### **Gradient Descent**

#### Nicolas Le Roux

#### Optimiza

Pooloo

Approximations to Newto method

Stochastic Optimization

Learning (Bottor

#### TONG

latural Gradien

Online Natural Gradier

Results

# Using Gradient Descent for Optimization and Learning

Nicolas Le Roux

15 May 2009

#### Ontimization

Basi

method

Stochastic Optimization

Learning (Bottou

#### **TONGA**

Natural Gradier

Online Natural Gradient

Results

# Why having a good optimizer?

- plenty of data available everywhere
- extract information efficiently

Approximations to Newton

Natural Gradient

Online Natural Gradient

# Why gradient descent?

- cheap
- suitable for large models
- it works

#### . . . .

Basic

method

Otooriaatio Optimization

Learning (Botto

#### TONG

Natural Gradie

Online Natural Gradient

Online Natural Gradie

## Optimization vs Learning

## Optimization

- function f to minimize
- time  $T(\rho)$  to reach error level  $\rho$

method

Learning (Botto

#### TONG

Natural Gradie

Online Natural Gradier

# Optimization vs Learning

## Optimization

- function f to minimize
- time  $T(\rho)$  to reach error level  $\rho$

## Learning

- measure of quality f (cost function)
- get training samples  $x_1, \ldots, x_n$  from p
- choose a model  $\mathcal F$  with parameters  $\theta$
- minimize  $E_p[f(\theta, x)]$

### Optimization

Basics

method

-----

Learning (Botto

#### TONG

Natural Gradie

Online Natural Gradier

Results

## **Outline**

Optimization

**Basics** 

Approximations to Newton method Stochastic Optimization

- 2 Learning (Bottou)
- 3 TONGA

Natural Gradient
Online Natural Gradient
Results

Stochastic Optimization

Learning (Botto

#### TONG

Vatural Gradie

Online Natural Gradie

D 1:

## The basics

Taylor expansion of *f* to the first order:

$$f(\theta + \varepsilon) = f(\theta) + \varepsilon^{\mathsf{T}} \nabla_{\theta} f + o(\|\varepsilon\|)$$

Best improvement obtained with

$$\varepsilon = -\eta \nabla_{\theta} f \qquad \eta > 0$$

## **Gradient Descent**

#### Nicolas Le Roux

### Ontimization

### Basics

method

Stochastic Optimization

Learning (Bottou

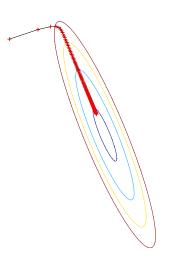
#### TONG

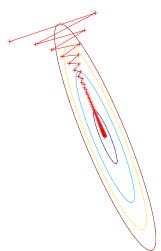
Natural Gradient

Online Natural Gradient

Results

## Quadratic bowl





$$\eta = .1$$

### Optimization

### Basics

method

Stochastic Optimization

#### Learning (Botto

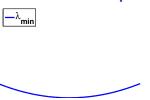
#### **TONG**

Natural Gradie

Online Natural Gradier

D 1

# Optimal learning rate



$$\eta_{\mathsf{min,opt}} = rac{1}{\lambda_{\mathsf{min}}} \ \eta_{\mathsf{min,div}} = rac{2}{\lambda_{\mathsf{min}}}$$

$$-\lambda_{\mathsf{max}}$$

$$\begin{array}{l} \eta_{\rm max,opt} = \frac{1}{\lambda_{\rm max}} \\ \eta_{\rm max,div} = \frac{2}{\lambda_{\rm max}} \end{array}$$

$$egin{array}{lcl} \eta &=& \eta_{
m max,opt} \ \kappa &=& rac{\eta}{\eta_{
m min,opt}} = rac{\lambda_{
m max}}{\lambda_{
m min}} \end{array}$$

$$T(\rho) = O\left(d\kappa \log \frac{1}{\rho}\right)$$

### Basics

method

Stochastic Optimization

Learning (Botto

#### **TONG**

Natural Gradier

Online Natural Gradie

Doculte

# Limitations of gradient descent

- Speed of convergence highly dependent on  $\kappa$
- Not invariant to linear transformations

$$\theta' = k\theta 
f(\theta' + \varepsilon) = f(\theta') + \varepsilon^T \nabla_{\theta'} f + o(\|\varepsilon\|) 
\varepsilon = -\eta \nabla_{\theta'} f$$

But 
$$\nabla_{\theta'} f = \frac{\nabla_{\theta} f}{k}$$
!

