Early Stopping is Nonparametric Variational Inference



Dougal Maclaurin, David Duvenaud, Ryan Adams

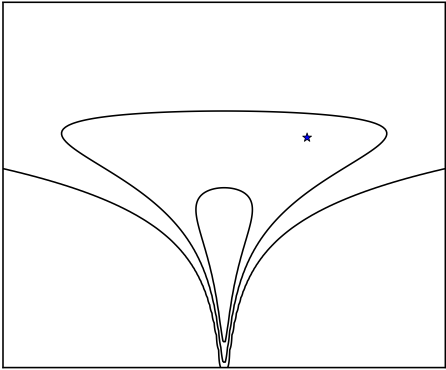
HARVARD
School of Engineering
and Applied Sciences

Inference is moving to stochastic optimization

- First: Full-batch MCMC
- Then: Variational Bayes (optimization)
- Then: Stochastic variational inference (minibatches)
- Then: SVI for deep GPs (neural networks)
- Looks like training a (Bayesian) neural net by SGD
- What's next?

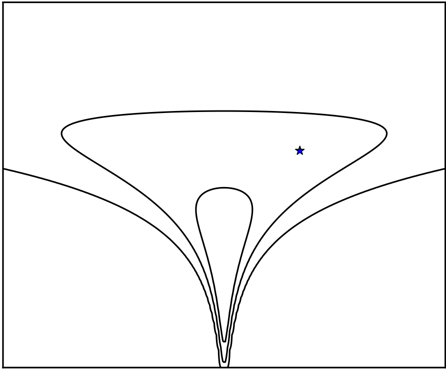
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



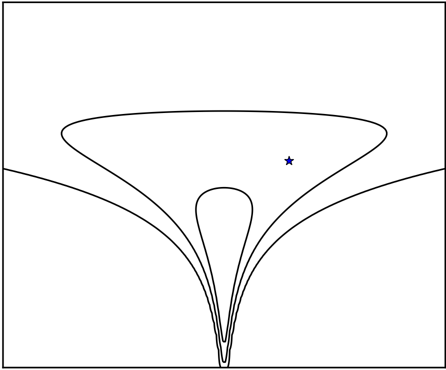
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



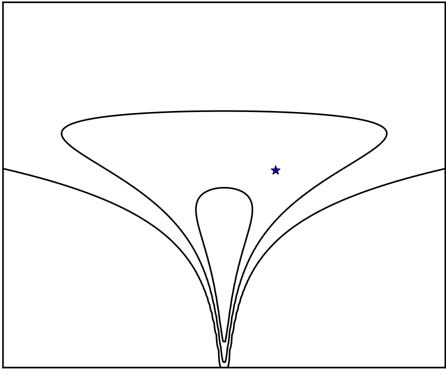
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



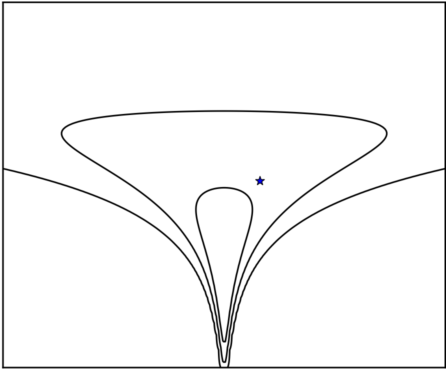
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



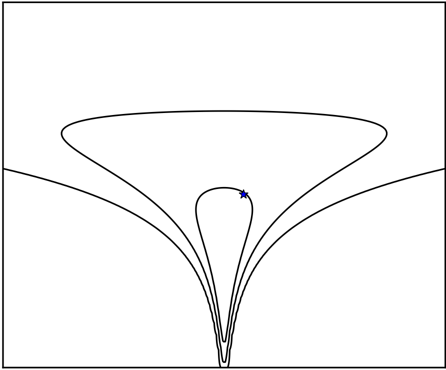
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



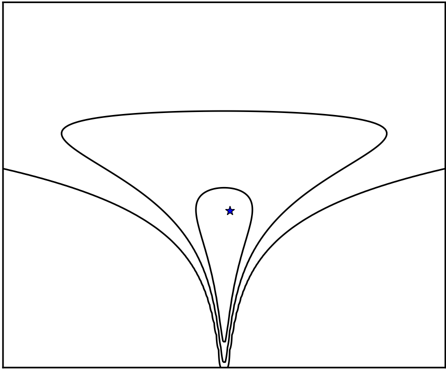
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



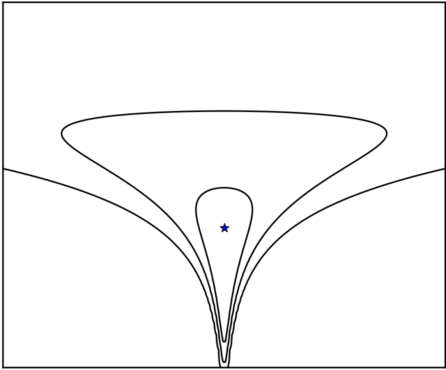
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



