Learning with Differentiable Perturbed Optimizers

Quentin Berthet

Google Research

Youth in High-dimensions - ICTP - 2020

Q. Berthet

M.Blondel

O.Teboul

M. Cuturi

J-P. Vert

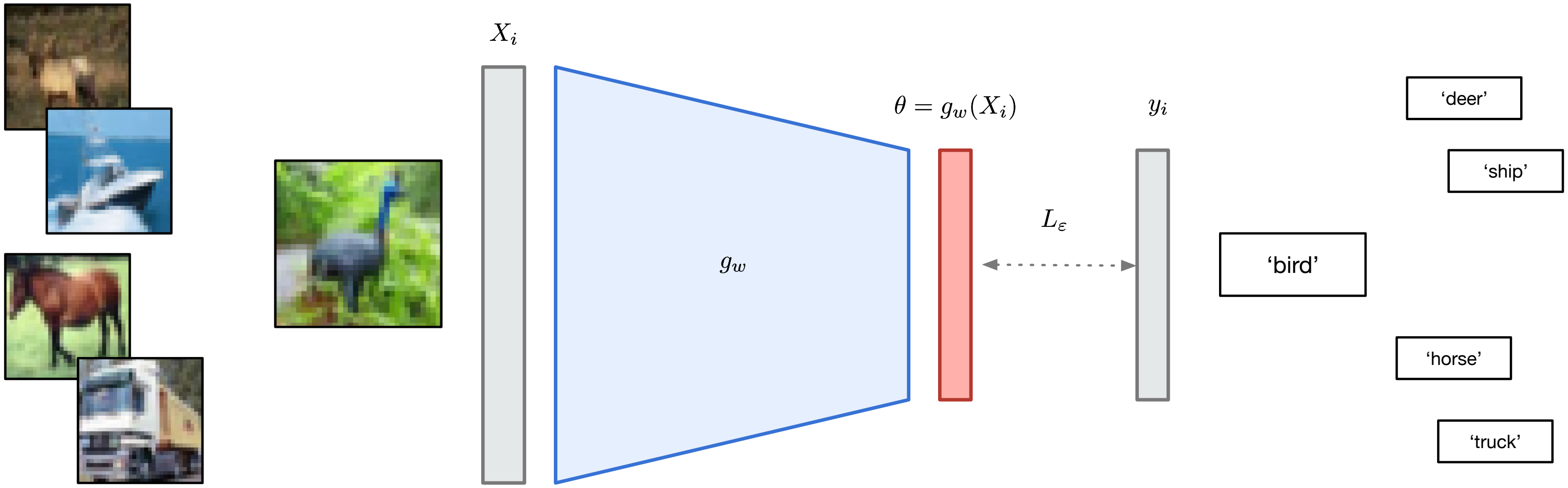
F.Bach

- **Learning with Differentiable Perturbed Optimizers**

  Preprint: **arXiv:2002.08676**

[A lot of] Machine learning these days

Supervised learning: couples of inputs/responses (X_i, y_i), a model g_w



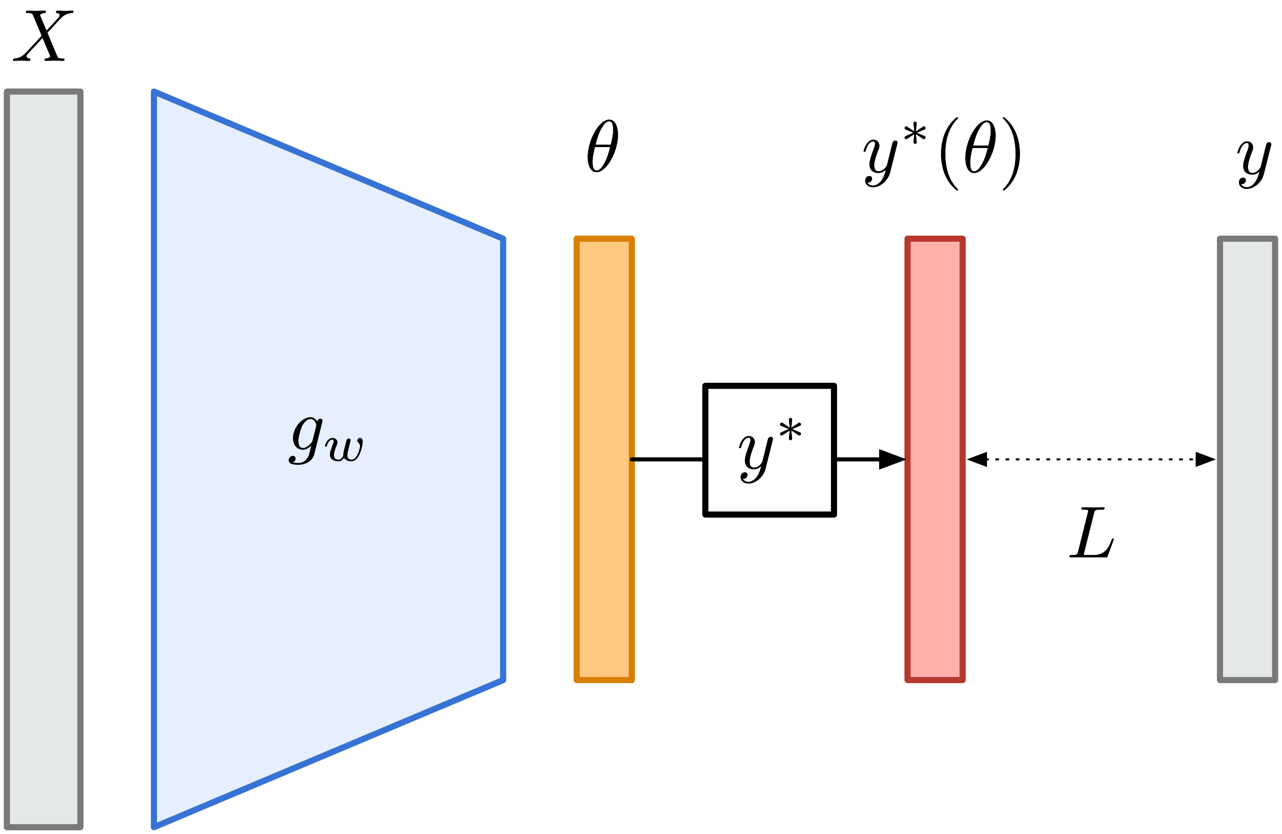
Goal: Optimize parameters $w \in \mathbf{R}^d$ of a function g_w such that $g_w(X_i) \approx y_i$