Stochastic Optimizatio

Learning (Bottou

#### TONGA

latural Gradie

Online Natural Gradie

- ·

## **Newton method**

Second-order Taylor expansion of f around  $\theta$ :

$$f(\theta + \varepsilon) = f(\theta) + \varepsilon^{\mathsf{T}} \nabla_{\theta} f + \frac{\varepsilon^{\mathsf{T}} H \varepsilon}{2} + o(\|\varepsilon\|^{2})$$

Best improvement obtained with

$$\varepsilon = -\eta H^{-1} \nabla_{\theta} f$$

### Basics

Approximations to Ne method

Stochastic Optimization

Learning (Bottou

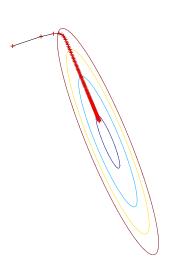
#### TONGA

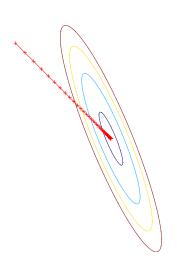
Natural Gradient

Online Natural Gradient

Results

## Quadratic bowl





Basics

Approximations to Newton method

Stochastic Optimization

Learning (Bottou

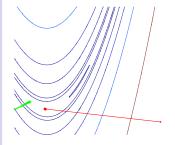
#### TONG

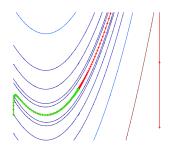
Natural Gradient

Online Natural Gradient

Results

## Rosenbrock function





#### Optiiii

#### Basics

method

Stochastic Optimization

Learning (Bottou

#### TONG

Notural Cradia

Online Natural Gradie

Dogulto

# Properties of Newton method

- Newton method assumes the function is locally quadratic (Beyond Newton method, Minka)
- H must be positive definite
- storing H and finding  $\varepsilon$  are in  $d^2$

$$T(
ho) = O\left(d^2 \log \log \frac{1}{
ho}\right)$$

#### Optimization

### Basics

Approximations to Newton method

Stochastic Optimizatio

Learning (Bottou

#### TONGA

Natural Gradien

Online Natural Gradient

# Limitations of Newton method

Newton method looks powerful

but...

## Basics

Approximations to Newton method

Stochastic Optimizatio

Learning (Botto

#### IONGA

Natural Gradier

Online Natural Gradie

D. II

# Limitations of Newton method

Newton method looks powerful

## but...

- H may be hard to compute
- it is expensive

Stochastic Optimization

Learning (Bottor

#### TONG/

Natural Gradier

Online Natural Gradie

## **Gauss-Newton**

Only works when f is a sum of squared residuals!

$$f(\theta) = \frac{1}{2} \sum_{i} r_{i}^{2}(\theta)$$

$$\frac{\partial f}{\partial \theta} = \sum_{i} r_{i} \frac{\partial r_{i}}{\partial \theta}$$

$$\frac{\partial^{2} f}{\partial \theta^{2}} = \sum_{i} \left[ r_{i} \frac{\partial^{2} r_{i}}{\partial \theta^{2}} + \left( \frac{\partial r_{i}}{\partial \theta} \right) \left( \frac{\partial r_{i}}{\partial \theta} \right)^{T} \right]$$

$$\theta = \theta - \eta \left( \sum_{i} \left( \frac{\partial r_{i}}{\partial \theta} \right) \left( \frac{\partial r_{i}}{\partial \theta} \right)^{T} \right)^{-1} \frac{\partial f}{\partial \theta}$$