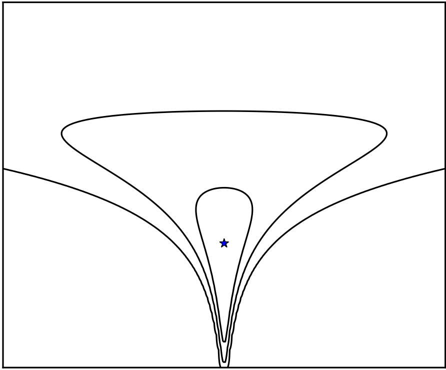
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



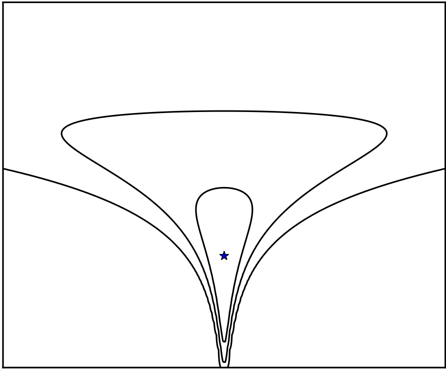
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



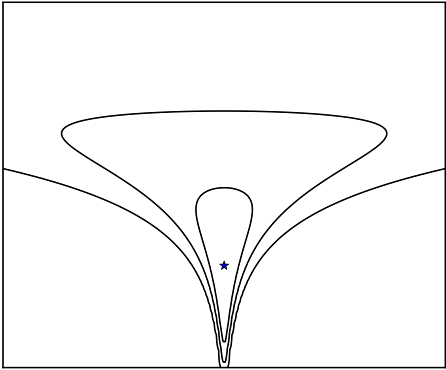
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



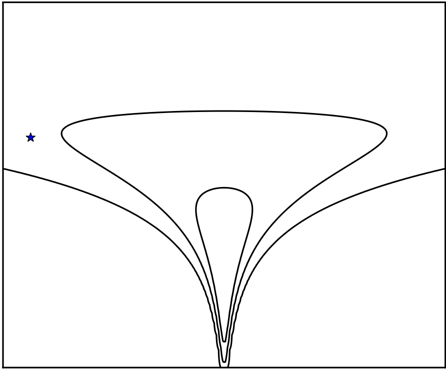
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



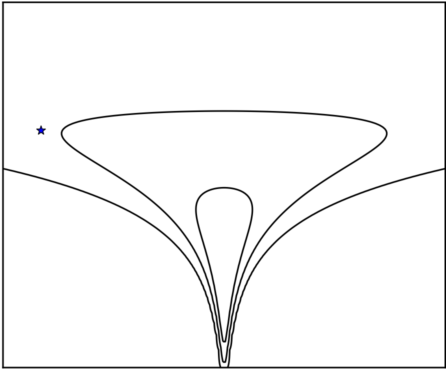
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



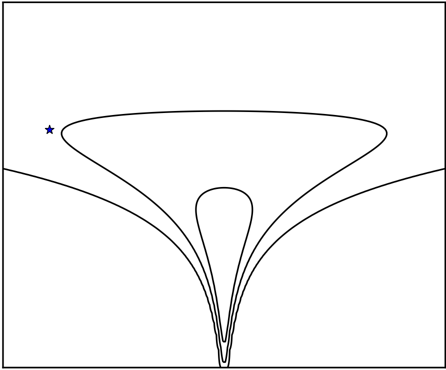
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



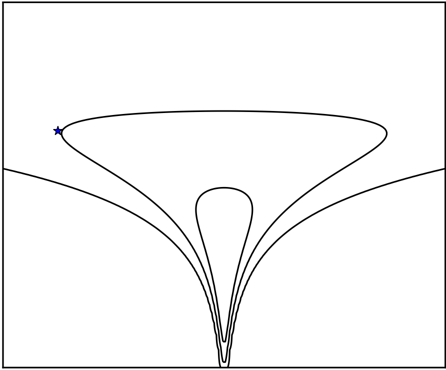
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



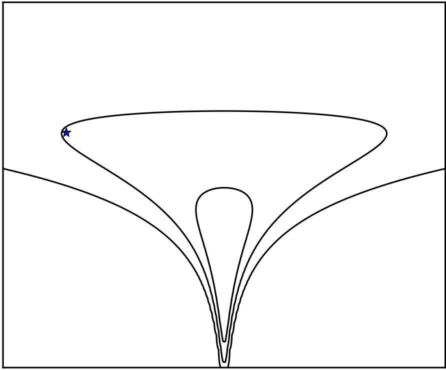
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



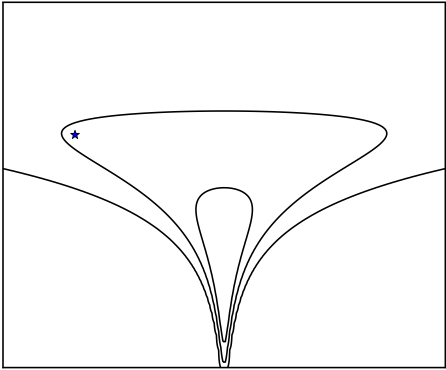
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



