DATASCI 347 Machine Learning

Lecture 18: Neural Networks

Ruoxuan Xiong

Suggested reading: ISL Chapter 10

Announcements

- Extension of HW3 deadline to Friday 4/3

Lecture plan

- Gradient descent
- Clustering methods

Single-layer neural network

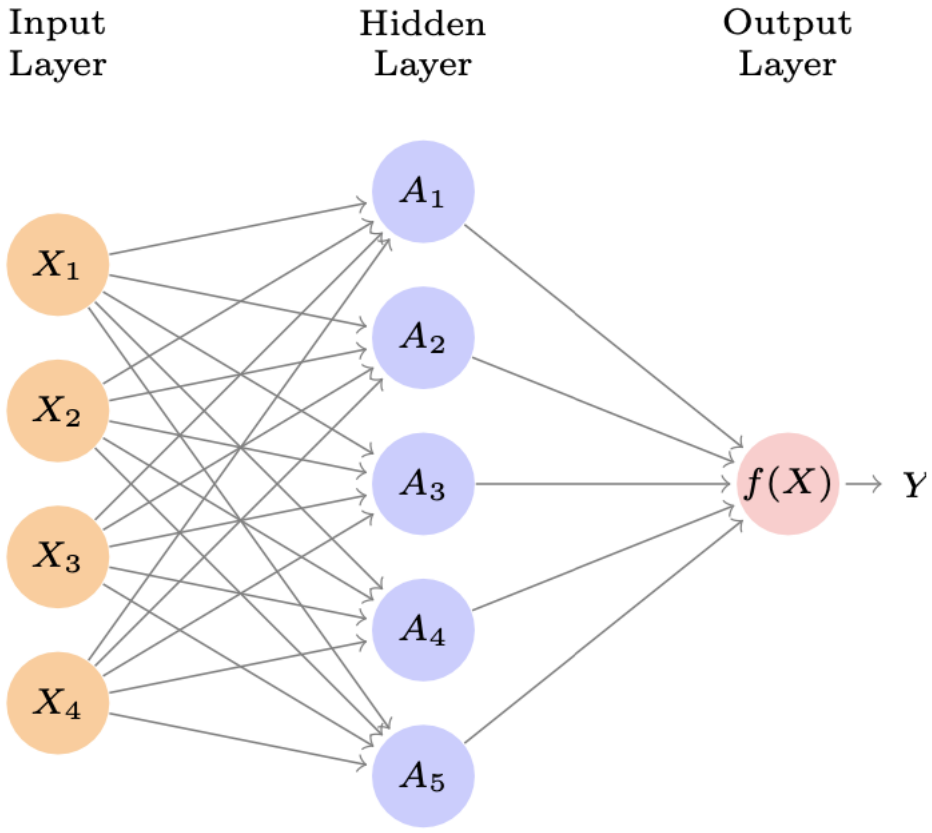
- The neural network model has the form

$$f(X) = \beta_0 + \sum_{k=1}^{K} \beta_k h_k(X)$$

Here $h_k(X)$ is a neuron / hidden unit (analogous to neurons in the brain)

$$A_k = h_k(X) = g\left(w_{k0} + \sum_{j=1}^{p} w_{kj} X_j\right)$$

g is a nonlinear activation function



Loss function to fit a neural network

- Fitting a neural network requires estimating the unknown parameters (β_k and w_{kj})

- For a regression problem, typically use *squared-error*

$$\sum_{i=1}^{n}\left(y_i - f(x_i)\right)^2$$

- For a classification problem, typically use *cross-entropy (negative multinomial log-likelihood)*

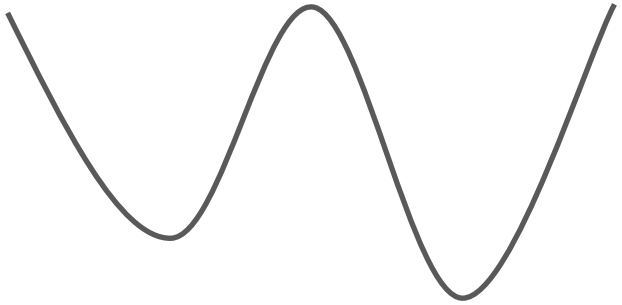
$$-\sum_{i=1}^{n}\sum_{m} y_{im}\log(f_m(x_i))$$

