# CS 360: Machine Learning

Prof. Sara Mathieson

Fall 2020

# Admin

- Office hours **today 4:30-6pm**
- **Study guide** posted for the exam
  - Exam posted mid next week, due Dec 18

- **Lab Thursday**: discuss projects with each group
  - Lab attendance is optional if you're taking the exam

- **Project**
  - Email proposal by Thursday Dec 3
  - Presentation: last day of classes or Dec 18 (morning)
  - Git repo should be finalized by Dec 18 at noon

# CS360 Final Project

Proposal Email: Thursday, December 3 at 11:59pm

Short Presentation: Friday, December 11 (in class)

Code + Lab Notebook: Friday, December 18 at noon

There will be an alternative presentation time on Dec 18 (morning)

## Outline for a typical project:

- Find a dataset (see project writeup)
- Run an algorithm we've discussed on the dataset
- Try to do a comparison
  - run the algorithm in multiple ways
  - different data pre-processing
  - try a different algorithm
- Record and interpret the results

# Project Deliverables

- Main deliverable: presentation

  - Group of 1: 5 min
  - Group of 2: 9 min
  - Group of 3: 12 min

- On git:
  - Lab Notebook
  - Project Code

# Outline for December 1

- Training a NN: backpropagation
- Begin unsupervised learning
- K-means clustering
- Next time
  - Gaussian Mixture Models (GMM)
  - Principal Component Analysis (PCA)

# Outline for December 1

- Training a NN: backpropagation
- Begin unsupervised learning
- K-means clustering
- Next time
  - Gaussian Mixture Models (GMM)
  - Principal Component Analysis (PCA)

# Backpropagation

- *High-level goal:* we want to know how the output depends on the input
- *Issue:* network is very complicated and overall gradient may be difficult to compute
- *Idea:* use the chain rule to compute local gradients throughout the network
- *Takeaway:* nodes can know about their value and local gradient without knowing about the network they are imbedded in

# Backpropagation

$J(\vec{w})$ loss/cost

Idea: want to know how output depends on input

weights $\vec{w}$

example

$$f(x, y, z) = (x + y)z = q \cdot z$$

$q = x + y$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} \cdot \frac{\partial q}{\partial x} = z$$