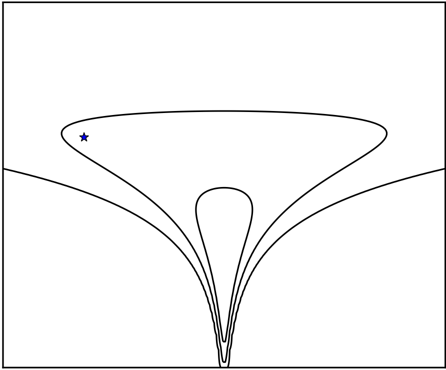
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



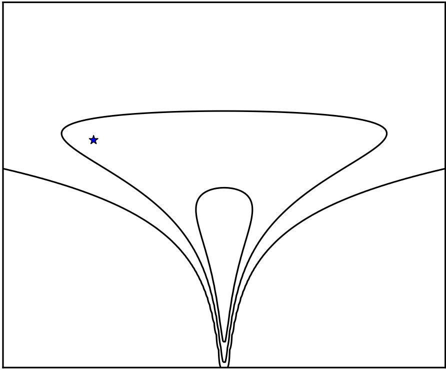
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



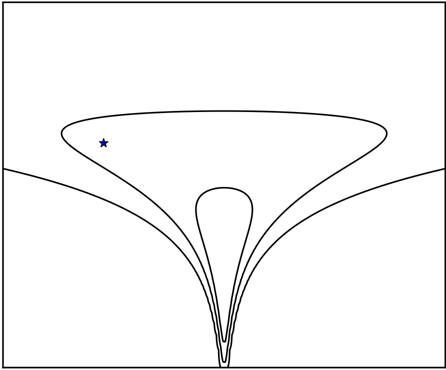
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



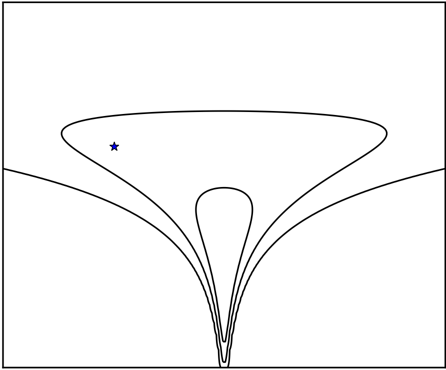
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



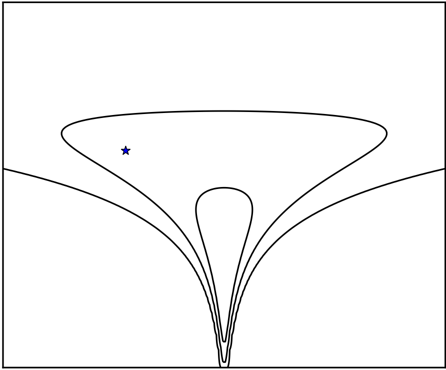
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



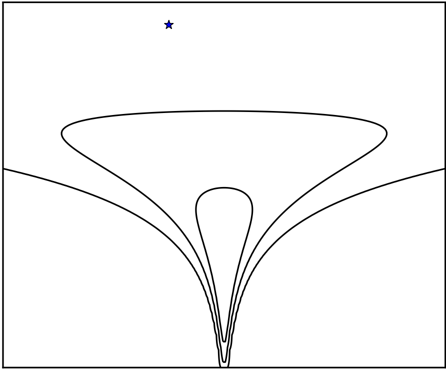
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



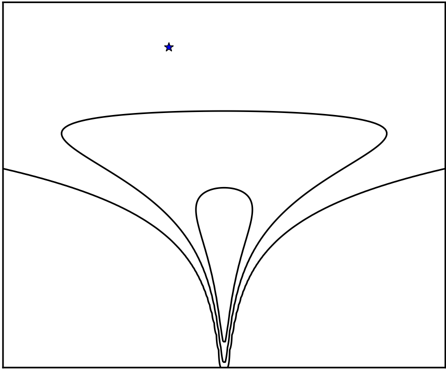
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



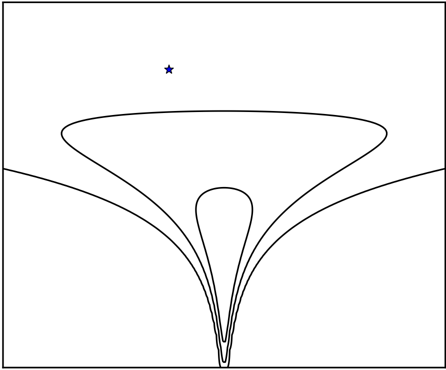
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



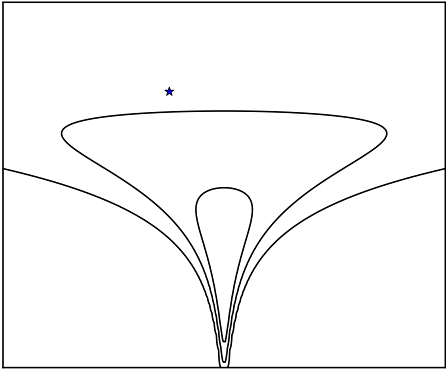
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