Optimization

-----

Approximations to Newton method

Stochastic Optimization

Learning (Botto

#### TONGA

Natural Grad

Online Natural Gradie

Regults

# Properties of Gauss-Newton

$$\theta = \theta - \eta \left( \sum_{i} \left( \frac{\partial r_{i}}{\partial \theta} \right) \left( \frac{\partial r_{i}}{\partial \theta} \right)^{T} \right)^{-1} \frac{\partial f}{\partial \theta}$$

Discarded term:  $r_i \frac{\partial^2 r_i}{\partial \theta^2}$ 

- Does not require the computation of H
- Only valid close to the optimum where  $r_i = 0$
- Computation cost in  $O(d^2)$

method

Learning (Botto

#### TONG

Natural Gradie

Online Natural Gradie

Danisha

## Levenberg-Marquardt

$$\theta = \theta - \eta \left( \sum_{i} \left( \frac{\partial r_{i}}{\partial \theta} \right) \left( \frac{\partial r_{i}}{\partial \theta} \right)^{T} + \lambda I \right)^{-1} \frac{\partial f}{\partial \theta}$$

- "Damped" Gauss-Newton
- Intermediate between Gauss-Newton and Steepest Descent
- Slower optimization but more robust
- Cost in *O*(*d*<sup>2</sup>)

- 1 -

Approximations to Newton method

Stochastic Optimization

Learning (Botto

#### TONG

Vatural Gradien

Online Natural Gradie

## **Quasi-Newton methods**

- Gauss-Newton and Levenberg-Marquardt can only be used in special cases
- What about the general case?

Approximations to Newton

## Quasi-Newton methods

- Gauss-Newton and Levenberg-Marguardt can only be used in special cases
- What about the general case?
- H characterizes the change in gradient when moving in parameter space
- Let's find a matrix which does the same!

Online Natural Gradient

## **BFGS**

We look for a matrix B such that

$$abla_{ heta}f( heta+arepsilon)-
abla_{ heta}f( heta)=B_{t}^{-1}arepsilon$$
 : Secant equation

....

Learning (Botto

#### TONG

National Condi

Online Natural Gradie

Online Natural Grad

## **BFGS**

## We look for a matrix B such that

$$abla_{ heta}f( heta+arepsilon)-
abla_{ heta}f( heta)=B_{t}^{-1}arepsilon$$
 : Secant equation

- Problem: this is underconstrained
- Solution: set additional constraints: small
   ||B<sub>t+1</sub> B<sub>t</sub>||<sub>W</sub>

Basic

Approximations to Newton method

Stochastic Optimization

Learning (Botto

#### TONG

Natural Gradie

Online Natural Gradie

Doculte

# BFGS - (2)

- $B_0 = I$
- while not converged:
  - $p_t = -B_t \nabla f(\theta_t)$
  - $\eta_t = linemin(f, \theta_t, p_t)$
  - $s_t = \eta_t p_t$  (change in parameter space)
  - $\theta_{t+1} = \theta_t + s_t$
  - $y_t = \nabla f(\theta_{t+1}) \nabla f(\theta_t)$  (change in gradient space)
  - $\rho_t = (s_t^T y_t)^{-1}$
  - $B_{t+1} = (I \rho_t s_t y_t^T) B_t (I \rho_t y_t s_t^T) + \rho_t s_t s_t^T$  (stems from Sherman-Morrisson formula)

.

Approximations to Newton method

otoonaotio optimizatio

Learning (Botto

#### **TONGA**

Natural Gradier

Online Natural Gradient

Danilla

# BFGS - (3)

- Requires a line search
- No matrix inversion required
- Update in O(d²)
- Can we do better than  $O(d^2)$ ?

method

#### Learning (Botto

#### TONGA

Natural Gradier

Online Natural Gradie

## L-BFGS

- Low-rank estimate of B
- Based on the last m moves in parameters and gradient spaces
- Cost O(md) per update
- Same ballpark as steepest descent!

