Neural Networks training inspired by convex optimization over measures

Karl Hajjar

PhD student, LMO, équipe Probas & Stat
Supervised by Lénaïc Chizat (EPFL) & Christophe Giraud (LMO)

Outline

Setting and General Background

We consider objectives of the form

$$\min_{\mu \in \mathcal{M}(\mathbb{S}^{d-1})} F(\mu)$$

with:

- $\mathcal{M}(\mathbb{S}^{d-1})$ space of signed measures over the unit sphere
- $F = J + \lambda H$ convex
- J convex and smooth, H convex and potentially non-smooth, $\lambda \geq 0$

Goals

- Problem 1: how to obtain global minimum of F with explicit CV rate ?
- Problem 2: how to obtain algos that are practical computationally ?
- Wasserstein GD enjoys good practical behaviour but not always global CV guarantees (Wojtowytsch, 2020; Chizat, 2022) (no explicit rate, local convergence, ...)
- We propose algos inspired from finite-dimensional convex optimization to obtain global convergence guarantees

Plan of the talk

- For smooth F (*i.e.*, $\lambda = 0$), coordinate descent (in $L^2(\mathbb{S}^{d-1})$) has explicit convergence guarantee in $O(k^{-1/d})$
- Problem: memory & compute grow linearly with iteration $k \implies$ impractical...
- Idea: penalize objective to encourage sparsity. *e.g.*:
  (i) total variation penalty akin to L^1 penalty (non-smooth)
  (ii) attractive / repulsive kernel penalties
- (i) leads to proximal algorithms in space of measures
  (ii) described by a PDE called Wasserstein-Fisher-Rao GF

Link with neural networks

Link with neural networks

- Consider the objective over two-layer neural networks

$$\min_{\mathbf{a},\mathbf{b}\in\mathbb{R}^m\times\mathbb{R}^{m\times d}} \Big\{ J_m(\mathbf{a},\mathbf{b}) := \mathbb{E}_{x\sim\rho}\left[\ell(f^*(x), f_m(\mathbf{a},\mathbf{b};x))\right] \Big\},$$

$$f_m(\mathbf{a},\mathbf{b};x) = \sum_{j=1}^{m} a_j \sigma(b_j^\top x)$$

- If σ positively homogeneous (*e.g.*, ReLU),
  $f_m(\mathbf{a},\mathbf{b};x) = \int \sigma(u^\top x)\mathrm{d}\mu_m =: f(\mu_m;x)$,
  $\mu_m := \sum_{j=1}^m \frac{a_j}{||b_j||}\delta_{b_j/||b_j||} \in \mathcal{M}(\mathbb{S}^{d-1})$

- Then, $J_m(\mathbf{a},\mathbf{b}) = J(\mu_m)$, $J(\mu) := \mathbb{E}_{x\sim\rho}\left[\ell(f^*(x), f(\mu;x))\right]$.

- Usual training of NNs (GD) can be describe by the Wasserstein GF over $\nu_t \in \mathcal{P}_2(\mathbb{R}^{d+1})$

$$\partial_t \nu_t = -\mathrm{div}(-\nabla V[\nu_t]\nu_t)$$

- Define $\mu_t^{\pm} \in \mathcal{M}_+(\mathbb{S}^{d-1})$ through, for any test function φ

$$\int \varphi \mathrm{d}\mu_t^{\pm} = \int_{\pm a \geq 0, b} |a|||b||\varphi\left(\frac{b}{||b||}\right) \mathrm{d}\nu_t(a,b)$$

- Then through homogeneity, Wasserstein GF $\implies$ Wasserstein-Fisher-Rao GF (Chizat, 2022; Hajjar and Chizat, 2023 for more details):

$$\partial_t \mu_t^{\pm} = -\mathrm{div}(-\mathrm{proj}_{\tan(\mathbb{S}^{d-1})}(\nabla V[\nu_t])) \pm 2V[\nu_t]\mu_t^{\pm}$$

Link with neural networks, use of homogeneity

- Actually needs homogeneity + ν_0 initialized on the cone $\{|a| = ||b||\} \implies$ "conic" GD

