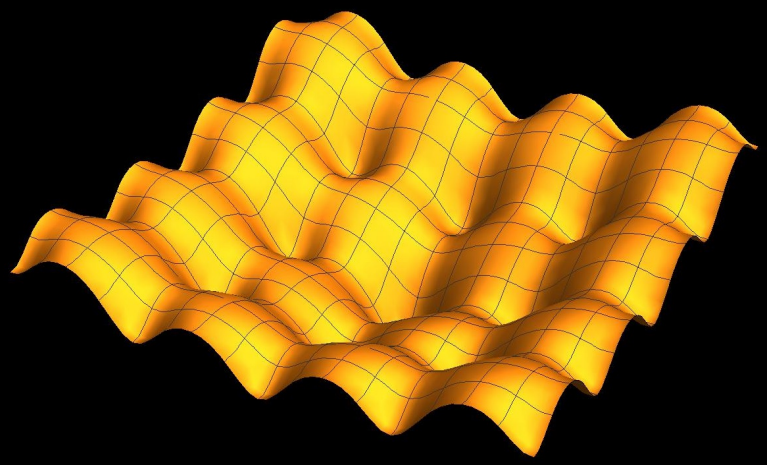
Nonconvex Optimization by Complexity Progression

Hossein Mobahi
Google Research

EECS Seminar, UC Merced
November 4th 2016

Learning as Optimization



A Typical Learning Scenario

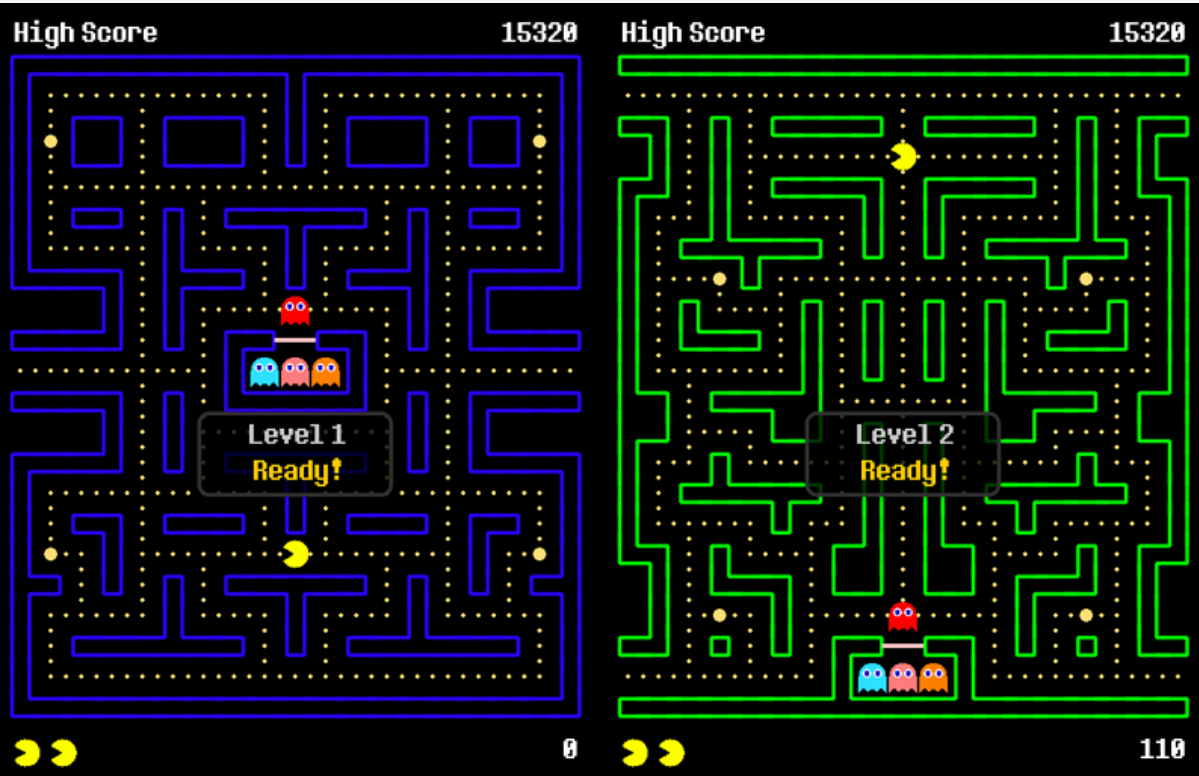
- Training data $\mathcal{X}$.
- Solution space $(\theta_1, \ldots, \theta_n)$.
- Cost criterion $f(\mathcal{X}, \theta_1, \ldots, \theta_n)$.

Learning seeks $(\theta_1, \ldots, \theta_n)$ which minimizes $f(\mathcal{X}, \theta_1, \ldots, \theta_n)$.
When f is convex, any local minimum is a global minimum (good for local search algorithms).

Learning Complex Tasks

Progressive Strategy

- Start from what we can solve.
- Gradually increase the complexity of the task.
- Move to the more complex task when the current one is learned.



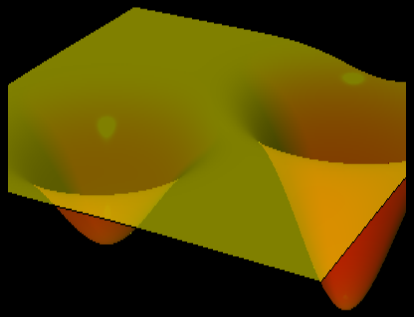
Extensively Used in Machine Learning

Examples

- Graduated NonConvexity (GNC).
- Deterministic Annealing.
- Mean Field Approximation.
- Curriculum Learning
- Layerwise Pretraining

Optimization by Continuation

$$f(\boldsymbol{x}) \triangleq$$



$$k_\sigma(\boldsymbol{x}) \triangleq \frac{1}{(\sqrt{2\pi}\sigma)^{\dim(\boldsymbol{x})}} e^{-\frac{\|\boldsymbol{x}\|^2}{2\sigma^2}}$$

$$\boldsymbol{g}(\boldsymbol{x}, t) \triangleq t f(\boldsymbol{x}) + (1-t)(x^2 + y^2)$$

$$\boldsymbol{g}(\boldsymbol{x}, t) \triangleq [f \star k_{\frac{1}{t}-1}](\boldsymbol{x})$$

Evolution Equation

- A formalism to express changes over time.
- Initial condition $g(\boldsymbol{x}, t = 0) = f(\boldsymbol{x})$.