$$\min_w \sum_i L(g_w(X_i), y_i)\,.$$

Workhorse: first-order methods, based on $\nabla_w L(g_w(X_i), y_i)$, backpropagation

Problem: What if these models contain **nondifferentiable*** operations?

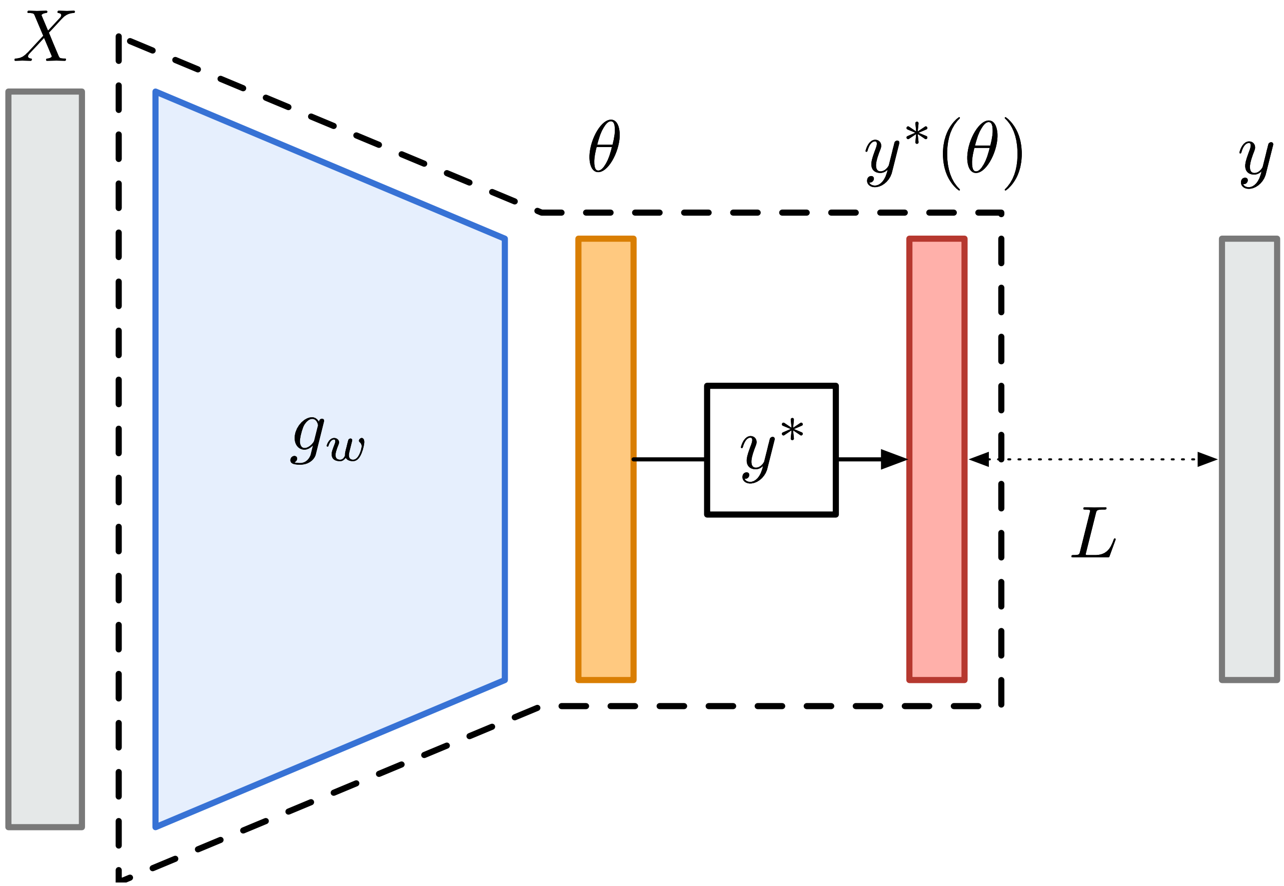
Discrete decisions in Machine learning



Examples: discrete operations (e.g. max, rankings), break autodifferentiation

- $\theta =$ scores for k products, $y^* =$ vector of ranks e.g. $[5, 2, 4, 3, 1]$
- $\theta =$ edge costs, $y^* =$ shortest path between two points
- $\theta =$ classification scores for each class, $y^* =$ one-hot vector

Discrete decisions in Machine learning



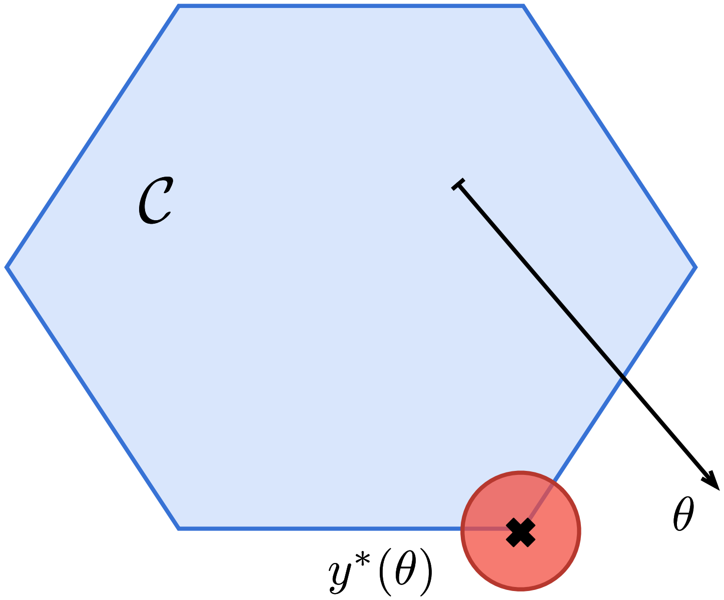
Examples: discrete operations (e.g. max, rankings), break autodifferentiation