- It holds $\int a\sigma(b^\top x)\mathrm{d}\nu_t(a,b) = \int \sigma(u^\top x)\mathrm{d}\mu_t(u)$, $\mu_t = \mu_t^+ - \mu_t^- \in \mathcal{M}(\mathbb{S}^{d-1})$

- **Remark**: usual NNs trained with (Wasserstein) GD with fixed # neurons. In this talk it evolves dynamically during optimization

Brief overview of (some) convex optimization methods

Coordinate descent methods

- Consider convex $f : \mathbb{R}^m \to \mathbb{R}$ and smooth, *i.e.*, $||\nabla f(x) - \nabla f(y)|| \leq L||x - y||$
- GD converges to global minimum in $O(1/k)$ but requires m operations at each iteration
- Random coordinate descent: at each iteration k, select coordinate $i_k \sim \mathcal{U}(\{1, \ldots, m\})$ and $x_{k+1} = x_k - \eta \nabla_{i_k} f(x_k)$
- CV to a global minimum in expectation in $O(m/k)$ but requires $O(1)$ operations at each iteration $\implies$ m iterations to compute a full gradient, slower CV but cheaper iteration

- Many proof techniques but essentially reduces to proving a condition akin to a Łojasiewicz condition:
  $\frac{1}{2}\mathbb{E}[f(x_k) - M^*]^2 \leq \tau\mathbb{E}[||\nabla f(x_k)||]^2$

- Łojasiewicz condition (PL = {L & $\gamma = 1$}):
  $\frac{1}{2}||\nabla f(x)||^2 \geq \tau(f(x) - M^*)^\gamma, \quad \tau, \gamma > 0$

- $\implies$ CV of coordinate descent to global min in
  $O\left(\left(\frac{\tau(\gamma-1)}{mL}k\right)^{-\frac{1}{\gamma-1}}\right)$ if $\gamma > 1$

- Plain coordinate descent (without Łojasiewicz assumption) essentially same (in expectation) as Łojasiewicz with $\gamma = 2$

- What if $g = f + h$, f convex smooth, h convex non-smooth but separable $h(x) = \sum_{i=1}^{m} h_i(x_i)$ and "easy" to optimize (*e.g.*, $h(x) = ||x||_1$)?

- Proximal methods start with upper bound:

$$g(y) - g(x) \leq \langle \nabla f(x), y - x \rangle + \frac{L}{2}||y - x||^2 + h(y) - h(x) \tag{1}$$

- Plugging $y = x_k + te_{i_k}$ and minimizing the RHS over $t \in \mathbb{R}$ yields the proximal step $x_{k+1} = x_k + t_k e_{i_k}$ providing a descent step: $g(x_{k+1}) \leq g(x_k)$

Convergence rates of proximal methods

Convergence of the proximal coordinate descent method:

- Łojasiewicz condition $\implies$ CV to global min in $O\left(\left(\frac{mL}{\tau k}\right)\right)^{1/(\gamma-1)}\Big)$
- In general, if we assume only $||x_k||$ bounded, CV in $O(mL/(\tau k))$ (same as Łojasiewicz with $\gamma = 2$).
- Boundedness assumption on $||x_k||$ holds as soon as h is an increasing function of some norm
- **Q:** can we adapt / generalize those methods to the infinite-dim space of measures for NN training ?

Random coordinate descent in the space of signed measures

- Objective $F : \mathcal{M}(\mathbb{S}^{d-1}) \to \mathbb{R}$ convex and smooth, *i.e.*, admits *continuous* first variation $V[\mu] : \mathbb{S}^{d-1} \to \mathbb{R}$, and $||V[\nu] - V[\mu]||_\infty \leq L|\nu - \mu|_{TV}$
- The first variation is the "derivative" of the functional F: $\frac{d}{dt} F(\mu + t\nu)\Big|_{t=0} = \int V[\mu] \mathrm{d}\nu$
- Analogous to $\frac{d}{dt} f(x + ty)\Big|_{t=0} = \langle \nabla f(x), y \rangle$ in finite-dim

Upper bound from smoothness

- Similarly to the upper bound $f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}||y - x||^2$ in finite-dim:

$$F(\nu) - F(\mu) \leq \int V[\mu] \mathrm{d}(\nu - \mu) + \frac{L}{2}|\nu - \mu|_{TV}^2 \qquad (2)$$

- Pb: compared to finite-dim, minimum over $\nu \in \mathcal{M}(\mathbb{S}^{d-1})$ of the RHS is not always tractable (Frank-Wolfe step)...