## Conjugate Gradient

We want to solve

$$Ax = b$$

- Relies on conjugate directions
- u and v are conjugate if  $u^T A v = 0$ .
- d mutually conjugate directions form a base of  $R^d$
- Goal: to move along conjugate directions close to steepest descent directions

Ontimization

Basics

Approximations to Newton method

Stochastic Optimization

Learning (Botto

#### TONGA

Natural Gradie

Online Natural Gradie

Paculte

# Nonlinear Conjugate Gradient

- Extension to non-quadratic functions
- Requires a line search in every direction (Important for conjugacy!)
- Various direction updates (Polak-Ribiere)

Basi

method

Stochastic Optimization

Learning (Bottou

#### TONGA

Natural Gradien

Online Natural Gradie

D ----It-

# Going from batch to stochastic

- dataset composed of n samples
- $f(\theta) = \frac{1}{n} \sum_{i} f_i(\theta, \mathbf{x}_i)$

Do we really need to see all the examples before making a parameter update?

#### \_ .. . ..

Pool

method

Stochastic Optimization

Learning (Bottou

#### TONGA

Natural Gradie

Online Natural Gradient

## Stochastic optimization

- Information is redundant amongst samples
- We can afford more frequent, noisier updates
- But problems arise...

#### Optimization

Optimizati

method

Stochastic Optimization

Learning (Bottou

#### TONG

Natural Gradie

Online Natural Gradien

Online Natural Gradien

# Stochastic (Bottou)

## Advantage

 much faster convergence on large redundant datasets

## Disadvantages

- Keeps bouncing around unless  $\eta$  is reduced
- Extremely hard to reach high accuracy
- Theoretical definitions for convergence not as well defined
- Most second-orders methods will not work

Stochastic Optimization

Learning (Botton

### TONG

Natural Gradie

Online Natural Gradien

Online Natural Grad

## Batch (Bottou)

## Advantage

- Guaranteed convergence to a local minimum under simple conditions
- Lots of tricks to speed up the learning

## Disadvantage

Painfully slow on large problems

Basi

method

Stochastic Optimization

Learning (Bottou

#### TONGA

Natural Gradi

Online Natural Gradie

\_ .

# Problems arising in stochastic setting

- First order descent:  $O(dn) \Longrightarrow O(dn)$
- Second order methods:  $O(d^2 + dn) \Longrightarrow O(d^2n)$
- Special cases: algorithms requiring line search
  - BFGS: not critical, may be replaced by a one-step update
  - Conjugate Gradient: critical, no stochastic version

#### Ontimization

-1.

method

Stochastic Optimization

Learning (Bottou

#### TONGA

Natural Gradier

Online Natural Gradie

Doculto

# Successful stochastic methods

- Stochastic gradient descent
- Online BFGS (Schraudolph, 2007)
- Online L-BFGS (Schraudolph, 2007)

#### . . . .

Dooi

method

Stochastic Optimization

Learning (Botton

#### **TONG**

Natural Gradie

Online Natural Gradier

Online Natural Gradie

# Conclusions of the tutorial

## **Batch methods**

- Second-order methods have much faster convergence
- They are too expensive when d is large
- Except for L-BFGS and Nonlinear Conjugate Gradient

Stochastic Optimization

Learning (Bottou

#### **TONG**

Natural Gradie

Online Natural Gradie

Offilitie (Vatural Orau)

# Conclusions of the tutorial

## Stochastic methods

- Much faster updates
- Terrible convergence rates
  - Stochastic Gradient Descent:  $T(
    ho) = O\left(rac{d}{
    ho}
    ight)$
  - Second-order Stochastic Descent:

$$T(
ho) = O\left(\frac{d^2}{
ho}\right)$$

# Optimizatio

Basics

method

оснавис Оринидаи

# Learning (Bottou)

## . . . . . . .

Natural Gradie

Online Natural Gradie

Regulto

# **Outline**

1 Optimization

Basics

Approximations to Newton method Stochastic Optimization

- 2 Learning (Bottou)
- **3** TONGA

Natural Gradient

Online Natural Gradient

Results

# Rasi

method

Stochastic Optimizatio

# Learning (Bottou)

# TONGA

Natural Gradie

Online Natural Gradient

Danisha

# Learning

- Measure of quality f ("cost function")
- Get training samples  $x_1, \ldots, x_n$  from p
- Choose a model  $\mathcal F$  with parameters  $\theta$
- Minimize  $E_p[f(\theta, x)]$
- Time budget T

