

with $A = \sum_{k=1}^{K}\alpha_k^2 + \sum_{k=K+1}^{n}\|u_k(0)\|^2$ and $B = \sum_{k=1}^{K}\beta_k^2$. By integration we have (15), i.e.:

$$\|w_r(t)\| \leqslant \frac{\sqrt{n}}{\sqrt{m}}\Phi(t)$$

with

$$\Phi(t) = \frac{2}{\lambda^*}(\sqrt{A}\sinh^{-1}(\sqrt{\frac{A}{B}}e^{+\frac{1}{2}\lambda^* t}) - \sqrt{A + Be^{-\lambda^* t}}) + c$$

Theorem 2 (upper bound on $|\mathbf{w}_r(t)|$): $\|w_r(t)\| \leqslant \frac{\sqrt{n}}{\sqrt{m}}\Phi(t)$ with

[Martin-Chamoin-F 2023]

$$\Phi(t) = \frac{2}{\lambda^*}\left(\sqrt{A}\sinh^{-1}\left(\sqrt{\frac{A}{B}}e^{+\frac{1}{2}\lambda^* t}\right) - \sqrt{A + Be^{-\lambda^* t}}\right) + c$$

$$A = \sum_{k=1}^{K}\alpha_k^2 + \sum_{k=K+1}^{n}\|u_k(0)\|^2, \quad B = \sum_{k=1}^{K}\beta_k^2,$$

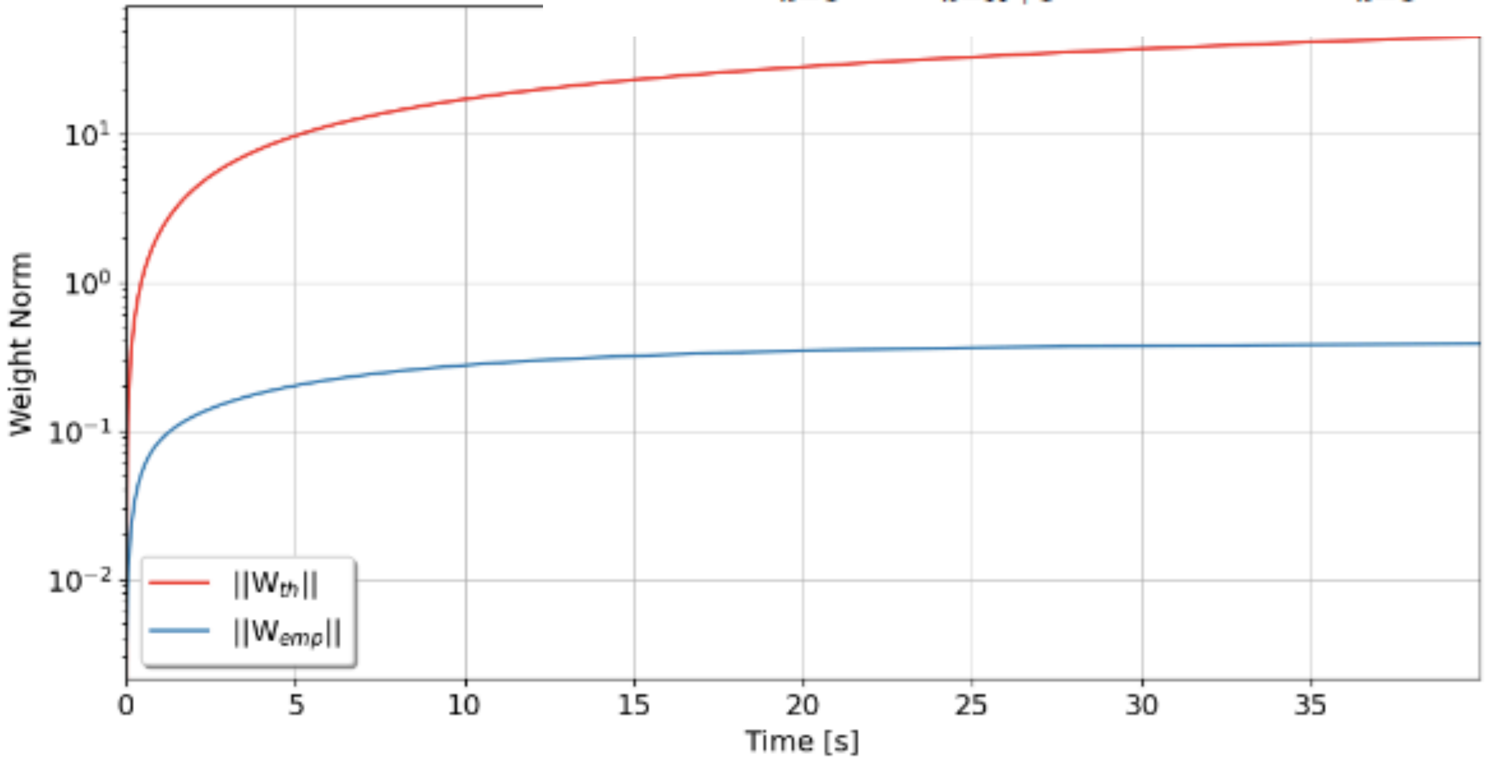
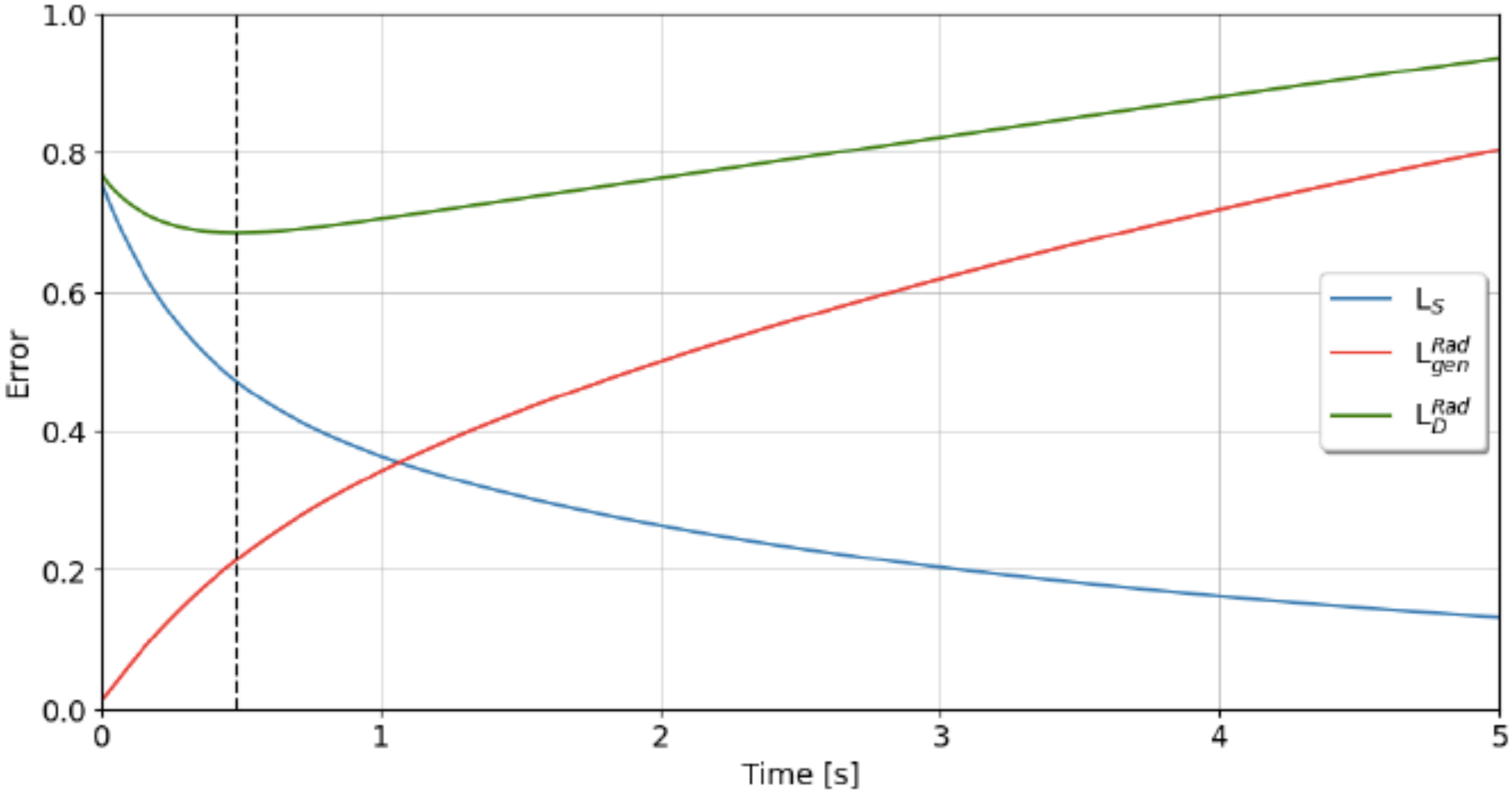


