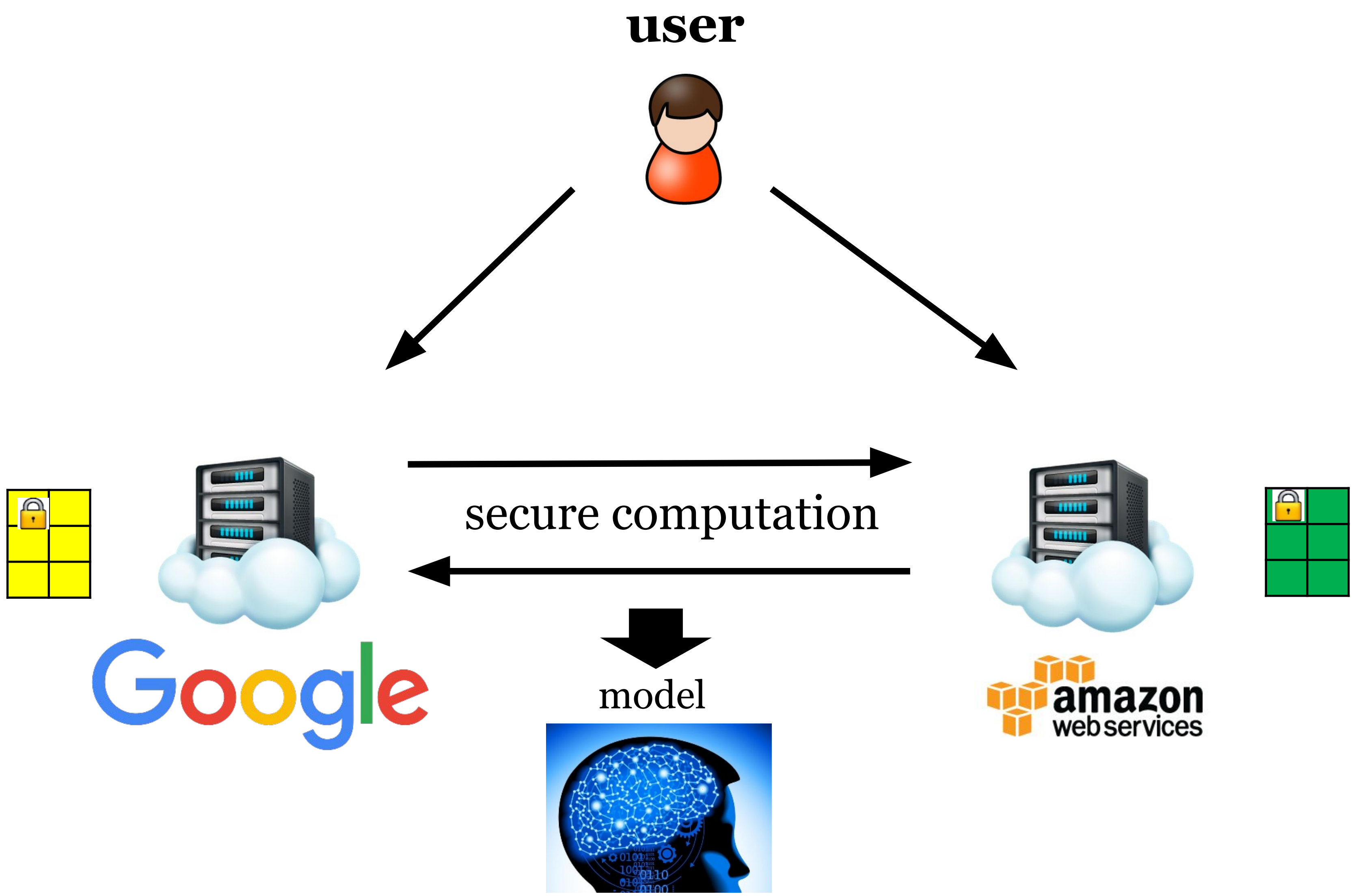
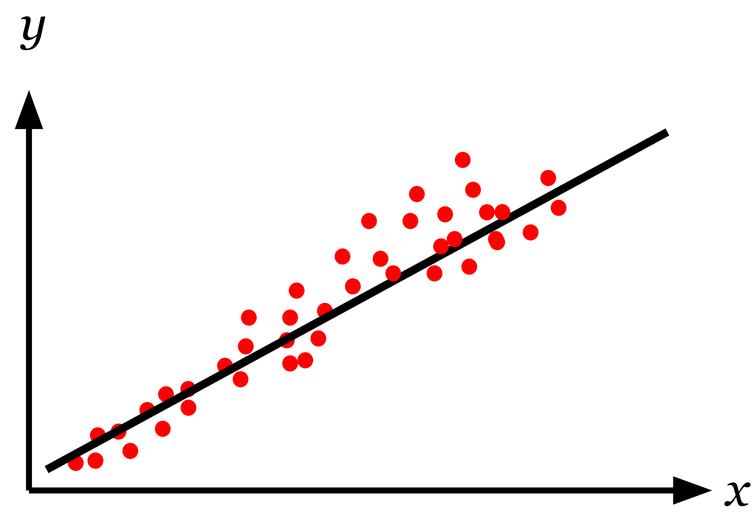
Privacy-preserving Machine Learning