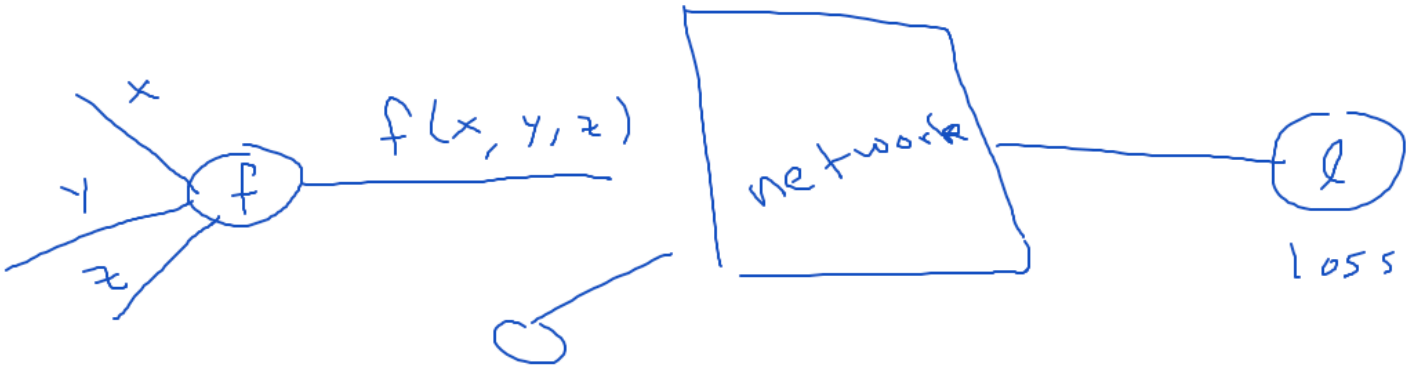
$z \cdot 1$

SGD

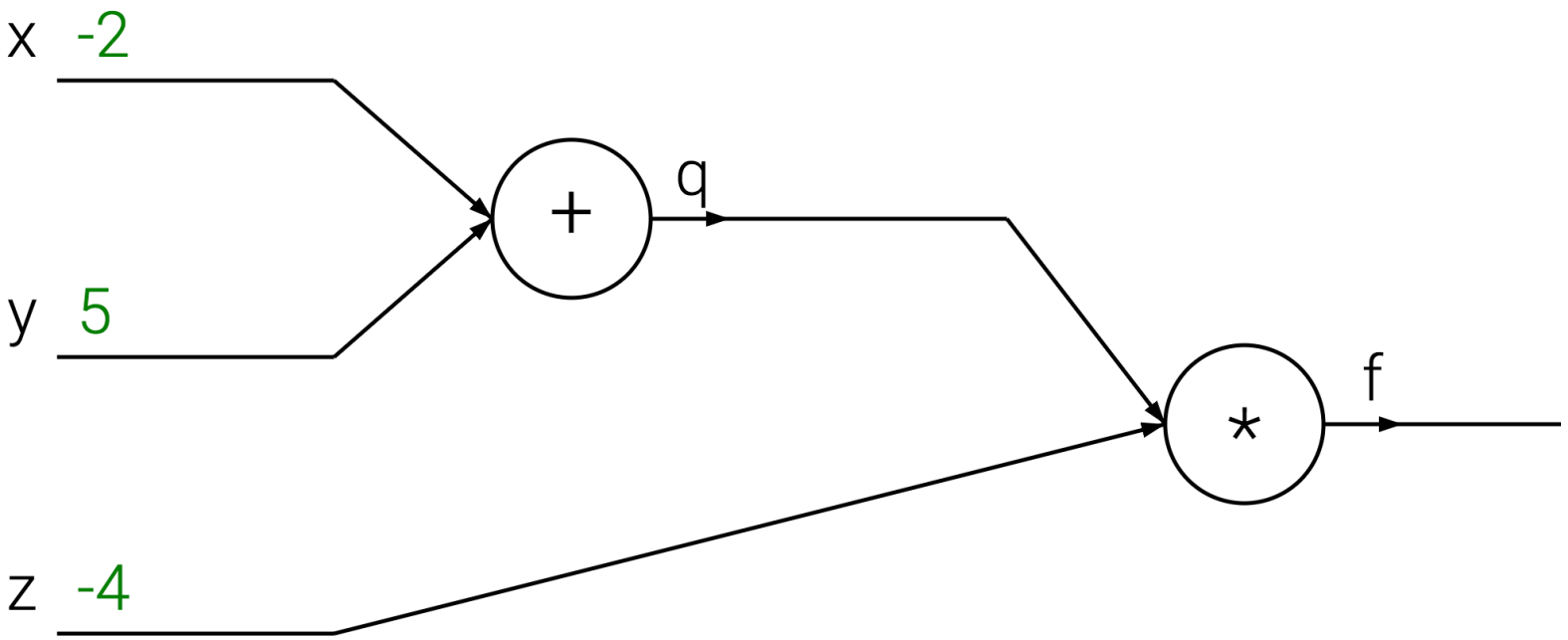
$$x \leftarrow x - \alpha \frac{\partial f}{\partial x}$$