Fig. 2. Evolution of the empirical weight norm $\|W_{emp}\|$ (blue) in comparison to the upper bound $\|W_{th}\|$ (red).

Early Stopping: stop GD at t = **t*** when $Ł_D = Ł_S + Ł_{gen}$ **minimal**

t* $= 1/\lambda_K \ln[\, B / (A + (4n/\lambda_K)^2)\,]$ with $A = \sum_{k=1}^{K} \alpha_k^2 + \sum_{k=K+1}^{n} \|u_k(0)\|^2, \quad B = \sum_{k=1}^{K} \beta_k^2,$



Recapitulation

- GD *minimizes* **training error** using training set *S*
- Useful to *minimize* also **w**(t) to *reduce* **generalization error** (tradeoff **bias-variance**)
- → **early stopping** strategy

Contribution

1. Via theory of **NTK**, computation of analytic **upperbound** on the **training error**
2. Via **Rademacher's complexity**, analytic **upperbound** on the **generalization error**
3. → analytic estimation of **optimal** time for **stopping** GD.

Moral of the story: SYNthesize the ***L***east COmplex ***P***ossible Parameters

SYNCOP → SYN***L***CO***P***P

References

1 P. L. Bartlett, S. Mendelson.
Rademacher and Gaussian complexities: Risk bounds and structural results.
J. Mach. Learn. Res. 3, 2002.

2 A. Jacot, C. Hongler, F. Gabriel.
Neural Tangent Kernel: Convergence and generalization in neural networks.
NeurIPS 2018.

3 S. Du, X. Zhai, B. Póczos, A. Singh.
Gradient descent provably optimizes over-parameterized neural networks.
ICLR 2019.

4 J. Jerray, A. Saoud, L. Fribourg.
Using Euler's method to prove the convergence of neural networks.
IEEE Control. Syst. Lett. 6, 2022.

5 D. M. Xavier, L. Chamoin, L. Fribourg.
Training and generalization errors for underparameterized neural networks.
IEEE Control. Syst. Lett. 7, 2023.

THANKS !