## Optimization

Basi

method

Stochastic Optimization

# Learning (Bottou)

## **TONGA**

Natural Gradier

Online Natural Gradie

Results

# Small-scale vs. Large-scale Learning

- Small-scale learning problem: the active budget constraint is the number of examples *n*.
- Large-scale learning problem: the active budget constraint is the computing time T.

tochastic Optimization

# Learning (Bottou)

# TONG

Natural Gradie

Online Natural Gradie

D II

# Which algorithm should we use?

# $T(\rho)$ for various algorithms:

- Gradient Descent:  $T(\rho) = O\left(nd\kappa\log\frac{1}{\rho}\right)$
- Second Order Gradient Descent:

$$T(
ho) = O\left(d^2 \log \log \frac{1}{
ho}\right)$$

- Stochastic Gradient Descent:  $T(\rho) = O\left(\frac{d}{\rho}\right)$
- Second-order Stochastic Descent:

$$T(
ho) = O\left(\frac{d^2}{
ho}\right)$$

Second Order Gradient Descent seems a good choice!

## Optimizatio

\_ .

method

Stochastic Optimization

# Learning (Bottou)

## TONGA

Natural Gradier

Online Natural Gradient

Results

# Large-scale learning

- We are limited by the time T
- We can choose between  $\rho$  and n
- Better optimization means fewer examples

# Learning (Bottou)

# Generalization error

$$E(\tilde{f}_n) - E(f^*) = E(f_F^*) - E(f^*)$$
 Approximation error  $+ E(f_n) - E(f_F^*)$  Estimation error  $+ E(\tilde{f}_n) - E(f_n)$  Optimization error

There is no need to optimize thoroughly if we cannot process enough data points!

# Learning (Bottou)

Natural Gradient

# Which algorithm should we use?

Time to reach  $E(\tilde{f}_n) - E(f^*) < \epsilon$ :

- Gradient Descent:  $O\left(\frac{d^2\kappa}{\epsilon^{1/\alpha}}\log^2\frac{1}{\epsilon}\right)$
- Second Order Gradient Descent:  $O\left(\frac{d^2\kappa}{\epsilon^{1/\alpha}}\log\frac{1}{\epsilon}\log\log\frac{1}{\epsilon}\right)$
- Stochastic Gradient Descent:  $O\left(\frac{d\kappa^2}{\epsilon}\right)$
- Second-order Stochastic Descent:  $O\left(\frac{d^2}{\epsilon}\right)$ with  $\frac{1}{2} \le \alpha \le 1$  (statistical estimation rate).

## Optimization

Ориници

method

Stochastic Optimization

# Learning (Bottou)

# TONG/

Natural Gradie

Online Natural Gradient

# In a nutshell

- Simple stochastic gradient descent is extremely efficient
- Fast second-order stochastic gradient descent can win us a constant factor
- Are there other possible factors of improvement?

Approximations to Newto method

Stochastic Optimization

# Learning (Bottou)

## **TONGA**

Natural Gradier

Online Natural Gradie

Poculte

# What we did not talk about (yet)

- we have access to  $x_1, \ldots, x_n \sim p$
- we wish to minimize  $E_{x \sim p}[f(\theta, x)]$
- can we use the uncertainty in the dataset to get information about p?

