# GOSSIP COMMUNICATIONS FOR DEEP LEARNING DISTRIBUTED OPTIMIZATION

18 mars 2017

Michael Blot[1], David Picard[2], Matthieu Cord[1], Nicolas Thome[1]

(1) Université Pierre et Marie Curie, UPMC–Sorbonne Universités, LIP6, Paris, France

# Outline

# Outline

# Expected Loss Optimization

## Objective

We want to find $x^*$ that minimize

$$L(x) = \mathbb{E}_Y[\ell(x, Y)]$$

- $Y \sim \mathcal{I}$ is a random couple (input, label) variable following **natural images** distribution
- $x$ is the network parameters to optimize
- $\ell$ the error function or loss

# Stochastic gradient descent

## Gradient descent

$$x \leftarrow x - \eta \nabla L(x)$$

## Stochastic estimation of the gradient

Monte Carlo method :

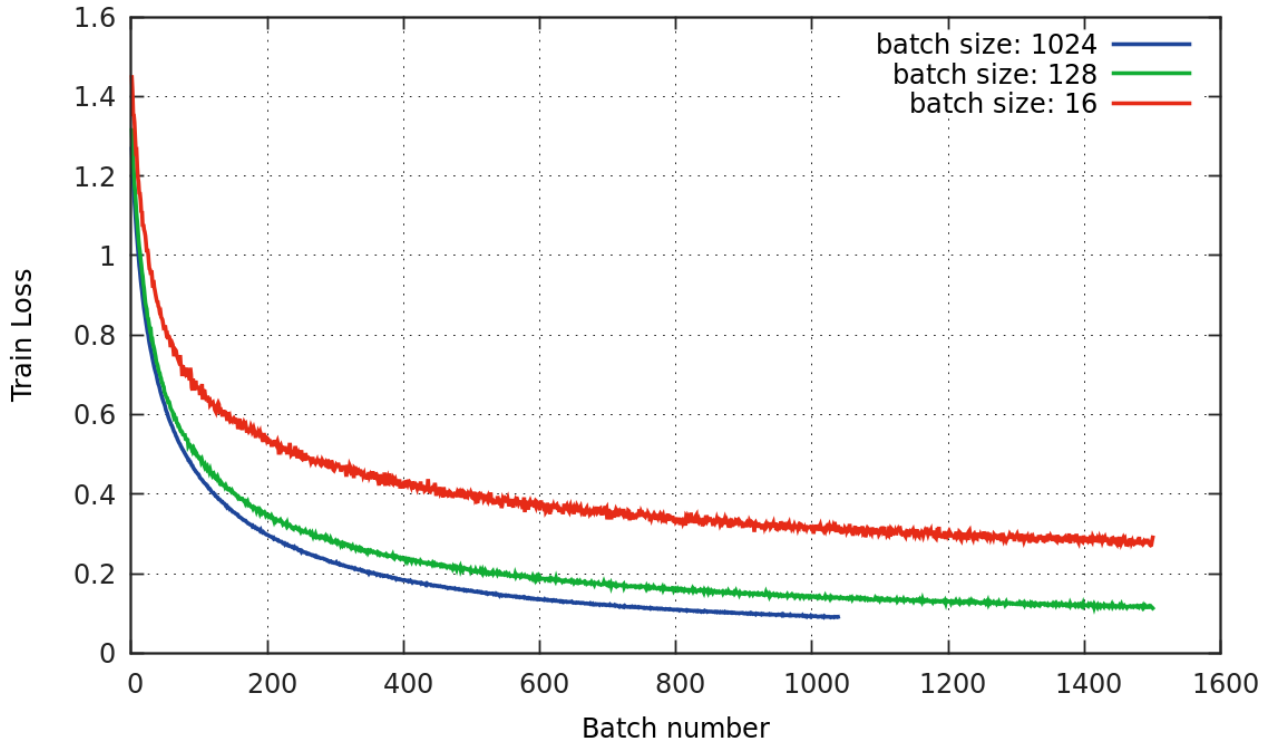
$$\nabla L(x) = \nabla_x \left( \mathbb{E}_Y [\ell(x, Y)] \right)$$