- $\theta =$ scores for k products, $y^* =$ vector of ranks e.g. $[5, 2, 4, 3, 1]$
- $\theta =$ edge costs, $y^* =$ shortest path between two points
- $\theta =$ classification scores for each class, $y^* =$ one-hot vector

Perturbed maximizer

Discrete decisions: optimizers of linear program over $\mathcal{C}$, convex hull of $\mathcal{Y} \subseteq \mathbf{R}^d$

$$F(\theta) = \max_{y \in \mathcal{C}} \langle y, \theta \rangle \,, \quad \text{and} \quad y^*(\theta) = \operatorname*{argmax}_{y \in \mathcal{C}} \langle y, \theta \rangle = \nabla_\theta F(\theta) \,.$$

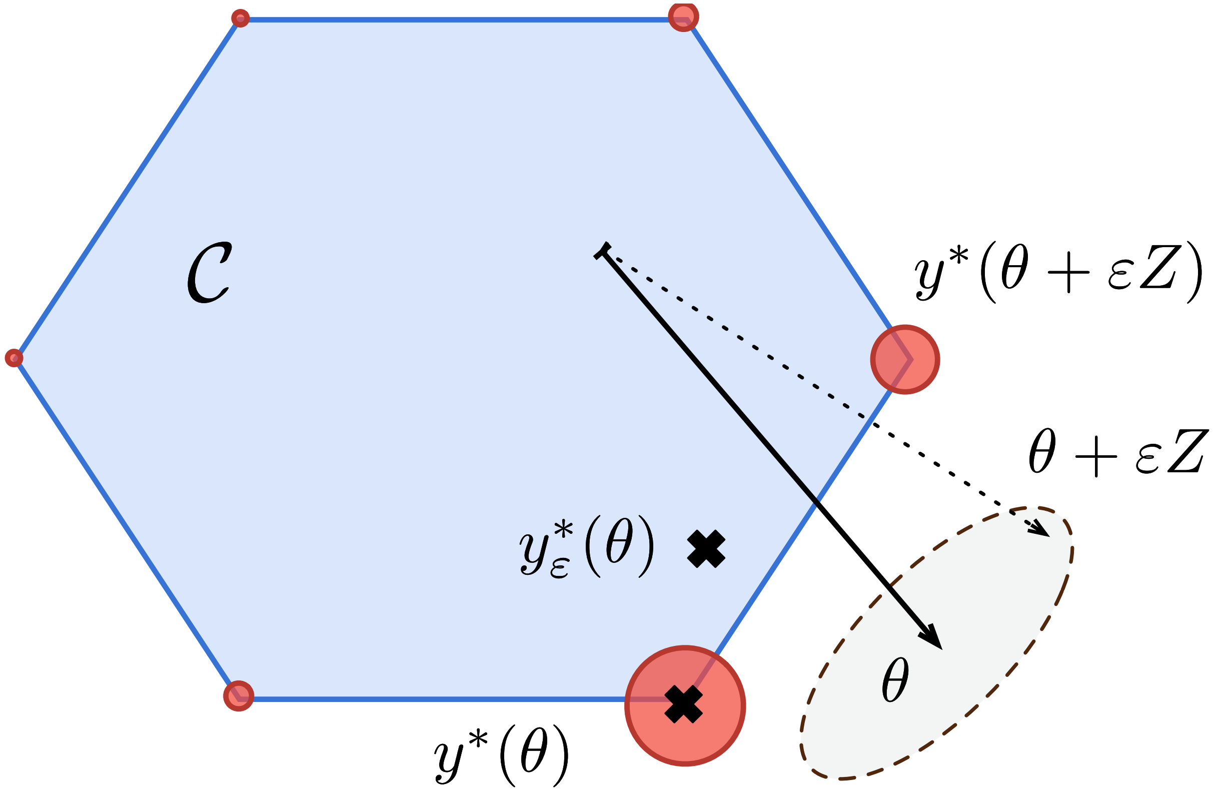


Perturbed maximizer: average of solutions for inputs with noise εZ

$$F_\varepsilon(\theta) = \mathbf{E}[\max_{y \in \mathcal{C}} \langle y, \theta + \varepsilon Z \rangle] \,,\ y^*_\varepsilon(\theta) = \mathbf{E}[y^*(\theta + \varepsilon Z)] = \mathbf{E}[\operatorname*{argmax}_{y \in \mathcal{C}} \langle y, \theta + \varepsilon Z \rangle] = \nabla_\theta F_\varepsilon(\theta) \,.$$

Perturbed maximizer

Discrete decisions: optimizers of linear program over $\mathcal{C}$, convex hull of $\mathcal{Y} \subseteq \mathbf{R}^d$



Perturbed maximizer: average of solutions for inputs with noise εZ

$$F_\varepsilon(\theta) = \mathbf{E}[\max_{y \in \mathcal{C}} \langle y, \theta + \varepsilon Z \rangle]\,,\ y^*_\varepsilon(\theta) = \mathbf{E}[y^*(\theta + \varepsilon Z)] = \mathbf{E}[\operatorname*{argmax}_{y \in \mathcal{C}} \langle y, \theta + \varepsilon Z \rangle] = \nabla_\theta F_\varepsilon(\theta)\,.$$