$$y \leftarrow y - \alpha \frac{\partial f}{\partial y}$$

$$z \leftarrow z - \alpha \frac{\partial f}{\partial z}$$

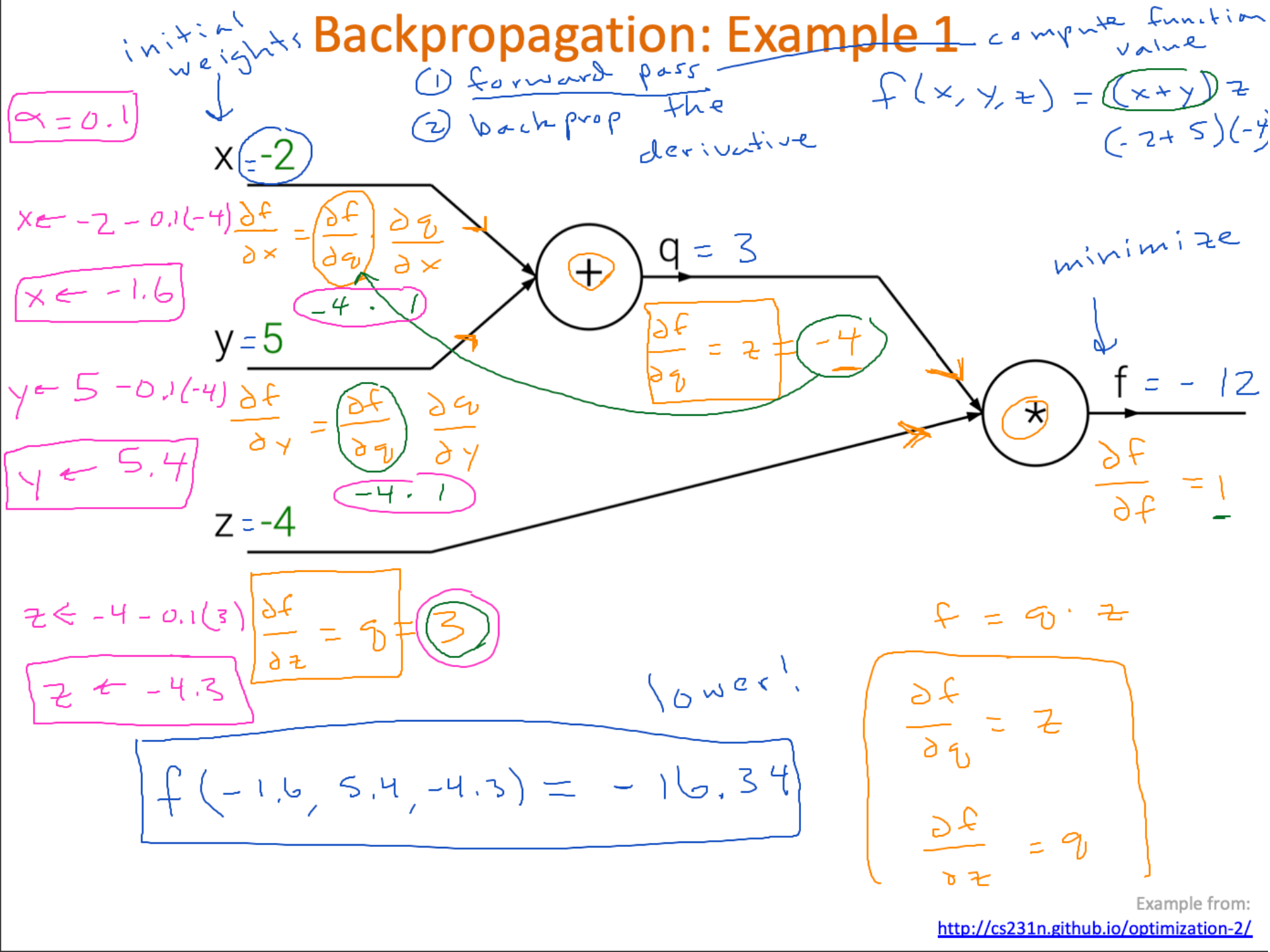


# Backpropagation: Example 1



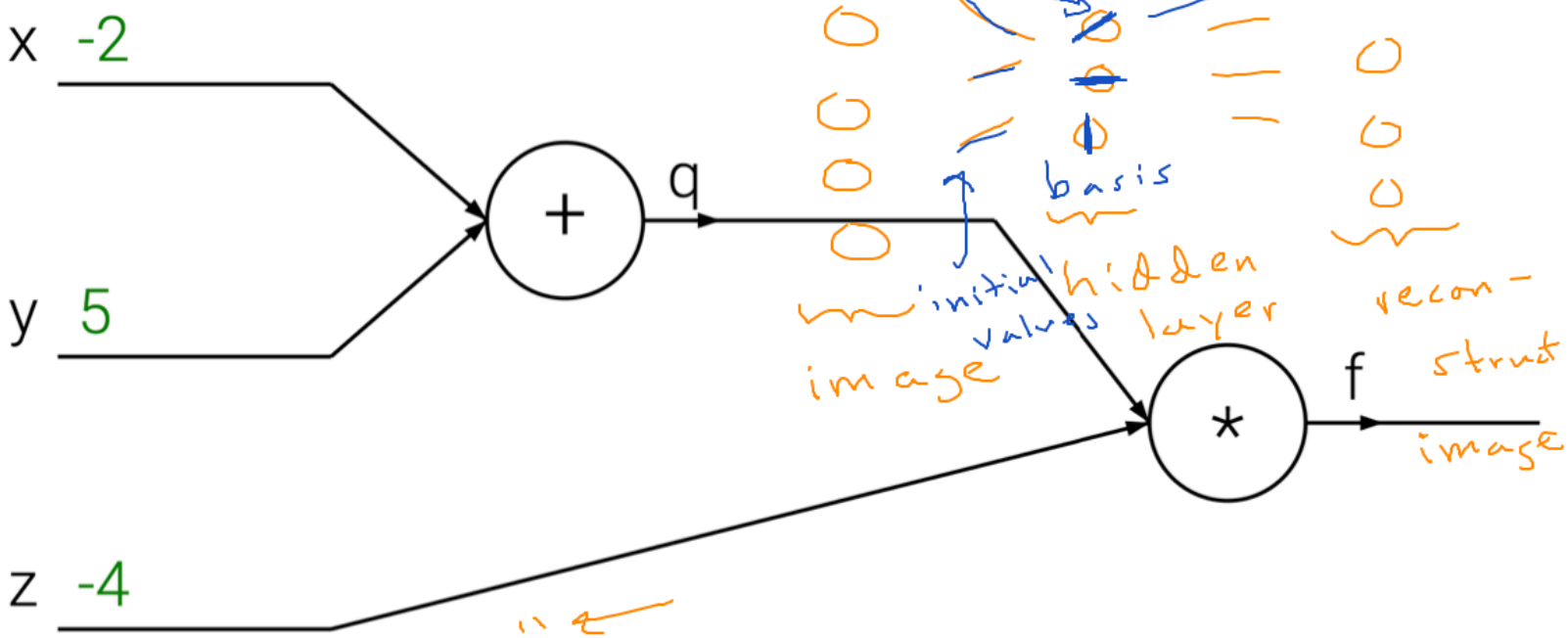