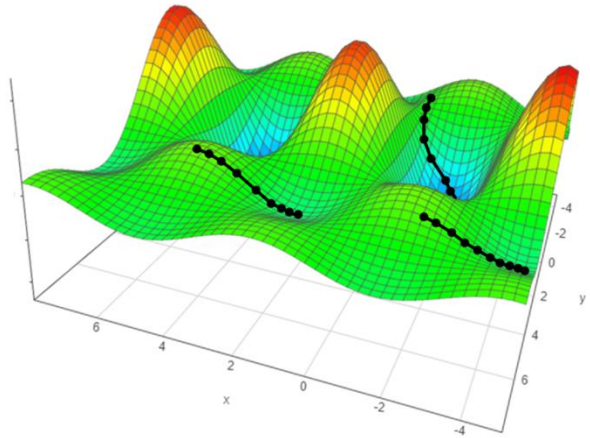
Gradient descent

- After setting up the loss $l(f(x_i), y_i)$, we set up an algorithm to minimize the empirical loss, measured as the averaged loss on the training set

$$\hat{L}(f_\theta) = \frac{1}{n} \sum_{1 \le i \le n} l(f_\theta(x_i), y_i)$$

- We use optimization algorithms like gradient descent that are quick to run





The gradient descent algorithm

- Let θ_t be the parameters of a neural network
- Let f_{θ_t} be the neural network
- Let $\nabla\hat{L}(f_{\theta_t})$ be the gradient of the training loss at θ_t