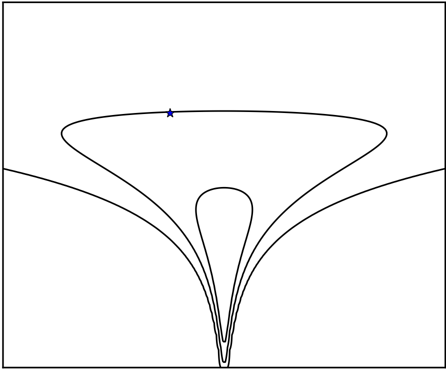
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



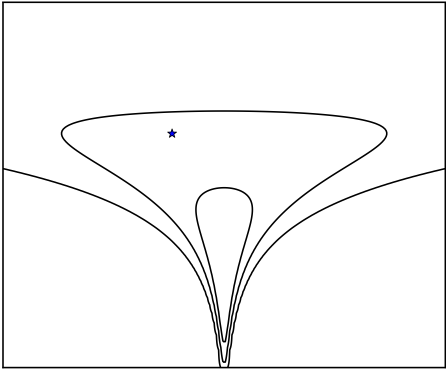
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



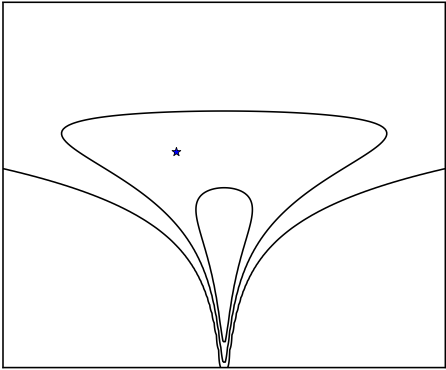
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



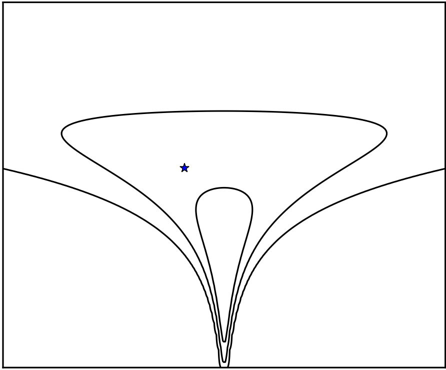
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



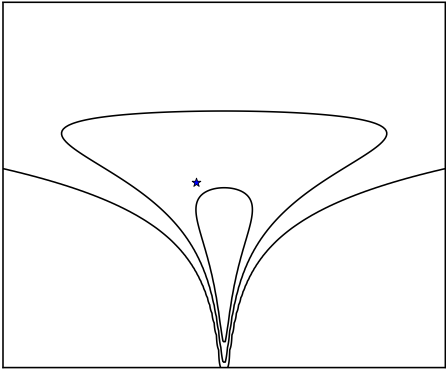
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



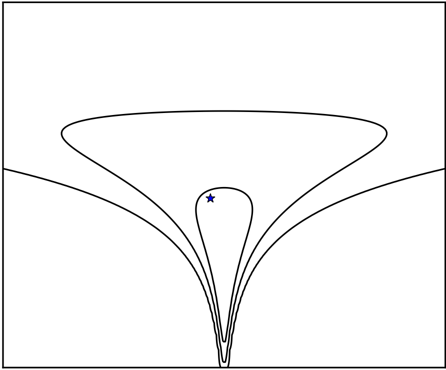
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



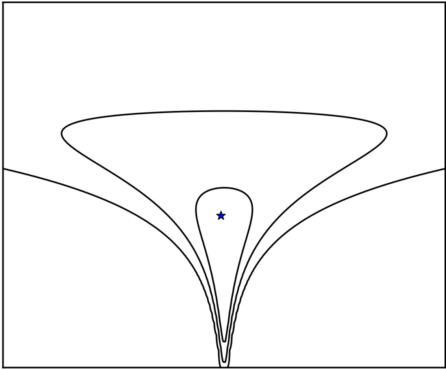
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



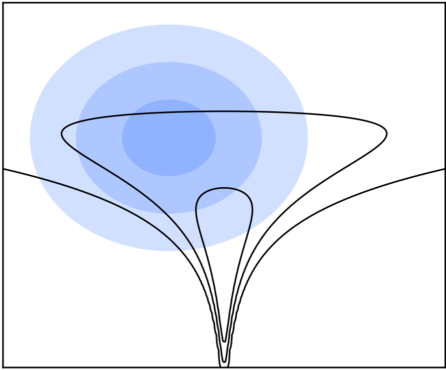
Gradient Descent as Inference

- Optimization paths start from random init, and converges to modes...



