Large Scale Learning

John Langford

Yahoo! Research

For COMS-4771@Columbia

Demonstration: Vowpal Wabbit

For more details: http://hunch.net/~vw/

What is VW?

Start with $\forall i: \quad w_i = 0$, Repeatedly:

1. Get example $x \in (-\infty, \infty)^*$.

2. Make prediction $\hat{y} - \frac{\sum_i w_i x_i}{\sqrt{|\{i:x_i \neq 0\}|}}$ clipped to interval $[0,1]$.

3. Learn truth $y \in [0,1]$ with importance I or goto (1).

4. Update $w_i \leftarrow w_i + \frac{\eta 2(y-\hat{y}) I x_i}{\sqrt{|\{i:x_i \neq 0\}|}}$ and go to (1).

What does this do?

Squared loss = $(y - \widehat{y})^2$ Derivative is:

$$\frac{\partial}{\partial w_i}(y - \widehat{y}(x,w))^2$$

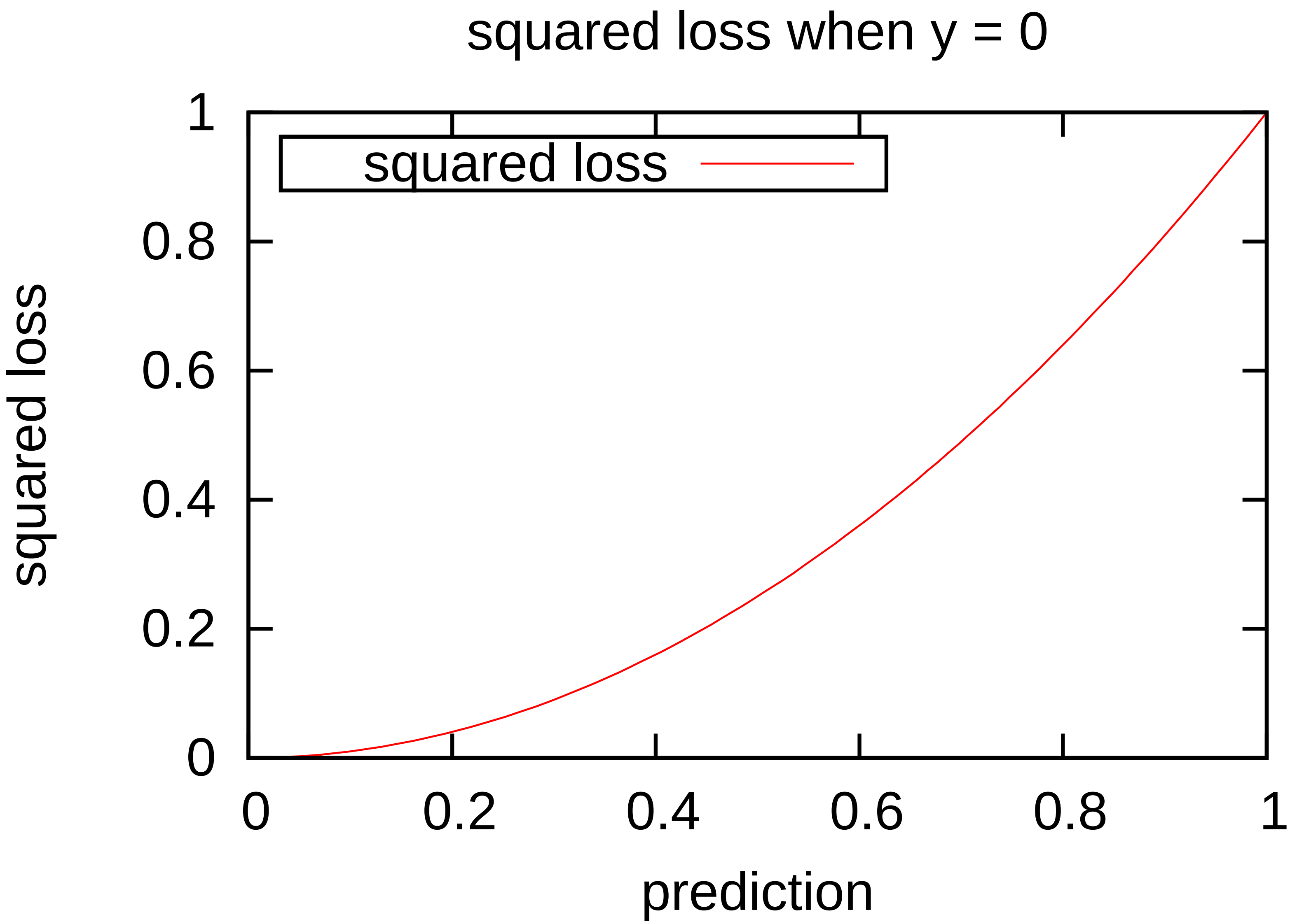
$$= -2(y - \widehat{y}(x,w))\frac{\partial}{\partial w_i}\widehat{y}(x,w)$$