$$= \mathbb{E}_Y [\nabla_x \ell(x, Y)]$$

$$\approx \frac{1}{N} \sum_{i=1}^{N} \nabla_x \ell(x, Y_i)$$

# Convergence improvement

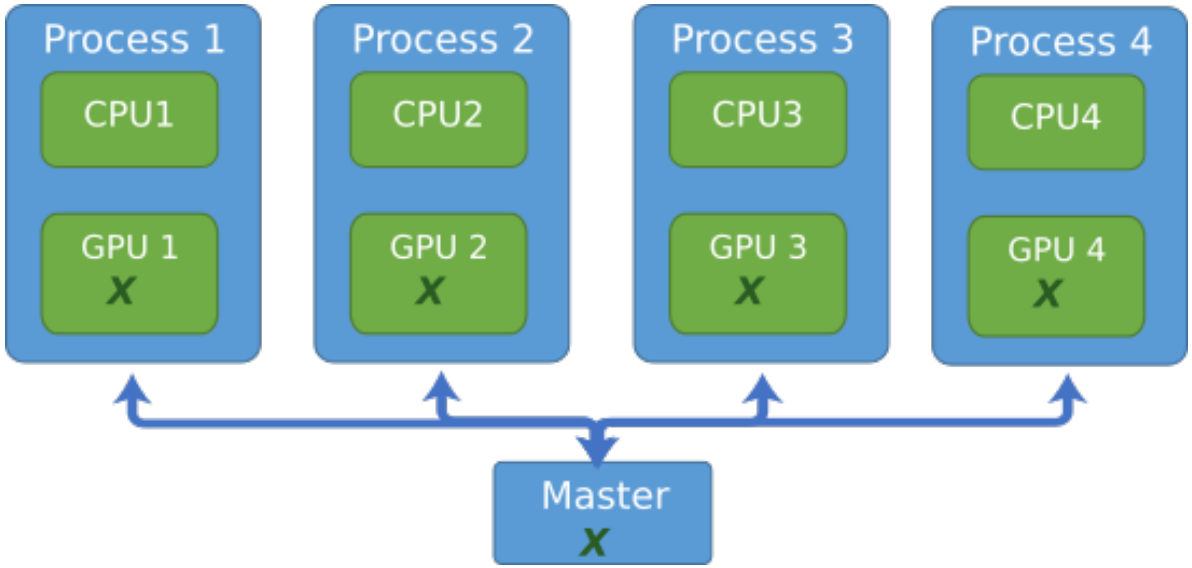
Comparison of convergence at same number of gradient descent iteration for different batch size : 16, 128, 1024




CIFAR10 : Image size $32 \times 32$ RGB pixels

# Outline

# Map Reduce Framework




M threads :

## Synchronous aggregation of the gradient

$$\widehat{\nabla} L_N(x) = \frac{1}{M} \sum_{m=1}^{M} \widehat{\nabla} L^m_{N'}(x)$$

# Distributed gradient descent

## Synchronous framework variable

$$x^{t+1} = x^0 - \sum_{\tau=0}^{t} \eta^\tau \sum_{m=1}^{M} \frac{1}{M} \widehat{\nabla} L^m_{N'}(x^\tau)$$

## General framework

$$x^{t+1} = x^0 - \sum_{\tau=0}^{t} \eta^\tau \sum_{m=1}^{M} a^\tau_m \widehat{\nabla} L^m_{N'}(x^\tau_m)$$