Heat Equation

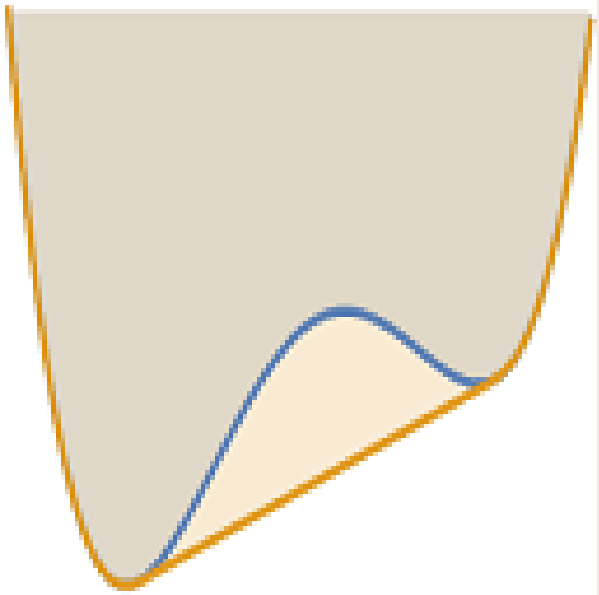
$$\frac{\partial}{\partial t} g = \Delta g$$

Schrodinger's Equation

$$\frac{\partial}{\partial t} g = i \Delta g$$

Choice of Simplified Optimization

Convex Envelope



- Any global minimizer of original function is also a global minimizer of convex envelope.
- Computation of convex envelope is generally intractable.

Convex Envelope by PDE

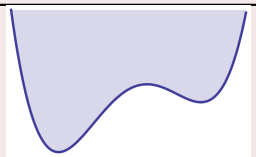
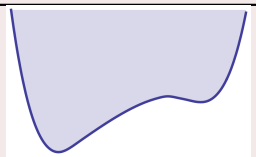
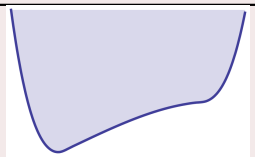
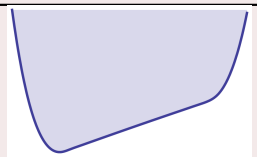
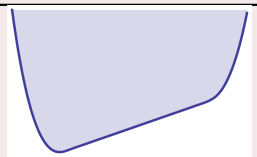
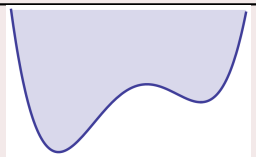
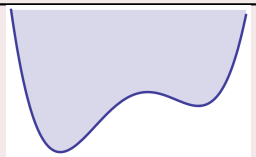
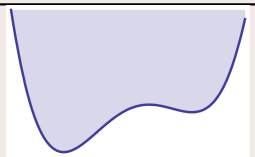
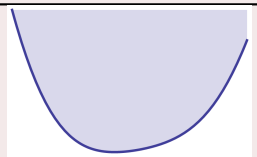
Vese's Equation

$$\frac{\partial}{\partial t} h = \sqrt{1 + \|\nabla h\|^2}\, \min\{0, \lambda_{\mathsf{min}}(\nabla^2 h)\} \quad , \quad \text{s.t. } h(\,.\,;0) = f(\,.\,)\,.$$

Heat Equation

Connection to Heat Equation [Mobahi14]

$$\begin{aligned} \frac{\partial}{\partial t} h &= \sqrt{1+\|\nabla h\|^2}\,\min\{0, \lambda_{\text{min}}(\nabla^2 h)\} \quad \text{s.t. } h(\,.\,;0) = f(\,.\,) \\ \approx \quad \frac{\partial}{\partial t} h &= \frac{1}{n+1}[\Delta h](\boldsymbol{x}) \quad \text{s.t. } h(\,.\,;0) = f(\,.\,) \end{aligned}$$

t	0	0.1	0.5	1	4
Original					
Linearized					

Diffusion of Ackley's Function

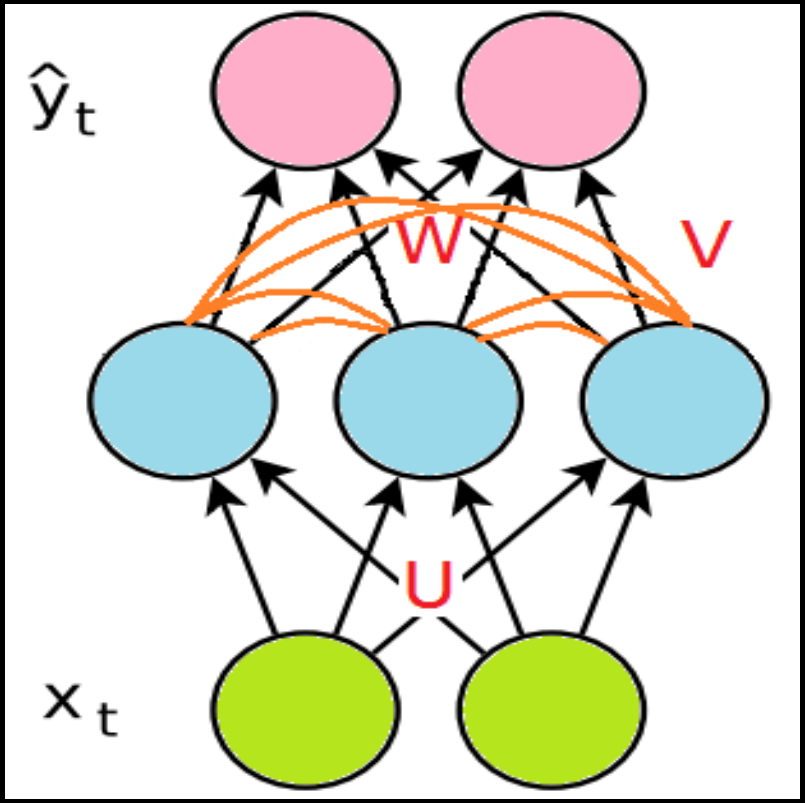
Solution to Heat Equation

Gaussian Convolution

It has closed form for some interesting function [Mobahi16], e.g. polynomials.