Privacy-preserving machine learning



Linear Regression

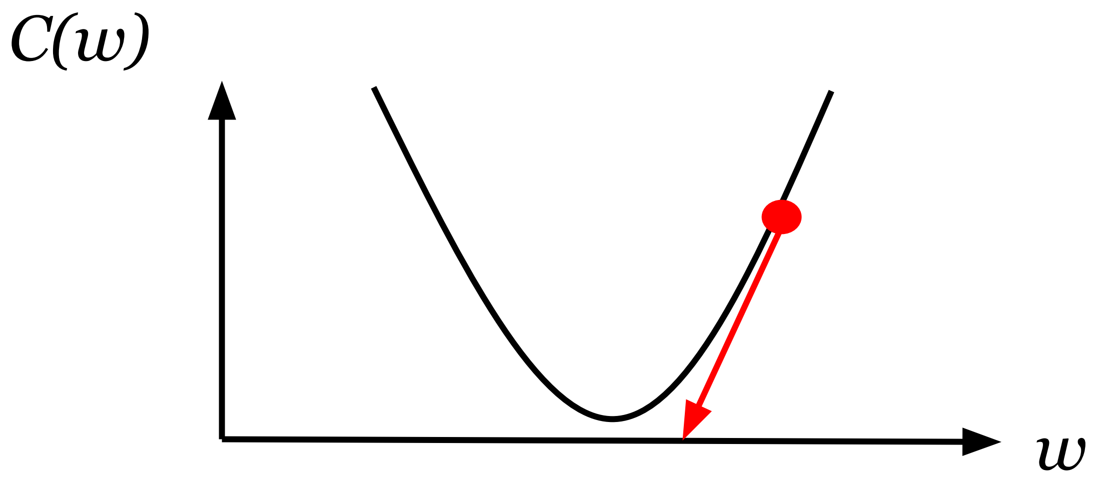


Input: data value pairs (x, y)s

Output: model w

$$y^* = \sum_i w_i x_i = w \cdot x \approx y$$

Stochastic gradient decent (SGD)