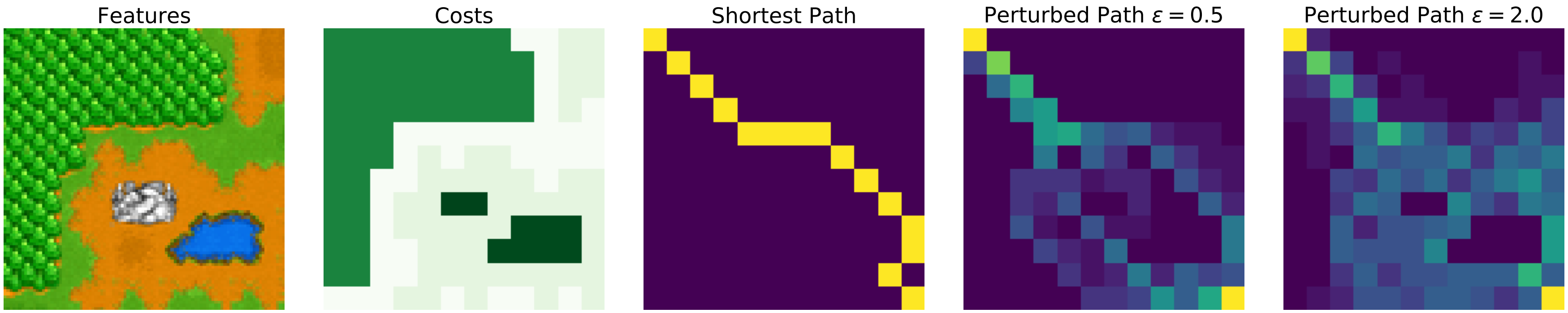
Perturbed model

Model of optimal decision under uncertainty Luce (1959), McFadden et al. (1973)

$$Y = \underset{y \in \mathcal{C}}{\operatorname{argmax}} \langle y, \theta + \varepsilon Z \rangle$$

Follows a **perturbed model** with $Y \sim p_\theta(y)$, expectation $y^*_\varepsilon(\theta) = \mathbf{E}_{p_\theta}[Y]$.

Perturb and map Papandreou & Yuille (2011), FT Perturbed L Kalai & Vempala (2003)



Example. Over the unit simplex $\mathcal{C} = \Delta^d$ with Gumbel noise Z, Gibbs distribution.

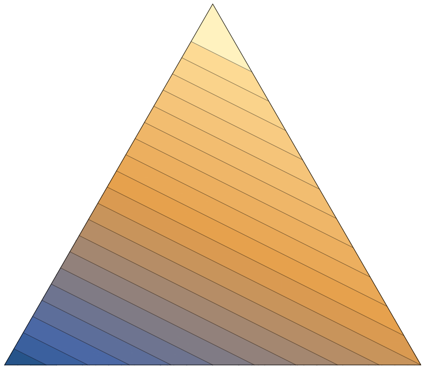
$$F_\varepsilon(\theta) = \varepsilon \log \sum_{i \in [d]} e^{\frac{\theta_i}{\varepsilon}}\,, \qquad p_\theta(e_i) \propto \exp(\langle \theta, e_i \rangle / \varepsilon)\,, \qquad [y^*_\varepsilon(\theta)]_i = \frac{e^{\frac{\theta_i}{\varepsilon}}}{\sum e^{\frac{\theta_j}{\varepsilon}}}$$

Properties

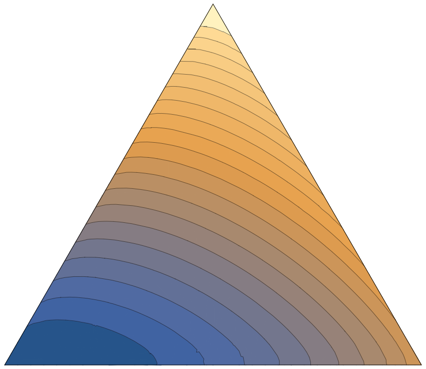
Link with regularization: $\varepsilon\,\Omega = \big(F_\varepsilon\big)^*$ is a convex function with domain $\mathcal{C}$

$$y^*_\varepsilon(\theta) = \operatorname*{argmax}_{y \in \mathcal{C}} \big\{ \langle y, \theta \rangle - \varepsilon \Omega(y) \big\} \,.$$

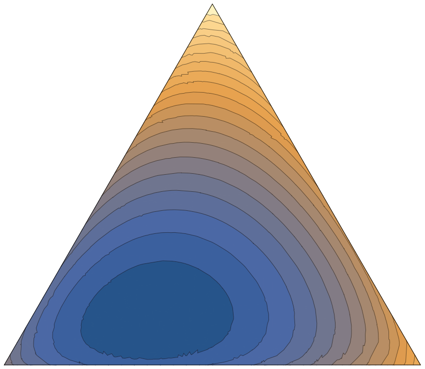
Consequence of duality and $y^*_\varepsilon(\theta) = \nabla_\varepsilon F_\varepsilon(\theta)$. Generalized entropy Ω



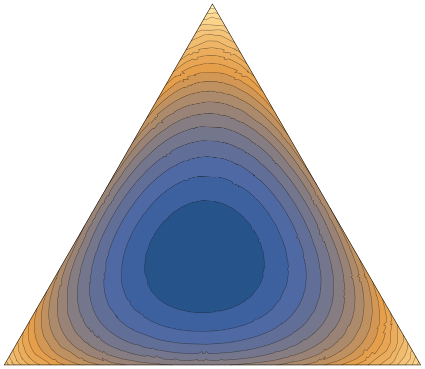
$\varepsilon = 0$



tiny ε



small ε



large ε

Extreme temperatures. When $\varepsilon \to 0$, $y^*_\varepsilon(\theta) \to y^*(\theta)$ for unique max.