Original	Convolved
x	x
x^2	$\sigma^2 + x^2$
x^3	$3\sigma^2 x + x^3$
$\vdots$	$\vdots$
x^{10}	$945\sigma^{10} + 4725\sigma^8 x^2 + 3150\sigma^6 x^4 + 630\sigma^4 x^6 + 45\sigma^2 x^8 + x^{10}$

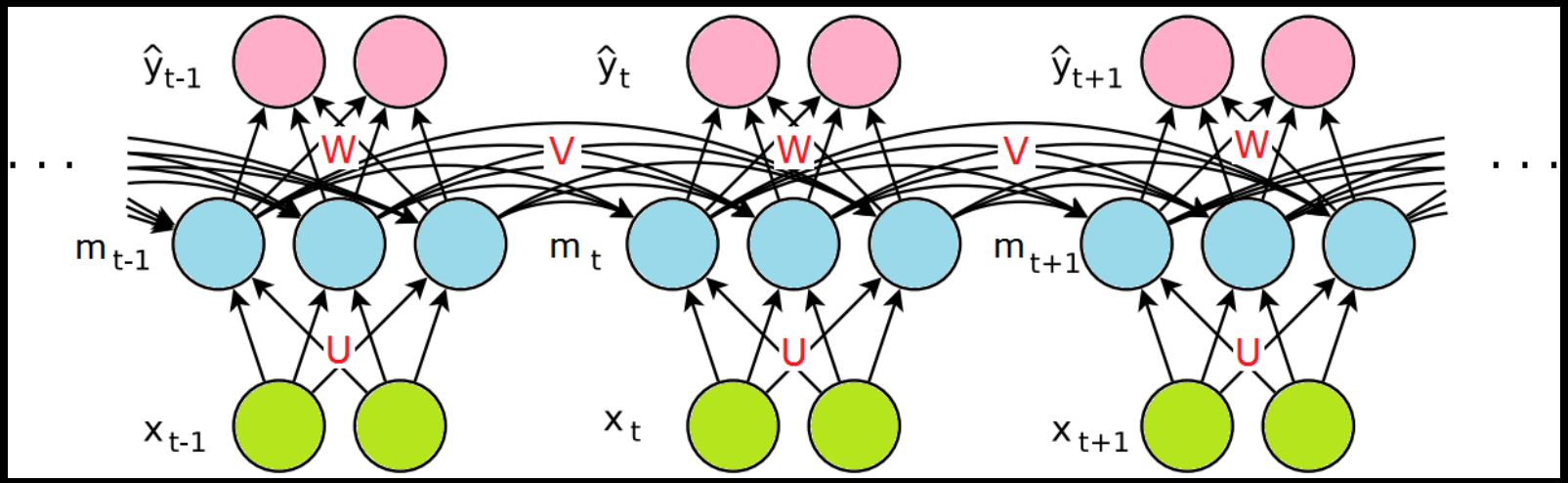
Recurrent Neural Networks



Dynamic

$$\text{for } t = 1, \ldots, T \quad \begin{cases} \hat{\boldsymbol{y}}_t & \triangleq h(\boldsymbol{n}_t) + \boldsymbol{b} \\ \boldsymbol{n}_t & \triangleq \boldsymbol{W}\, h(\boldsymbol{m}_t) \\ \boldsymbol{m}_t & \triangleq \boldsymbol{U}\boldsymbol{x}_t + \boldsymbol{V} h(\boldsymbol{m}_{t-1}) + \boldsymbol{a} \end{cases}$$

RNN Unfolded in Time



Learning

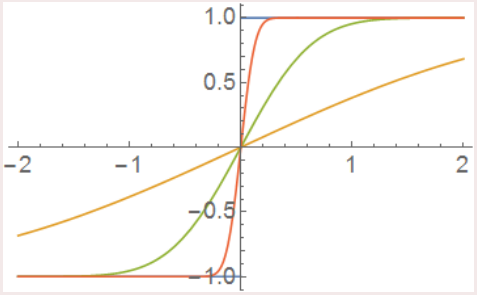
$$\min_{\boldsymbol{a},\boldsymbol{b},\boldsymbol{U},\boldsymbol{V},\boldsymbol{W}} \sum_{t=1}^{T} f(\hat{\boldsymbol{y}}_t - \boldsymbol{y}_t)$$

for $t = 1, \ldots, T$ $\quad \begin{cases} \hat{\boldsymbol{y}}_t & \triangleq h(\boldsymbol{n}_t) \\ \boldsymbol{n}_t & \triangleq \boldsymbol{W}\, h(\boldsymbol{m}_t) + \boldsymbol{b} \\ \boldsymbol{m}_t & \triangleq \boldsymbol{U}\boldsymbol{x}_t + \boldsymbol{V} h(\boldsymbol{m}_{t-1}) + \boldsymbol{a} \end{cases}$

