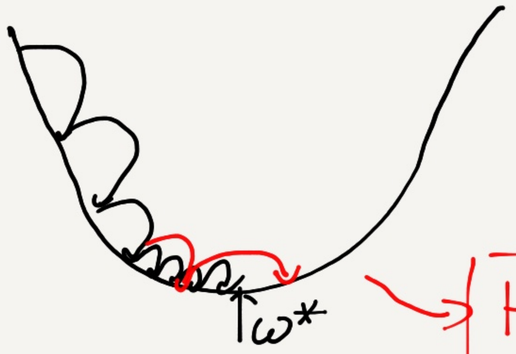
Lecture 8.

1. SGD (mini-batch) review.



Hw2:
lr = 0.001 for GD ??

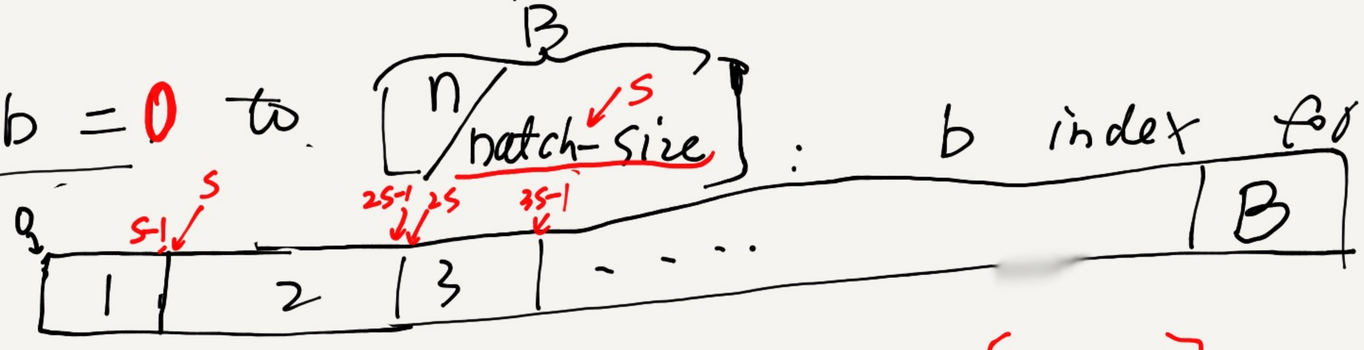
For $i = 1$ to max_epoch:

1) shuffle the training set, (X_train, y_train)

X_train: x_1, x_2, ..., x_n

y_train: y_1, y_2, ..., y_n

2) for b = 0 to $\lfloor n / \text{batch\_size} \rfloor$: b index for batches.



2.1 mini-batch $X\_batch = X\_train\_shuffled[b*S, (b+1)*S-1]$
$y\_batch = y\_train\_shuffled[b*S, (b+1)*S-1]$

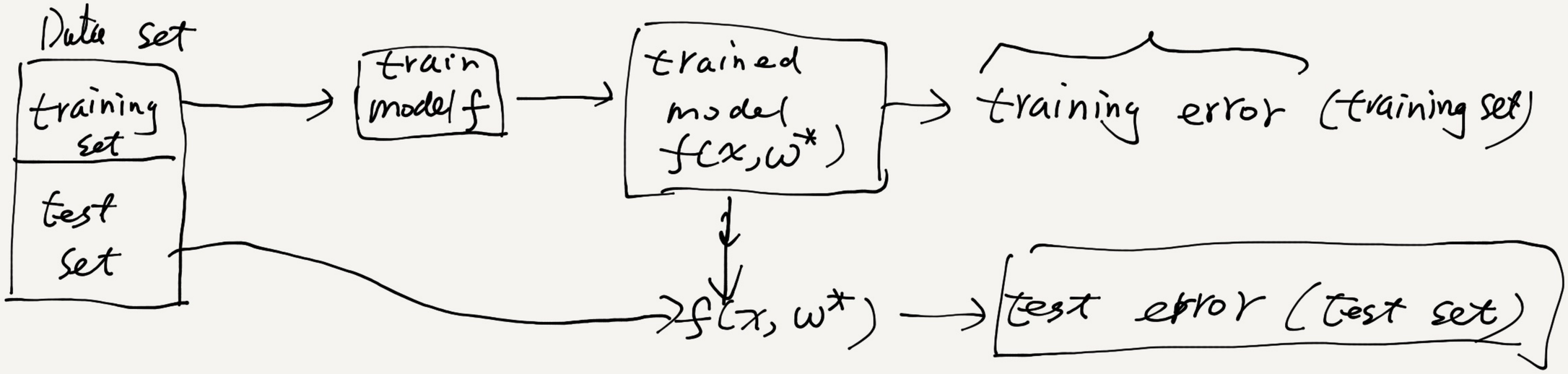
2.2 prepare matrix A and b. using the bth batch.