- However, if we restrict to $L^2(\omega)$, the RHS is upper bounded (Jensen) by $\int V[\mu](f_\nu - f_\mu)\mathrm{d}\omega + \frac{L}{2}||f_\nu - f_\mu||_{L^2(\omega)}^2$

- Minimum over $f_\nu \in L^2(\omega)$ is obtained (pointwise) for $f_\nu^* = f_\mu - \frac{1}{L}V[\mu]$ and the min is $-\frac{1}{2L}||V[\mu]||_{L^2(\omega)}^2$

Coordinate descent intuition

Back to our minimization problem

- In $\mathcal{M}(\mathbb{S}^{d-1})$, no clear notion of basis, coordinate or projection as in finite-dim
- However, intuitively δ_u is a good candidate for a "basis" vector and the mass μ "puts" at $u \in \mathbb{S}^{d-1}$ good candidate for coordinate along δ_u
- We plug $\nu = \mu + t\delta_u$ into the previous upper bound (2) and minimize over $t \in \mathbb{R}$
- Min is obtained for $t = -\frac{1}{L}V[\mu](u)$ and equal to $-\frac{1}{2L}V[\mu](u)^2$.

Coordinate descent algorithm

- Starting from $\mu_0 = c_0 \delta_{u_0}$, for each iteration $k \geq 1$, do:
  1. Sample $u_k \sim \omega_d := \mathcal{U}(\mathbb{S}^{d-1})$
  2. Set $c_k := -\frac{1}{L} V[\mu_{k-1}](u_k)$ and $\mu_k = \mu_{k-1} + c_k \delta_{u_k}$

- Since $c_k = \operatorname{argmin}_{t \in \mathbb{R}} tV[\mu_{k-1}](u_k) + \frac{L}{2} t^2$, this is reminiscent of finite-dim coordinate descent, except we never circle back to the same coordinate twice...
- In expectation, this yields $\mathbb{E}[F(\mu_k)|\mu_{k-1}] \leq F(\mu_{k-1}) - \frac{1}{2L} ||V[\mu_{k-1}]||^2_{L^2(\omega)}$

Link with L^2 geometry

Coordinate descent and L^2 geometry

- Given μ_{k-1}, it holds for any $f \in L^2(\omega_d)$

$$F(\mu_{k-1} + f\omega_d) - F(\mu_{k-1}) \leq \int V[\mu_{k-1}] f \mathrm{d}\omega_d + \frac{L}{2}||f||^2_{L^2(\omega_d)}$$

- Minimizing the upper bound on the RHS yields
$f^* = -\frac{1}{L}V[\mu_{k-1}]$ and
$F(\mu_{k-1} + f^*\omega_d) - F(\mu_{k-1}) \leq -\frac{1}{2L}||V[\mu_{k-1}]||^2_{L^2(\omega_d)}$

- Thus, a step of coordinate descent is (in expectation) equivalent to a minimization in $L^2(\omega_d)$ geometry

- V is the gradient *w.r.t* $L^2(\omega_d)$ geometry

Brief outline for the convergence proof

- Assumptions:
  1. $(|\mu_k|_{TV})_{k\geq 0}$ is bounded
  2. There is $K > 0$ such that for any $u, v \in \mathbb{S}^{d-1}$ and $\mu \in \mathcal{M}(\mathbb{S}^{d-1})$, $|V[\mu](v) - V[\mu](u)| \leq K||v - u||$

- Łojasiewicz inequality:

Lemma (Chizat, Hajjar & Giraud (2023))