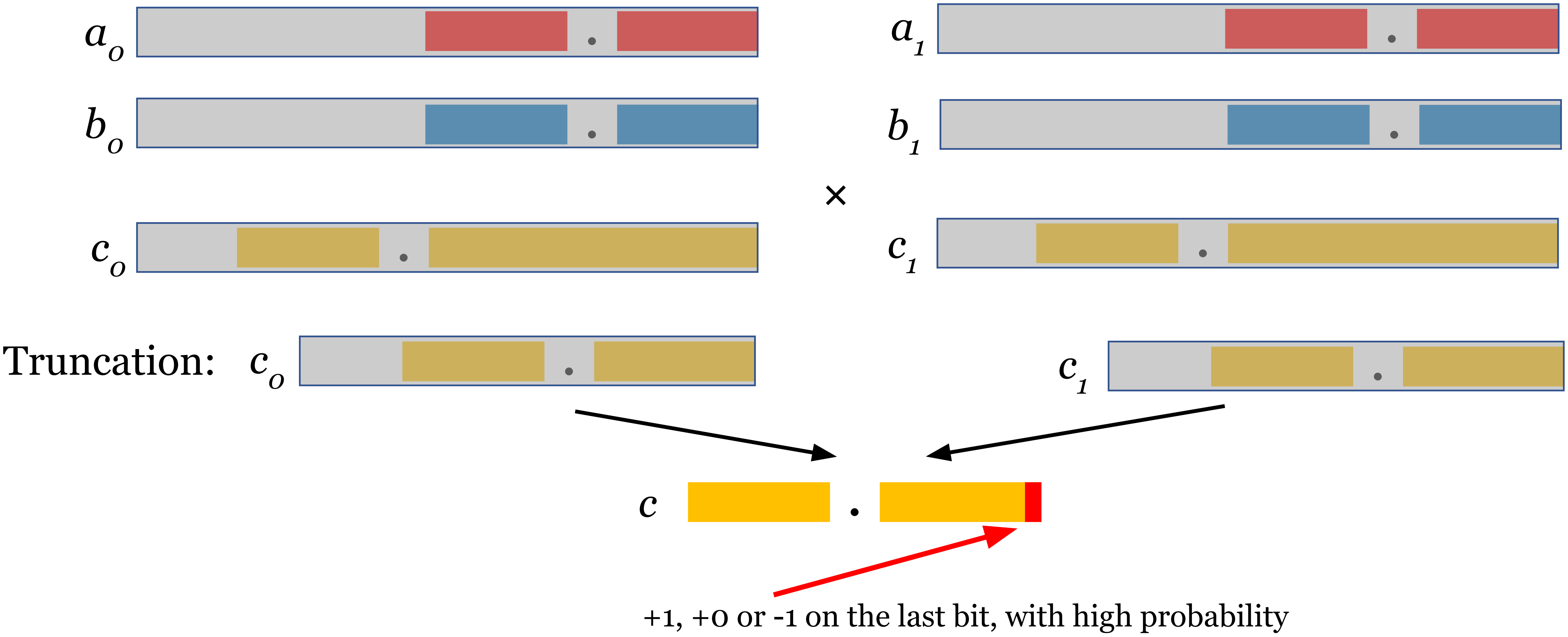


1. Initialize w randomly
2. Select a random sample (x, y), compute derivative of $C_x(w)$
3. Update w

$$w = w - \alpha (x \cdot w - y) x$$

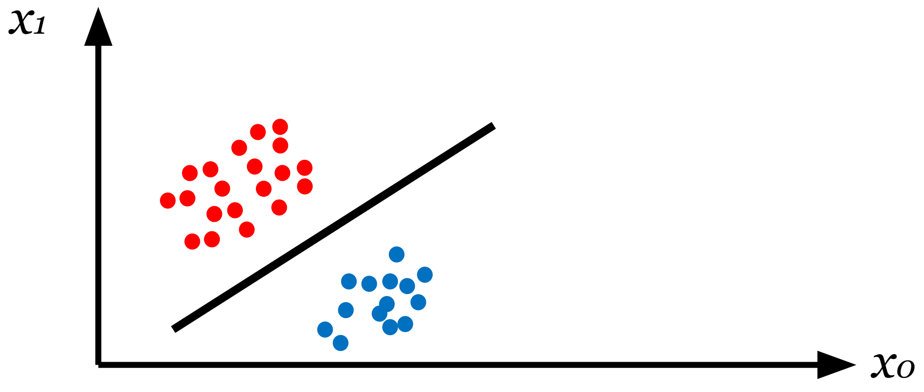
$$w_i = w_i - \alpha (x \cdot w - y) x_i$$

