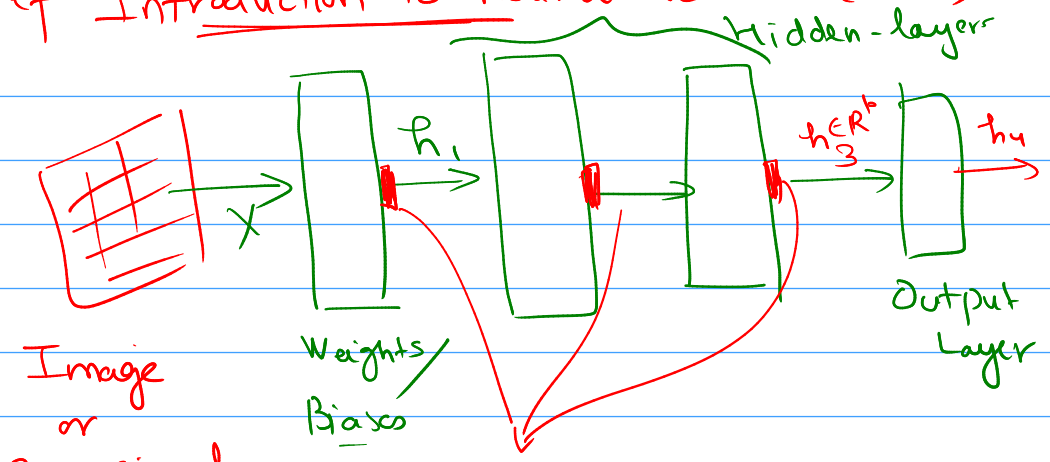
Brief Introduction to Neural Networks (NNs)



$$h_1 = \sigma(WX + b)$$

$h_1 \in \mathbb{R}^m$, $WX + b \in \mathbb{R}^n$

$$\underset{W^{m\times(n+1)}}{\tilde{X}} = \begin{bmatrix} X \\ 1 \end{bmatrix}$$

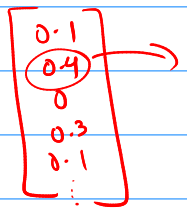
$W \in \mathbb{R}^{m\times n}$

$b \in \mathbb{R}^m$

Output layer → usually of size $1 \times p$ (Regression)

→ K-fold classification problem ($K \times p$)

K classes



$\sum p_i = 1$

$h_4 = W h_3$

(logits → Soft-max activation)

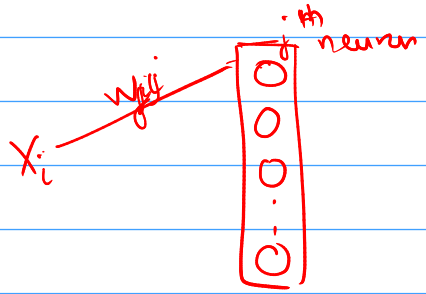
$$y_i = \frac{\exp(h_4(i))}{\sum_j \exp(h_4(j))}$$

$\left(h_i - \max_{i \in [p]} \{h_4(i)\}\right)$ → Then take Softmax

$$Y = f_\theta(X)$$

θ : weights of the NN

Theorem: Any piecewise continuous function can be approximated arbitrarily close using a four-layer NN with ReLU activations.



$\{x_i, y_i\}_{i=1}^{N}$

$W \rightarrow$ weights of a NN

* How do we train a NN?

↳ Loss functions

- Regression → $(\hat{y}_i - y_i)^2$
- Classification → Cross-entropy loss. $-\sum_{i=1}^{k} t_i \log p_i$

$\ell(W; \xi)$

↳ minimize this loss function

Gradient-descent algorithm:

$$\min_{x \in \mathbb{R}^n} f(x)$$

$$\boxed{x(k+1) = x(k) - \eta_k \nabla f(x_k)} \rightarrow \text{Gradient descent}$$

Stochastic Gradient descent (SGD):