$$= -2(y - \widehat{y}(x,w))\frac{x_i}{N_x}$$

So update = negative gradient step.

= step towards minimum of squared loss

= step towards expected value of y


squared loss when y = 0
squared loss
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
prediction
squared loss

The Batch Optimization vs. Online Optimization Debate

Suppose you see the examples:

$((1, 0, 1, 1, 1), 1)$
$((1, 0, 1, 0, 1), 0)$
$((1, 0, 1, 1, 1), 1)$
$((1, 0, 1, 1, 1), 1)$
$((1, 0, 1, 0, 1), 0)$
$((1, 0, 1, 1, 1), 1)$
$\ldots.$

How many more do you need in order to update your predictor?

Batch answer: All of them

Online answer: One is enough

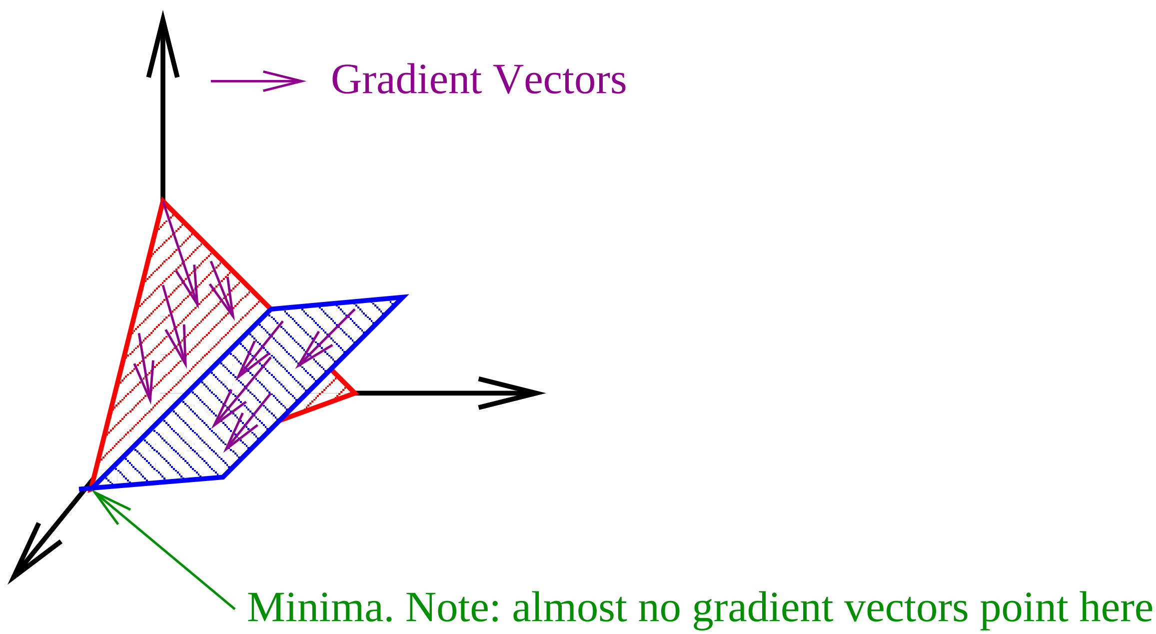
Batch vs. Online Learning

1. Update rules for Batch learning are easier.

2. Nonlinear learning is easiear for Batch learning.

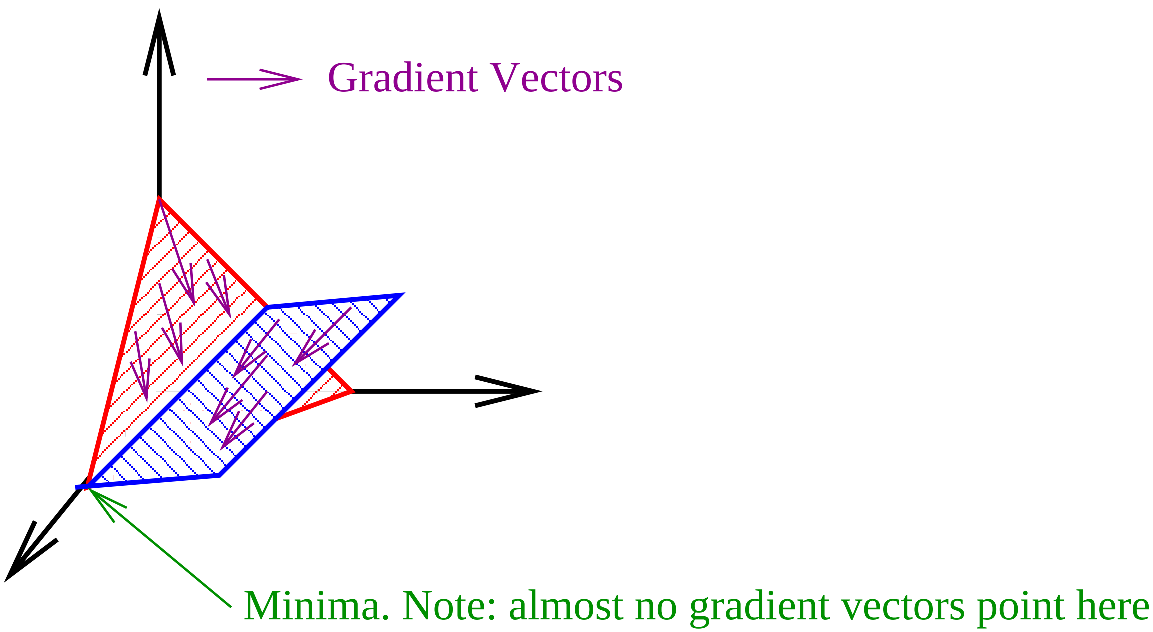
3. Batch learning is slow—potentially $O(m)$ slower. (This isn't just slow, it's also inefficient.)

The Optimization Picture



Even with linear constraints & global optima, gradient descent can't get there in one step.

Why we will all do Online Optimization, eventually



Options: Make multiples data passes (slow!) or use online optimization (fast!)