There is a constant $\tau > 0$ such that:

$$\frac{1}{2}||V[\mu_k]||^2_{L^2(\omega_d)} \geq \tau(F(\mu_k) - F^*)^{d+1}.$$

Theorem (Chizat, Hajjar & Giraud (2023))

Let μ_k be the iterates generated by the coordinate descent algorithm in $\mathcal{M}(\mathbb{S}^{d-1})$. Then, under the previous assumptions, there is a constant $C > 0$ such that, for any $k \geq 1$:

$$0 \leq \mathbb{E}[F(\mu_k) - F^*] \leq \frac{C}{k^{1/d}}.$$

- Issue: the number of atoms of μ_k grows linearly with k $\implies$ computationally impractical because new particle added at each iteration
- Idea: add sparsity-inducing penalties such as total variation norm !
- Mix global CV steps with Wasserstein GD "conic" steps which have good local convergence properties

Proximal algorithms for the total variation penalty

Total variation penalty

- TV norm for measures is analogous to L^1 penalty in finite-dim: $|\sum_j c_j \delta_{u_j}|_{TV} = \sum_j |c_j|$
- As in finite-dim it encourages sparsity (see next slide)
- We consider the objective functional $F(\mu) = J(\mu) + \lambda |\mu|_{TV}$, J convex smooth, $|-|_{TV}$ is convex but not smooth
- Diff with finite-dim: $|-|_{TV}$ is not separable, cannot write $|\mu|_{TV} = \int f_\mu \mathrm{d}\omega_d$ in general

Proximal bound and sparsity

As in finite-dim, proximal upper bound:

$$F(\nu) - F(\mu) \leq \int V[\mu] \mathrm{d}(\nu - \mu) + \frac{L}{2}|\nu - \mu|_{TV}^2 + \lambda|\nu|_{TV} - \lambda|\mu|_{TV} \tag{3}$$

- Plug $\nu = \mu + t\delta_u$ for $u \in \mathbb{S}^{d-1}$ and minimize the RHS *w.r.t* $t \in \mathbb{R}$
- If u not in $\mathrm{supp}(\mu)$, we get (similarly to finite-dim), $t^* = -\frac{V[\mu](u)}{L} \max\left(0, 1 - \frac{\lambda}{|V[\mu](u)|}\right)$
- Large $\lambda \implies$ sparsity (at least support does not grow)

Basic variant of the proximal algorithm: starting from $\mu_0 = c_0 \delta_{u_0}$, for each iteration $k \geq 1$, do

1. Sample $u_k \sim \omega_d = \mathcal{U}(\mathbb{S}^{d-1})$

2. Set $c_k = -\frac{V[\mu_{k-1}](u_k)}{L} \max\left(0, 1 - \frac{\lambda}{|V[\mu](u_k)|}\right)$

3. If $c_k = 0$, $\mu_k = \mu_{k-1}$, else $\mu_k = \mu_{k-1} + c_k \delta_{u_k}$

Discussion

Issues:

1. No real sparsity because support size can still grow as αk, $\alpha \in (0, 1)$...

2. Global convergence guarantee is lost (no smoothness $\Longrightarrow$ previous proof does not work, and same technique as finite-dim does not apply because of TV)

3. Worse, trade-off between sparsity and descent:
   $F(\mu_{k+1}) - F(\mu_k) \leq -\frac{1}{2L} \max\left(0, |V[\mu_k](u_{k+1})| - \lambda\right)^2$,
   objective decrease **only** if new atom added...