Truncation on shared values



Logistic Regression

Logistic regression

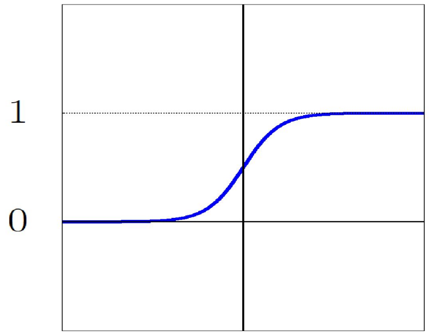


Input: data value pairs (x, y)s y=0 or 1

Output: model w

$$y^* = f(w \cdot x) \approx y$$

$$f(u) = \frac{1}{1+e^{-u}}$$



Cost function

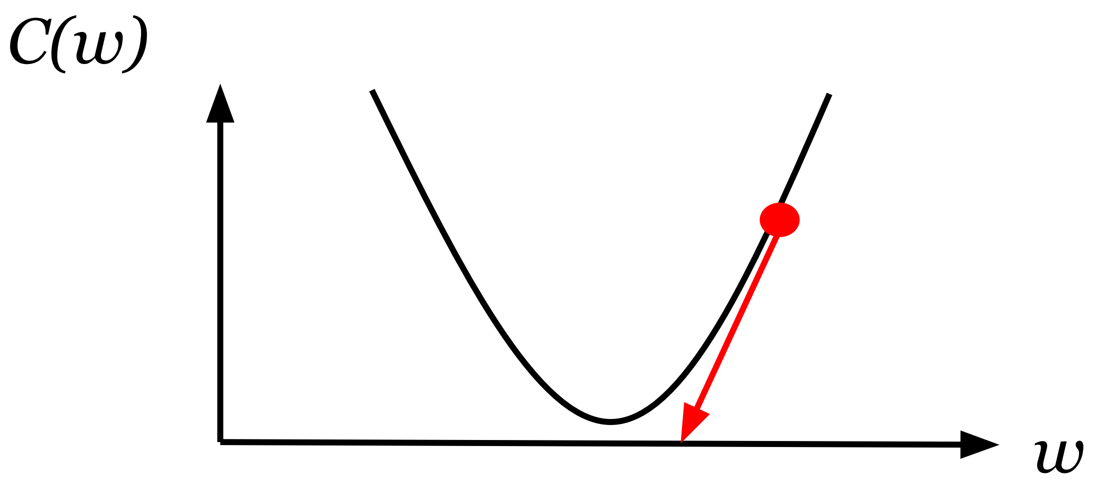
$$y^* = f(w \cdot x) \approx y \qquad f(u) = \frac{1}{1+e^{-u}}$$

- Cross entropy: $C_x(w) = -(y \log y^* + (1-y)\log(1-y^*))$

$$C(w) = \frac{1}{n}\sum_x C_x(w)$$

$$\arg\min_w C(w)$$

Stochastic gradient decent (SGD)



