BLUE WATERS

SUSTAINED PETASCALE COMPUTING

7/11/19

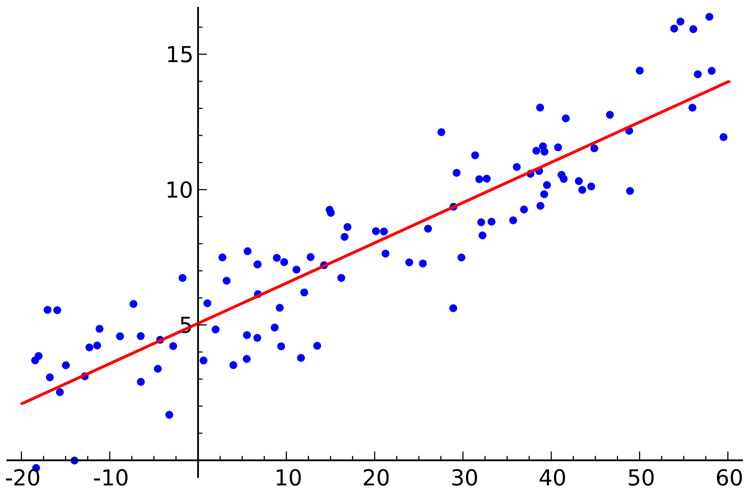
Distributed Training on HPC

Presented By: Aaron D. Saxton, PhD

GREAT LAKES CONSORTIUM FOR PETASCALE COMPUTATION

Statistics Review

- Simple $y = m \cdot x + b$ regression
  - Least Squares to find m,b
  - With data set $\{(x_i, y_i)\}_{i=1,\ldots,n}$
    - Very special, often hard to measure y_i
  - Let the error be
    - $R = \sum_{i=1}^{n}[(y_i - (m \cdot x_i + b)]^2$
  - Minimize R with respect to m and b.
    - Simultaneously Solve
      - $R_m(m, b) = 0$
      - $R_b(m, b) = 0$
    - Linear System
- We will consider more general $y = f(x)$
  - $R_m(m, b) = 0$ and $R_b(m, b) = 0$ may not be linear