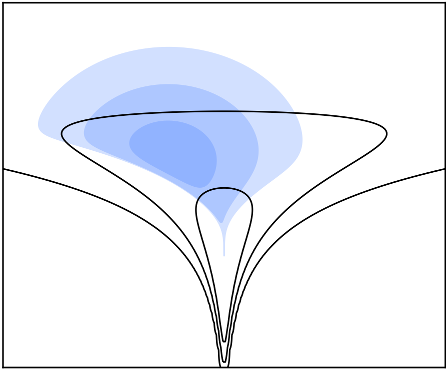
Implicit Distributions

- What about the implicit distribution of parameters after optimizing for t steps?
- Starts as a bad approximation (prior dist)
- Ends as a bad approximation (point mass)
- Choosing best intermediate dist $=$ early stopping
- Taking multiple samples from dist $=$ ensembling



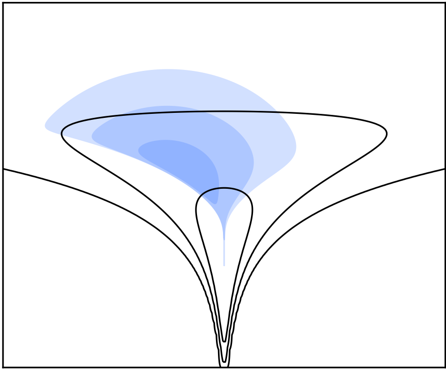
Implicit Distributions

- What about the implicit distribution of parameters after optimizing for t steps?
- Starts as a bad approximation (prior dist)
- Ends as a bad approximation (point mass)
- Choosing best intermediate dist $=$ early stopping
- Taking multiple samples from dist $=$ ensembling



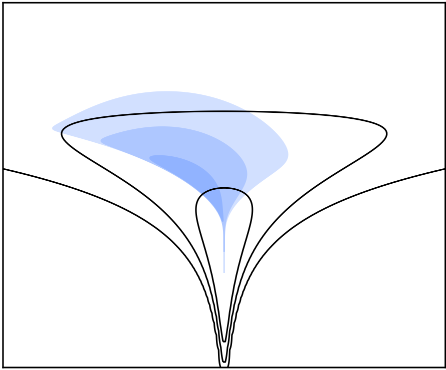
Implicit Distributions

- What about the implicit distribution of parameters after optimizing for t steps?
- Starts as a bad approximation (prior dist)
- Ends as a bad approximation (point mass)
- Choosing best intermediate dist $=$ early stopping
- Taking multiple samples from dist $=$ ensembling



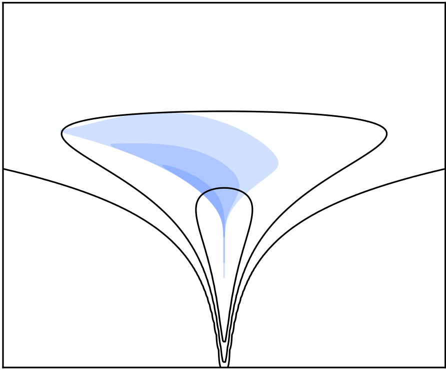
Implicit Distributions

- What about the implicit distribution of parameters after optimizing for t steps?
- Starts as a bad approximation (prior dist)
- Ends as a bad approximation (point mass)
- Choosing best intermediate dist $=$ early stopping
- Taking multiple samples from dist $=$ ensembling



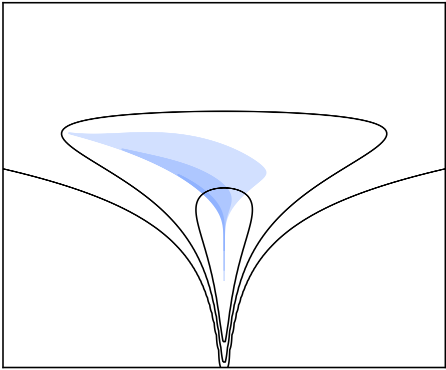
Implicit Distributions

- What about the implicit distribution of parameters after optimizing for t steps?
- Starts as a bad approximation (prior dist)
- Ends as a bad approximation (point mass)
- Choosing best intermediate dist $=$ early stopping
- Taking multiple samples from dist $=$ ensembling



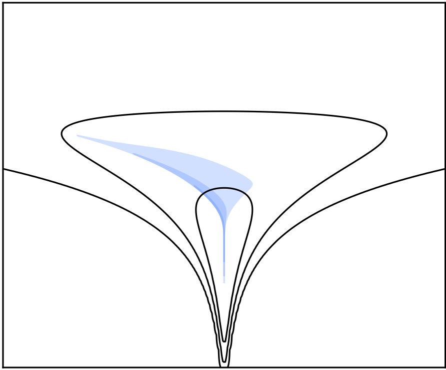
Implicit Distributions

- What about the implicit distribution of parameters after optimizing for t steps?
- Starts as a bad approximation (prior dist)
- Ends as a bad approximation (point mass)
- Choosing best intermediate dist $=$ early stopping
- Taking multiple samples from dist $=$ ensembling