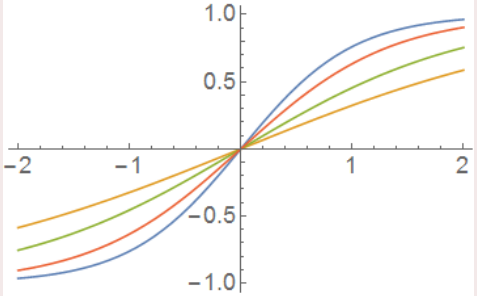
Diffused Cost

Has Closed Form Expression

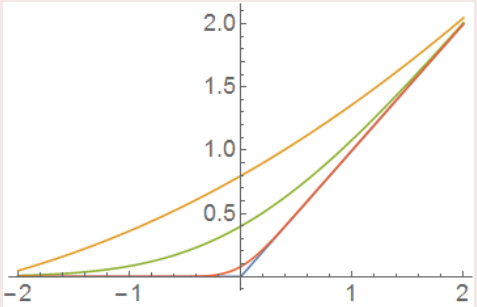
Name	Original	Diffused
Sign	$\mathrm{sign}(x)$	$\mathrm{erf}(\frac{x}{\sqrt{2}\sigma})$
Error	$\mathrm{erf}(ax)$	$\mathrm{erf}(\frac{ax}{\sqrt{1+2(a\sigma)^2}})$
Tanh	$\tanh(x)$	$\tanh(\frac{x}{\sqrt{1+\frac{\pi}{2}\sigma^2}})$
ReLU	$\max(0,x)$	$\frac{\sigma}{\sqrt{2\pi}}e^{-\frac{x^2}{2\sigma^2}}+\frac{1}{2}x\left(1+\mathrm{erf}(\frac{x}{\sqrt{2}\sigma})\right)$



$Sign$



$Tanh$



$ReLU$

Popular Training Gadgets

Emerge Spontaneously from Theory

- Annealing Learning Rate
- Careful Initialization
- Noise Injection (SGD, Dropout)
- Saturation Control (Batch Normalization)
- Preventing Exploding/Vanishing Gradient (LSTM)
- Layerwise Pretraining

Popular Training Gadgets

Emerge Spontaneously from Theory

- Annealing Learning Rate
- Careful Initialization
- Noise Injection (SGD, Dropout)
- Saturation Control (Batch Normalization)
- Preventing Exploding/Vanishing Gradient (LSTM)
- Layerwise Pretraining

Taylor's Remainder Theorem

$$\begin{aligned} \hat{f}(\boldsymbol{w}) &\triangleq f(\boldsymbol{w}_0) + (\boldsymbol{w} - \boldsymbol{w}_0)^T \nabla f(\boldsymbol{w}_0) \\ |\hat{f}(\boldsymbol{w}) - f(\boldsymbol{w})| &\leq \frac{1}{2}\|\boldsymbol{w} - \boldsymbol{w}_0\|^2 \Lambda \end{aligned}$$

Popular Training Gadgets

Emerge Spontaneously from Theory

- Annealing Learning Rate
- Careful Initialization
- Noise Injection (SGD, Dropout)
- Saturation Control (Batch Normalization)
- Preventing Exploding/Vanishing Gradient (LSTM)
- Layerwise Pretraining

Popular Training Gadgets

Emerge Spontaneously from Theory

- Annealing Learning Rate
- Careful Initialization
- Noise Injection (SGD, Dropout)
- Saturation Control (Batch Normalization)
- Preventing Exploding/Vanishing Gradient (LSTM)
- Layerwise Pretraining

MCMC Integration

$$(f \star k_\sigma)[\boldsymbol{w}] = \int_{\mathbb{R}^n} f(\boldsymbol{w} - \boldsymbol{t}) k_\sigma(\boldsymbol{t}) \approx \frac{1}{N} \sum_{i=1}^{N} f(\boldsymbol{w} - \boldsymbol{t}_i) \text{ s.t. } \boldsymbol{t}_i \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$$

Popular Training Gadgets

Emerge Spontaneously from Theory

- Annealing Learning Rate
- Careful Initialization
- Noise Injection (SGD, Dropout)
- Saturation Control (Batch Normalization)
- Preventing Exploding/Vanishing Gradient (LSTM)
- Layerwise Pretraining

Input Normalization

Example: squashing function $f(y) \triangleq \mathrm{sign}(y)$:

$$[\mathrm{sign}(\square^T \boldsymbol{x}) \star k_\sigma(\square)](\boldsymbol{w}) = \mathrm{erf}(\frac{\boldsymbol{w}^T \boldsymbol{x}}{\sqrt{2}\sigma\|\boldsymbol{x}\|})$$

Popular Training Gadgets

Emerge Spontaneously from Theory

- Annealing Learning Rate
- Careful Initialization
- Noise Injection (SGD, Dropout)
- Saturation Control (Batch Normalization)
- Preventing Exploding/Vanishing Gradient (LSTM)
- Layerwise Pretraining

High Orders Disappear

Recursive updates... expressions like:

$$\boldsymbol{I} + \boldsymbol{V}\,\mathrm{diag}\left(\widetilde{h}'_\sigma(\boldsymbol{m}_t)\right) + \boldsymbol{V}^2\,\mathrm{diag}\left(\widetilde{h}'_\sigma(\boldsymbol{m}_t) \bullet \widetilde{h}'_\sigma(\boldsymbol{m}_{t-1})\right) + \ldots$$

When $\sigma \to \infty$

$$\boldsymbol{I} + \boldsymbol{V}\epsilon + \boldsymbol{V}^2\epsilon^2 + \ldots$$

Popular Training Gadgets

Emerge Spontaneously from Theory

- Annealing Learning Rate
- Careful Initialization
- Noise Injection (SGD, Dropout)
- Saturation Control (Batch Normalization)
- Preventing Exploding/Vanishing Gradient (LSTM)
- Layerwise Pretraining

High Orders Disappear

Recursive updates... expressions like:

$$\boldsymbol{I} + \boldsymbol{V}\,\mathrm{diag}\left(\widetilde{h}'_\sigma(\boldsymbol{m}_t)\right) + \boldsymbol{V}^2\,\mathrm{diag}\left(\widetilde{h}'_\sigma(\boldsymbol{m}_t) \bullet \widetilde{h}'_\sigma(\boldsymbol{m}_{t-1})\right) + \ldots$$