2.3 Calculate gradient $\hat{g} = A^T(Aw - b)$

2.4. update $w = w + \varepsilon \cdot (-\hat{g})$

3) Decay the learning rate $\varepsilon = \varepsilon * 0.9$.

2. Generalization:



Test performance is also used to evaluate model's ability to perform on new/unseen data.

↓ generalizability.

Error metrics:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

($\hat{y}_i$: prediction, y_i: target)

MAE: mean absolute error

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|$$

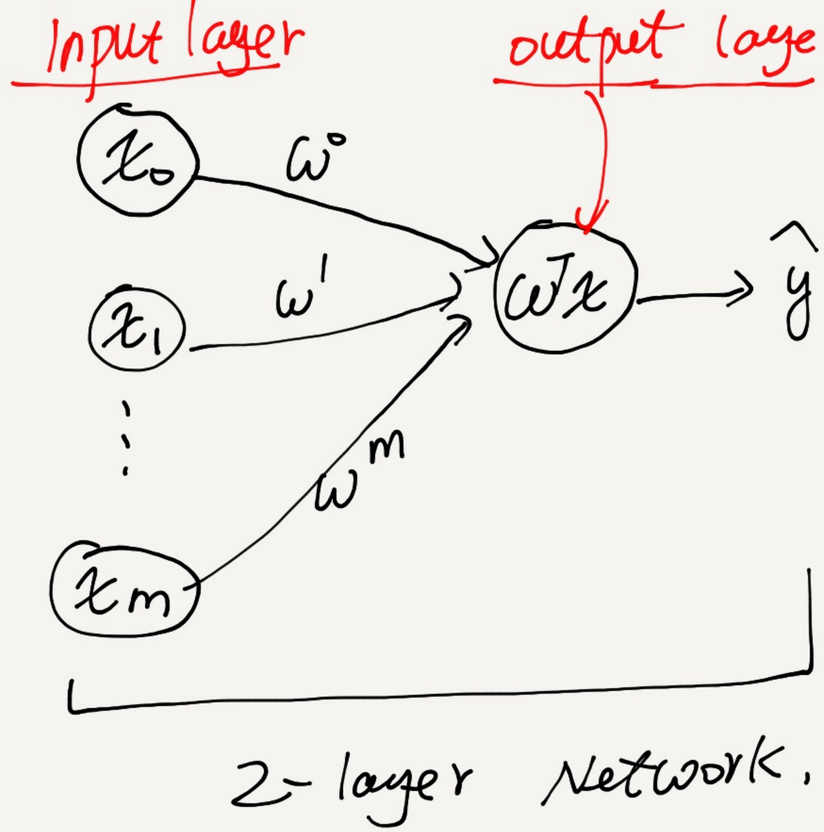
(abs.)

3. Artificial Neural Networks (ANNs / NNs)

1) From Linear Regression (LR) to NNs

(1) model in LR: $f(x) = \omega^T \cdot x$ $\quad x = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_m \end{pmatrix}$ $\quad \omega = \begin{pmatrix} \omega_0 \\ \omega_1 \\ \vdots \\ \omega_m \end{pmatrix}$

(2) use a graph to represent $f(x)$



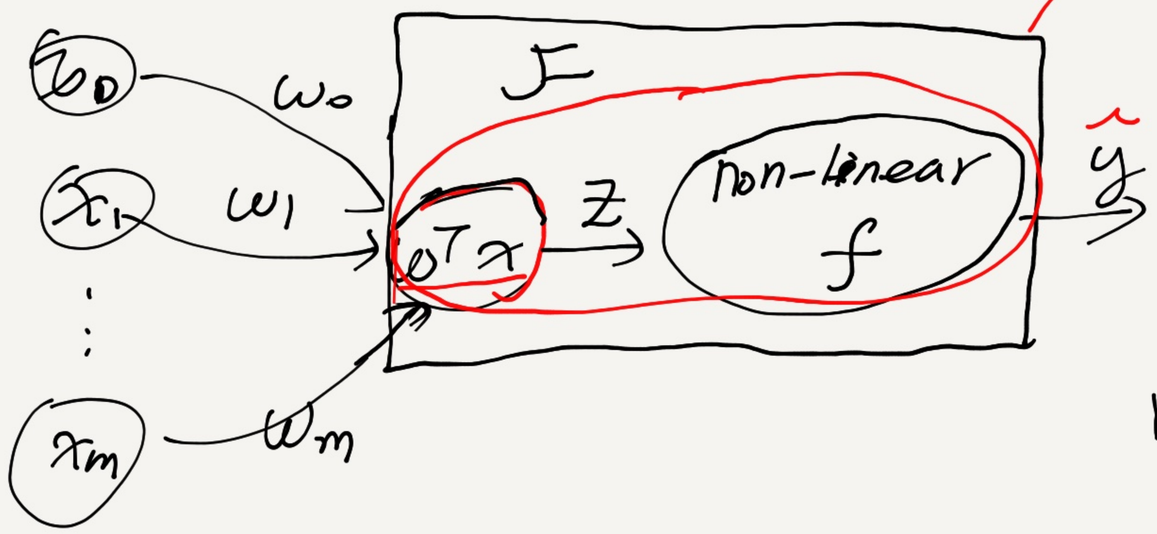
2-layer Network.