## Optimization

Optimizati

method

oondotto optimizatio

Learning (Botto

# **TONGA**

Natural Gradie

Online Natural Gradie

Results

# **Outline**

Optimization
 Basics

Approximations to Newton method Stochastic Optimization

- 2 Learning (Bottou)
- **3** TONGA

Natural Gradient
Online Natural Gradient
Results

# MICOIAS LE INOU

Basics

Approximations to Newton method

Stochastic Optimizatio

Learning (Bottou

# TONG

## Natural Gradient

Online Natural Gradie

Danisha

# **Cost functions**

# **Empirical cost function**

$$f(\theta) = \frac{1}{n} \sum_{i} f(\theta, \mathbf{x}_i)$$

True cost function

$$f^*(\theta) = E_{\mathsf{X} \sim p}[f(\theta, \mathsf{X})]$$

# Natural Gradient

# Gradients

# **Empirical** gradient

$$g = \frac{1}{n} \sum_{i} \nabla_{\theta} f(\theta, \mathbf{x}_i)$$

True gradient

$$g^* = E_{x \sim p}[\nabla_{\theta} f(\theta, x)]$$

Central-limit theorem:

$$g|g^* \sim \mathcal{N}\left(g^*, \frac{C}{n}\right)$$

method

Learning (Botto

# TONG

## Natural Gradient

Online Natural Gradier

# Posterior over *g*\*

Using

$$g^* \sim \mathcal{N}(0, \sigma^2 I)$$

we have

$$g^*|g \sim \mathcal{N}\left(\left(I + \frac{C}{n\sigma^2}\right)^{-1}g, \left(\frac{I}{\sigma^2} + nC^{-1}\right)^{-1}\right)$$

and then

$$\varepsilon^{T} g^{*} | g \sim \mathcal{N} \left( \varepsilon^{T} \left( I + \frac{C}{n\sigma^{2}} \right)^{-1} g, \varepsilon^{T} \left( \frac{I}{\sigma^{2}} + nC^{-1} \right)^{-1} \varepsilon \right)$$

## Natural Gradient

# Aggressive strategy

$$\varepsilon^{T} g^{*} | g \sim \mathcal{N} \left( \varepsilon^{T} \left( I + \frac{C}{n\sigma^{2}} \right)^{-1} g, \varepsilon^{T} \left( \frac{I}{\sigma^{2}} + nC^{-1} \right)^{-1} \varepsilon \right)$$

- we want to minimize  $E_{q^*}[\varepsilon^T q^*]$
- Solution:

$$\varepsilon = -\eta \left( I + \frac{C}{n\sigma^2} \right)^{-1} g$$

method

Stochastic Optimization

Learning (Bottou

# TONG

# Natural Gradient

Online Natural Gradie

# Aggressive strategy

$$\varepsilon^{T} g^{*} | g \sim \mathcal{N} \left( \varepsilon^{T} \left( I + \frac{C}{n\sigma^{2}} \right)^{-1} g, \varepsilon^{T} \left( \frac{I}{\sigma^{2}} + nC^{-1} \right)^{-1} \varepsilon \right)$$

- we want to minimize  $E_{g^*}[\varepsilon^T g^*]$
- Solution:

$$\varepsilon = -\eta \left( I + \frac{C}{n\sigma^2} \right)^{-1} g$$

This is the regularized natural gradient.



# 0---------------

Baci

method

Stochastic Optimization

Learning (Bottou

# TONG

## Natural Gradient

Online Natural Gradie

# Conservative strategy

$$\varepsilon^{T} g^{*} | g \sim \mathcal{N} \left( \varepsilon^{T} \left( I + \frac{C}{\sigma^{2}} \right)^{-1} g, \varepsilon^{T} \left( \frac{I}{\sigma^{2}} + nC^{-1} \right)^{-1} \varepsilon \right)$$

- we want to minimize  $Pr(\varepsilon^T g^* > 0)$
- Solution:

$$\varepsilon = -\eta \left(\frac{C}{n}\right)^{-1} g$$

Otochastic Optimization

Learning (Bottou

# TONGA

## Natural Gradient

Online Natural Gradier

# Conservative strategy

$$\varepsilon^{T} g^{*} | g \sim \mathcal{N} \left( \varepsilon^{T} \left( I + \frac{C}{\sigma^{2}} \right)^{-1} g, \varepsilon^{T} \left( \frac{I}{\sigma^{2}} + nC^{-1} \right)^{-1} \varepsilon \right)$$