Example from:
http://cs231n.github.io/optimization-2/

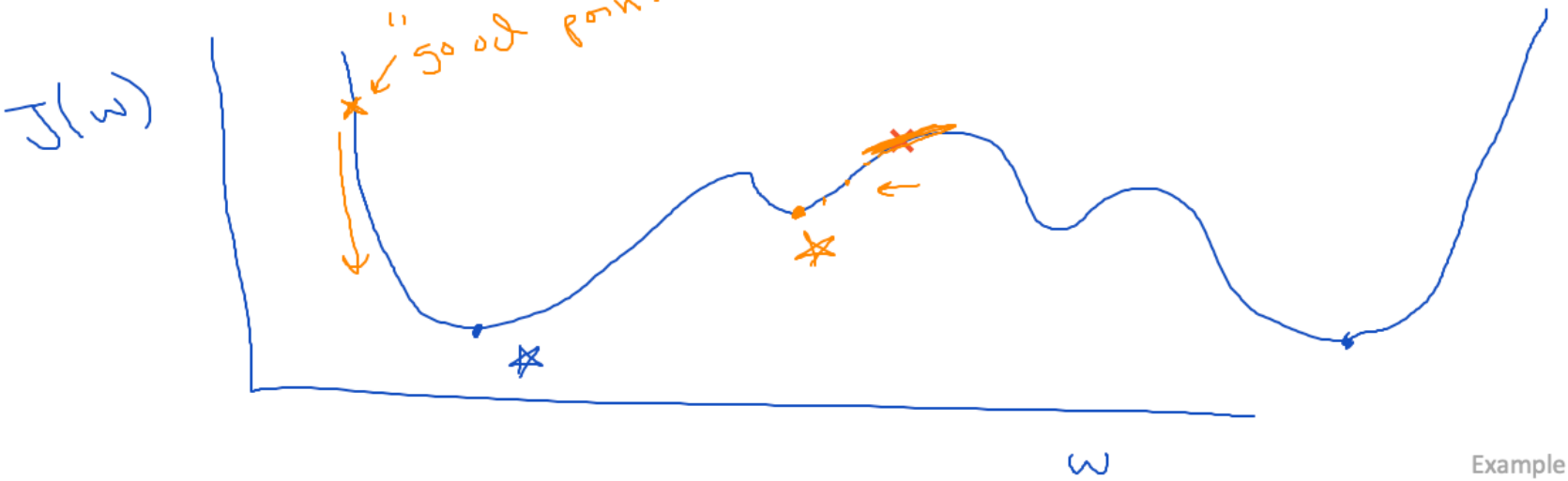
# Backpropagation: Example 1




Example from:
http://cs231n.github.io/optimization-2/