$$y^* = f(w \cdot x) \qquad f(u) = \frac{1}{1+e^{-u}}$$

- $C_x(w) = -(y \log y^* + (1-y)\log(1-y^*))$

1. Initialize w randomly
2. Select a random sample (x, y), compute derivative of $C_x(w)$
3. Update w

$$w_i = w_i - \alpha(f(x \cdot w) - y)x_i$$

Privacy-preserving logistic regression

SGD: $w_i = w_i - \alpha(f(x \cdot w) - y)x_i$

1. Users secret share data and values (x,y)

2. Servers initialize and secret share the model w

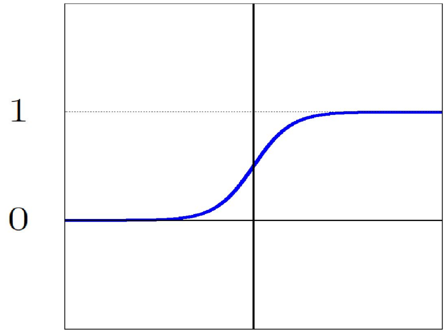
3. Run SGD using GMW protocol

4. Truncate the shares after every multiplication

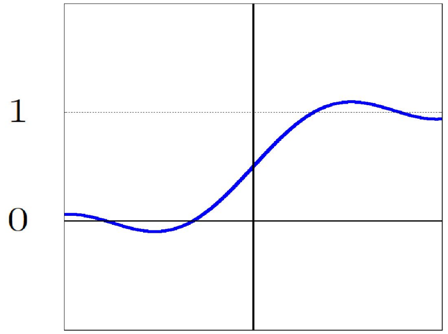
Privacy-preserving Logistic Regression

SGD: $w_i = w_i - \alpha(f(x \cdot w) - y)x_i$

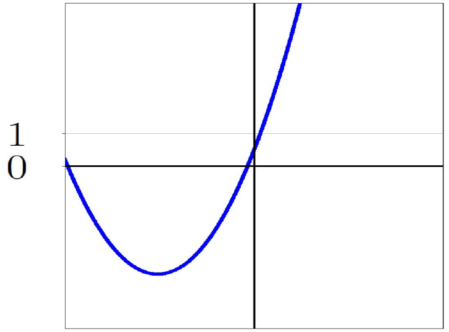
Logistic function



degree 10 polynomial



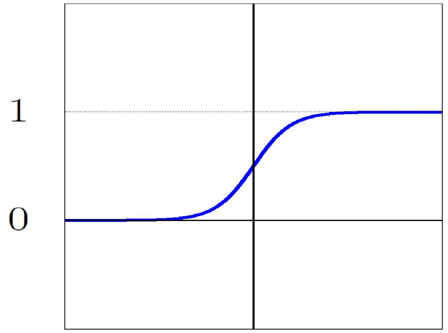
degree 2 polynomial



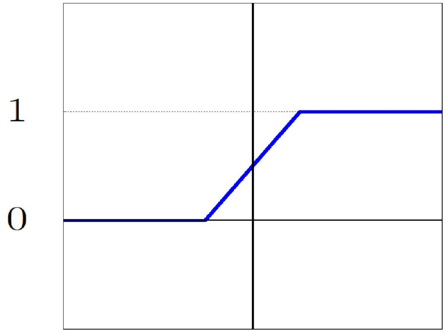
Privacy-preserving Logistic Regression

SGD: $w_i = w_i - \alpha(f(\mathbf{x} \cdot \mathbf{w}) - y)x_i$

Logistic function



Our function



Almost the same accuracy as logistic function

Much faster than polynomial approximation

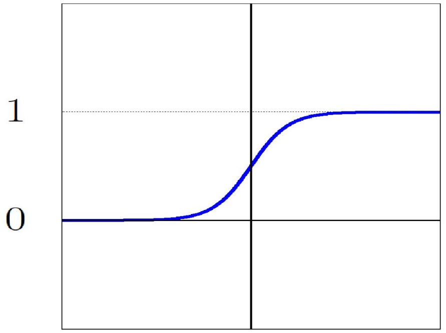
Secure-computation-friendly activation function

$$f(x) = \begin{cases} 0, & \text{if } x < -\frac{1}{2} \\ x + \frac{1}{2}, & \text{if } -\frac{1}{2} \leq x \leq \frac{1}{2} \\ 1, & \text{if } x > \frac{1}{2} \end{cases}$$

