Kernelized Wasserstein Natural Gradient

Michael Arbel [1] Arthur Gretton [1] Wuchen Li [2] Guido Montufar [2,3]

[1]Gatsby Computational Neuroscience Unit, UCL, London

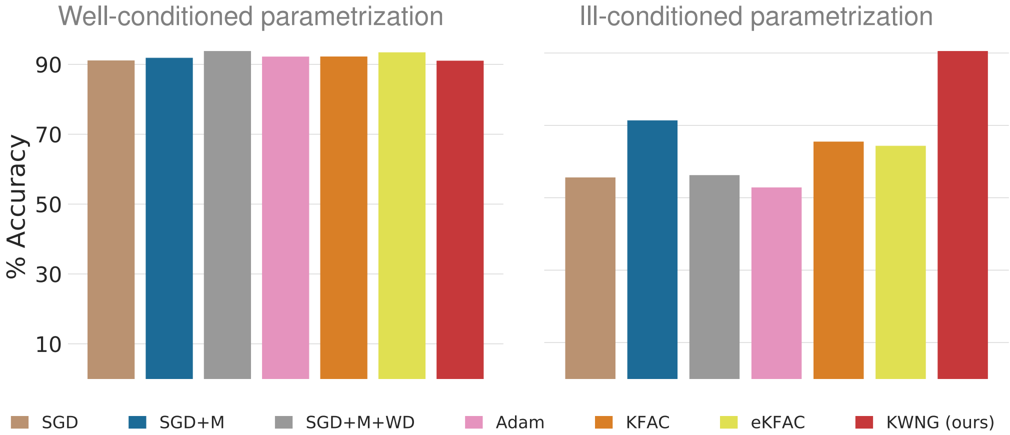
[2]University of California, Los Angeles

[3]Max Planck Institute for Mathematics in the Sciences, Leipzig

April 8, 2020

KWNG: A natural gradient optimizer with built in Optimal Transport Geometry.

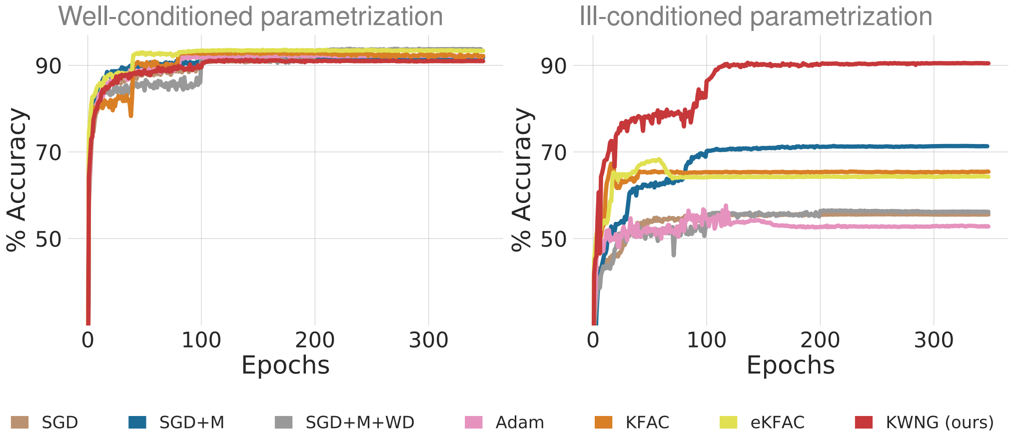
✓ Approximately Invariant to re-parametrization



Cifar10 classification task using ResNet-18 networks.

KWNG: A natural gradient optimizer with built in Optimal Transport Geometry.