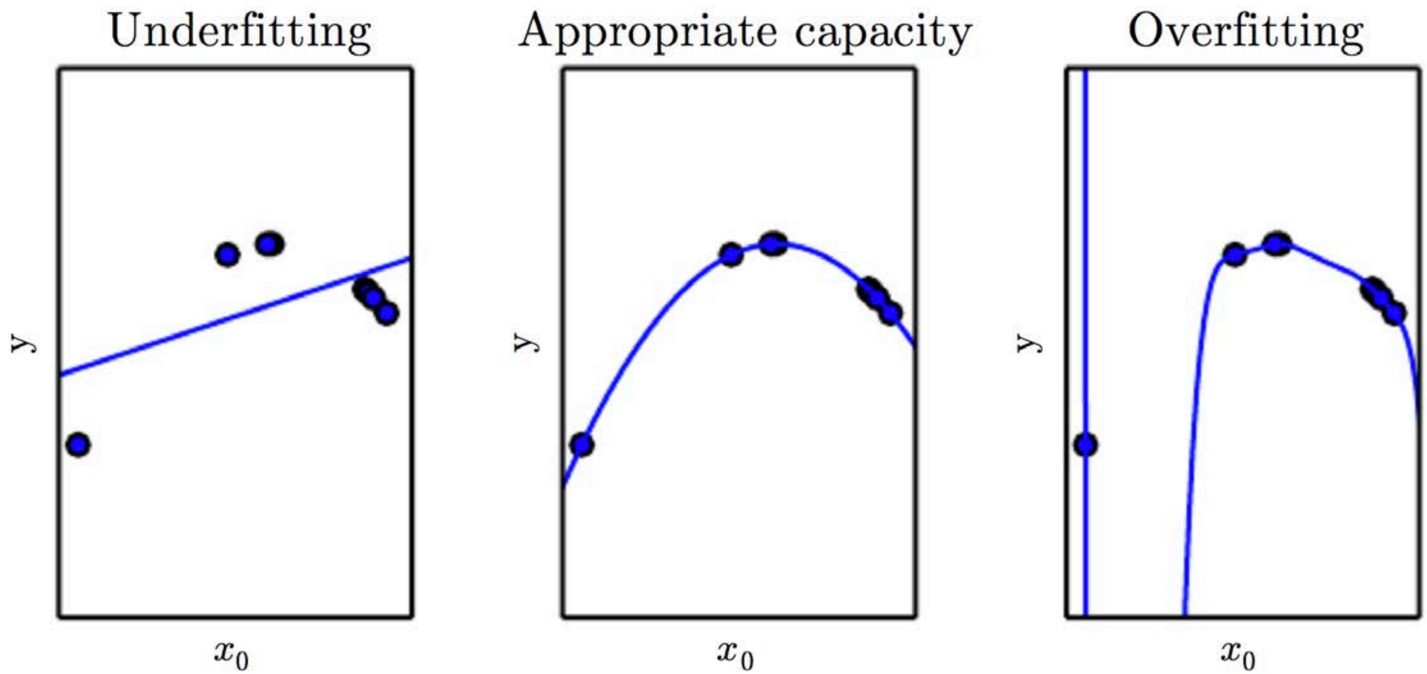


Statistics Review

- Regressions with parameterized sets of functions. e.g.
  - $y = ax^2 + bx + c$ (quadratic)
  - $y = \sum a_i \ x^i$ (polynomial)
  - $y = Ne^{rx}$(exponential)
  - $y = \frac{1}{1+e^{-(a+bx)}}$ (logistic)

Statistics Review

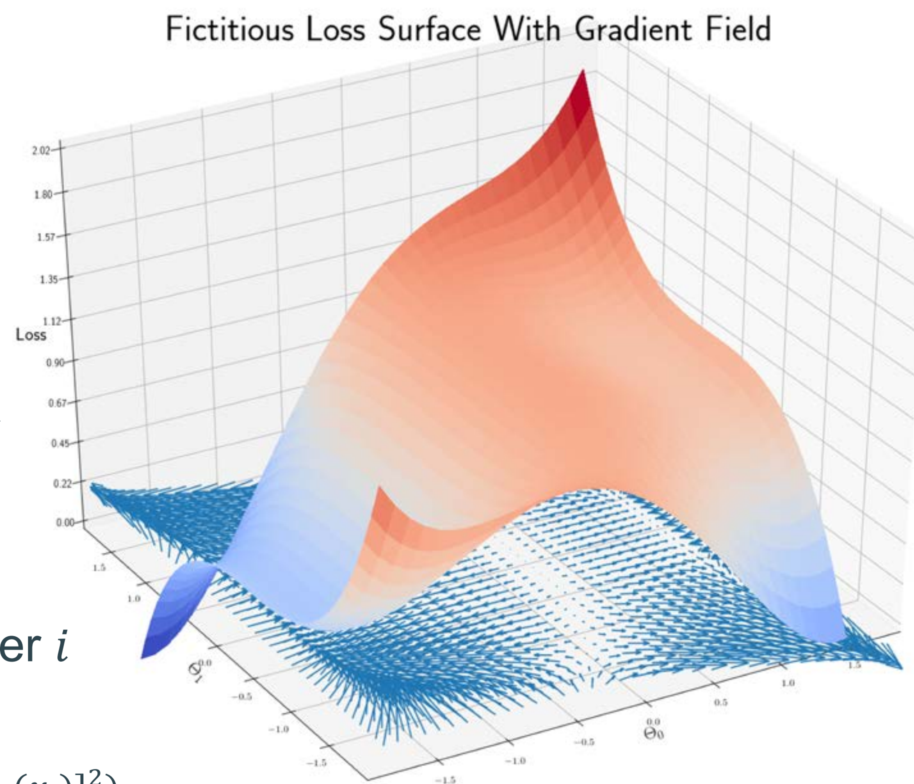
- Polynomial model of degree ‘n’
  - “degrees of freedom” - models capacity



