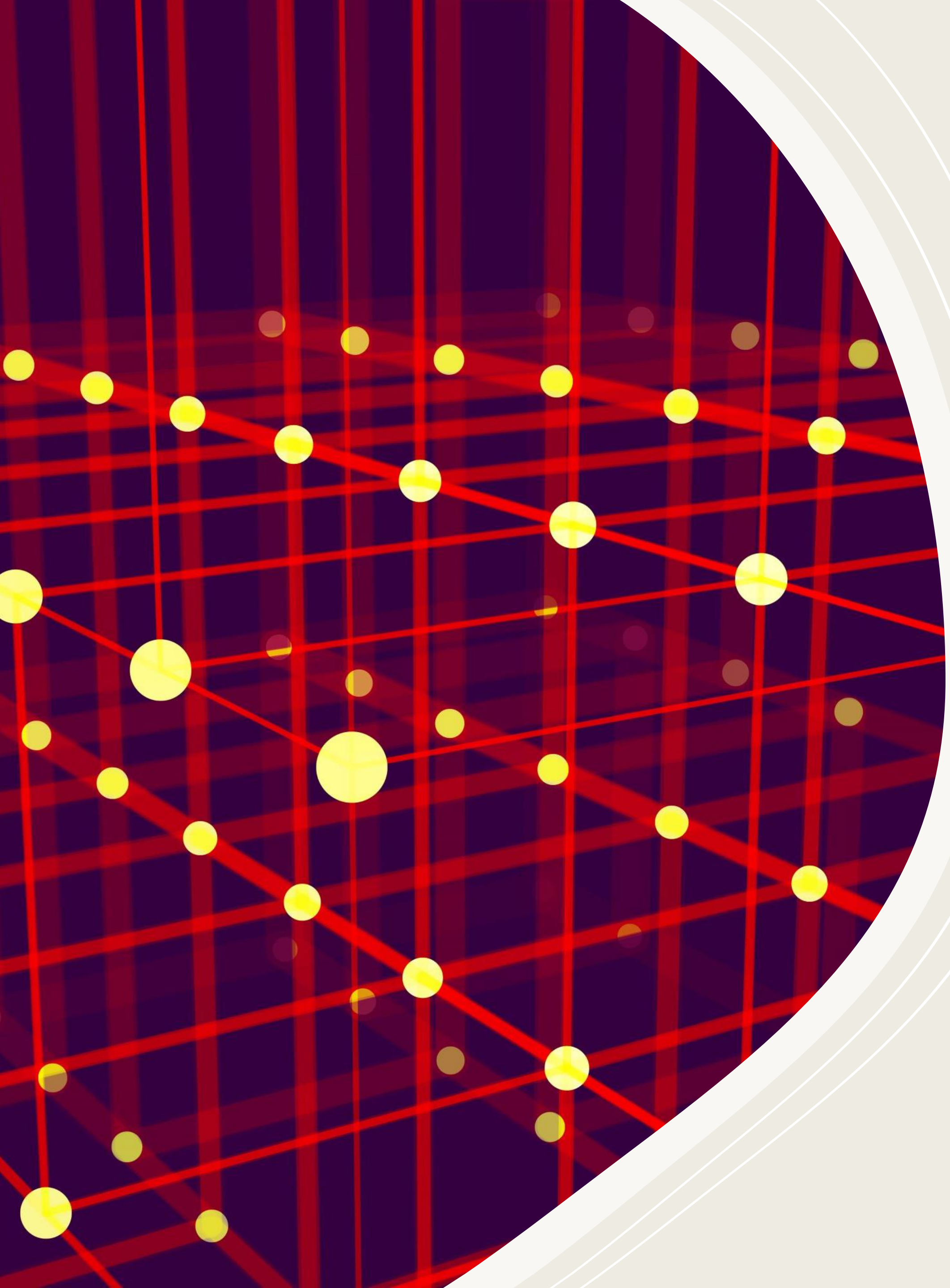


CPSC 340 – Tutorial 8

Lironne Kurzman
lironnek@cs.ubc.ca

Slides courtesy of Nam Hee Kim

University of British Columbia

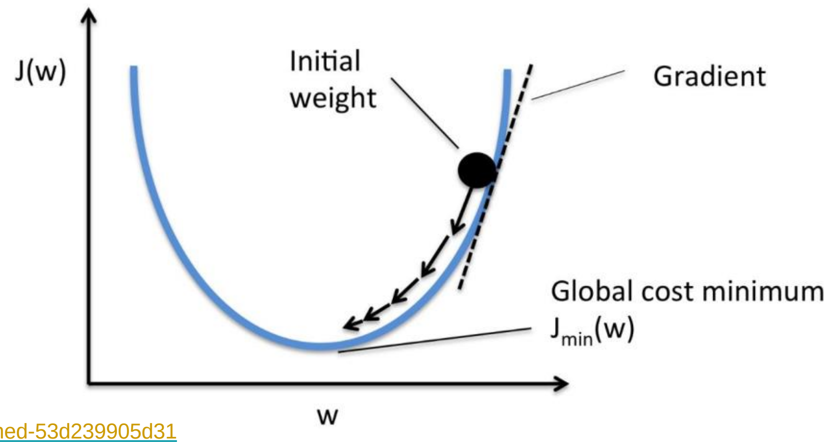
November 15th, 2021

Agenda

1. Gradient Descent?

Gradient Descent

"Gradient descent is an iterative algorithm, that starts from a random point on a function and travels down its slope in steps until it reaches the lowest point of that function."