- we want to minimize  $Pr(\varepsilon^T g^* > 0)$
- Solution:

$$\varepsilon = -\eta \left(\frac{C}{n}\right)^{-1} g$$

This is the natural gradient.

method

Stochastic Optimization

Learning (Bottou

# TONGA

Natural Gradier

# Online Natural Gradient

.

# Online Natural Gradient

- Natural gradient has nice properties
- Its cost per iteration is in d<sup>2</sup>
- Can we make it faster?

Approximations to Newton

method

Stochastic Optimizatio

Learning (Bottor

## **TONGA**

Natural Gradien

## Online Natural Gradient

. .

# Goals

$$\varepsilon = -\eta \left( I + \frac{C}{n\sigma^2} \right)^{-1} g$$

We must be able to update and invert *C* whenever a new gradient arrives.

D--:--

Approximations to Newton method

Stochastic Optimizatio

Learning (Bottor

# **TONG**

Natural Gradient

Online Natural Gradient

Results

# Plan of action

Updating (uncentered) C:

$$C_t \propto \gamma C_{t-1} + (1-\gamma)g_t g_t^T$$

Stochastic Optimizatio

Learning (Bottou

# **TONG**

Natural Gradier

Online Natural Gradient

Regulto

# Plan of action

Updating (uncentered) C:

$$C_t \propto \gamma C_{t-1} + (1-\gamma)g_t g_t^T$$

# Inverting C:

 maintain low-rank estimates of C using its eigendecomposition.

D--:-

method

Stochastic Optimization

Learning (Bottor

# TONG

National Conditions

Online Natural Gradient

.

# Eigendecomposition of C

- Computing k eigenvectors of C = GG<sup>T</sup> is O(kd<sup>2</sup>)
- But computing k eigenvectors of  $G^TG$  is  $O(kp^2)$ !
- Still too expensive to compute for each new sample (and p must not grow)

\_ .

method

Stochastic Optimization

Learning (Bottor

# TONG

Material Condition

Online Natural Gradient

Ommio reacardi Or

# Eigendecomposition of C

- Computing k eigenvectors of C = GG<sup>T</sup> is O(kd<sup>2</sup>)
- But computing k eigenvectors of  $G^TG$  is  $O(kp^2)$ !
- Still too expensive to compute for each new sample (and p must not grow)
- Done only every b steps

Pool

method

Otochastic Optimizatio

Learning (Botto

# TONG

Natural Gradian

Online Natural Gradient

Offilia Natural Oraci

# To prove I'm not cheating

• eigendecomposition every b steps

$$C_{t} = \gamma^{t}C + \sum_{k=1}^{t} \gamma^{t-k} g_{k} g_{k}^{T} + \lambda t \qquad t = 1, \dots, b$$

$$v_{t} = C_{t}^{-1} g_{t}$$

$$X_{t} = \left[ \gamma^{\frac{t}{2}} U \quad \gamma^{\frac{t-1}{2}} g_{1} \quad \dots \quad \gamma^{\frac{1}{2}} g_{t-1} \quad g_{t} \right]$$

$$C_{t} = X_{t} X_{t}^{T} + \lambda t \qquad v_{t} = X_{t} \alpha_{t} \qquad g_{t} = X_{t} y_{t}$$

$$\alpha_{t} = (X_{t}^{T} X_{t} + \lambda t)^{-1} y_{t}$$

$$v_{t} = X_{t} (X_{t}^{T} X_{t} + \lambda t)^{-1} y_{t}$$

....