When $\varepsilon \to \infty$, $y^*_\varepsilon(\theta) \to \operatorname{argmin}_y \Omega(y)$. Nonasymptotic results.

Differentiability. Smoothness in the inputs, Jacobian as simple expectations.

Learning and Fenchel-Young losses

Learning from $Y_1, \dots, Y_n$ for a model p_θ.

Gibbs distribution $\propto \exp(\langle \theta, Y \rangle)$: minimize negative log-likelihood

$$L_{\text{Gibbs}}(\theta; Y) = -\frac{1}{n} \sum_{i=1}^{n} \langle \theta, Y_i \rangle + \log Z(\theta)$$

Stochastic gradient and full (batch) gradient: moment matching

$$\nabla_\theta L_{\text{Gibbs}}(\theta; Y_i) = \mathbf{E}_{\text{Gibbs},\theta}[Y] - Y_i \,, \quad \nabla_\theta L_{\text{Gibbs}}(\theta; Y) = \mathbf{E}_{\text{Gibbs},\theta}[Y] - \bar{Y}_n \,.$$

Algorithmic challenge: replace by perturbed model Papandreou, Yuille (2011)

$$\nabla_\theta L_i(\theta) = \mathbf{E}_{p_\theta}[Y] - Y_i = y^*_\varepsilon(\theta) - Y_i \,.$$

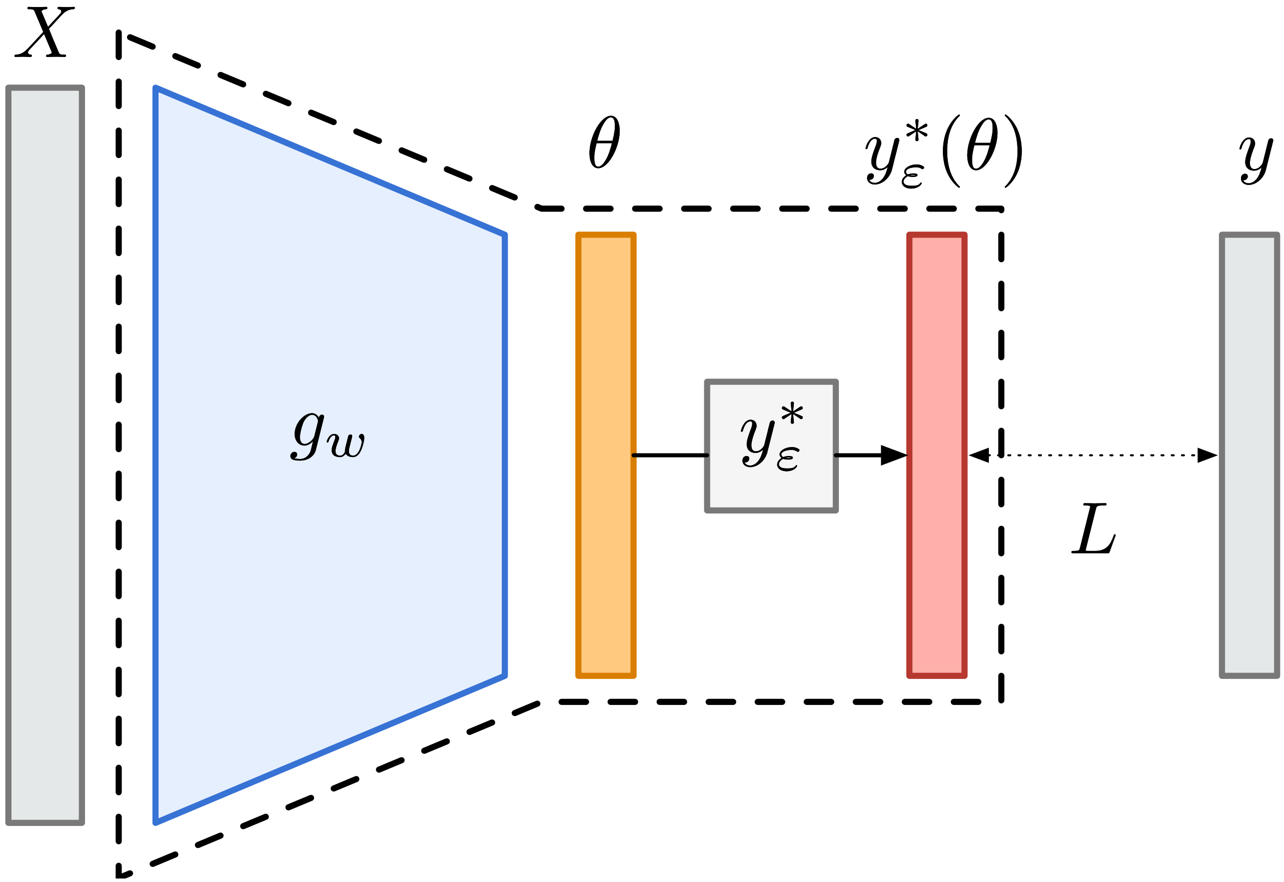
Stochastic gradient of modified functional in θ, not a log-likelihood

$$L_\varepsilon(\theta; y) = -\frac{1}{n} \sum_{i=1}^{n} \langle \theta, Y_i \rangle + F_\varepsilon(\theta) \,.$$

Fenchel-Young loss Blondel et al. (2019), good properties (convexity, randomness).