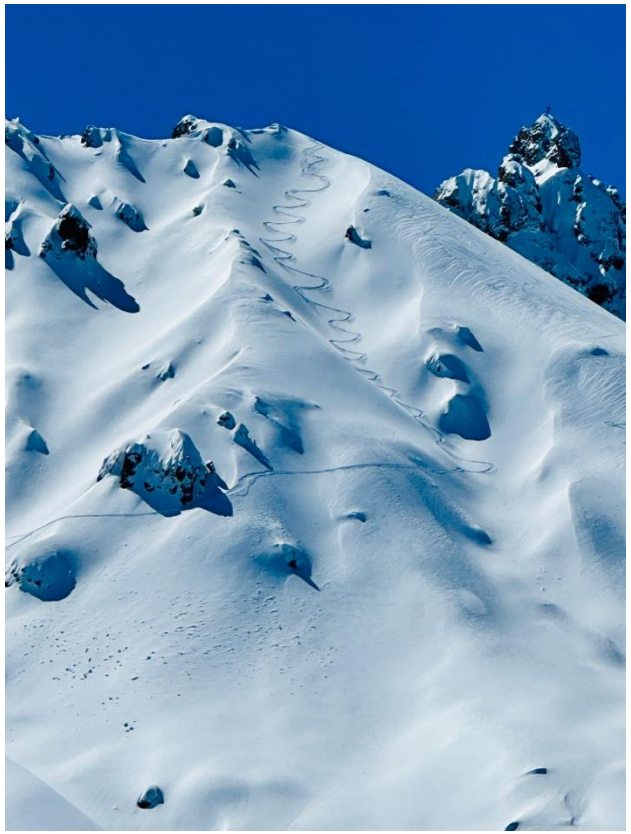
- Let η be a learning rate parameter

$$\theta_{t+1} \leftarrow \theta_t - \eta \cdot \nabla\hat{L}(f_{\theta_t})$$

- Example: gradient descent for quadratic loss

Stochastic gradient descent

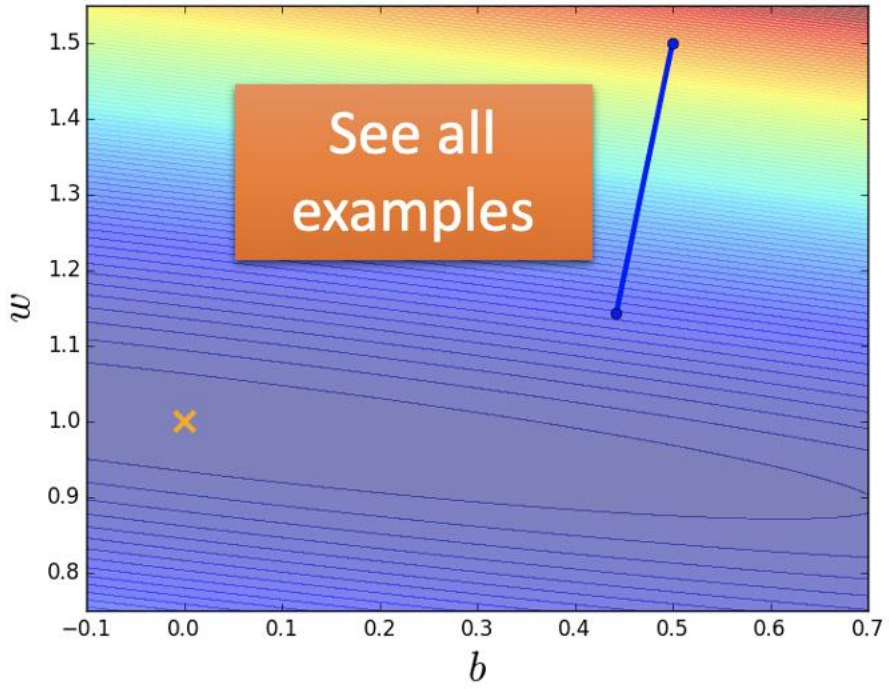
- **Motivation:** If the dataset is highly redundant, the gradient on the first half is almost identical to the gradient on the second half
  - Mini-batch stochastic gradient descent