A modified proximal algorithm

Idea:

- Proximal update with TV penalty $\implies$ we don't always add a new atom
- If we sample from existing atoms maybe we can "kill" some atoms (same effect as L^1 penalty in finite-dim) ?
- Thus sample new atom half the time and existing atom half the time
- Need to recompute the upper bound when sampling existing atom

A modified proximal algorithm

- Let $\mu_k = \sum_{j=0}^{k} c_j \delta_{u_j}$, $j \in \{0, \dots, k\}$. RHS of (3) with $\nu = \mu_k + t\delta_{u_j}$ is $tV[\mu_k](u_j) + \frac{L}{2}t^2 + \lambda|c_j + t| - \lambda|c_j|$

- Min *w.r.t* $t \in \mathbb{R}$ same as prox step with L^1 penalty in finite dim: Iterative Soft Thresholding

$$t^* = -c_j + \left(c_j - \frac{V[\mu_k](u_j)}{L}\right) \max\left(0, 1 - \frac{\lambda}{|V[\mu_k](u_j) - Lc_j|}\right)$$

- Total weight on u_j after update is $c_j + t^*$: can be 0 for large $\lambda \implies$ decrease of the number of atoms !

Modified proximal algorithm: starting from $\mu_0 = c_0\delta_{u_0}$, for each iteration $k \geq 1$, do

1. **if** k odd: sample $u_k \sim \omega_d = \mathcal{U}(\mathbb{S}^{d-1})$,
   else: sample $u_k \sim \mathcal{U}(\{u_0, \ldots, u_{k-1}\})$

2. Set c_k accordingly depending on parity of k

3. **if** total mass on u_k after update is 0: $\mu_k = \mu_{k-1}$,
   else: $\mu_k = \mu_{k-1} + c_k\delta_{u_k}$

Discussion

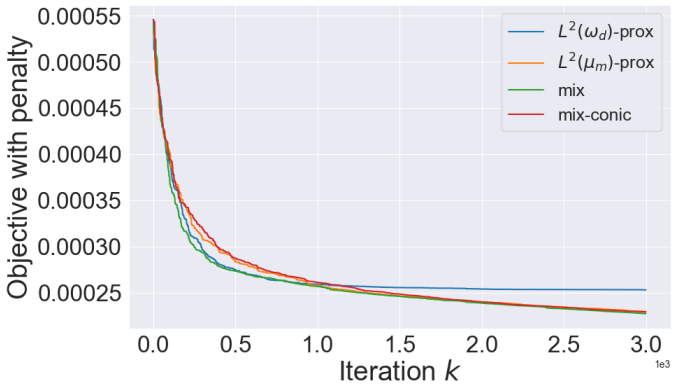
- The proximal algo is a true descent algo: $F(\mu_{k+1}) \leq F(\mu_k)$
- Odd steps are strict descent only if new neuron is added
- Even steps are strict descent (in expectation) as soon as μ_{2k+1} is not optimal among measures with the same support
- Unfortunately no global convergence guarantee and no explicit control over number of atoms
- **But**, good empirical behaviour ! (see next slides)

Numerical experiments for proximal coordinate descent with TV penalty

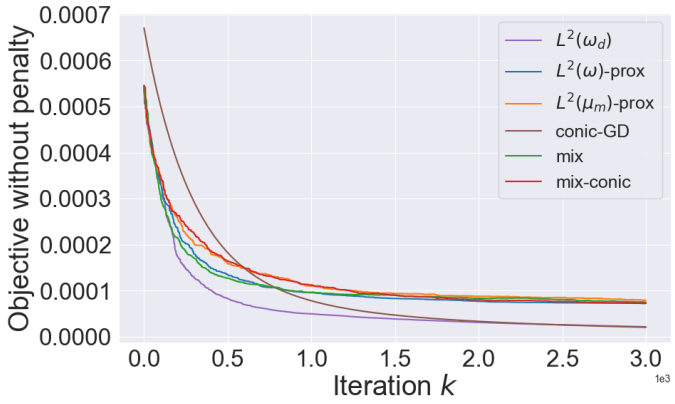
- Mix algos with "conic" Wasserstein GD steps (usual GD for NNs) which often have good behaviour in practice

- Algos: pure coord descent, prox-TV, prox-TV fixed support, modified prox-TV, modified prox-TV + conic, pure conic GD (no penalty)