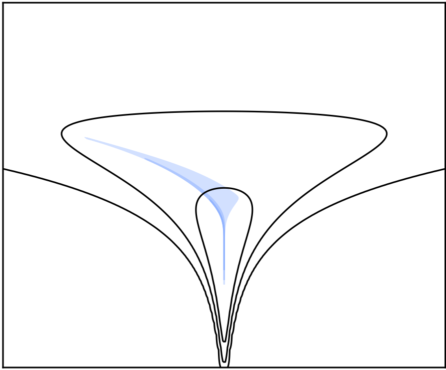
Implicit Distributions

- What about the implicit distribution of parameters after optimizing for t steps?
- Starts as a bad approximation (prior dist)
- Ends as a bad approximation (point mass)
- Choosing best intermediate dist $=$ early stopping
- Taking multiple samples from dist $=$ ensembling



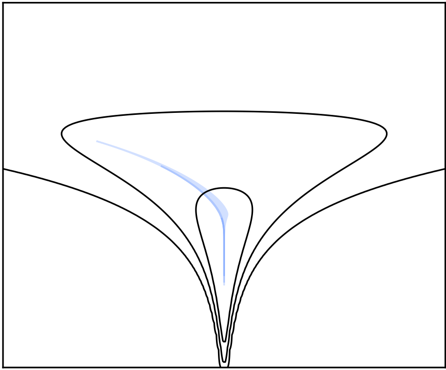
Implicit Distributions

- What about the implicit distribution of parameters after optimizing for t steps?
- Starts as a bad approximation (prior dist)
- Ends as a bad approximation (point mass)
- Choosing best intermediate dist $=$ early stopping
- Taking multiple samples from dist $=$ ensembling



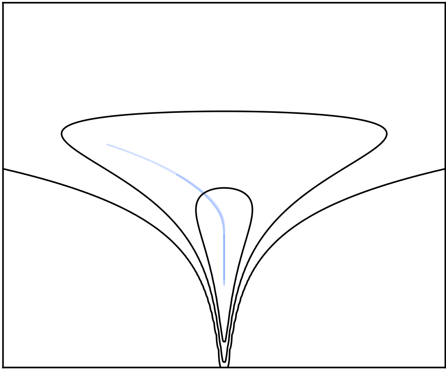
Implicit Distributions

- What about the implicit distribution of parameters after optimizing for t steps?
- Starts as a bad approximation (prior dist)
- Ends as a bad approximation (point mass)
- Choosing best intermediate dist $=$ early stopping
- Taking multiple samples from dist $=$ ensembling



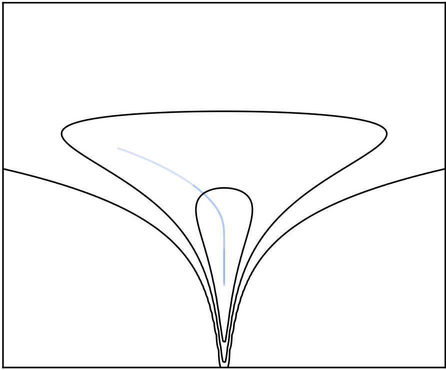
Implicit Distributions

- What about the implicit distribution of parameters after optimizing for t steps?
- Starts as a bad approximation (prior dist)
- Ends as a bad approximation (point mass)
- Choosing best intermediate dist $=$ early stopping
- Taking multiple samples from dist $=$ ensembling



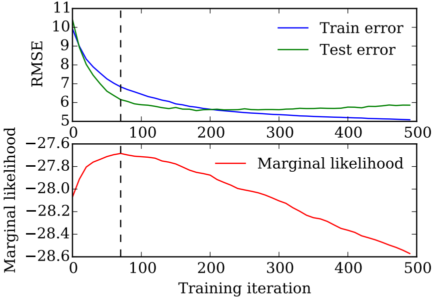
Implicit Distributions

- What about the implicit distribution of parameters after optimizing for t steps?
- Starts as a bad approximation (prior dist)
- Ends as a bad approximation (point mass)
- Choosing best intermediate dist $=$ early stopping
- Taking multiple samples from dist $=$ ensembling



Cross Validation vs Marginal Likelihood

- Currently, hyperparameters chosen by cross-validation.
- What if we could evaluate marginal likelihood of implicit distribution?
- Could choose all hypers to maximize marginal likelihood
- No need for validation set?



Variational Lower Bound

$$\log p(\mathbf{x}) \geq -\underbrace{\mathbb{E}_{q(\theta)}\left[-\log p(\theta, \mathbf{x})\right]}_{\text{Energy } E[q]} \quad \underbrace{-\mathbb{E}_{q(\theta)}\left[\log q(\theta)\right]}_{\text{Entropy } S[q]}$$

Likelihood estimated from optimized objective function:

$$\mathbb{E}_{q(\theta)}\left[-\log p(\theta, \mathbf{x})\right] \approx \log p(\hat{\theta}_T, \mathbf{x})$$

Entropy estimated by tracking change at each iteration:

$$-\mathbb{E}_{q(\theta)}\left[\log q(\theta)\right] \approx S[q_0] + \sum_{t=0}^{T-1} \log |J(\theta_t)|$$