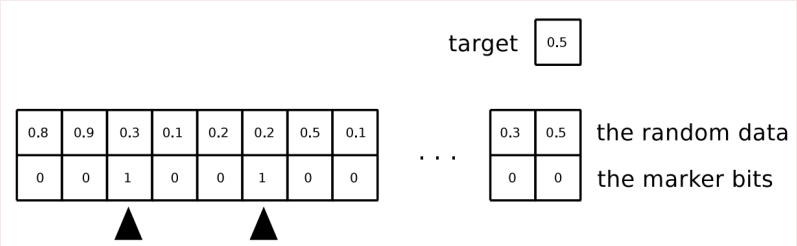
When $\sigma \to \infty$

$$\boldsymbol{I} + \boldsymbol{V}\epsilon + \boldsymbol{V}^2\epsilon^2 + \ldots$$

Ignores time dependencies at first, and then gradually reconsiders them.

Example

Learning to Add



Result

SGD Hyperparameters (Annealed Learning Rate, Mini Batch Size, Momentum Coefficient) carefully searched.

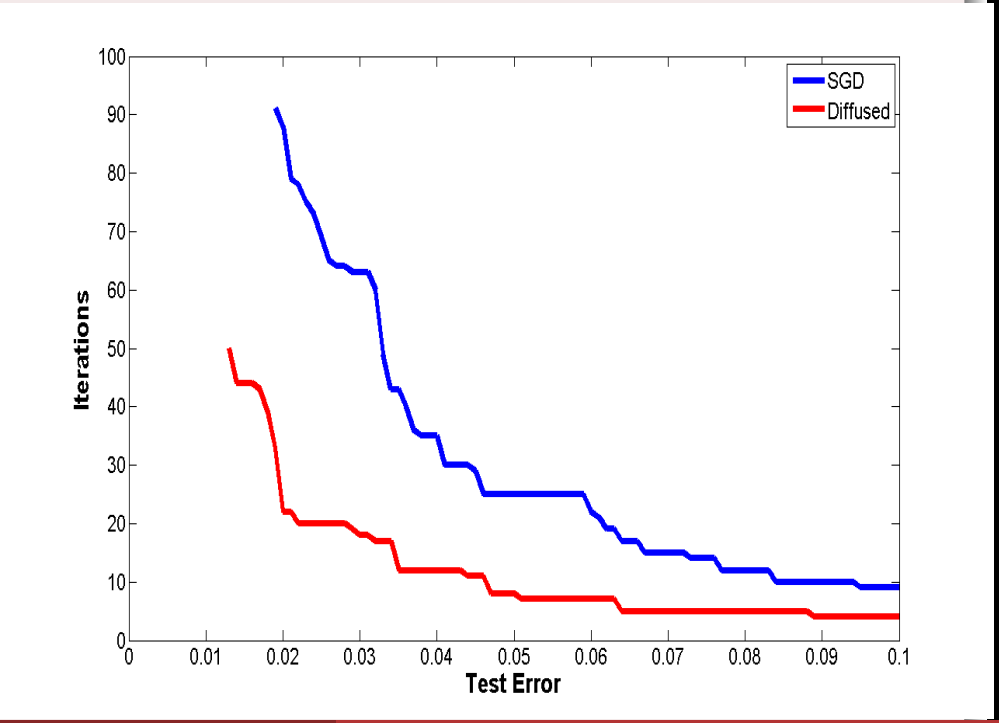


Image Alignment

Alignment as Energy Minimization

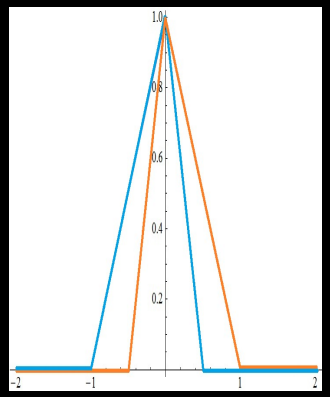
$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \int_{\mathcal{X}} \Big(f_1(\boldsymbol{\tau}(\boldsymbol{x};\boldsymbol{\theta})) - f_2(\boldsymbol{x}) \Big)^2 d\boldsymbol{x}$$

This is non-convex in variable $\boldsymbol{\theta}$.

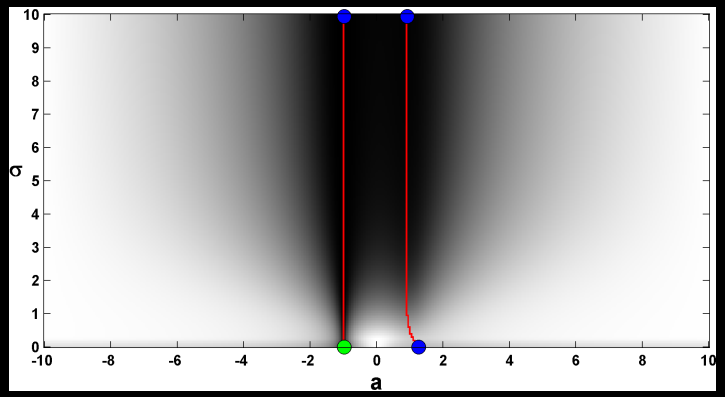
Smoothing the Objective for Alignment

1-d Scale Alignment

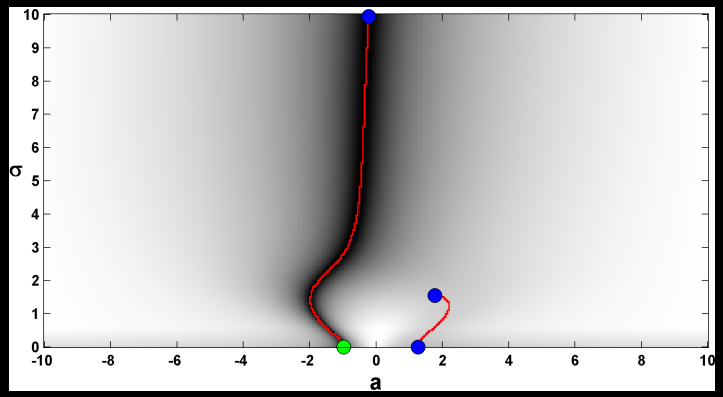
$$\min_a \int_{\mathbb{R}} (f_1(ax) - f_2(x))^2 dx$$