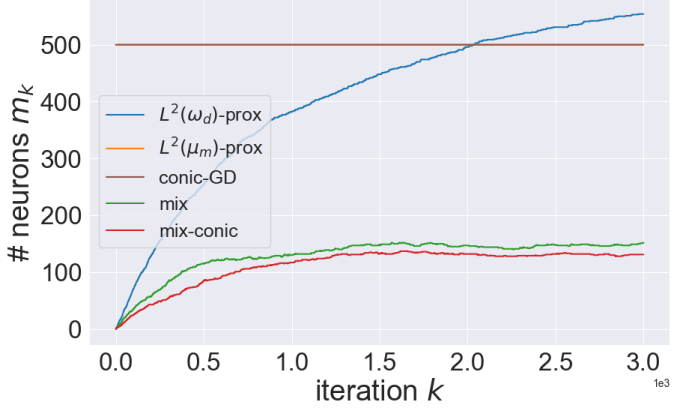
- Setting $d = 10$, $m = 500$ atoms for pure conic and fixed support, $3,000$ iterations



(a) Penalized objective



(b) Original objective



(c) # atoms

Figure 1: Empirical performance of different algorithms

Discussion on numerical simulation

- Modified prox-TV and conic variant have low objective and penalized objective values
- # atoms seem to be bounded compared to pure coord descent and basic prox-TV
- pure coord descent and pure conic GD have distinctly lower objective value **but** former computation cost grows linearly and latter has no CV rate
- modified prox-TV and conic variant seem to strike a balance between theoretical soundness and computational cost (dynamical adaptation of # atoms)

Kernel penalties

- Objective functional

$$F(\mu) = J(\mu) + \lambda \int_{u,v} K(u,v)\mathrm{d}|\mu|(u)\mathrm{d}|\mu|(v)$$

- J convex smooth, $K(u,v) = \kappa(\langle u, v\rangle)$, $\kappa : \mathbb{R} \to \mathbb{R}_+$, $\lambda > 0$

- *Attractive* kernel (*resp.* repulsive) if κ decreasing (*resp.* increasing)

- *e.g.*, $\kappa_{a,\sigma}(s) = 1 - e^{(s-1)/\sigma^2}$ or $\kappa_{r,\sigma}(s) = e^{(s-1)/\sigma^2}$

Intuition

- Idea: by pushing particles closer together or far apart, some particles will aggregate
- Then, merging particles which are at distance less than ϵ $\implies$ sparsity
- Theoretically motivated approach but no guarantees unfortunately...

- Evolution equations come from Wasserstein GF lifted on the sphere (called Wasserstein-Fisher-Rao GF)
- Starting from $\mu_0 \in \mathcal{M}(\mathbb{S}^{d-1})$, PDEs (distributionally)

$$\partial_t \mu_t^{\pm} = -\mathrm{div}(\pm \tilde{v}_t^{\pm} \mu_t^{\pm}) \pm 2 g_t^{\pm} \mu_t^{\pm}$$

- $\mu_t^+, \mu_t^- \in \mathcal{M}_+(\mathbb{S}^{d-1})$ positive/negative part of $\mu_t \in \mathcal{M}(\mathbb{S}^{d-1})$
- Advection / reaction terms (V is first variation of J):

$$g_t^{\pm}(u) = -\left(\pm V[\mu_t](u) + \lambda \int K(u,v) \mathrm{d}|\mu_t|(v) \right),$$
$$\tilde{v}_t^{\pm}(u) = \mathrm{proj}_{\{u\}^{\perp}} \left(\nabla g_t^{\pm}(u) \right).$$

Inconclusive numerical results

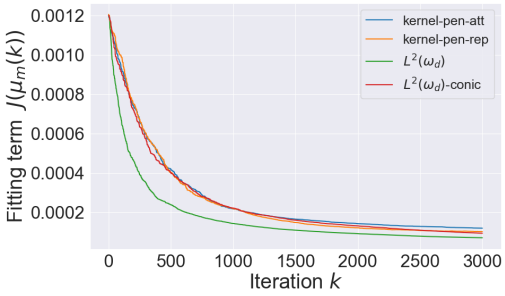
- PDEs can be discretized in time an provide iterates $(\mu_k)_{k\geq 0}$
- Still work in progress but numerical experiments are inconclusive at this stage
- Not obvious empirically that the dynamics induce sparsity
- Dynamics decrease the objective but less than pure coord descent or conic GD