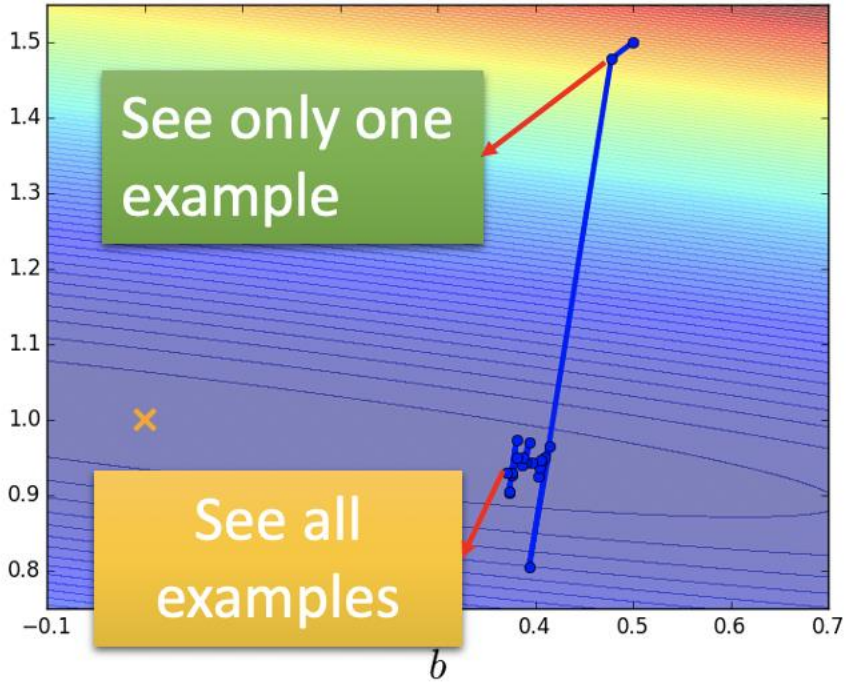
EMORY

GD vs. SGD

- **Gradient Descent:** Update after seeing all examples



- **Stochastic Gradient Descent:** Update for each example



EMORY

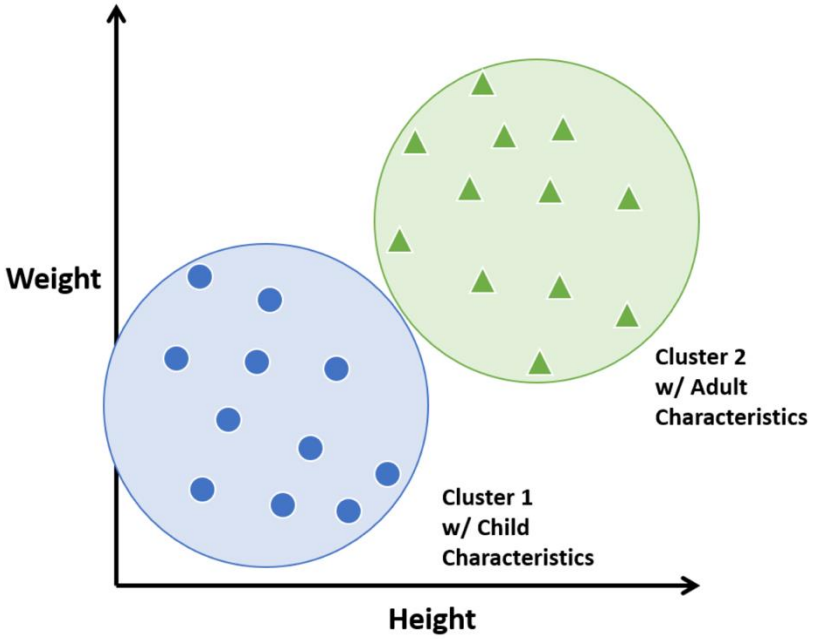
Lecture plan

- Gradient descent

- Clustering methods

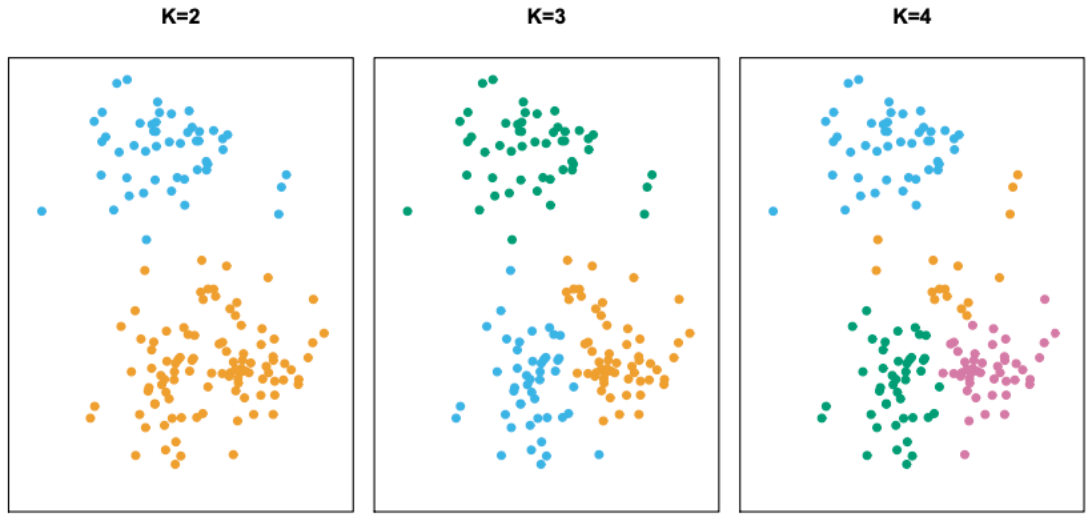
Clustering methods

- PCA looks to find a low-dimensional representation of observations
- Clustering looks to find homogeneous subgroups among observations
- A marketing example
  - Perform market segmentation by identifying subgroups who are more likely to purchase a particular product
  - Use features, e.g., median household income, occupation, distance form nearest urban area



