End-to-End Learning

with an

Optimization Model

$$\text{(inference)} = \underset{x \,\in\, \square}{\text{argmin}}\ \square + \square + \square$$

$\square$ = constraint

$\square$ = objective (analytic)

$\square$ = regularizer (analytic)

$\square$ = regularizer (data-driven)

Howard Heaton

Typal Academy

Overview

- **Setting**

  Problems where optimization models can be hand-crafted to roughly estimate solutions, but could be improved with data.

- **Model Structure**

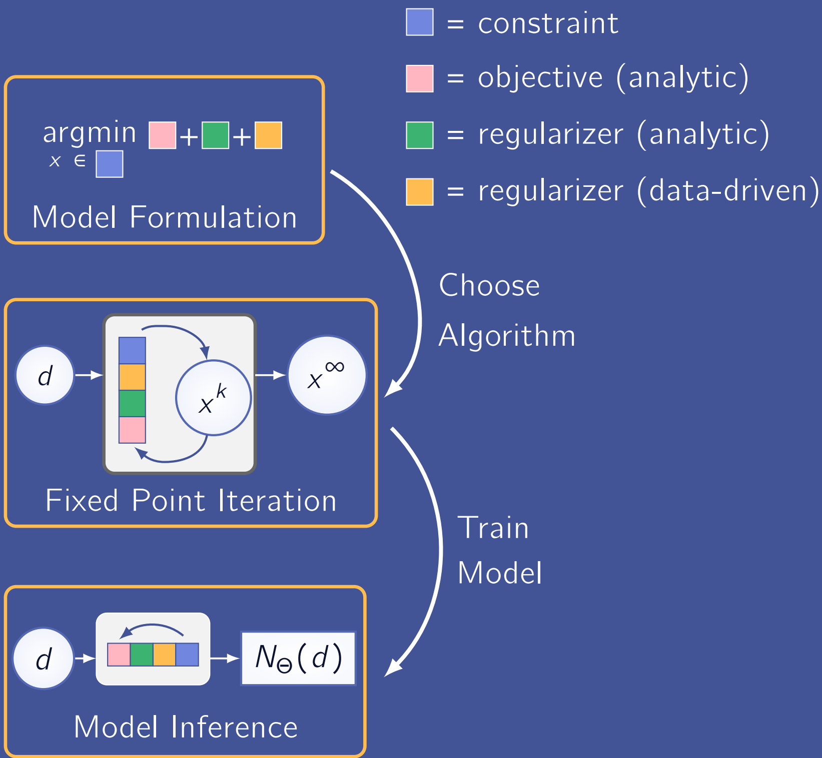
  Model includes prior knowledge (e.g. physical constraints) and include data-driven terms (e.g. parameterized regularizers, convolutions):

$$
\begin{aligned}
(\text{inference}) = \operatorname{argmin}\ &(\text{prior knowledge}) \\
&+ (\text{data-driven terms})
\end{aligned}
$$

These slides illustrate this via a toy inverse problem.

Modeling + Learning

Below is a schematic for building these models.



Modeling + Learning

Define Optimization Model

Set inferences $N_\Theta(d)$ to be optimizers:

$$N_\Theta(d) = \operatorname*{argmin}_x f_\Theta(x, d),$$

with f parameterized by weights Θ and input data d.

Construct Optimization Algorithm

Use first-order scheme such as gradient descent with

$x^{k+1} = x^k - \alpha \nabla f_\Theta(x^k, d)$ so that $N_\Theta(d) = \lim_{k\to\infty} x^k$.

End-to-End Training

Weights Θ are tuned so inferences minimize loss on given data. Mean square error is commonly used:

$$\min_\Theta \mathbb{E}_d \left[\|x_d^\star - N_\Theta(d)\|^2 \right].$$

Important Training Note

If inferences are computed as $N_\Theta(d) = x^n$ for large n,

then one cannot backprop through all n iterations!

Problem with Standard Backprop

Memory grows linearly with $n \implies$ memory blow up.

Solution

Only compute loss gradient for step from x^{n-1} to x^n.

Note

This is called "JFB" and is trivial to code in Pytorch.

Inverse Problem – Example Setup

- **Task**

  Recover signal $x_d^\star$ from measurements $d = Ax_d^\star$

- **Given Information**

  Linear system $Ax_d^\star = d$ is underdetermined

  Signal $x_d^\star$ has low-dimensional structure

- **Model**

  If $x_d^\star = Mz_d^\star$ for a matrix M and low-dimensional $z_d^\star$,

  there is a square matrix Θ for which $\Theta x_d^\star$ is sparse.