Learning with perturbations and F-Y losses

Within the same framework, possible to virtually bypass the optimization block

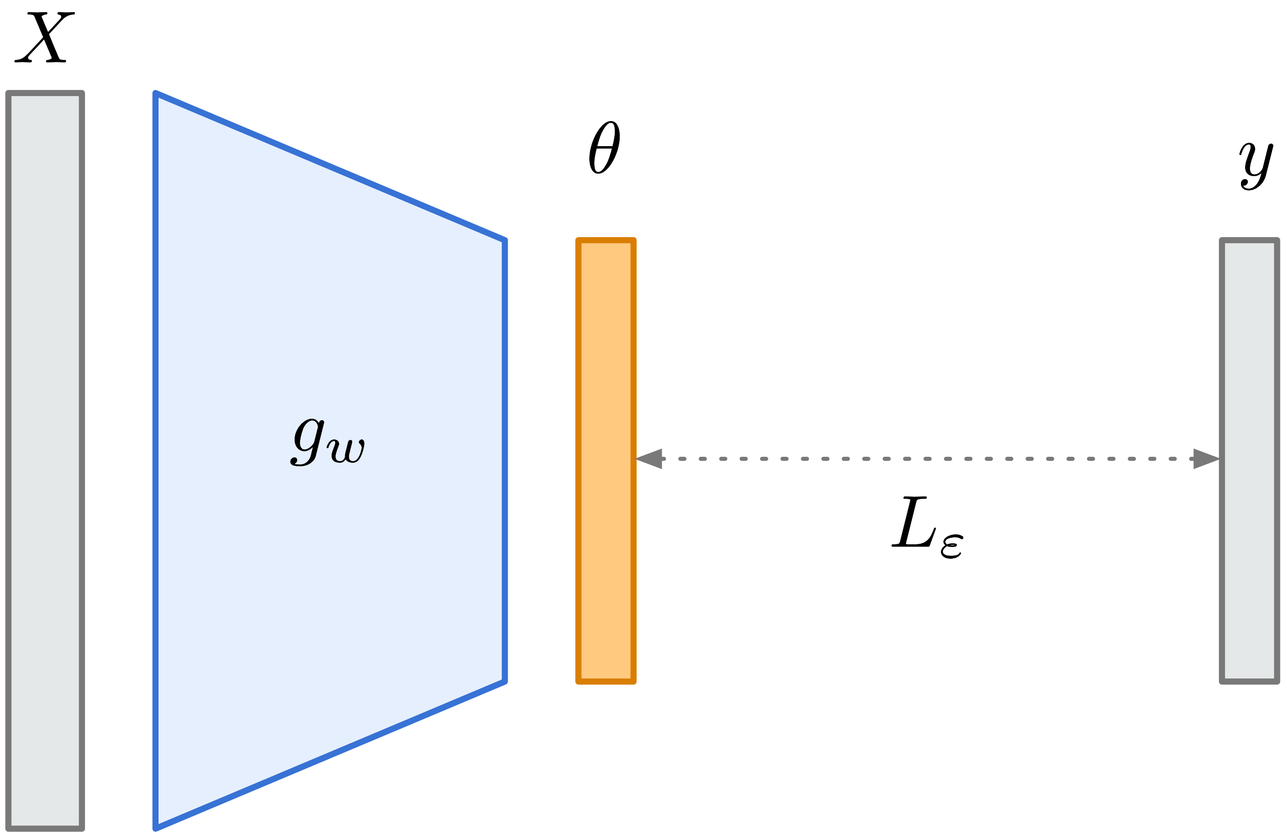


Easier to implement, no Jacobian of y^*_ε

Population loss minimized at ground truth for perturbed generative model.

Learning with perturbations and F-Y losses

Within the same framework, possible to virtually bypass the optimization block



Easier to implement, no Jacobian of y^*_ε

Population loss minimized at ground truth for perturbed generative model.

Computations

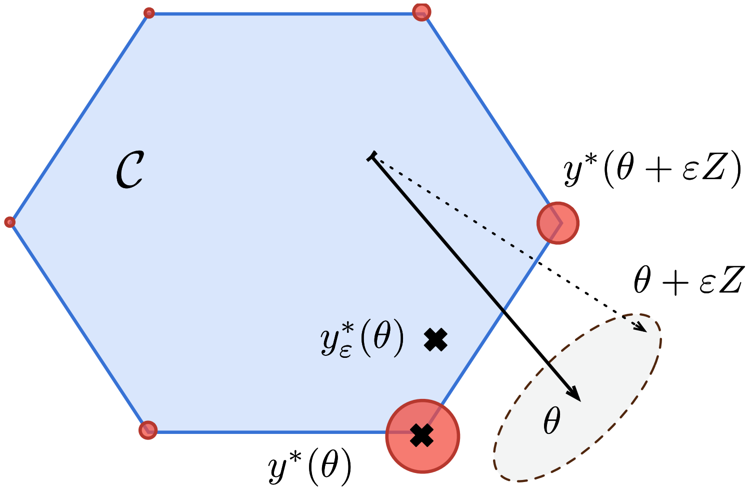
Monte Carlo estimates. Perturbed maximizer and derivatives as expectations.

For $\theta \in \mathbf{R}^d$, $Z^{(1)}, \dots, Z^{(M)}$ i.i.d. copies

$$y^{(\ell)} = y^*(\theta + \varepsilon Z^{(\ell)})$$

Unbiased estimate of $y^*_\varepsilon(\theta)$ given by

$$\bar{y}_{\varepsilon,M}(\theta) = \frac{1}{M} \sum_{\ell=1}^{M} y^{(\ell)} \, .$$



Supervised learning:

Features X_i, model output $\theta_w = g_w(X_i)$, prediction $y_{\text{pred}} = y^*_\varepsilon(\theta_w)$.