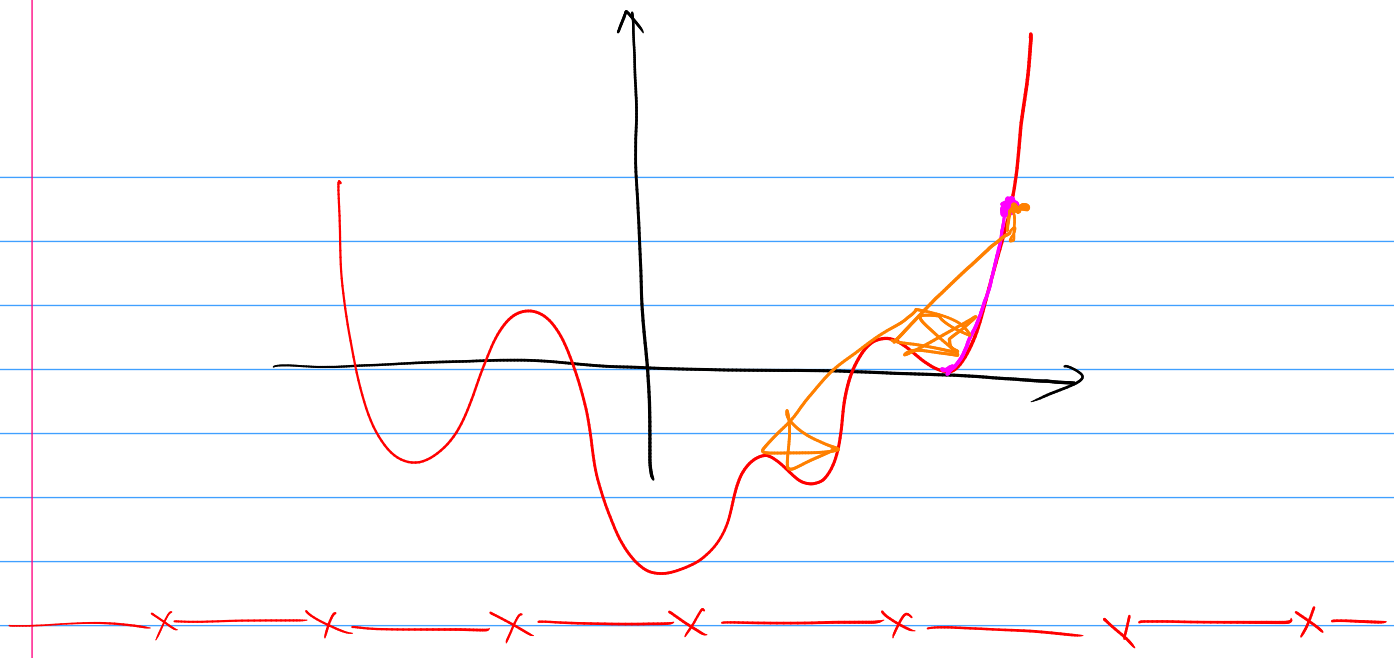
$$W(k+1) = W(k) - \eta_k \nabla L(W(k), \xi_k)$$

↳ Data points sampled in k^{th} iteration

Full-batch Gradient descent (GD):

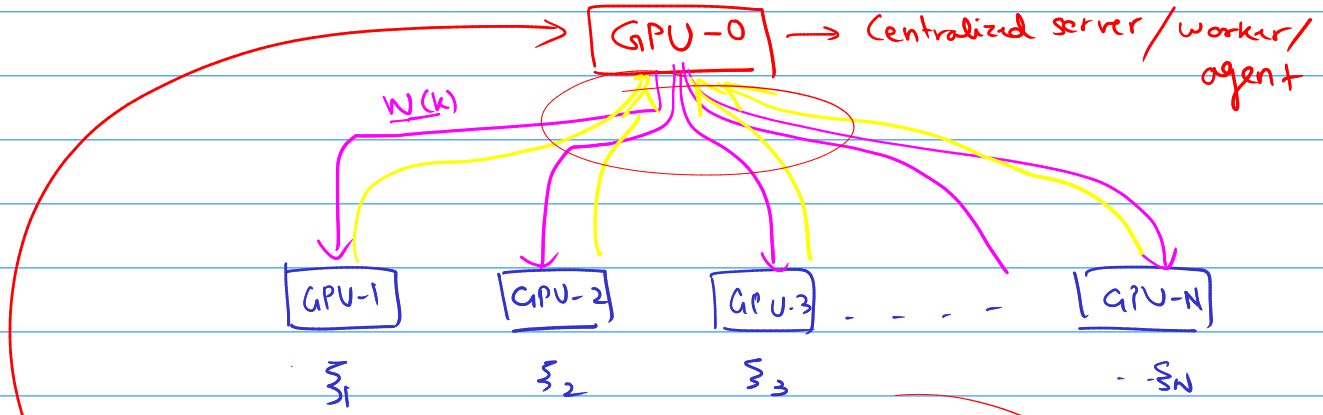
$$W(k+1) = W(k) - \eta_k \frac{1}{N} \sum_{i=1}^{N} \nabla l_{W'}(W(k), x_i)$$

↳ Too-slow (because of gradient computation on all data points)



* How do we work with large scale data?