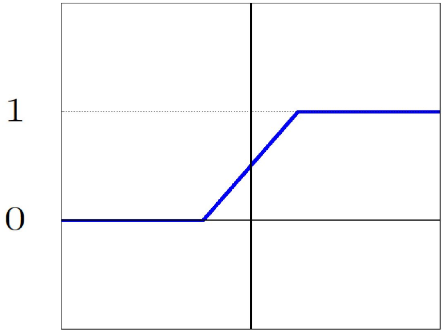
Privacy-preserving Logistic Regression

SGD: $w_i = w_i - \alpha(f(\mathbf{x} \cdot \mathbf{w}) - y)x_i$

Logistic function



Our function



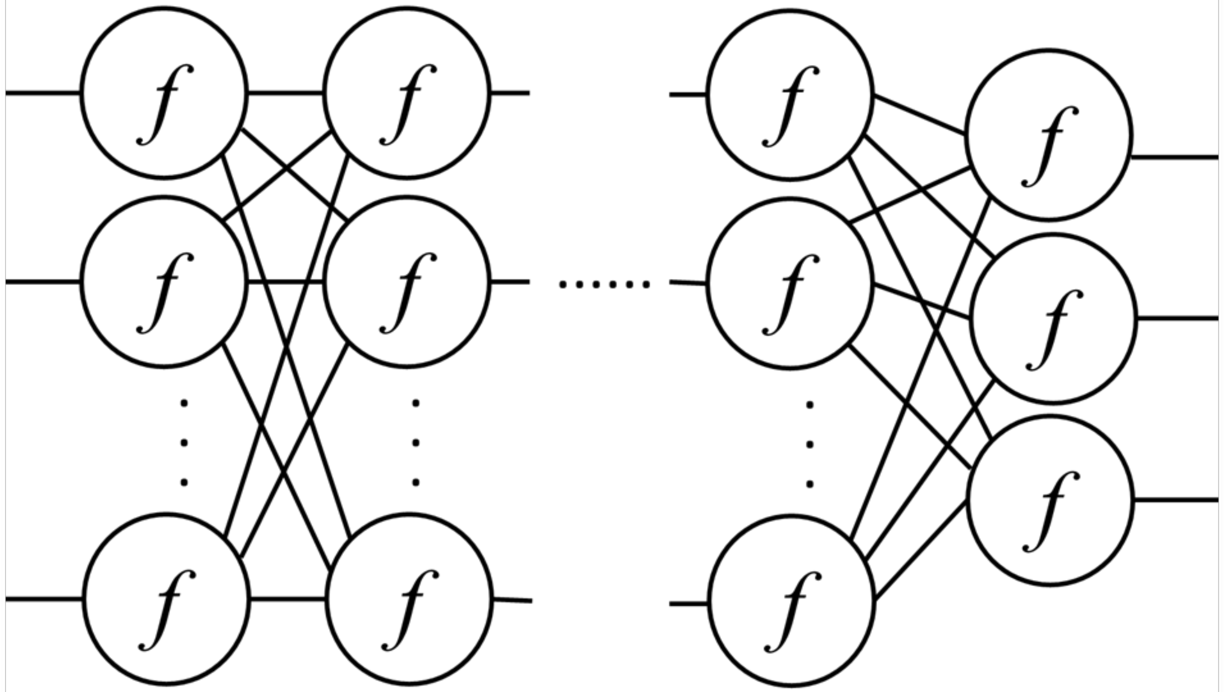
- Run our protocol for linear regression
- Switch to garbled circuit for f [DSZ15]
- Switch back to arithmetic secret sharing

Neural networks

m-1 hidden layers

$X_0 = X$

$B \times d_0$



W_1

$d_0 \times d_1$

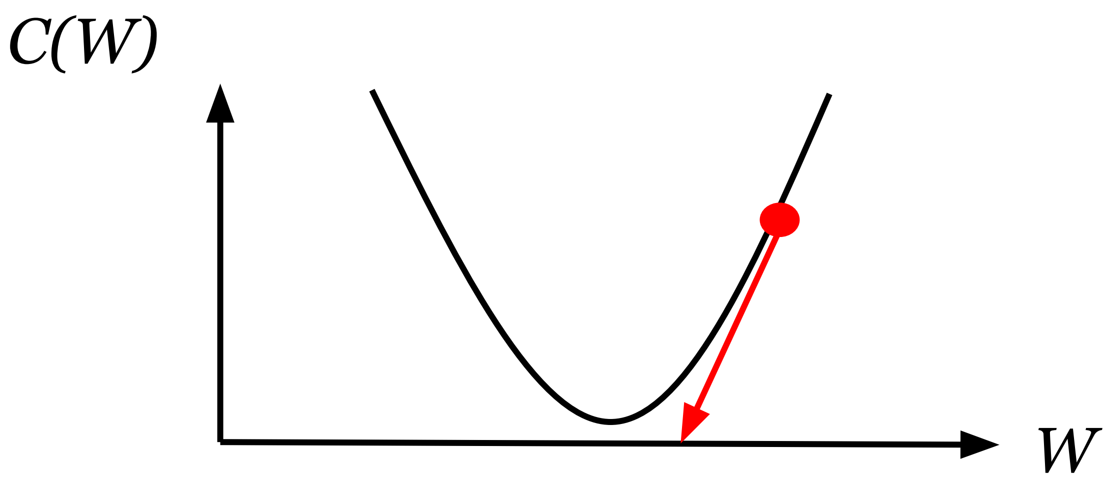
$$X_i = f(U_i) = f(X_{i-1} \times W_i)$$