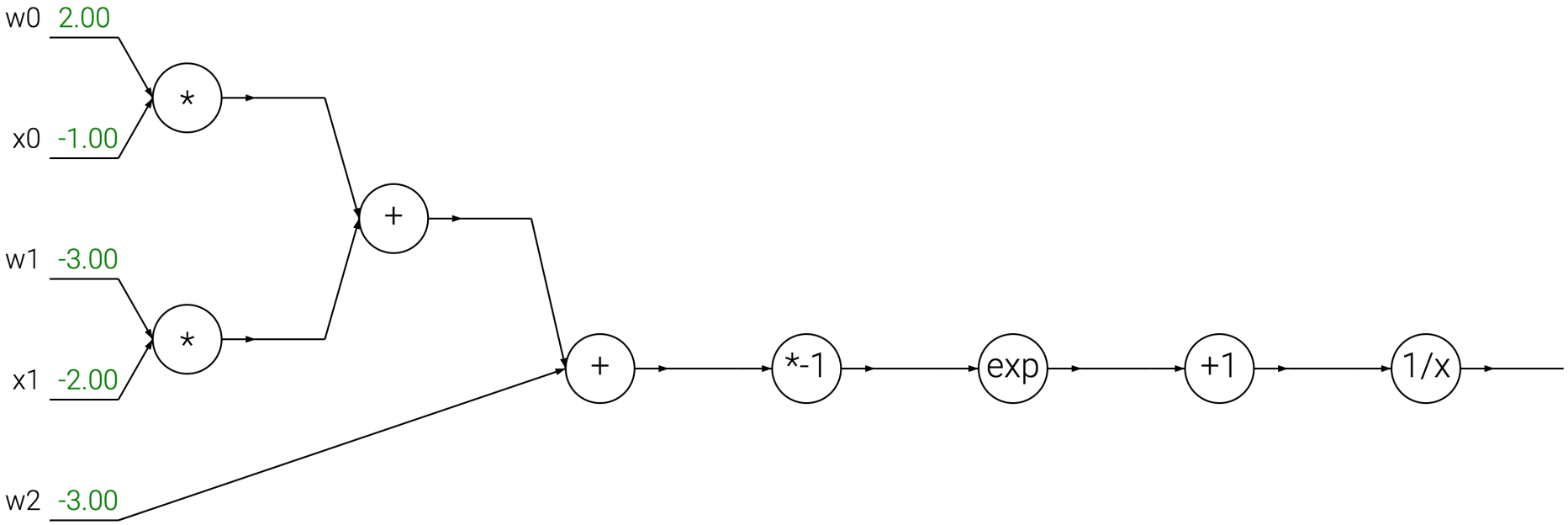
# Backpropagation: Example 1



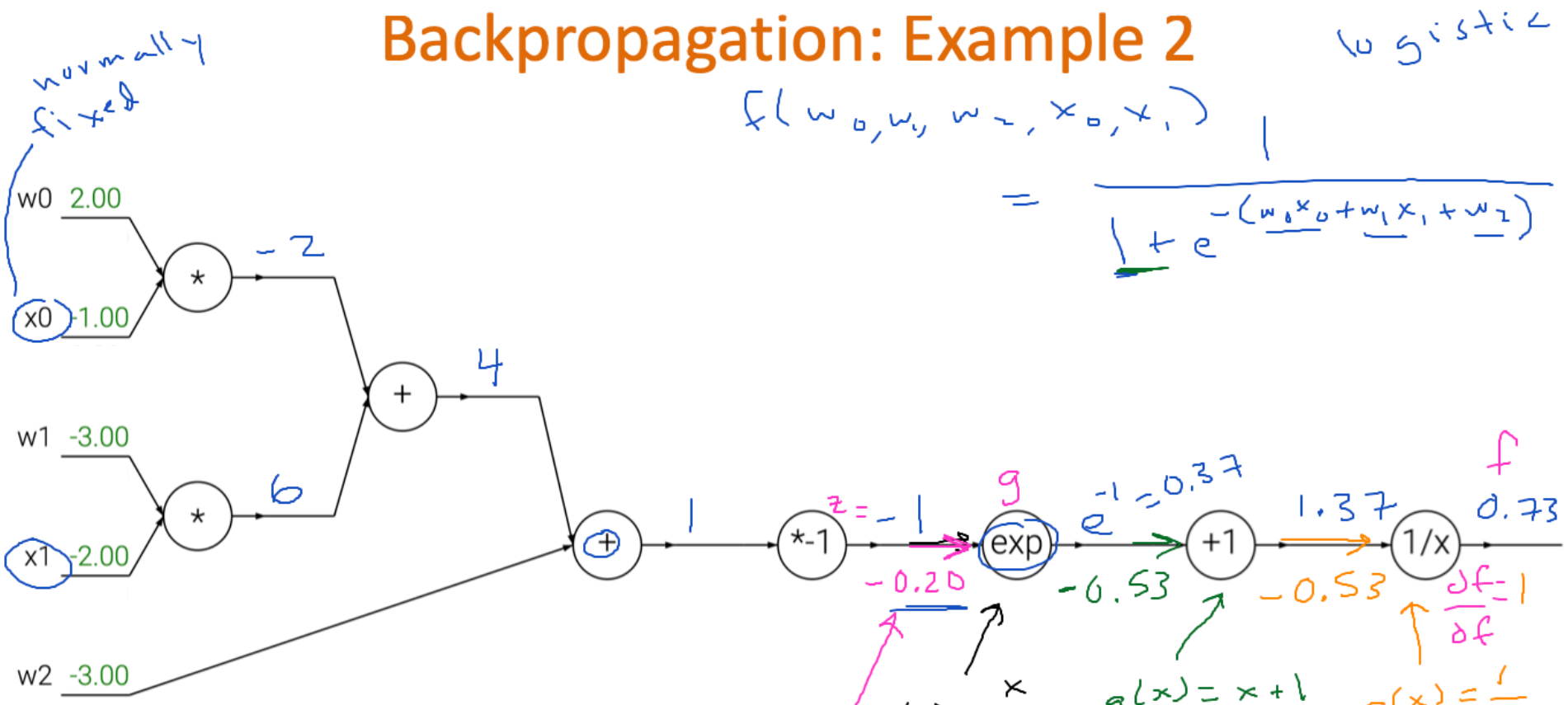

Example from:
http://cs231n.github.io/optimization-2/

# Backpropagation: Example 2




Example from:
http://cs231n.github.io/optimization-2/

# Backpropagation: Example 2




normally fixed

logistic

$$f(w_0, w_1, w_2, x_0, x_1) = \frac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2)}}$$