Open questions / future work

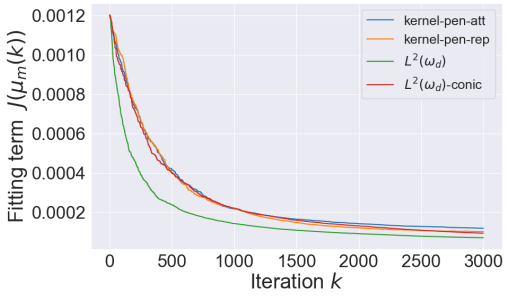
- Proof of the boundedness of $|\mu_k|_{TV}$ for the CV of pure coord descent
- Proof of CV for modified prox-TV in specific settings ?
- Control of the number of neurons for prox-TV and kernel penalties ?
- Empirical setting where kernel penalties are effective in terms of sparsity ?

Thank you!

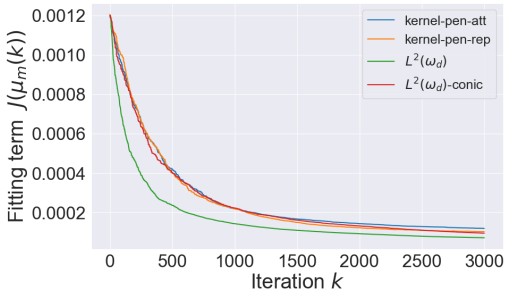
Kernel penalties performance



(a) $\epsilon = 0.1$

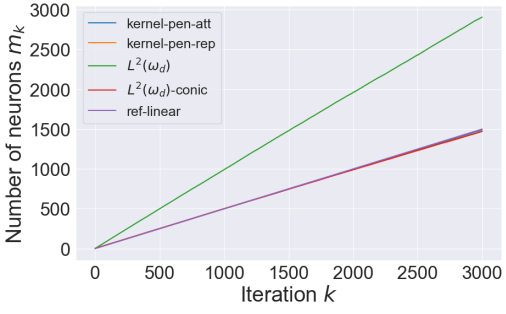


(b) $\epsilon = 0.15$

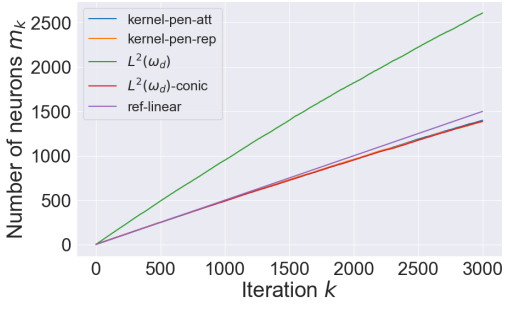


(c) $\epsilon = 0.25$

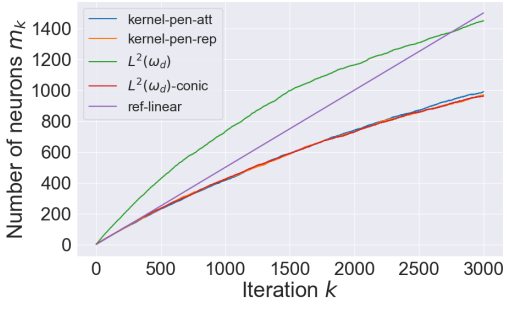
Figure 2: Initial objective $J(\mu_k)$ *vs.* k.



(a) $\epsilon = 0.1$



(b) $\epsilon = 0.15$



(c) $\epsilon = 0.25$

Figure 3: Number of neurons m_k *vs.* k.

References

Lenaic Chizat. Sparse optimization on measures with over-parameterized gradient descent. *Mathematical Programming*, 194(1):487–532, 2022.

Karl Hajjar and Lenaic Chizat. On the symmetries in the dynamics of wide two-layer neural networks, 2023.

Stephan Wojtowytsch. On the convergence of gradient descent training for two-layer relu-networks in the mean field regime. *arXiv preprint arXiv:2005.13530*, 2020.