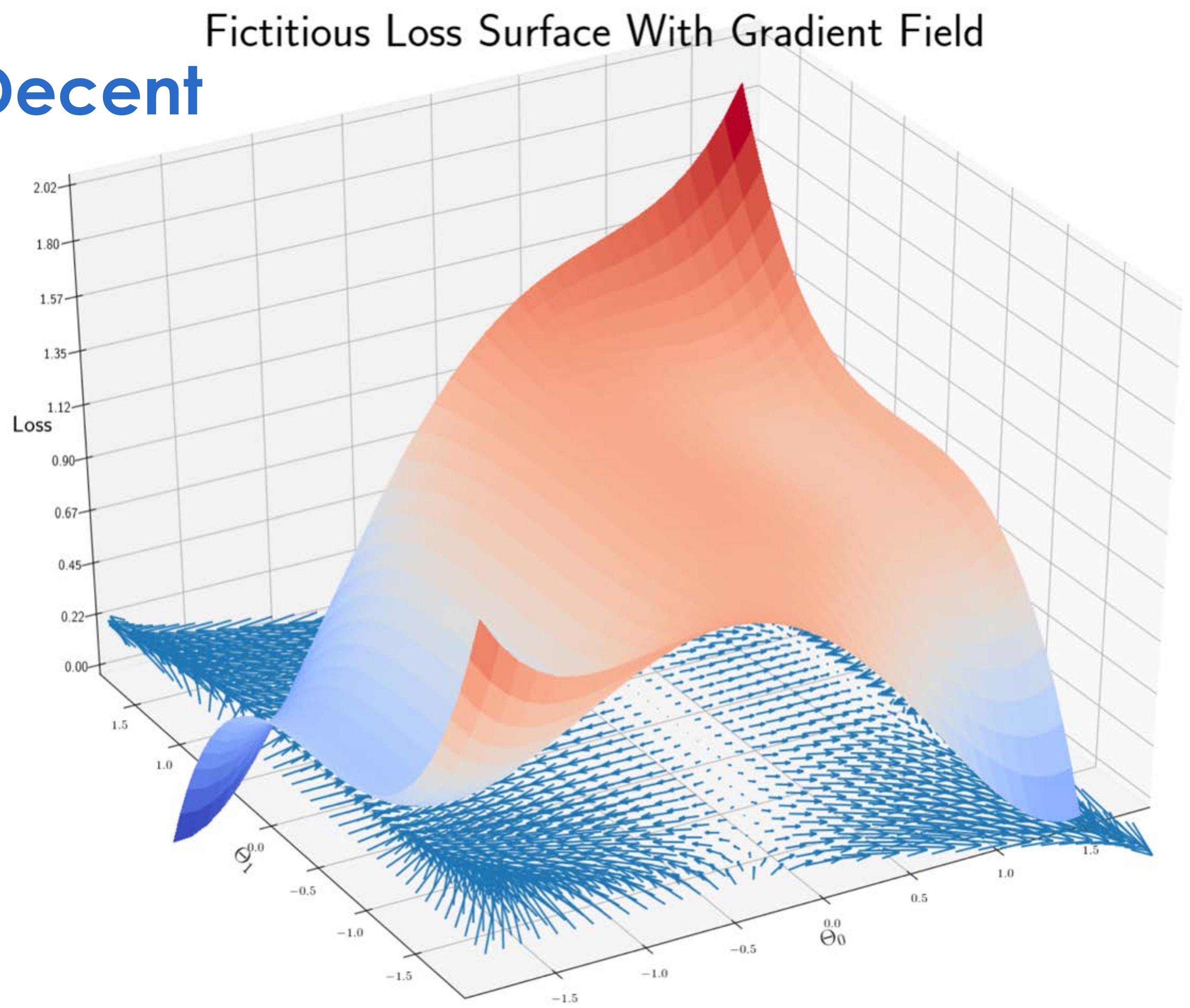
Deep Learning, Goodfellow et. al., MIT Press, http://www.deeplearningbook.org, 2016

Gradient Decent

- Searching for minimum
- $\nabla R = \langle R_{\theta_0}, R_{\theta_2}, \dots, R_{\theta_n} \rangle$
- $R(\vec{\theta}_{t+1}) = R(\vec{\theta}_t + \gamma \nabla R)$
- γ: Learning Rate
- Recall, Loss depends on data
  Expand notation,
  - $R(\vec{\theta}_t; \{(x_i, y_i)\}_n)$
  - Recall R and ∇R is a sum over i
- Intuitively, want R with
  ALL DATA ….. ? $(R = \sum_{i=1}^{n}[(y_i - f_{\theta_t}(x_i)]^2)$



Gradient Decent



Stochastic Gradient Decent

- Recall R is a sum over i ($R = \sum_{i=1}^{n}[(y_i - f_{\theta_t}(x_i)]^2)$
- Single training example, (x_i, y_i), Sum over only one training example
- $\nabla R_{(x_i,y_i)} = \langle R_{\theta_0}, R_{\theta_2}, \ldots, R_{\theta_n} \rangle_{(x_i,y_i)}$
- $R_{(x_i,y_i)}(\vec{\theta}_{t+1}) = R_{(x_i,y_i)}(\vec{\theta}_t + \gamma \nabla R_{(x_i,y_i)})$
- γ: Learning Rate
- Choose next (x_{i+1}, y_{i+1}), (Shuffled training set)