Stochastic gradient in w:

$$\nabla_w F_i(w) = J_w g_w(X_i) \cdot (y^*_\varepsilon(\theta) - Y_i)$$

Computations

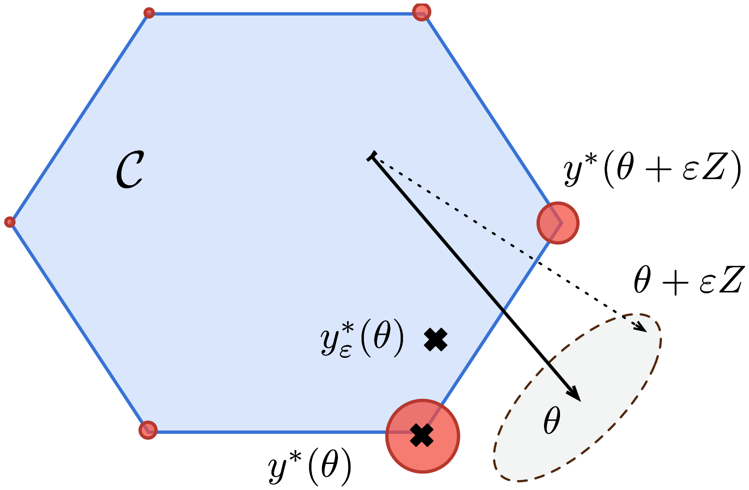
Monte Carlo estimates. Perturbed maximizer and derivatives as expectations.

For $\theta \in \mathbf{R}^d$, $Z^{(1)}, \dots, Z^{(M)}$ i.i.d. copies

$$y^{(\ell)} = y^*(\theta + \varepsilon Z^{(\ell)})$$

Unbiased estimate of $y^*_\varepsilon(\theta)$ given by

$$\bar{y}_{\varepsilon,M}(\theta) = \frac{1}{M} \sum_{\ell=1}^{M} y^{(\ell)} \, .$$



Supervised learning:

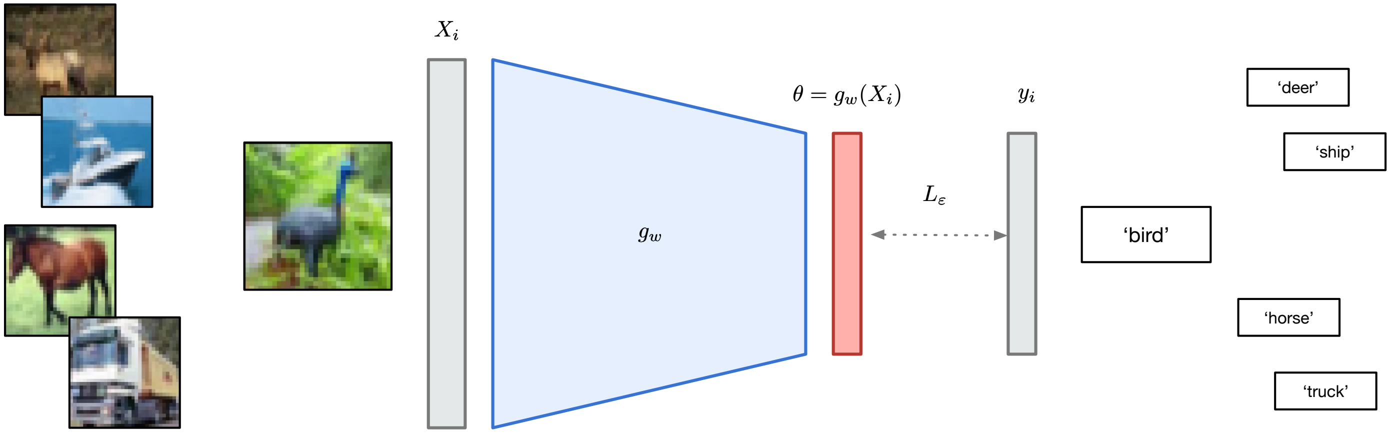
Features X_i, model output $\theta_w = g_w(X_i)$, prediction $y_{\text{pred}} = y^*_\varepsilon(\theta_w)$.

Stochastic gradient in w (doubly stochastic scheme)

$$\nabla_w F_i(w) = J_w g_w(X_i) \cdot \Big(\frac{1}{M} \sum_{\ell=1}^{M} y^*(\theta + \varepsilon Z^{(\ell)}) - Y_i \Big) \, .$$

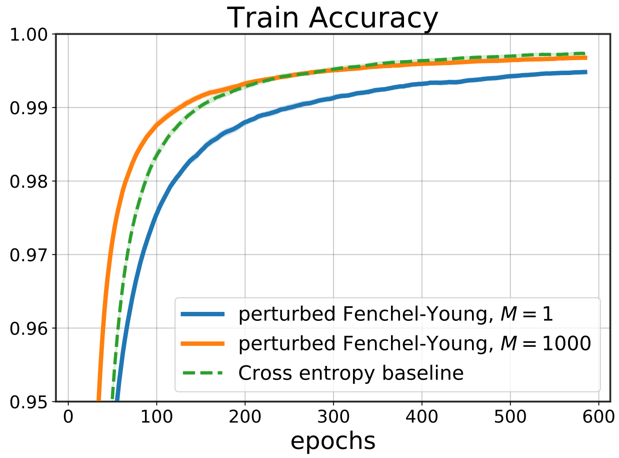
Experiments

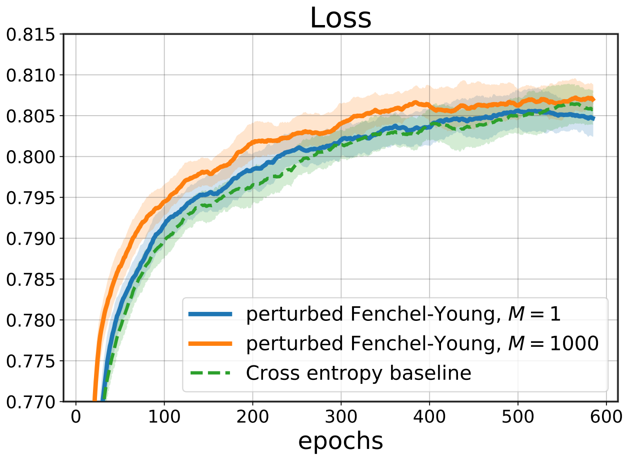
Classification: CIFAR-10 dataset of images with 10 classes - Toy comparison