Aishwarya V Srinivasan : https://towardsdatascience.com/stochastic-gradient-descent-clearly-explained-53d239905d31
Sarthak Gupta: https://hackernoon.com/dl03-gradient-descent-719aff91c7d6

Gradient Descent

The steps of the algorithm are

1. Find the slope of the objective function **with respect to each parameter/feature**. In other words, compute the gradient of the function.

$$\nabla_w f(w) = \left[\frac{\partial f(w)}{\partial w_1}, \frac{\partial f(w)}{\partial w_2}, \ldots, \frac{\partial f(w)}{\partial w_d}\right]^T$$

Aishwarya V Srinivasan : https://towardsdatascience.com/stochastic-gradient-descent-clearly-explained-53d239905d31
Sarthak Gupta: https://hackernoon.com/dl03-gradient-descent-719aff91c7d6

Gradient Descent

2. Pick a random initial value for the parameters.

$$w^{0}_{init} = [w^{0}_{1}, w^{0}_{2}, \dots, w^{0}_{d}]^{T}$$

3. Update the gradient function by plugging in the parameter values.

$$\nabla_{w} f(w_{init}) = [\frac{\partial f(w_{init})}{\partial w_1}, \frac{\partial f(w_{init})}{\partial w_2}, \dots, \frac{\partial f(w_{init})}{\partial w_d}]^{T}$$

Aishwarya V Srinivasan : https://towardsdatascience.com/stochastic-gradient-descent-clearly-explained-53d239905d31
Sarthak Gupta: https://hackernoon.com/dl03-gradient-descent-719aff91c7d6

Gradient Descent

4. Calculate the step sizes for each feature as : **step size = gradient * learning rate.**

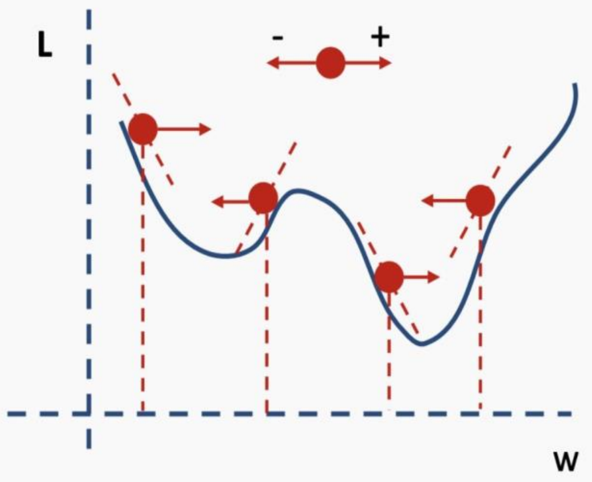
5. Calculate the new parameters as : **new params = old params -step size**

6. Repeat steps 3 to 5 until gradient is almost 0.

Aishwarya V Srinivasan : https://towardsdatascience.com/stochastic-gradient-descent-clearly-explained-53d239905d31
Sarthak Gupta: https://hackernoon.com/dl03-gradient-descent-719aff91c7d6

Gradient Descent

$$w^{(i+1)} = w^{(i)} - \lambda \frac{d\mathcal{L}}{dw}$$



Aishwarya V Srinivasan : https://towardsdatascience.com/stochastic-gradient-descent-clearly-explained-53d239905d31
Sarthak Gupta: https://hackernoon.com/dl03-gradient-descent-719aff91c7d6

Principle Component Analysis

- A linear latent-factor model:
  - Given input k, factorizes matrix X into two matrices Z and W:

$$\underset{n \times d}{X} \approx \underset{n \times k}{Z} W_{k \times d} \qquad x_i \approx w^T z_i \qquad x_{ij} \approx \langle w^j, z_i \rangle$$

- Some uses:
  - Dimensionality reduction: Replace X with low dimensional Z (k << d).
  - Outlier detection: If PCA gives poor approximation of example x_i, it could be outlier.
  - Partial least squares: Use Z as features in regression model.

- Objective function:

$$f(W, Z) = \sum_{i=1}^{n} \sum_{j=1}^{d} (\langle w^j, z_i \rangle - x_{ij})^2 = ||ZW - X||_F^2$$

Algorithm

- Training:
  - Center data: Each column of X should have mean 0
    i. Compute mean of each column: $\mu_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$
    ii. Center data: $x_{ij} \leftarrow x_{ij} - \mu_j$
  - Use alternating minimization:
    i. Randomly initialize Z and W
    ii. Optimize W with Z fixed (solve gradient w.r.t W)

$$\nabla_W f(W, Z) = Z^T(ZW - X) = 0 \Rightarrow W = (Z^T Z)^{-1} Z^T X$$

    i. Optimize Z with W fixed (solve gradient w.r.t Z)

$$\nabla_Z f(W, Z) = (ZW - X)W^T = 0 \Rightarrow Z = XW^T(WW^T)^{-1}$$

    i. Repeat until convergence.
- At the end of training, we can keep only μ_j and W.

Algorithm

- Prediction
  - Center data using mean of training data :

$$\tilde{X}_{ij} \leftarrow \tilde{X}_{ij} - \mu_j$$

  - Find $\tilde{Z}$ minimizing squared error:

$$\tilde{Z} = \tilde{X}W^T(WW^T)^{-1}$$

  - Now, the reconstruction loss is:

$$||\tilde{Z}W - \boxed{\tilde{X}}||_F^2$$

Centered
version