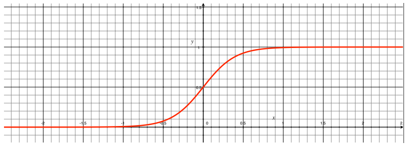
- SGD with mini batches
- Many training example, (x_i, y_i), Sum over many training example
  - Batch Size or Mini Batch Size (This gets ambiguous with distributed training)
- SGD often outperforms traditional GD, want small batches.
  - https://arxiv.org/abs/1609.04836, On Large-Batch Training … Sharp Minima
  - https://arxiv.org/abs/1711.04325, Extremely Large ... in 15 Minutes

Neural Networks

- Activation functions

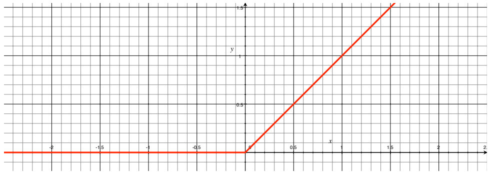
Logistic

$\sigma(x) =$



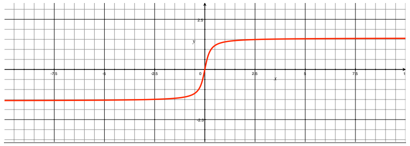
ReLU (Rectified Linear Unit)

$\sigma(x) =$



Arctan

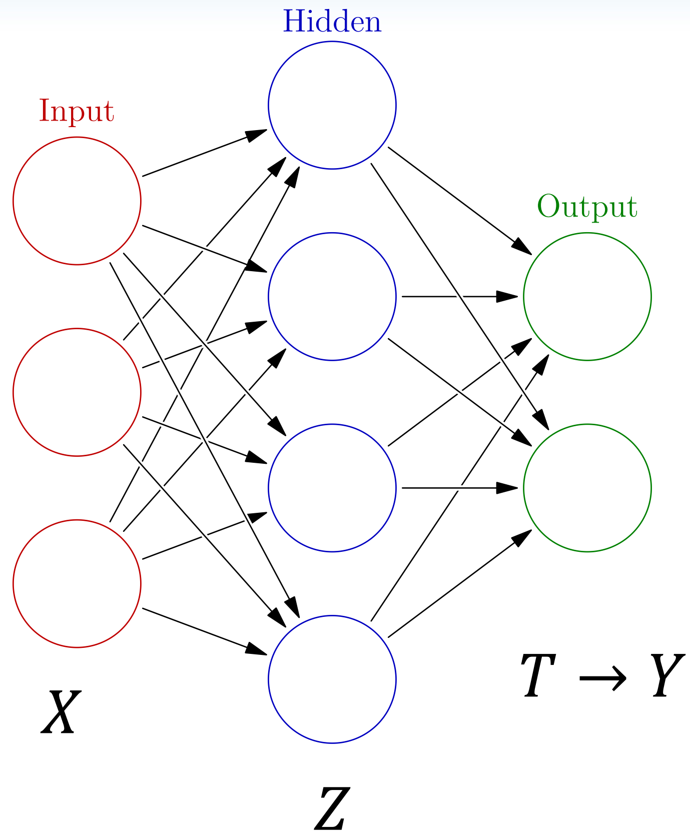
$\sigma(x) =$



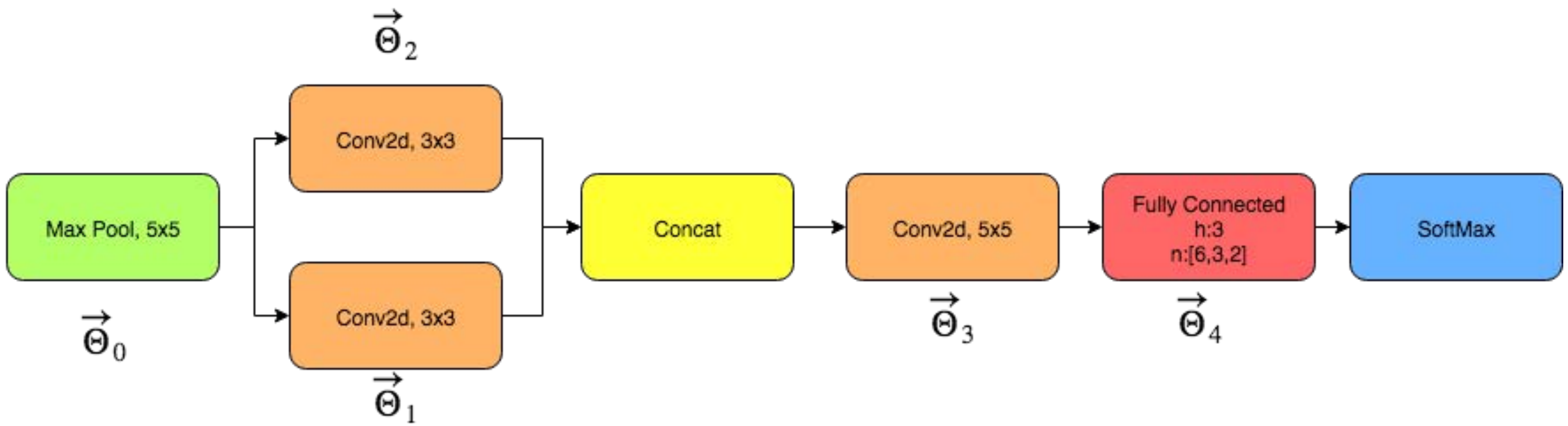
- Softmax
  - $g_k(x_1, x_2, \dots, x_N) = \dfrac{e^{x_k}}{\sum e^{x_i}}$