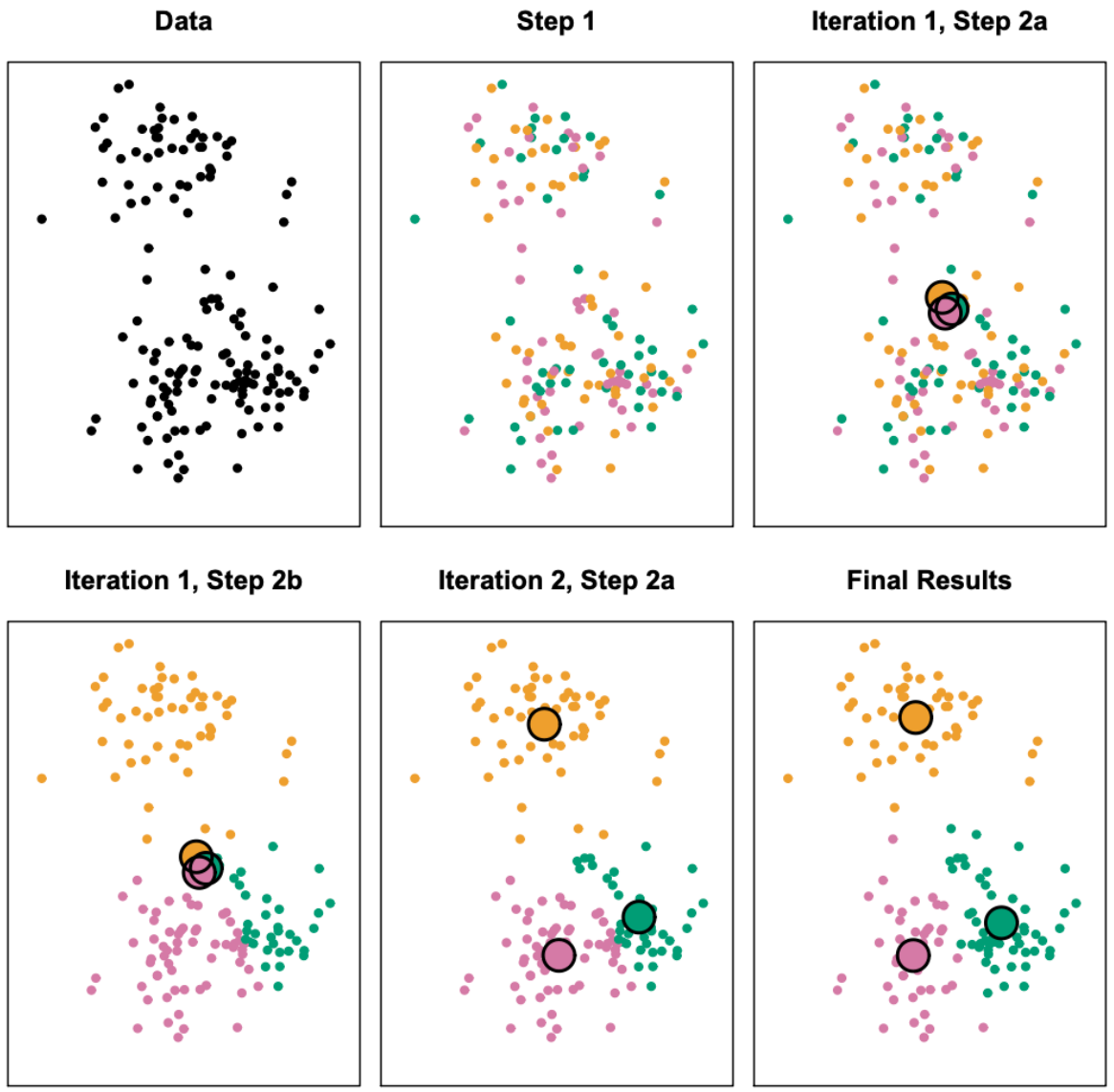
K-means clustering

- A *good clustering* is one for which the within-cluster variation is as small as possible
- **K-means clustering** is a simple and elegant approach for partitioning a data set into *K distinct, non-overlapping clusters*
  - Let $C_1, \cdots, C_K$ denote set of indices of observations in each cluster
    - $C_1 \cup \cdots \cup C_K = \{1, \cdots, n\}$: each observation belongs to at least one cluster
    - $C_k \cap C_{k'} = \emptyset$ for all $k \neq k'$: no observation belongs to more than one cluster



K-means clustering procedure

- *Randomly assign* a number, from 1 to K, to each observation, i.e., initial cluster assignment
- *Iterate* until the cluster assignments stop changing
  - For each cluster, *compute the cluster centroid*, i.e., vector of the p feature means for observations in the kth cluster
  - *Assign each observation* to the *cluster* whose *centroid is closest*



EMORY

Initialization of K-means clustering

- The *results* of K-means clustering will *depend on the initial cluster assignment*
- Solution: *run* the algorithm *multiple times* from different random initial configurations. Then one *selects the best solution*

K-means clustering performed six times with $K = 3$ and each time with a different initial cluster. Those labeled in red are optimal, with an objective value of 235.8.