Parameter - Server Approach



- Privacy concerns
- Limited memory

$$W(k+1) = W(k) - \eta_k \sum_{i=1}^{N} \frac{1}{N_i} \nabla_w L(W(k), \xi_i)$$

- Single point of failure.
- Communication bandwidth requirement scales with the number of servers

* (Baidu) Ring All Reduce algorithm

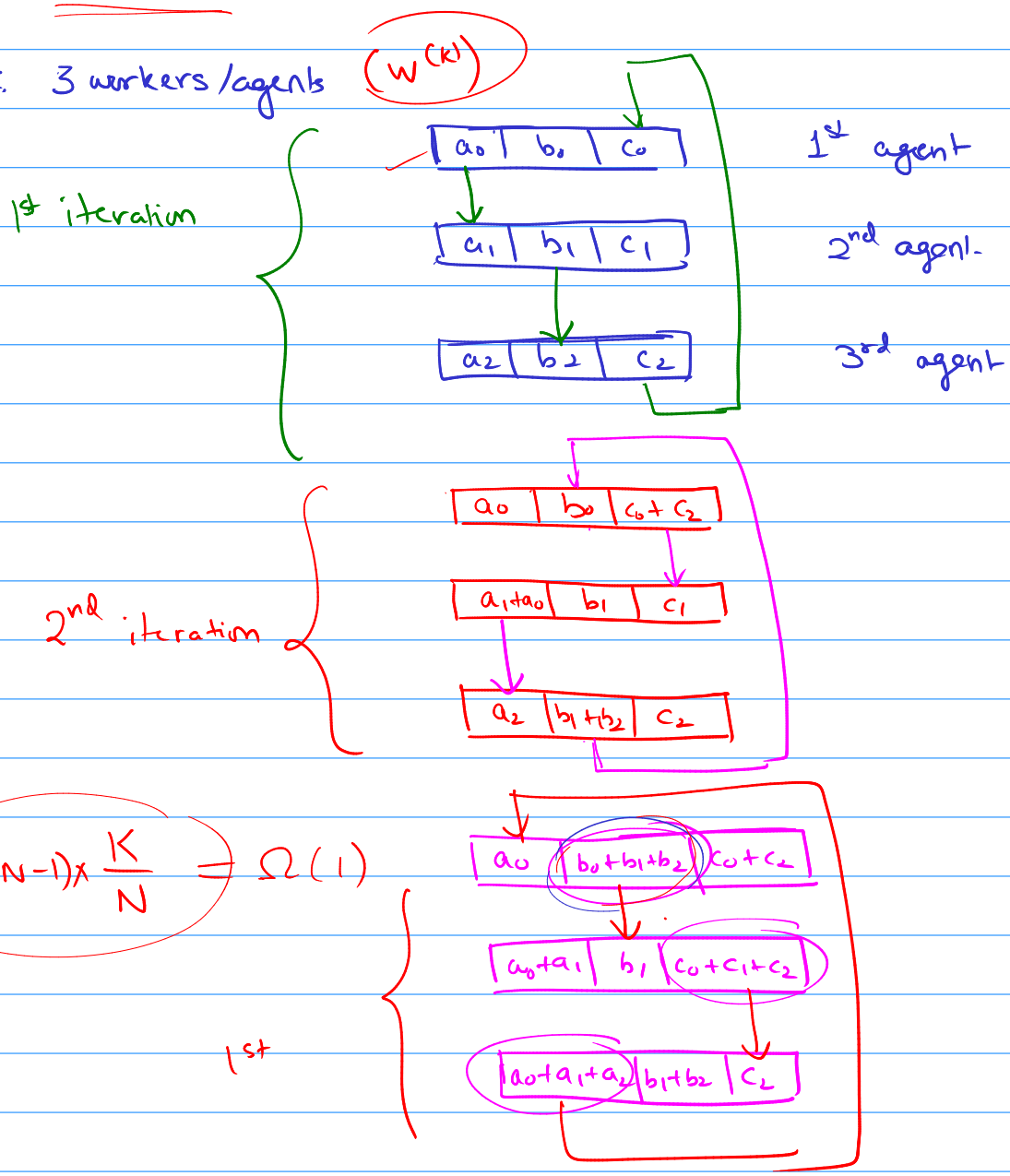
↳ Communication bandwidth requirement is constant in the number of agents.