Signals



Signal Smoothing



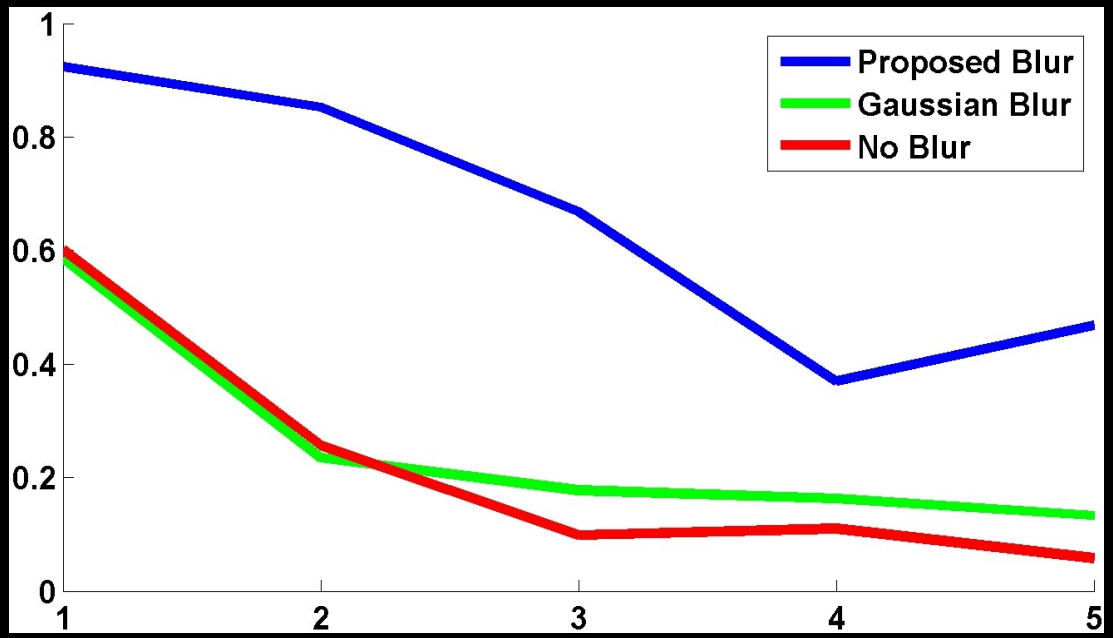
Objective Smoothing

Fovea Effect

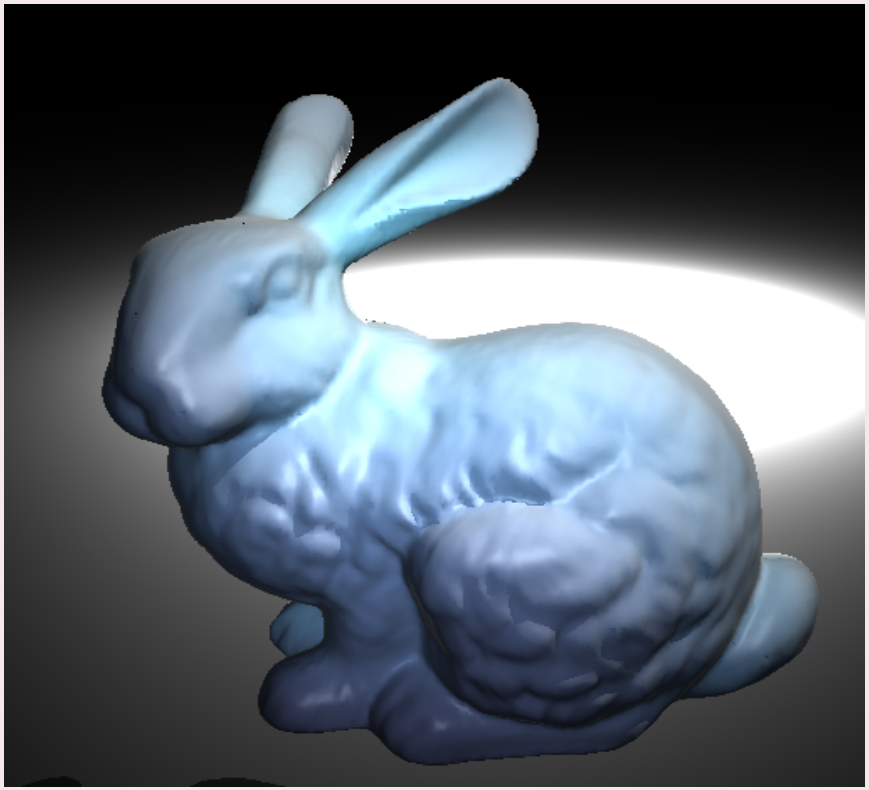
Equivalent Blur Operator

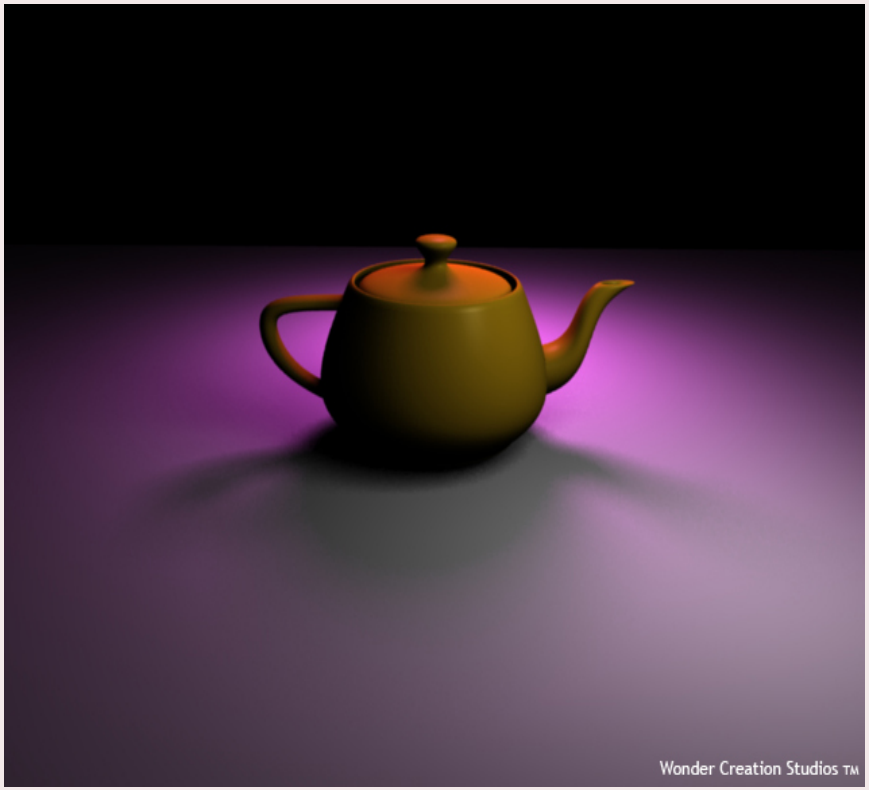
$$
\begin{aligned}
& [f(\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b}) \star k_\sigma](\boldsymbol{\theta}) \\
= & \int_{\mathcal{X}} f(\boldsymbol{y})\, k(\boldsymbol{A}\boldsymbol{x}+\boldsymbol{b}-\boldsymbol{y}; \sigma^2(1+\|\boldsymbol{x}\|^2))\, d\boldsymbol{y}
\end{aligned}
$$

Alignment Results [Mobahi12]



3D Point Cloud Alignment



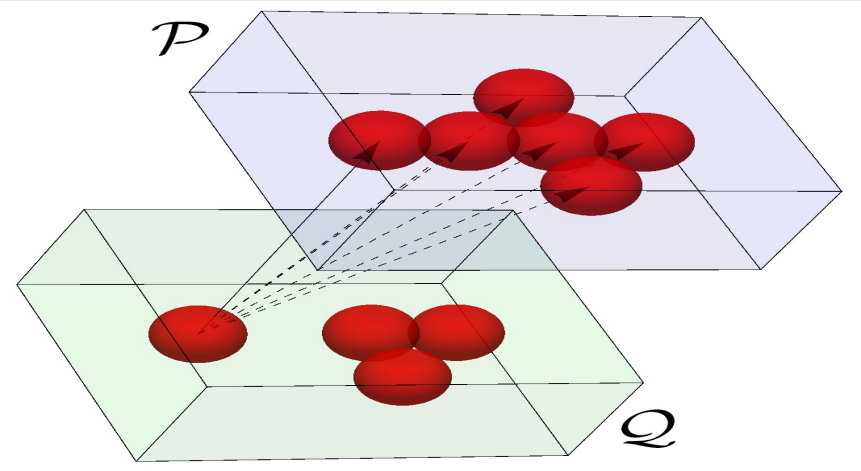