Learning (Botto

# **TONG**

Natural Gradier

Online Natural Gradient

- Valuar Orac

# Computation complexity

- *d* dimensions
- k eigenvectors
- b steps between two updates
- Computing the eigendecomposition (every b steps): k(k + b)<sup>2</sup>
- Computing the natural gradient:  $d(k+b) + (k+b)^2$
- if  $p \ll (k+b)$ , cost per example is O(d(k+b))

## D--!--

Approximations to Newton

method

Diochastic Optimization

Learning (Botto

# TONGA

Natural Gradier

# Online Natural Gradient

Dooult

# What next

- Complexity in O(d(k+b))
- We need a small k
- If  $d > 10^6$ , how large should k be?

# **Gradient Descent**

# Nicolas Le Roux

## Ontimization

Basic

Approximations to Newton method

Stochastic Optimizatio

Learning (Botto

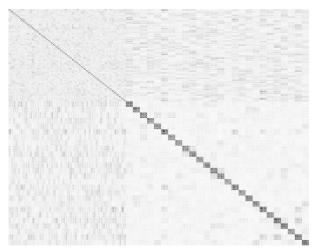
## TONG

Natural Gradient

Online Natural Gradient

Results

# Decomposing C



C is almost block-diagonal!



\_ .

Approximations to Newton method

Stochastic Optimization

Learning (Bottou

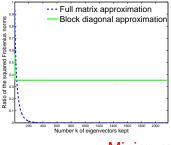
# TONGA

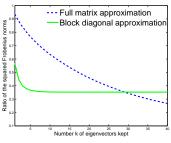
Material Condition

## Online Natural Gradient

Results

# Quality of approximation





Minimum with k = 5

## ......

Basics

method

Otochastic Optimization

Learning (Bottou

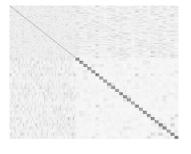
## TONG

Natural Gradient

## Online Natural Gradient

Results

# Evolution of C





## Optimization

Basic

Approximations to Newton method

Stochastic Optimization

## Learning (Botto

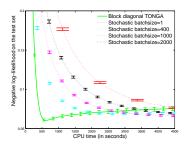
## TONGA

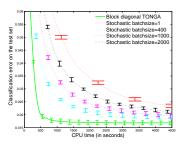
Natural Gradient

Online Natural Gradient

Results

# Results - MNIST



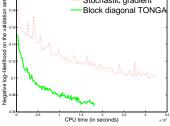


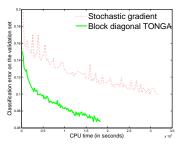
Approximations to Newton

Online Natural Gradient

Results

# Stochastic gradient Block diagonal TONGA





Results - Rectangles

Basics

Approximations to Newton method

Stochastic Optimizatio

# Learning (Bottou

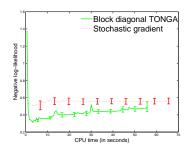
## TONG

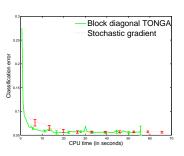
Natural Gradient

Online Natural Gradient

Results

# Results - USPS





method

Stochastic Optimization

Learning (Bottor

# **TONG**

Natural Gradier

Online Natural Gradient

Results

# **TONGA - Conclusion**

- Introducing uncertainty speeds up training
- There exists a fast implementation of the online natural gradient

# 0--------------

Basi

method

Stochastic Optimization

Learning (Bottou

# **TONGA**

atural Gradier

Online Natural Gradie

Results

# Difference between C and H

- H accounts for small changes in the parameter space
- C accounts for small changes in the input space
- C is not just a "cheap cousin" of H

## Optimizatio

method

Stochastic Optimization

Learning (Botto

# **TONG**

Natural Gradient

Online Natural Gradie

Results

# Future research

# Much more to do!

- Can we combine the effects of H and C?
- Are there better approximations of C and H?
- Anything you can think of!

# **Gradient Descent**

# Nicolas Le Roux

# Ontimization

Basics

Approximations to Newton method

Stochastic Optimization

Learning (Bottor

# TONGA

Natural Gradient

Online Natural Gradient

Results

Thank you!

Questions?