Online learning more important for bigger problems

1. More dimensions $\Rightarrow$ argument bites harder

2. More data $\Rightarrow$ argument bites harder

3. Nonlinearity $\Rightarrow$ argument bites harder

Integration into a Map-Reduce World

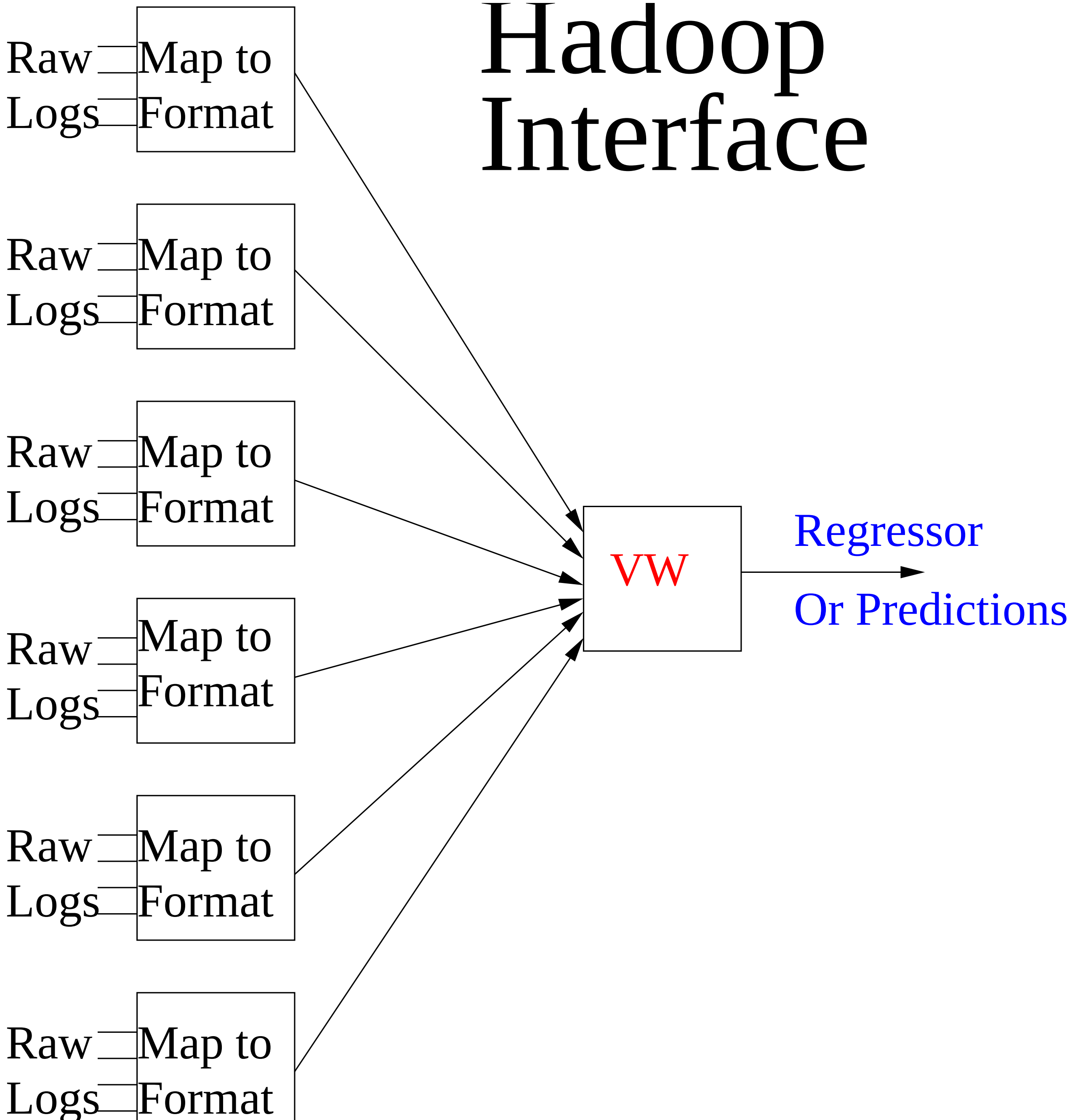
Map-Reduce = computational paradigm popularized by Google.