↳ There is no centralized server.

Scatter Reduce

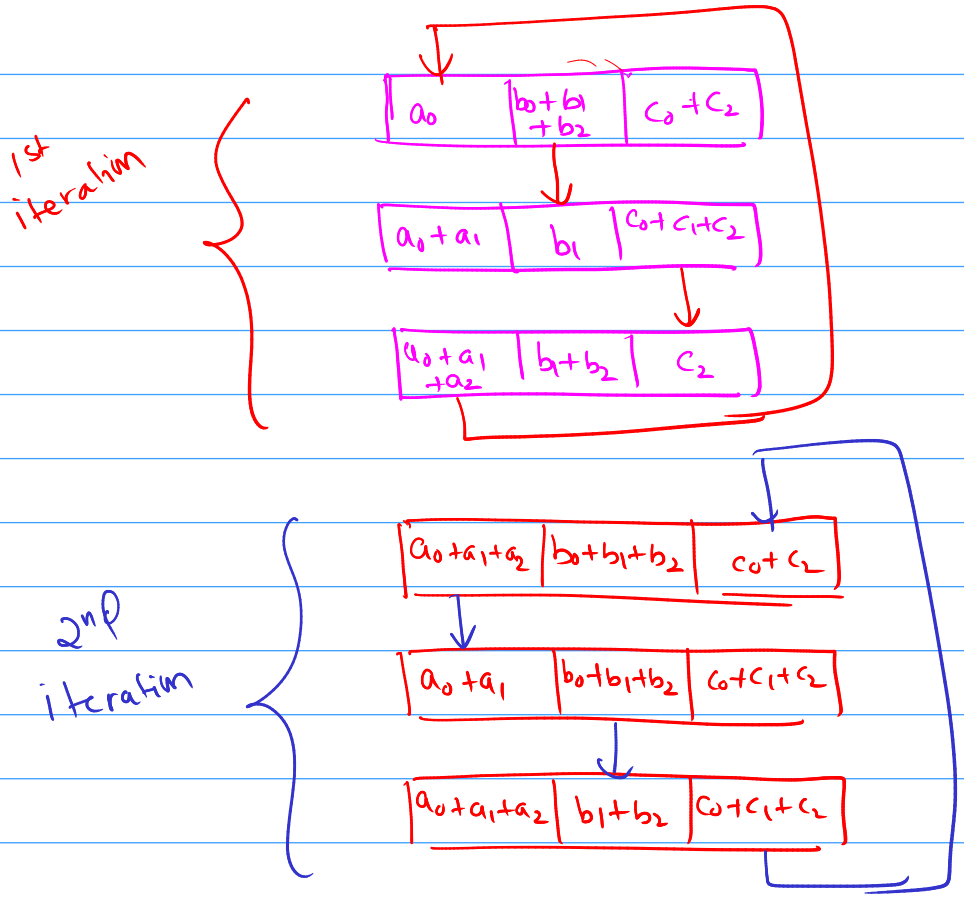
All Gather

Example: 3 workers/agents $(W^{(k)})$



$$2(N-1)\times\frac{K}{N} = \Omega(1)$$

All Gather



Result on SGD:

* Under the assumptions:

(i) The loss function $F(w;\xi)$ is L-smooth in terms of w.

(ii) Stochastic Gradient is unbiased and has bounded variance σ^2.

$$\mathbb{E}_{\xi\sim D}[\nabla F(w^{(k)};\xi)] = \nabla \mathbb{E}_{\xi\sim D}[F(w^{(k)};\xi)] = \nabla f(w^{(k)})$$

then, $\gamma = O\left(\frac{1}{\sqrt{T}}\right)$

$$\frac{1}{T}\sum_{k=0}^{T-1} \mathbb{E}\|\nabla f(w^{(k)})\|^2 = O\left(\frac{\sigma}{\sqrt{T}}\right)$$

ε