Using a single sample sometimes OK in high dimensions

Estimating Change in Entropy

Volume change given by Jacobian of optimizer's operator:

$$S[q_{t+1}] - S[q_t] = \mathbb{E}_{q_t(\theta_t)}\left[\log\left|J(\theta_t)\right|\right]$$

Gradient descent update rule:

$$\theta_{t+1} = \theta_t - \alpha\nabla L(\theta),$$

Has Jacobian:

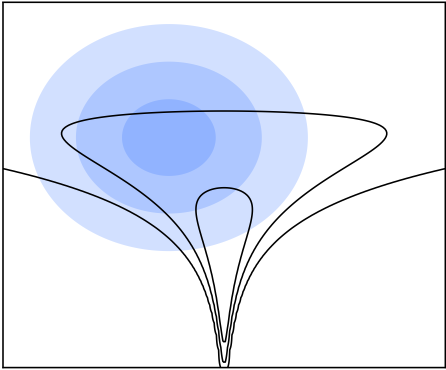
$$J(\theta_t) = I - \alpha\nabla\nabla L(\theta_t)$$

Entropy change estimate given by:

$$S[q_{t+1}] - S[q_t] \approx \log|I - \alpha\nabla\nabla L(\theta_t)|$$

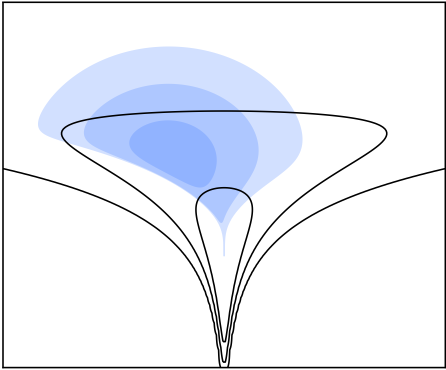
Estimating Change in Entropy

- Inuitively: High curvature makes entropy decrease
- Approximation good for small step-sizes



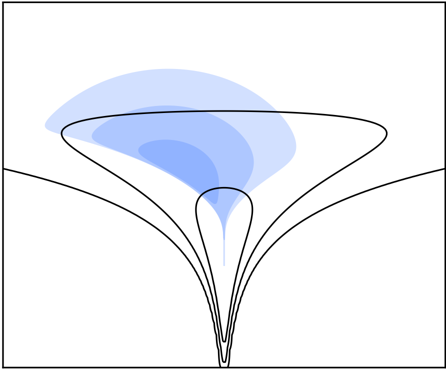
Estimating Change in Entropy

- Inuitively: High curvature makes entropy decrease
- Approximation good for small step-sizes



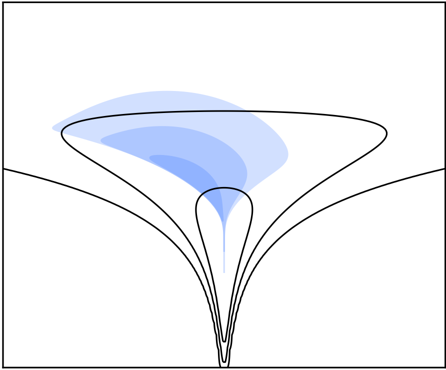
Estimating Change in Entropy

- Inuitively: High curvature makes entropy decrease
- Approximation good for small step-sizes



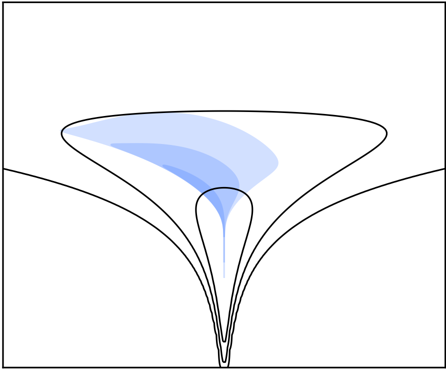
Estimating Change in Entropy

- Inuitively: High curvature makes entropy decrease
- Approximation good for small step-sizes



Estimating Change in Entropy

- Inuitively: High curvature makes entropy decrease
- Approximation good for small step-sizes



Final Algorithm

Stochastic Gradient Descent

input: Weight init scale σ_0, step size α, negative log-likelihood $L(\theta, t)$
initialize $\theta_0 \sim \mathcal{N}(0, \sigma_0 \mathbf{I}_D)$
for $t = 1$ **to** T **do**
  $\theta_t = \theta_{t-1} - \alpha \nabla L(\theta_t, t)$
output sample θ_T,

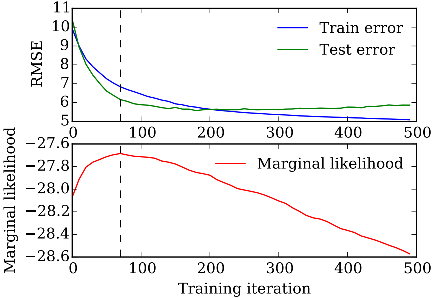
SGD with Entropy Estimate

input: Weight init scale σ_0, step size α, negative log-likelihood $L(\theta, t)$
initialize $\theta_0 \sim \mathcal{N}(0, \sigma_0 \mathbf{I}_D)$
initialize $S_0 = \frac{D}{2}(1 + \log 2\pi) + D \log \sigma_0$
for $t = 1$ **to** T **do**
  $S_t = S_{t-1} + \log |\mathbf{I} - \alpha H_{t-1}|$
  $\theta_t = \theta_{t-1} - \alpha \nabla L(\theta_t, t)$
output sample θ_T, entropy estimate S_T