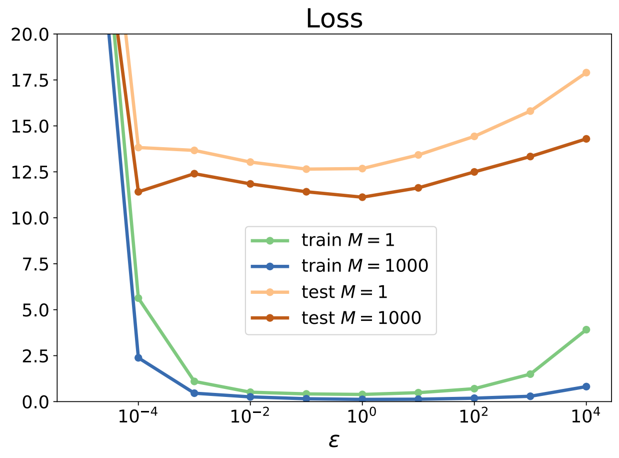


Architecture: vanilla-CNN made of 4 convolutional and 2 fully connected layers.

Training: 600 epochs with minibatches of size 32 - influence of M and ε

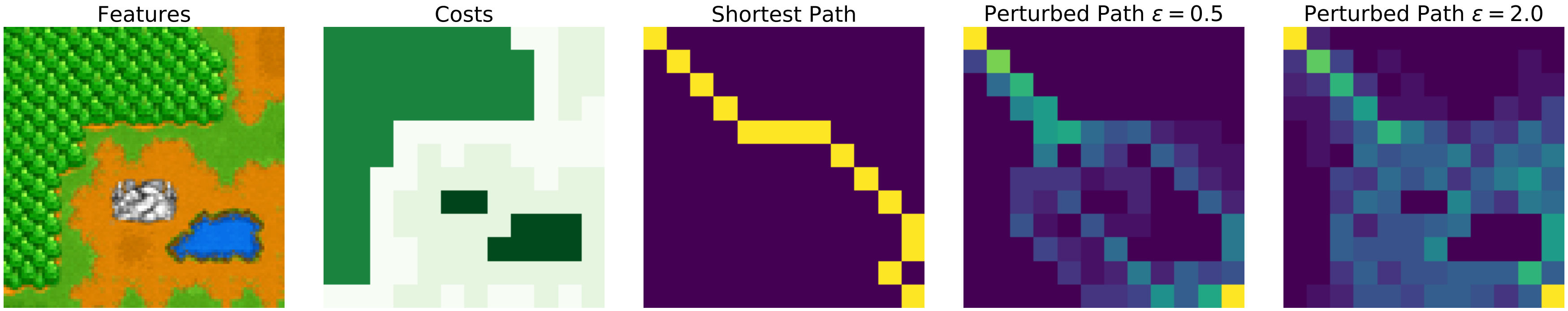




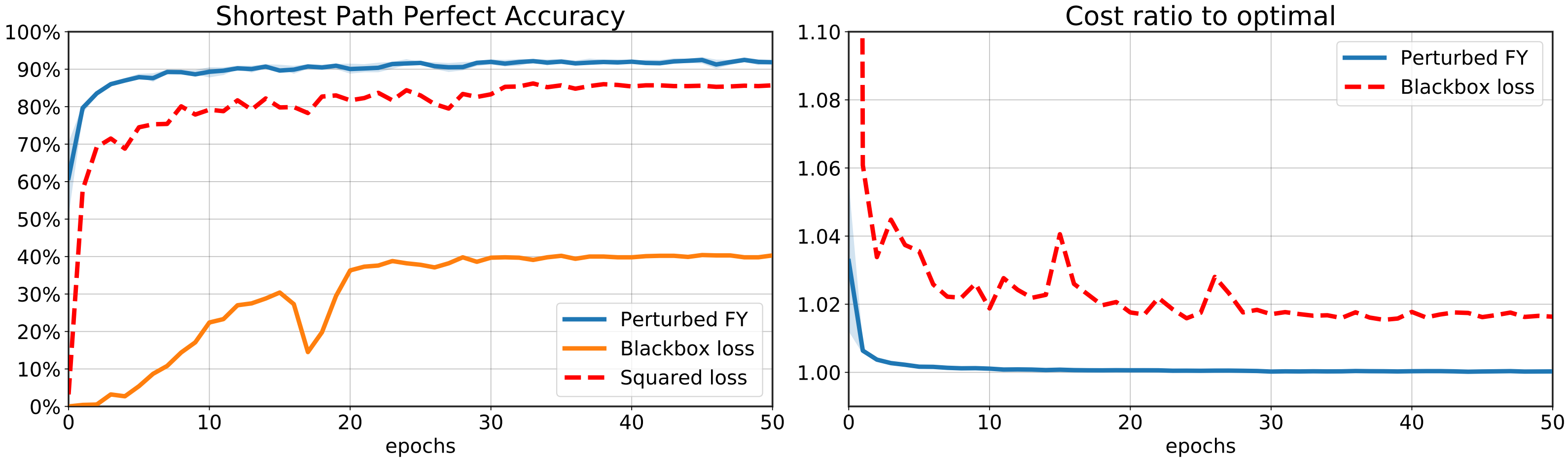


Experiments

Learning from shortest paths: From 10k examples of Warcraft 96×96 RGB images, representing 12×12 costs, and matrix of shortest paths. (Vlastelica et al. 19)


Features
Costs
Shortest Path
Perturbed Path $\varepsilon = 0.5$
Perturbed Path $\varepsilon = 2.0$

Train a CNN for 50 epochs, to learn costs recovery of optimal paths.



GRAZZIE