Neural Networks

- Parameterized function
  - $Z_M = \sigma(\alpha_{0m} + \alpha_m X)$
  - $T_K = \beta_{0k} + \beta_k Z$
  - $f_K(X) = g_k(T)$
- Linear Transformations with pointwise evaluation of nonlinear function, σ
- $\beta_{0i}, \beta_i, \alpha_{0m}, \alpha_m$
  - Weights to be optimized



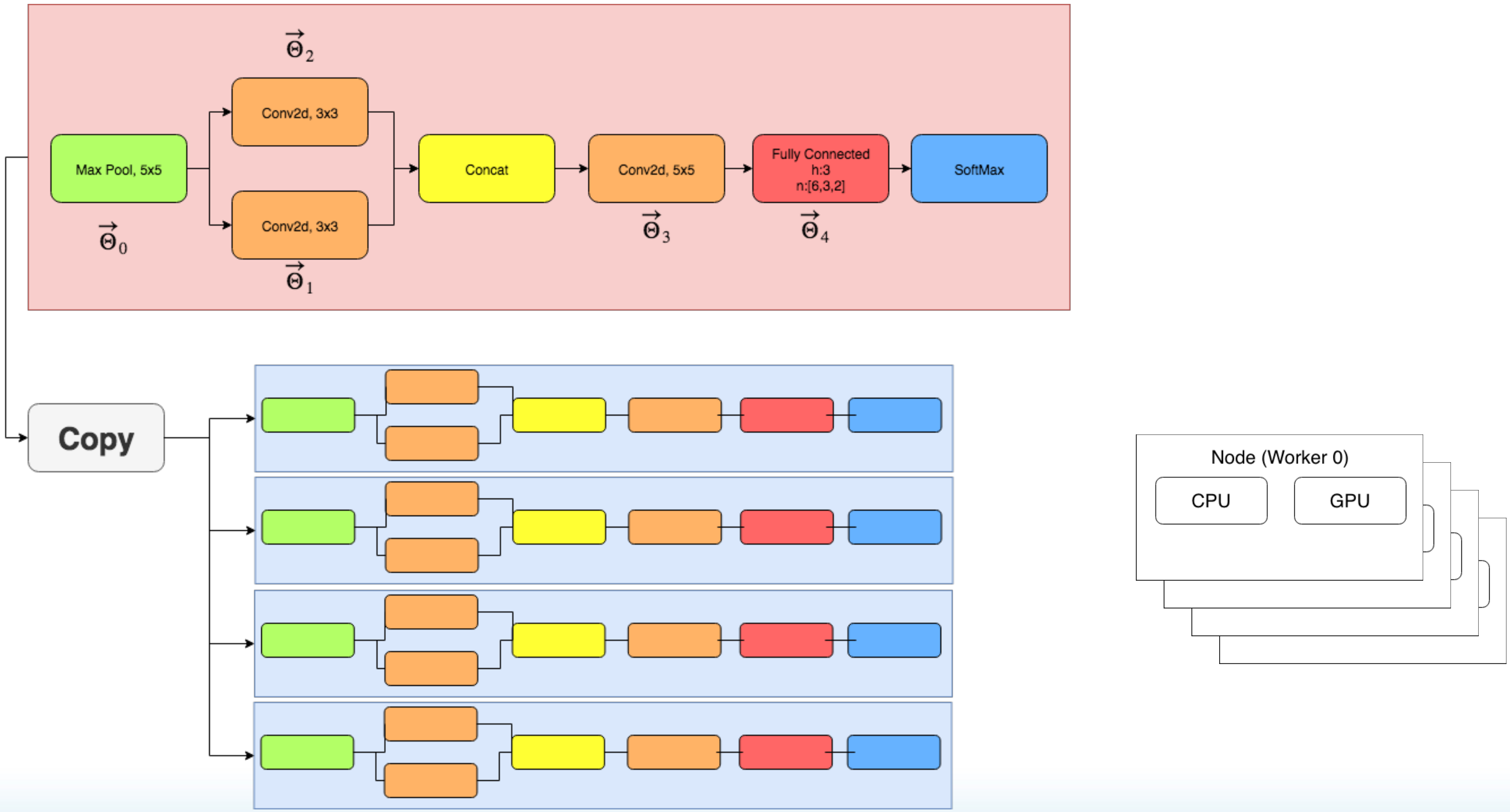
Faux Model Example