Map = an operation repeatedly applied to many objects

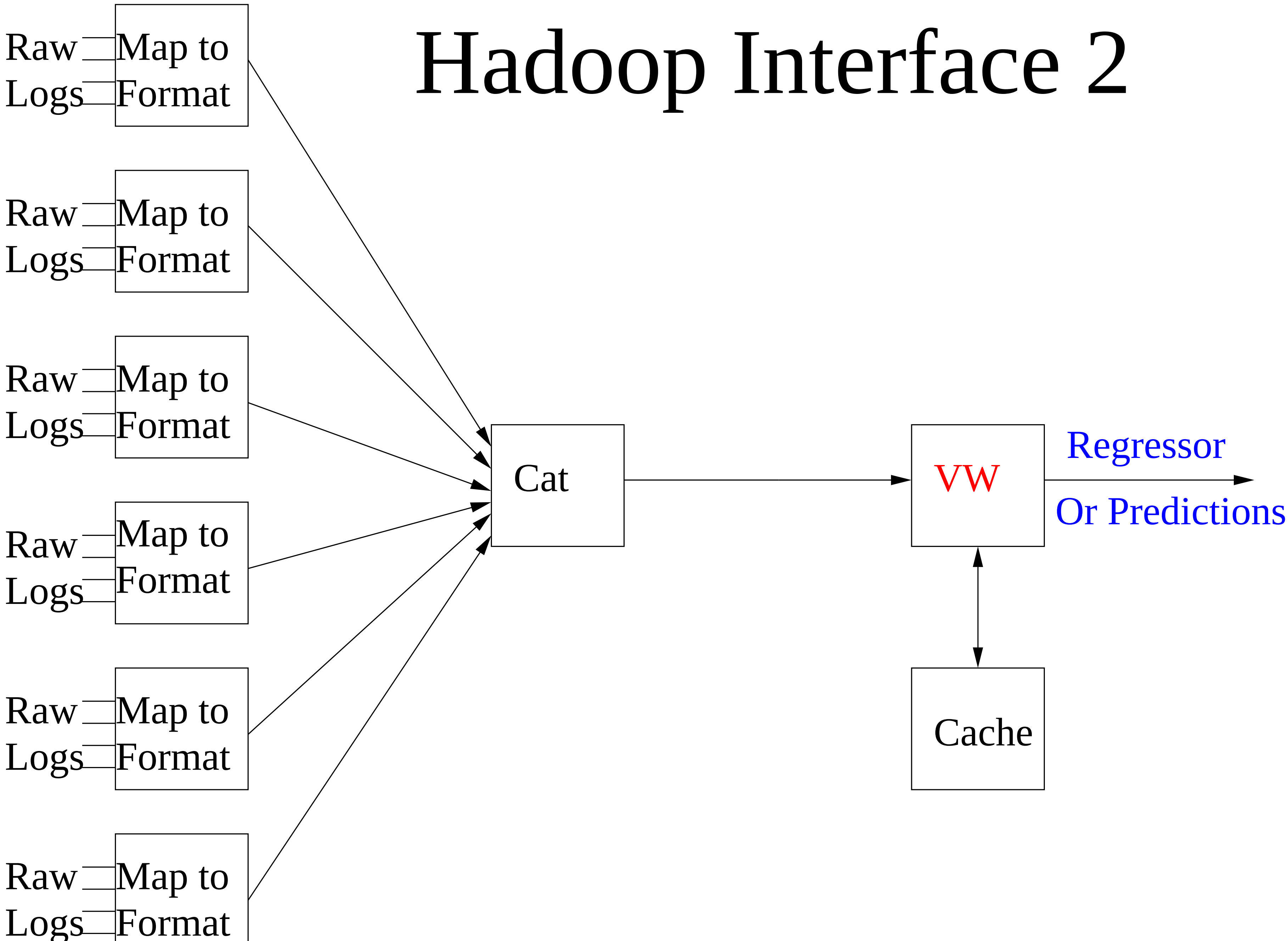
Reduce = operation which digests outputs of Map.

Hadoop = Open Source Map-Reduce (hadoop.apache.org), sponsored by Yahoo.

Hadoop Interface



Hadoop Interface 2



Things we know how to do fast

1. Vector multiply (= core prediction algorithm)
2. Hash (Built into parser)

Open Problems in Online Optimization for Learning

1. How do we efficiently learn nonlinearities?
2. How do we parallelize online learning over multiple nodes?