- Add entropy to likelihood to get lower bound estimate
- Efficient implementation uses Hessian-vector products
- Scales linearly in parameters and dataset size

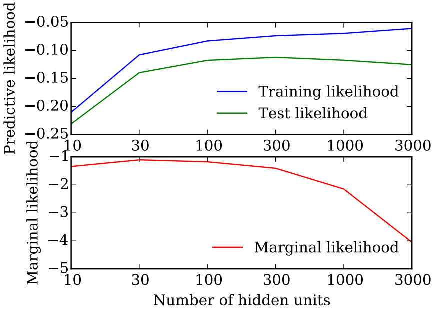
Experiments: Early Stopping

- Top: Training and test-set error on the Boston housing dataset.
- Bottom: SGD marginal likelihood estimates.



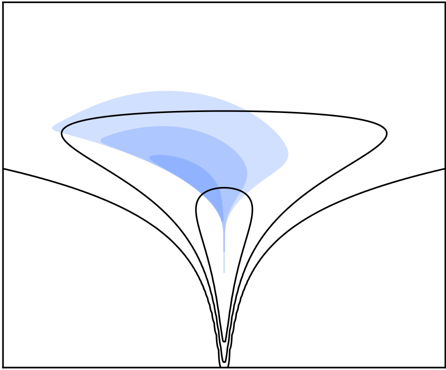
Experiments: Number of Hidden Units

- Top: Likelihood vs hidden units on MNIST
- Largest model has 2 million params
- Bottom: SGD marginal likelihood estimates
- Inter-sample variance is surprisingly low



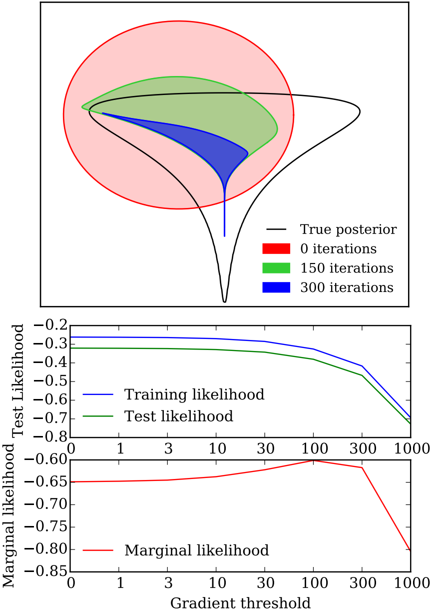
Limitations

- Entropy term gets arbitrarily bad due to concentration, but true performance only gets as bad as MLE
- Irrelevant parameters can cause low entropy estimate

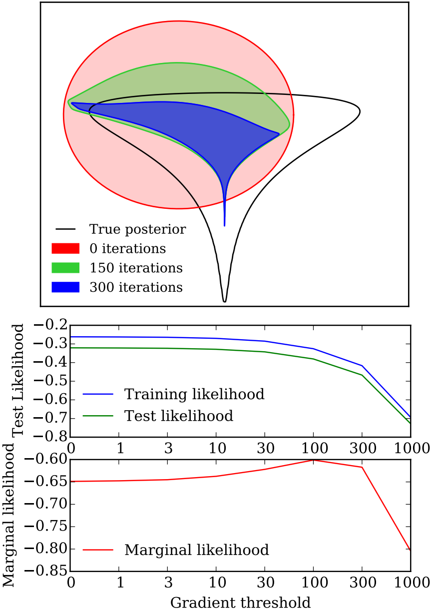


Experiments: Entropy-friendly Optimization

- Modified SGD to move slower near convergence
- Hurts performance, but gives higher bound estimate



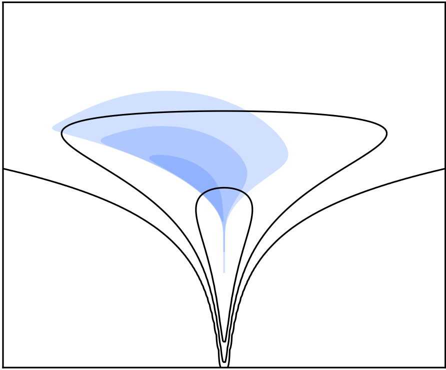
Experiments: Entropy-friendly Optimization



- Modified SGD to move slower near convergence
- Hurts performance, but gives higher bound estimate

More Limitations

- Entropy term gets arbitrarily bad due to concentration, but true performance only gets as bad as MLE
- Irrelevant parameters can cause low entropy estimate



Main Takeaways

- Optimization with random restarts implies nonparametric intermediate distributions
- Early stopping chooses among these distributions, ensembling samples from them
- Can scalably estimate lower bound on model evidence during optimization

Thanks!