$B \times d_i$

$Y = X_m$

$B \times d_m$

SGD: closed-form update for Wi



$$W_i = W_i - \alpha \frac{\partial C(W)}{\partial W_i}$$

$$X_i = f(U_i) = f(X_{i-1} \times W_i)$$

$$W_m = W_m - \alpha \frac{\partial C(W)}{\partial W_m} = W_m - \alpha \frac{\partial C(W)}{\partial X_m} \odot \frac{\partial X_m}{\partial U_m} \odot \frac{\partial U_m}{\partial W_m} = W_m - \frac{\alpha}{B} X_{m-1}^T \times (Y^* - Y)$$

$$W_{m-1} = W_{m-1} - \alpha \frac{\partial C(W)}{\partial W_{m-1}} = W_{m-1} - \alpha \frac{\partial C(W)}{\partial X_m} \odot \frac{\partial X_m}{\partial U_m} \odot \frac{\partial U_m}{\partial X_{m-1}} \odot \frac{\partial X_{m-1}}{\partial U_{m-1}} \odot \frac{\partial U_{m-1}}{\partial W_{m-1}}$$

$$W_i = W_i - \alpha \frac{\partial C(W)}{\partial W_i} = W_i - \alpha \frac{\partial C(W)}{\partial X_m} \odot \frac{\partial X_m}{\partial U_m} \odot \frac{\partial U_m}{\partial X_{m-1}} \odot \frac{\partial X_{m-1}}{\partial U_{m-1}} \odot \dots \odot \frac{\partial X_i}{\partial U_i} \odot \frac{\partial U_i}{\partial W_{i-1}}$$

Memorization

-

$$X_i = f(U_i) = f(X_{i-1} \times W_i)$$

$$Y_m = \frac{\partial C(W)}{\partial X_m} \odot \frac{\partial X_m}{\partial U_m}$$

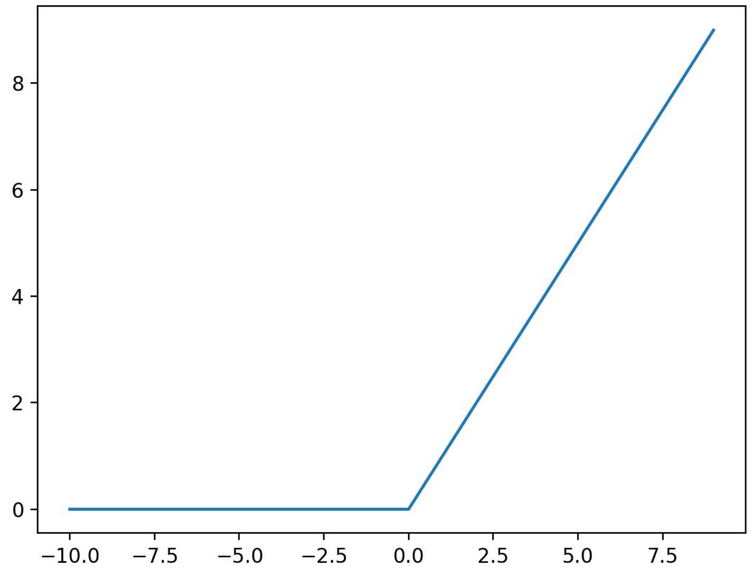
$$Y_i = \left(Y_{i+1} \times W_i^T\right) \odot \frac{\partial X_{m-1}}{\partial U_{m-1}}$$

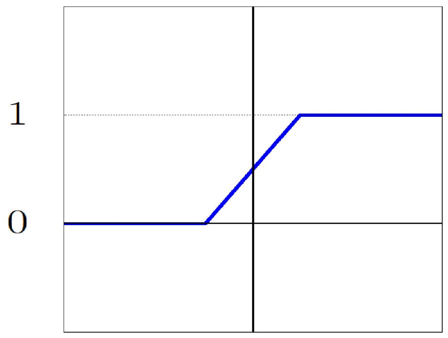
$$W_i = W_i - \frac{\alpha}{B} X_i^T \times Y_i$$