✓ Approximately Invariant to re-parametrization

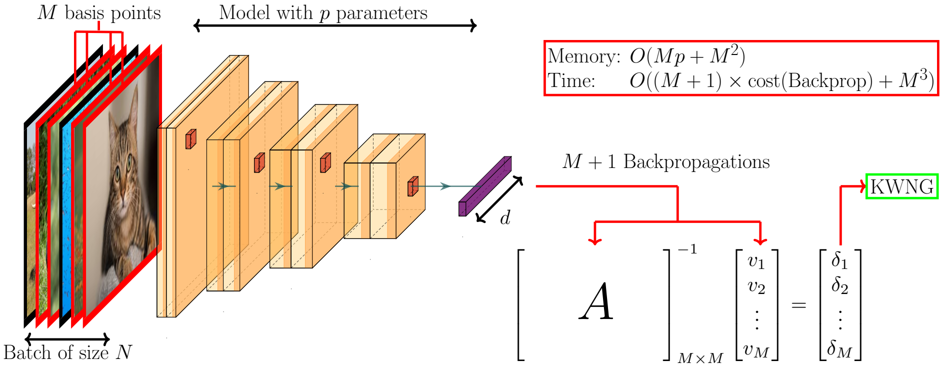


Cifar10 classification task using ResNet-18 networks.

KWNG: A natural gradient optimizer with built in Optimal Transport Geometry.

✓ Approximately invariant to re-parametrization

✓ Fast and scalable



KWNG: A natural gradient optimizer with built in Optimal Transport Geometry.

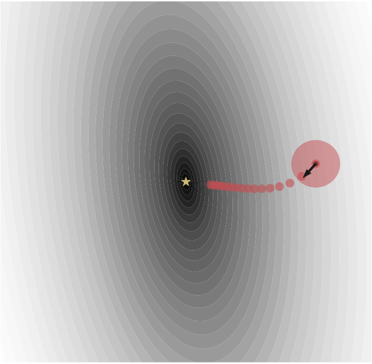
- ✓ Approximately invariant to re-parametrization
- ✓ Fast and scalable
- ✓ Can be used as a drop-in optimizer

```
from kwng import KWNG,  KWNGWrapper
from gaussian import Gaussian
kernel = Gaussian()
KWNGEstimator = KWNG (kernel,
                      num_basis= 10,
                      eps= 1e-4 )
w_optimizer = KWNGWrapper(optimizer,
                    criterion,
                    net,
                    KWNGEstimator)
loss,pred = w_optimizer.step(inputs, targets)
```

Motivation

- Learning problem: $\theta^* = \arg\min_\theta \mathcal{L}(p_\theta)$
- Update equation: $\theta_{k+1} = \theta_k + \lambda \mathcal{D}_k$

$$\mathcal{D}_k = \arg\min_u \boxed{\nabla_\theta \mathcal{L}(p_{\theta_k})^\top u} + \boxed{\frac{1}{2}\|u\|^2}$$

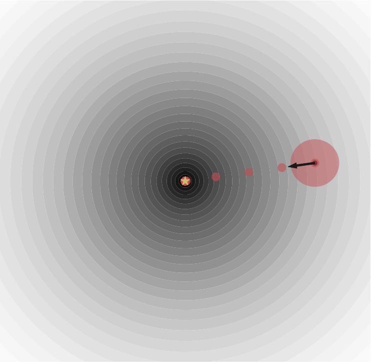


Motivation

- Learning problem: $\theta^* = \arg\min_\theta \mathcal{L}(p_\theta)$
- Update equation: $\theta_{k+1} = \theta_k + \lambda \mathcal{D}_k$

$$\mathcal{D}_k = \arg\min_u \boxed{\nabla_\theta \mathcal{L}(p_{\theta_k})^\top u} + \boxed{\frac{1}{2}\|u\|^2}$$

- Different re-parametrization: $\psi = s(\theta)$

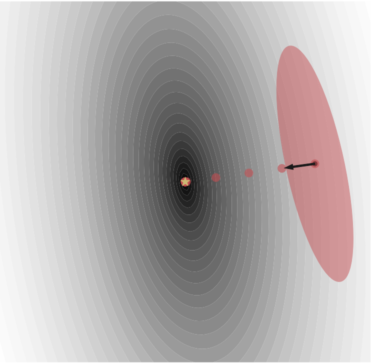


Fisher Natural Gradient

- Learning problem: $\theta^* = \arg\min_\theta \mathcal{L}(p_\theta)$
- Update equation: $\theta_{k+1} = \theta_k + \lambda \mathcal{D}_k$

$$\mathcal{D}_k = \arg\min_u \nabla_\theta \mathcal{L}(p_{\theta_k})^\top u + \frac{1}{2} \underbrace{u^\top G_F(\theta_k) u}_{\substack{\approx \\ KL(p_{\theta_k} || p_{\theta_k + u})}}$$