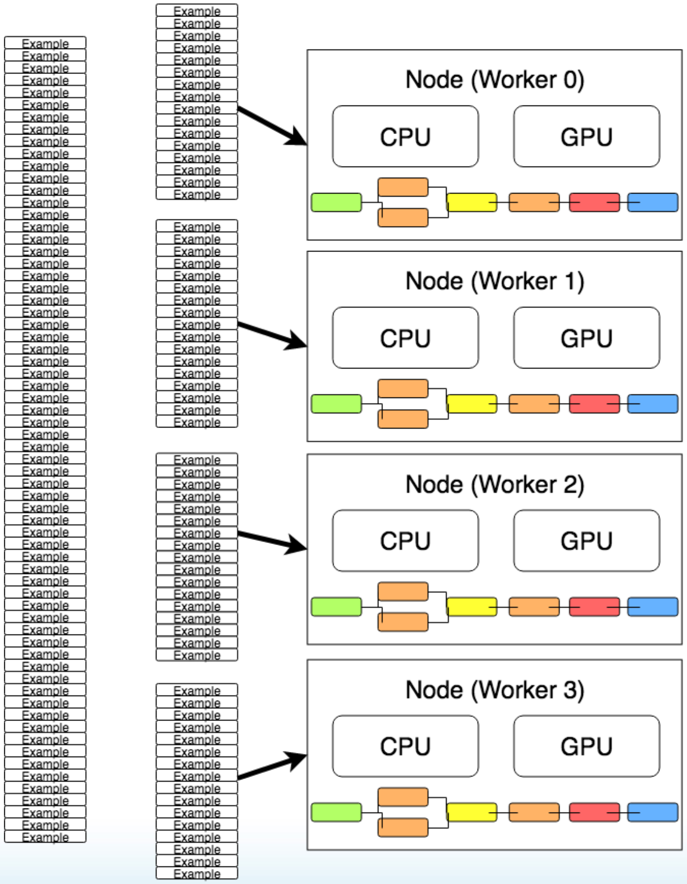
Trainable Weights

$$\left\{ \vec{\Theta}_i : i \in [0, 1, 2, 3, 4] \right\}$$

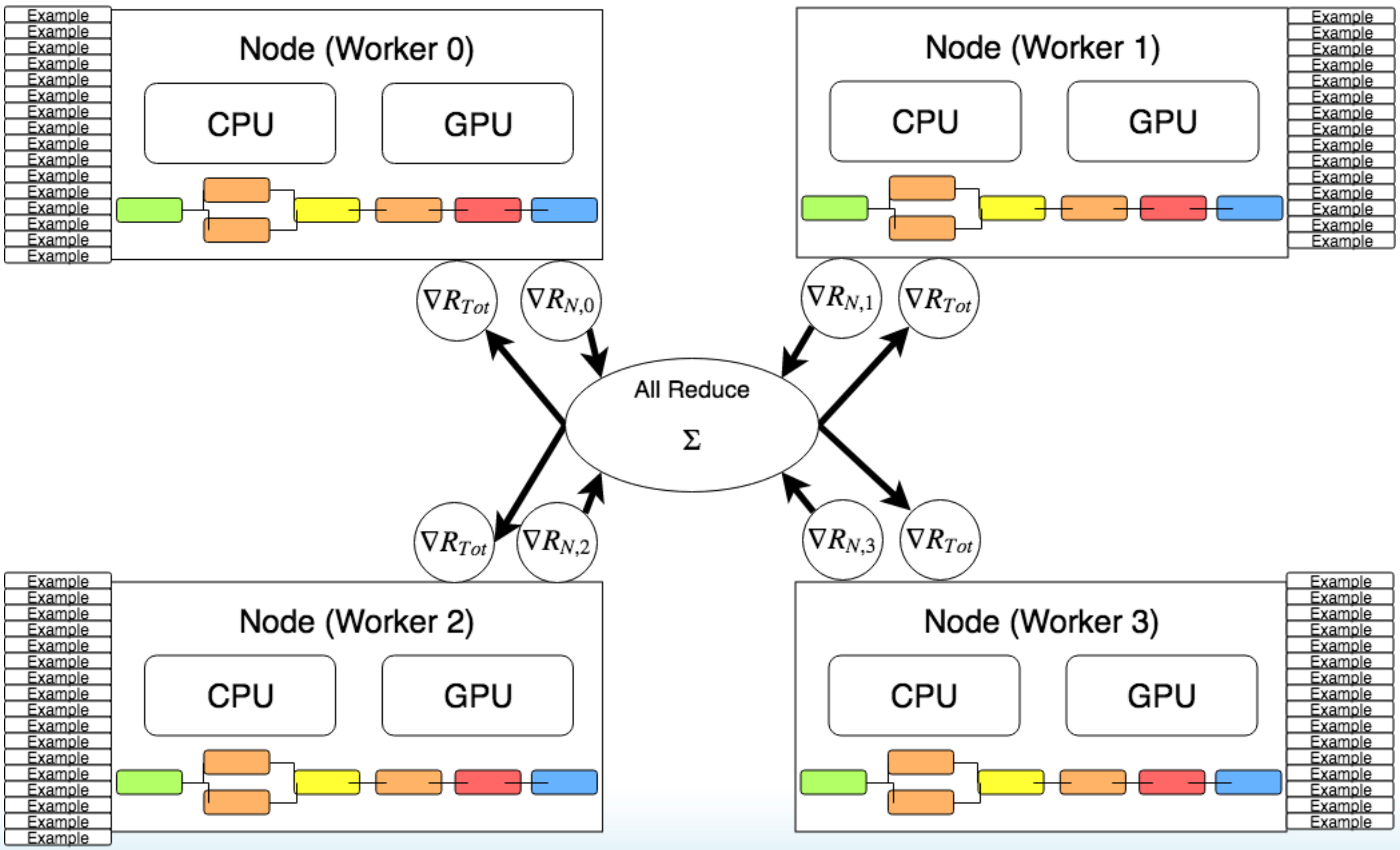
Distributed Training, data distributed



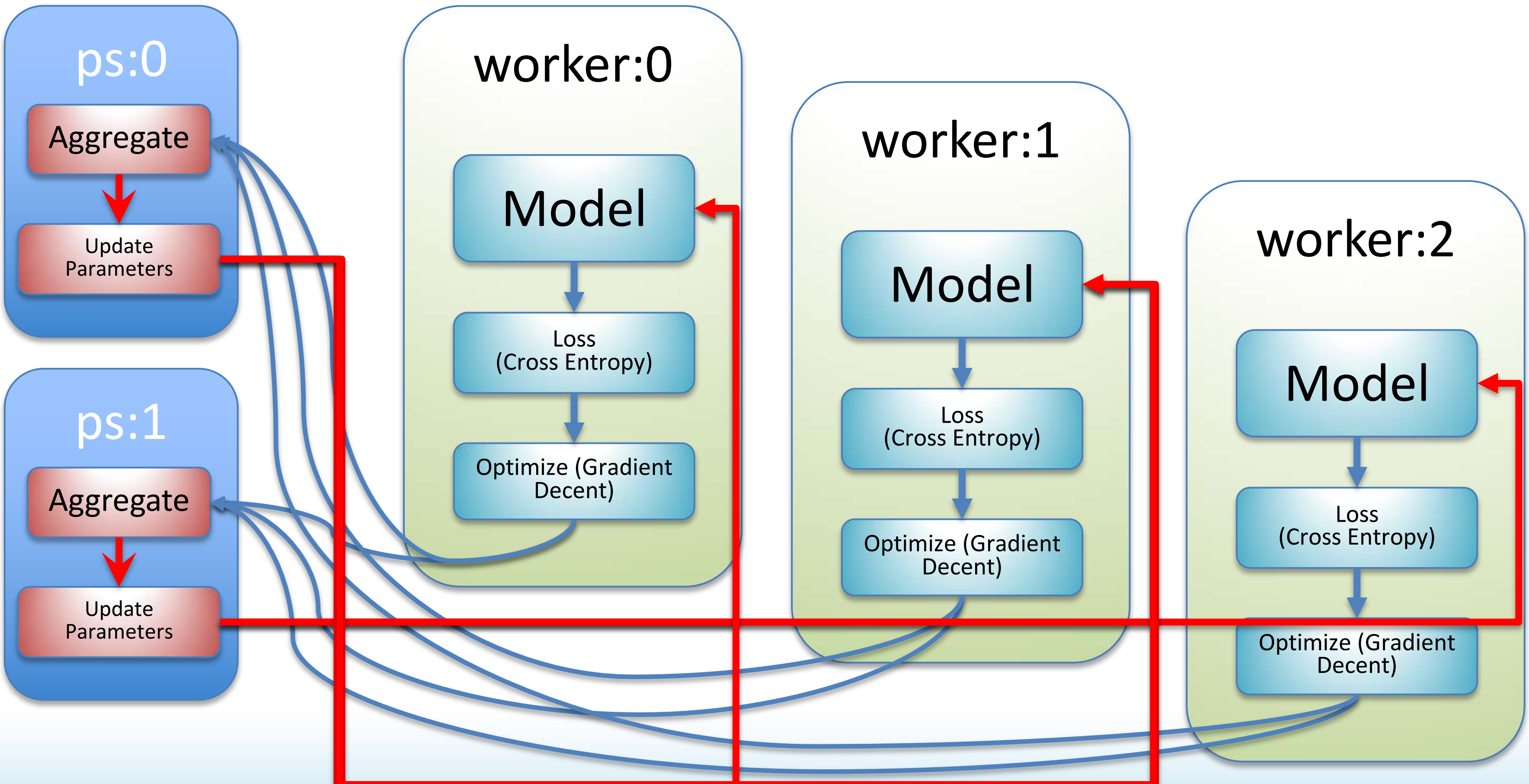
Distributed Training, data distributed



Distributed Training, All Reduce Collective