3D Point Cloud Alignment

Formulation

$$\min_{\boldsymbol{A},\boldsymbol{b},\boldsymbol{c}} \quad \sum_{i,j} c_{i,j} \|\boldsymbol{A}\boldsymbol{p}_i + \boldsymbol{b} - \boldsymbol{q}_j\|^2$$

$$\text{s.t.} \quad \sum_{i} c_{i,j} = 1 \quad , \quad c_{i,j} \in \{0,1\}$$

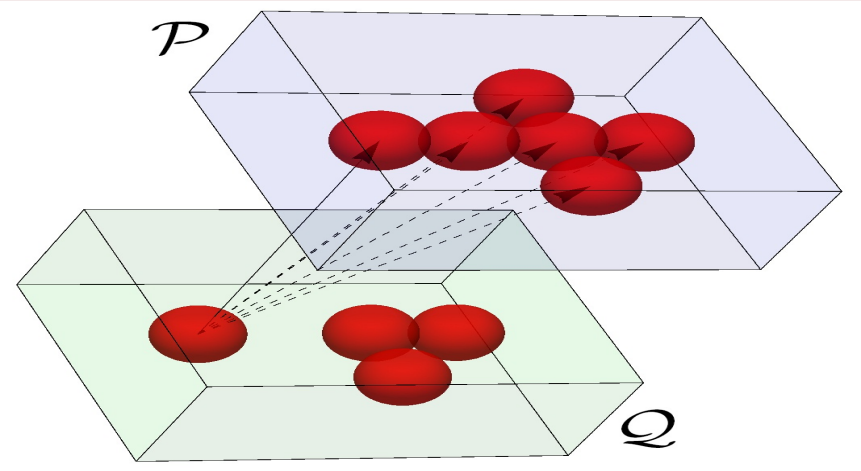


3D Point Cloud Alignment

Formulation

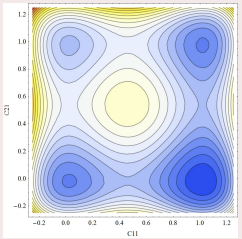
$$\min_{\boldsymbol{A},\boldsymbol{b},\boldsymbol{c}} \quad \sum_{i,j} c_{i,j} \|\boldsymbol{A}\boldsymbol{p}_i + \boldsymbol{b} - \boldsymbol{q}_j\|^2$$

$$\text{s.t.} \quad \sum_{i} c_{i,j} = 1 \quad , \quad c_{i,j}(1 - c_{i,j}) = 0$$