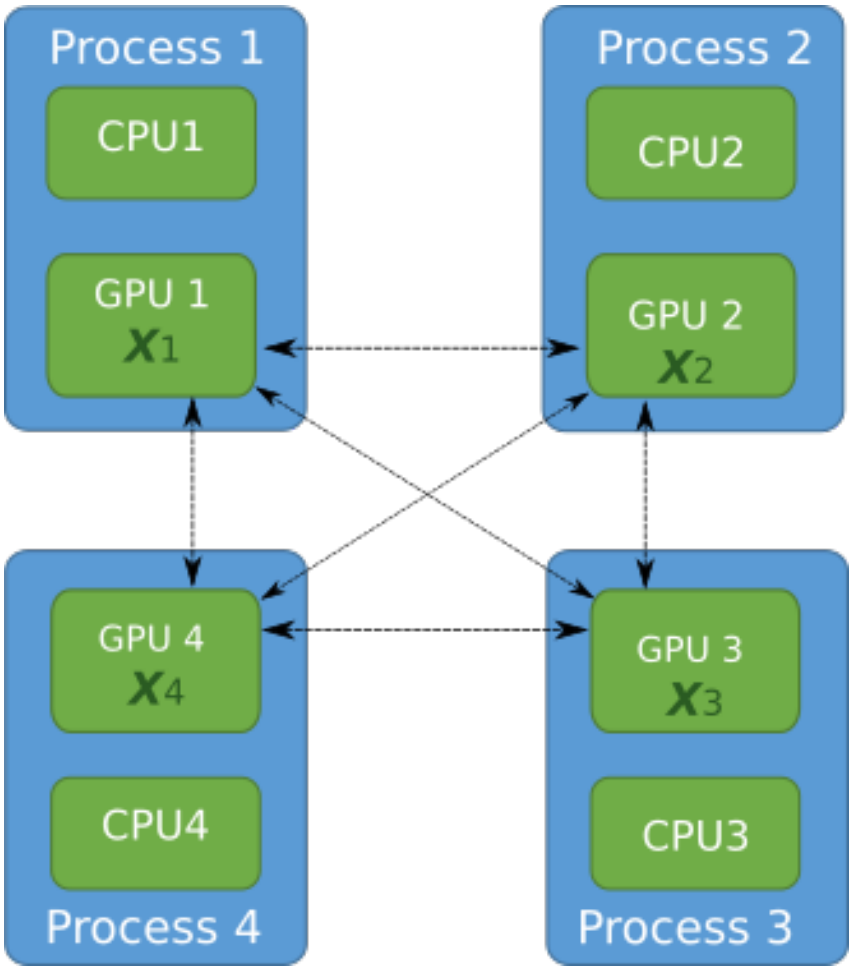
With $\forall \tau \ \sum_{m=1}^{M} a^\tau_m = 1$

Difficulty : how far are the gradient estimates :

$$||\widehat{\nabla} L^m_{N'}(x_m) - \widehat{\nabla} L^m_{N'}(x^\tau)|| < \epsilon$$

# Outline

# Gossip communication



# GoSGD pseudo-code

**Input :** $p$ : probability of exchange, $M$ : number of threads, $\eta$ : learning rate
**Initialize :** $x$ is initialized randomly, $x^i = x$, $\alpha_i = \frac{1}{M}$
**repeat**
  $x^i = x^i - \nu g(x^i, batch)$
  **if** $S \sim B(p)$ **then**
    $j = Random(M)$
    $\alpha^j = \frac{\alpha^j}{2}$
    $x^i = \frac{\alpha^i}{\alpha^i + \alpha^j} x^i + \frac{\alpha^j}{\alpha^i + \alpha^j} x^j$
    $\alpha_i = \alpha_i + \alpha_j$
  **end if**
**until** Forever
get $\frac{1}{M} \sum_{m=1}^{M} x_m$

# Outline

# Illustration : Gossip vs synchronous vs single thread

- Database : CIFAR10 (50000 images of $32 \times 32$ RGB pixels)
- Batch size : 20 images, learning rate : of 0.01
- Networks : 286154 parameters, 8 layers
- Number of thread (M) : 8