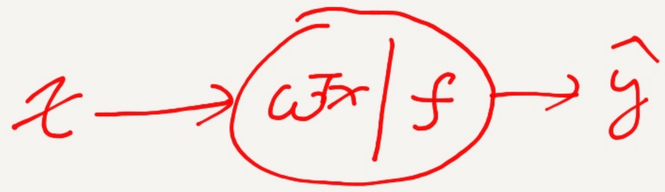
The linear model can only learn learn relationship between x and y.

Q1: How can we improve it to learn non-linear relationship?

Q2: Can we improve to model complex problems?

2. non-linear capacity.



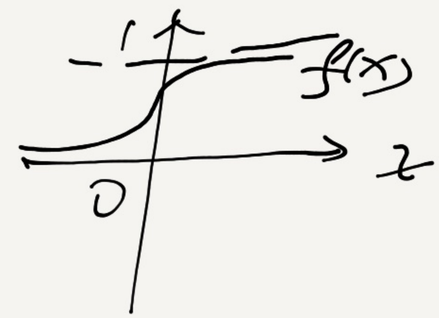


$z = w^T x$.

$F(x) = f(w^T x)$

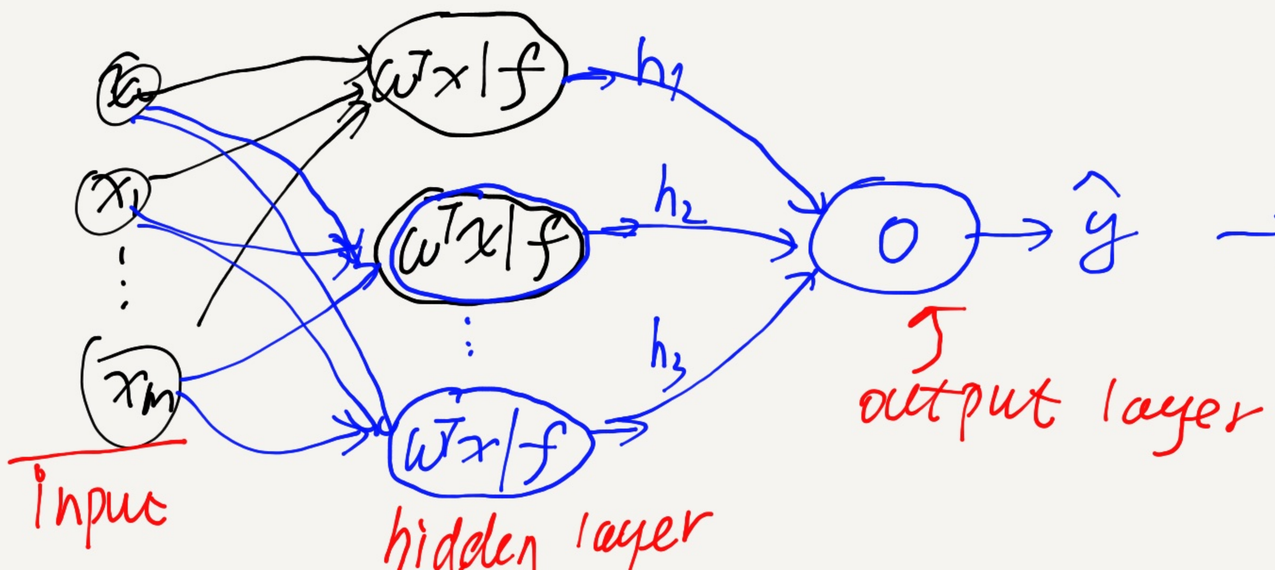
non linear f: activation function

sigmoid function: $f(x) = \frac{1}{1+e^{-x}}$



3, Enhance the learning capacity to deal with complex problems?

Add "building blocks"



→ 3-layer network