Distributed TensorFlow: Parameter Sever/Worker Default, Bad Way on HPC



Other models: Sequence Modeling

- Autoregression

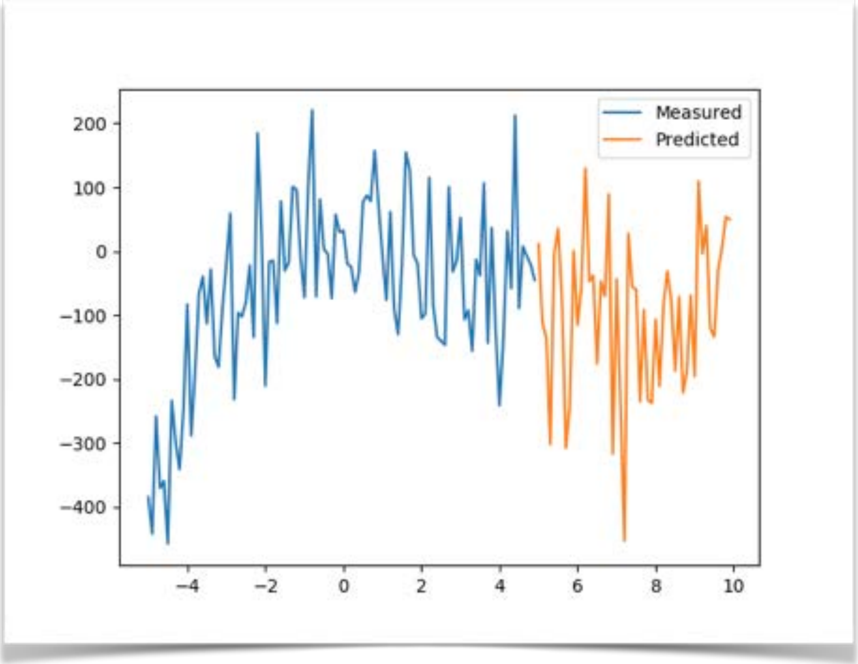
$$X_t = c + \sum_{i=1}^{p} \phi_i B^i X_t + \epsilon_t$$

Back Shift Operator: B^i

- Autocorrelation

$$R_{XX}(t_1, t_2) = E[X_{t_1} \overline{X_{t_2}}]$$

- Other tasks
  - Semantic Labeling



[art.]	[adj.]	[adj.]	[n.]	[v.]	[adverb]	[art.]	[adj.]	[adj.]	[d.o.]
The	quick	red	fox	jumps	over	the	lazy	brown	dog

Recurrent Neural Networks: Sequence Modeling

- Few projects use pure RNNs, this example is only for pedagogy
- RNN is a model that is as "deep" as the modeled sequence is long
- LSTM's, Gated recurrent unit,
- No Model Parallel distributed training on the market (June 2019)