Activation functions

- Softmax: $f(u_i) = \frac{e^{u_i}}{\sum_j e^{u_j}}$
  - Last layer of neural network, probability distribution

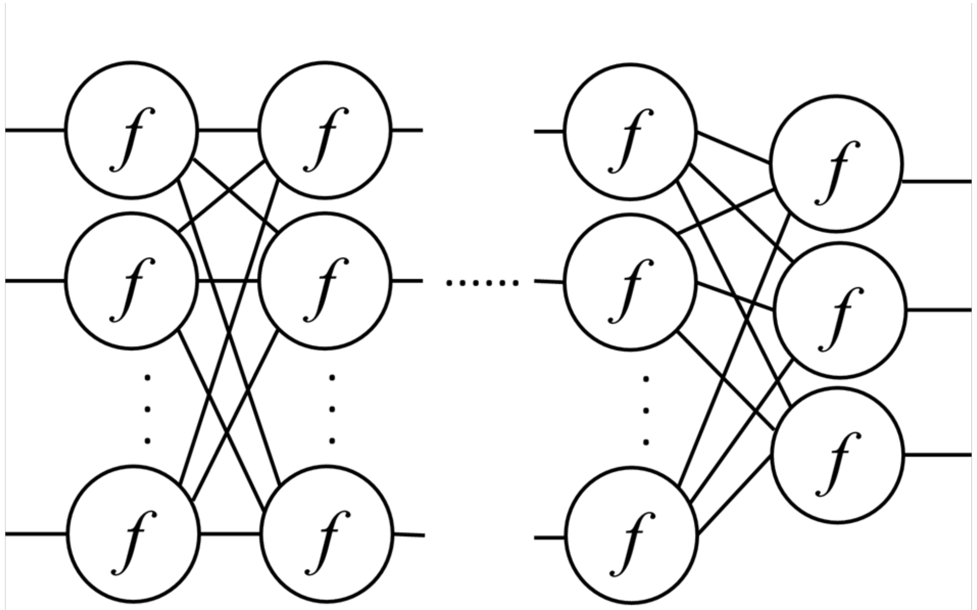
- ReLU:



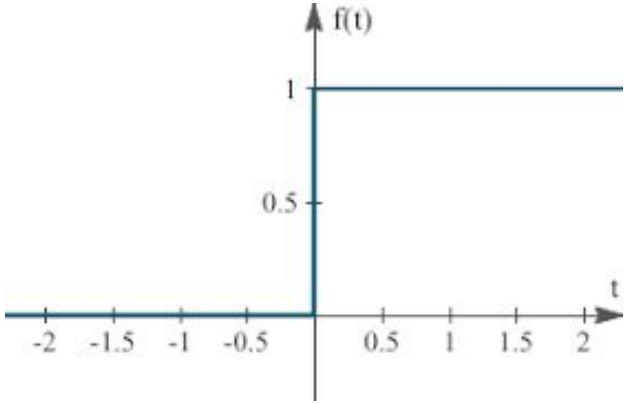


$$f(x) = \begin{cases} 0, & \text{if } x < -\frac{1}{2} \\ x + \frac{1}{2}, & \text{if } -\frac{1}{2} \leq x \leq \frac{1}{2} \\ 1, & \text{if } x > \frac{1}{2} \end{cases}$$

Neural networks



Step function:



Privacy-preserving neural networks

- Truncation on shared data
- Switch to Garbled circuit to evaluate ReLU
- Vectorization

Problems

- Efficiency
- Accuracy

Recent research

- Privacy-preserving neural network predictions
- Approximate/simplify activation functions
  - Polynomial approximation/square activation
  - Binarized NN/quantization
- Non-crypto preprocessing

Interesting directions

- Real data and applications

- Traditional machine learning
  - Decision tree and random forest (RAM model instead of circuit)