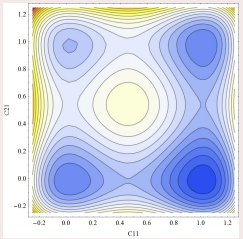


3D Point Cloud Alignment

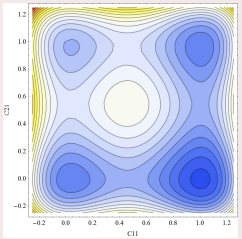
Optimization Landscape of Smoothed Objective



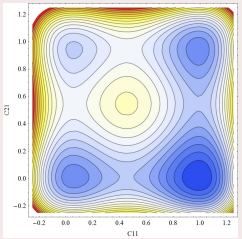
$\sigma = 0$



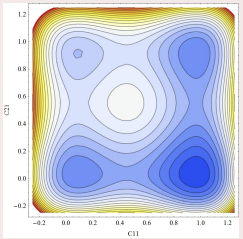
$\sigma = \frac{3}{88}$



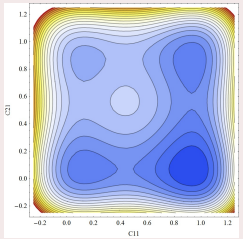
$\sigma = \frac{6}{88}$



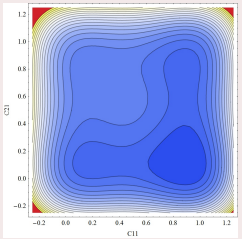
$\sigma = \frac{9}{88}$



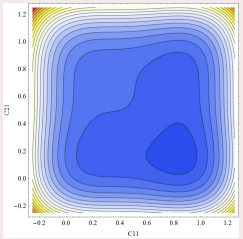
$\sigma = \frac{12}{88}$



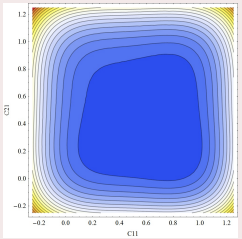
$\sigma = \frac{15}{88}$



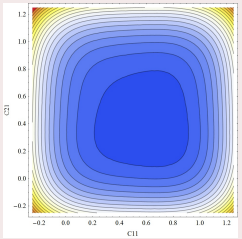
$\sigma = \frac{27}{88}$



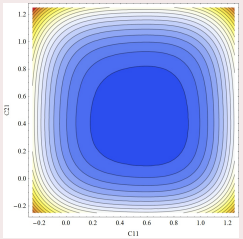
$\sigma = \frac{30}{88}$



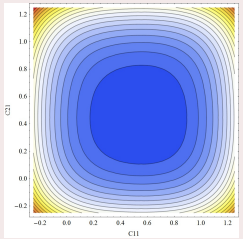
$\sigma = \frac{33}{88}$



$\sigma = \frac{18}{88}$



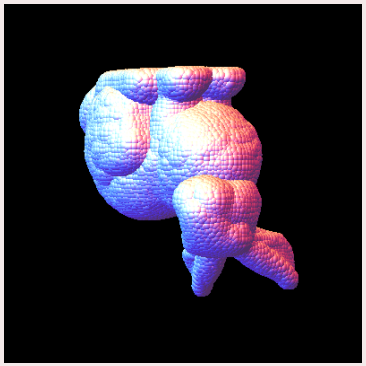
$\sigma = \frac{21}{88}$



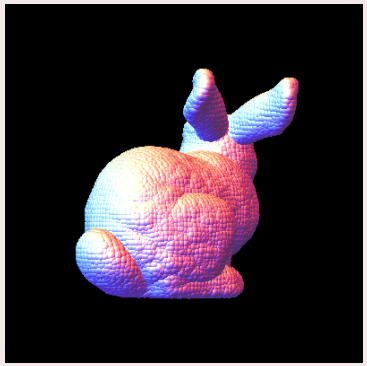
$\sigma = \frac{24}{88}$

Stanford 3D Point Cloud Dataset

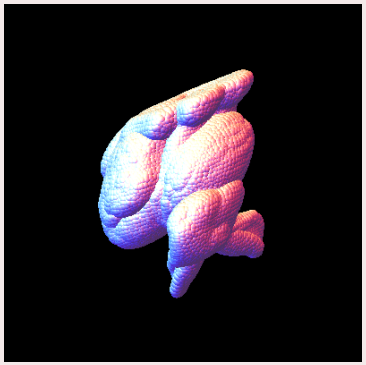
Transforming $\mathcal{P}$ to $\mathcal{Q}$



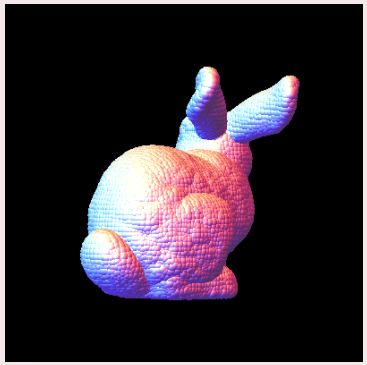
$\mathcal{P}$



$\mathcal{Q}$



ICP



Ours

Conclusion

- New theory for optimization by continuation presented.
- Efficient algorithms based on closed form expressions were discussed.
- Future Directions:
  - Investigating other applications (NLP, Robotics, Etc.).
  - Improving approximations used in deep learning application.
  - Improving the approximation quality of Vese's equation (E.g. by nonlinear PDEs with closed form, combination of analytical and numerical methods).

Acknowledgment

- John W. Fisher III (MIT)
- William T. Freeman (MIT)
- Yi Ma (UIUC/ShanghaiTech)
- Stefano Soatto (UCLA)
- Anima Anandkumar (UCI)
- Alan L. Yuille (UCLA)
- Rong Ge (Duke)
- John Wright (Columbia)