- Fisher information matrix: $G_F(\theta)$



Pros:

- Invariant to parametrization

Cons:

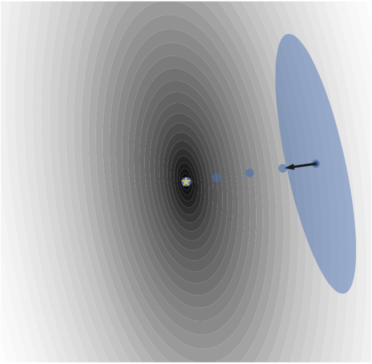
- Not scalable, but efficient approximations exist:
  [Martens and Grosse, 2015, Grosse and Martens, 2016]
- Requires the density of p_θ to be well defined.

Wasserstein Natural Gradient

- Learning problem: $\theta^* = \arg\min_\theta \mathcal{L}(p_\theta)$
- Update equation: $\theta_{k+1} = \theta_k + \lambda \mathcal{D}_k$

$$\mathcal{D}_k = \arg\min_u \nabla_\theta \mathcal{L}(p_{\theta_k})^\top u + \frac{1}{2} \underbrace{u^\top G_W(\theta_k) u}_{\approx W_2^2(p_{\theta_k}, p_{\theta_k+u})}$$

- Wasserstein information matrix: $G_W(\theta)$



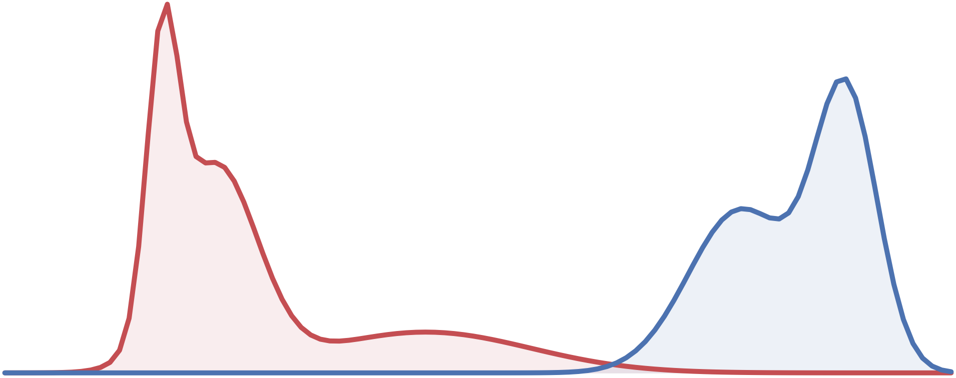
Pros:

- Invariant to parametrization
- Works with implicit model
- Scalable approximation

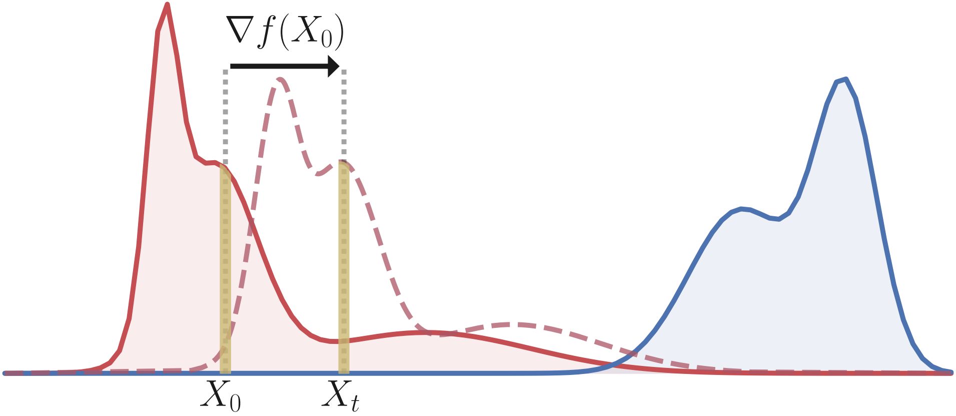
Cons:

- ~~Not scalable~~
- ~~Requires the density of p_θ to be well defined.~~

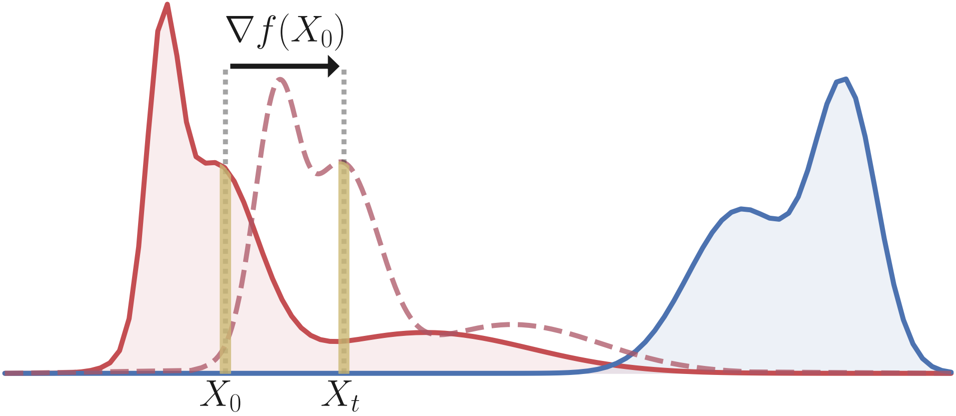
The Wasserstein Information Matrix



The Wasserstein Information Matrix

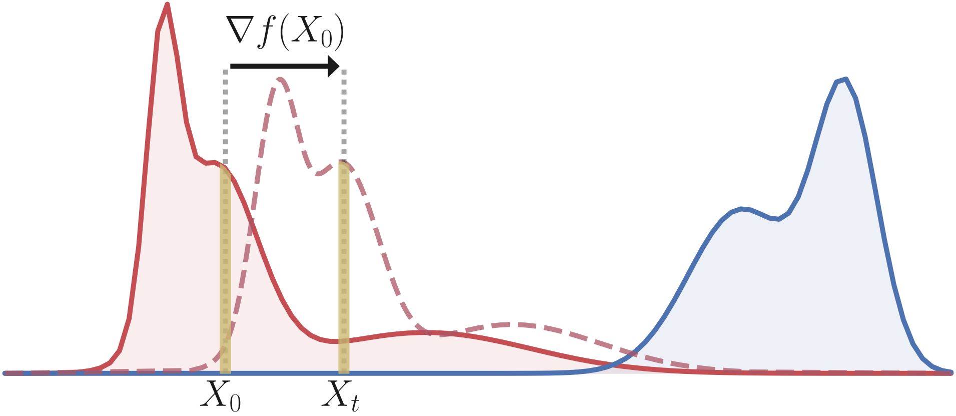


The Wasserstein Information Matrix



$$\sup_f \overbrace{\nabla_\theta \mathbb{E}_{p_\theta}\left[f(X)\right]^\top u}^{{\color{red}p_\theta} \to {\color{blue}p_{\theta+u}}} - \frac{1}{2}\overbrace{\mathbb{E}_{p_\theta}\left[\|\nabla f(X)\|^2\right]}^{\text{Gradient control}}$$

The Wasserstein Information Matrix



$$\frac{1}{2}u^\top G_W(\theta)u = \sup_f \overbrace{\nabla_\theta \mathbb{E}_{p_\theta}\left[f(X)\right]^\top u}^{{\color{red}p_\theta} \to {\color{blue}p_{\theta+u}}} - \frac{1}{2}\overbrace{\mathbb{E}_{p_\theta}\left[\|\nabla f(X)\|^2\right]}^{\text{Gradient control}}$$

Saddle-point formulation

$$\min_u \boxed{\nabla_\theta \mathcal{L}(p_\theta)^\top u} + \boxed{\frac{1}{2} u^\top G_W(\theta) u}$$

$$\Downarrow$$

$$\min_u \sup_{f \in \mathcal{H}_M} \boxed{\nabla_\theta \mathcal{L}(p_\theta)^\top u} + \boxed{\nabla_\theta \mathbb{E}_{p_\theta} [f(X)]^\top u - \frac{1}{2} \mathbb{E}_{p_\theta} \left[\|\nabla f(X)\|^2 \right]}$$