  Thus, assume $N_\Theta(d) \approx x_d^\star$, where $N_\Theta(d)$ solves

$$\min_x \ \|\Theta x\|_1 \quad \text{s.t.} \quad Ax = d,$$

  where the norm $\|\cdot\|_1$ is used to make Θx sparse.

Inverse Problem – Algorithm

N_Θ can be evaluated using linearized ADMM.* Here this uses step sizes α, β, λ, an auxiliary variable p, and dual variables ν and ω. Optimizer estimates x^k are iteratively updated via the batch of updates:†

$$p^{k+1} = \text{shrink}\left(p^k + \lambda[\nu^k + \alpha(\Theta x^k - p^k)],\ \lambda\right)$$

$$\nu^{k+1} = \nu^k + \alpha(\Theta x^k - p^{k+1})$$

$$\omega^{k+1} = \omega^k + \alpha(Ax^k - d)$$

$$x^{k+1} = x^k - \beta[\Theta^\top(2\nu^{k+1} - \nu^k) + A^\top(2\omega^{k+1} - \omega^k)].$$

The inference is $N_\Theta(d) = \lim_{k\to\infty} x^k \approx x^n$ for some large n.

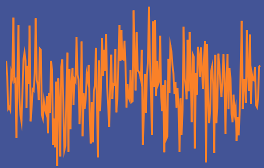
*See Appendices B and C in my paper for derivation details.
†Note $\text{shrink}(p, \lambda) = \text{sign}(p) \cdot \max(0,\ |p| - \lambda)$.

Inverse Problem – Training

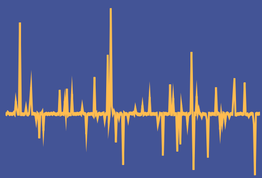
The tunble weights Θ in the model N_Θ form a square matrix. Given pairs $\{(d_n, x^\star_{d_n})\}_{n=1}^N$ for training data, optimal $\Theta^\star$ is found by solving the training problem

$$\min_{\Theta} \frac{1}{N}\sum_{n=1}^{N} \|x^\star_{d_n} - N_\Theta(d_n)\|^2.$$

Plots below use measurements d drawn from *test data*.



True Signal $x^\star_d$

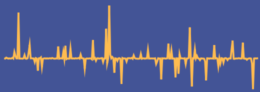


Sparsified Signal $\Theta x^\star_d$

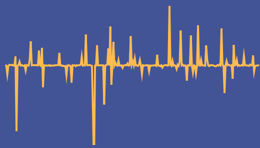
This shows optimal $\Theta^\star$ yields sparse $\Theta x^\star_d$, as desired.

Inverse Problem – Plots

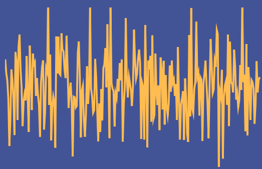
Plots show $\Theta^\star$ multiplied by various estimates of $x_d^\star$



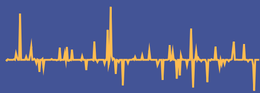
Ground Truth $x_d^\star$



Wrong Signal $x_p^\star$ with $p \neq d$



Least Squares $A^\top(AA^\top)^{-1}d$



Model Inference $N_\Theta(d)$

Estimate	Sparse Transform	Satisfy Constraint
Ground Truth	✓	✓
Wrong Signal	✓	✗
Least Squares	✗	✓
Model Inference	✓	✓

Inverse Problem – Summary

- Inference solves optimization problem:

$$N_\Theta(d) = \underset{x}{\operatorname{argmin}} \, \|\Theta x\|_1 \quad \text{s.t.} \quad Ax = d.$$

- Linear system $Ax = d$ is underdetermined
- Learned Θ sparsifies signal $x_d^\star$
- Signal $x_d^\star$ is *not* well-approximated via least squares

  i.e. low-rank structure must be exploited
- Signal $x_d^\star$ is well-approximated by $N_\Theta(d)$

Takeaways

When crafting a model N_Θ that is defined via an optimization problem and trained as shown:

- Model N_Θ can rigorously capture prior knowledge (*e.g.* hard constraints in physical systems)
- Model N_Θ is evaluated via an optimization algorithm (*e.g.* proximal gradient, ADMM, Davis-Yin splitting)
- Model parameters can be tuned for optimal performance on a specific distribution of data

Found this useful?

- Follow for more
- Repost to share with friends