- $\mathcal{H}_M$ contains functions of the form:

$$f(x) = \sum_{m=1}^{M} \alpha_m \phi_m(x)$$

Saddle-point formulation

$$\min_u \boxed{\nabla_\theta \mathcal{L}(p_\theta)^\top u} + \boxed{\frac{1}{2} u^\top G_W(\theta) u} + \boxed{\overbrace{\frac{\epsilon}{2}\|u\|^2}^{\text{damping}}}$$

$$\Downarrow$$

$$\min_u \sup_{f \in \mathcal{H}_M} \boxed{\nabla_\theta \mathcal{L}(p_\theta)^\top u} + \boxed{\nabla_\theta \mathbb{E}_{p_\theta}[f(X)]^\top u - \frac{1}{2}\mathbb{E}_{p_\theta}\left[\|\nabla f(X)\|^2\right]} + \boxed{\overbrace{\frac{\epsilon}{2}\|u\|^2}^{\text{damping}}}$$

- $\mathcal{H}_M$ contains functions of the form:

$$f(x) = \sum_{m=1}^{M} \alpha_m \phi_m(x)$$

Saddle-point formulation

$$\min_u \boxed{\nabla_\theta \mathcal{L}(p_\theta)^\top u} + \boxed{\frac{1}{2} u^\top G_W(\theta) u} + \boxed{\frac{\epsilon}{2}\|u\|^2}$$

$$\Downarrow$$

$$\sup_{f \in \mathcal{H}_M} \min_u \boxed{\nabla_\theta \mathcal{L}(p_\theta)^\top u + \nabla_\theta \mathbb{E}_{p_\theta}\left[f(X)\right]^\top u} - \boxed{\frac{1}{2}\mathbb{E}_{p_\theta}\left[\|\nabla f(X)\|^2\right]} + \boxed{\frac{\epsilon}{2}\|u\|^2}$$

- $\mathcal{H}_M$ contains functions of the form:

$$f(x) = \sum_{m=1}^{M} \alpha_m \phi_m(x)$$

- Optimal $f^\star$ obtained by solving a quadratic problem of size M in $(\alpha_1, ..., \alpha_M)$
- Wasserstein natural descent direction:

$$\widehat{\mathcal{D}}_k = -\frac{1}{\epsilon}\left(\nabla_\theta \mathcal{L}(p_{\theta_k}) + \nabla_\theta \mathbb{E}_{p_{\theta_k}}\left[f^\star(X)\right]\right)$$

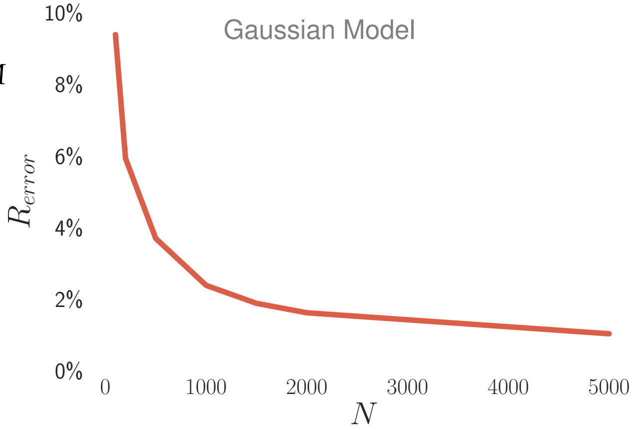
How small M can be and still be sure it works?

- Need fewer basis points M than data points N

$$M \approx \sqrt{N}$$

- Relative error decreases with more data ($N \to +\infty$)

$$R_{error} \sim \frac{1}{N^{\frac{1}{4}}}$$



Conclusion

KWNG: A natural gradient optimizer with built in Optimal Transport Geometry.

- ✓ Approximately invariant to re-parametrization
- ✓ Fast and scalable
- ✓ Can be used as a drop-in optimizer
- ✓ Comes with statistical guarantees

Code:

`https://github.com/MichaelArbel/KWNG`

Thank you !

Grosse, R. and Martens, J. (2016).
A Kronecker-factored Approximate Fisher Matrix for Convolution Layers.
In Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48, ICML'16, pages 573–582. JMLR.org.
event-place: New York, NY, USA.

Martens, J. and Grosse, R. (2015).
Optimizing Neural Networks with Kronecker-factored Approximate Curvature.
arXiv:1503.05671 [cs, stat].
arXiv: 1503.05671.