$$f(z) = f(g(z))$$

$g(x) = e^x$ $\quad g'(x) = e^x$ $\quad g'(-1) = e^{-1} = 0.37$

$g(x) = x + 1$ $\quad g'(x) = 1$ $\quad g'(0.37) = 1$

$g(x) = \frac{1}{x}$ $\quad g'(x) = -x^{-2}$ $\quad g'(1.37) = -0.53$

input to node g

back prop so far

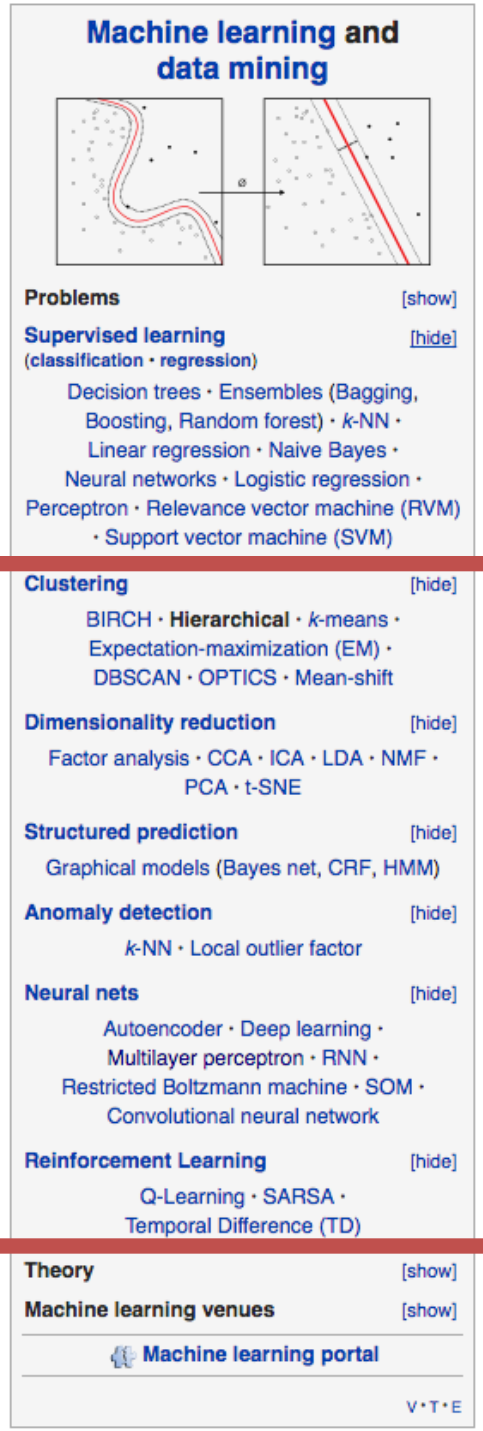
$$\frac{df}{dz} = \frac{dg}{dz} \cdot \frac{df}{dg} = (0.37)(-0.53)$$

Example from:
http://cs231n.github.io/optimization-2/

# Outline for December 1

- Training a NN: backpropagation
- Begin unsupervised learning
- K-means clustering
- Next time
  - Gaussian Mixture Models (GMM)
  - Principal Component Analysis (PCA)

**Supervised Learning:**
makes use of examples where we know the underlying "truth" (label/output)




**Unsupervised Learning:**
Learn underlying structure or features without labeled training data

Image: wikipedia

# Unsupervised learning: 3 main areas

1) Clustering: group data points into clusters based on features only

2) Dimensionality reduction: remove feature correlation, compress data, visualize data

3) Structured prediction: model latent variables (example: Hidden Markov Models)

# Unsupervised learning examples from biology: clustering

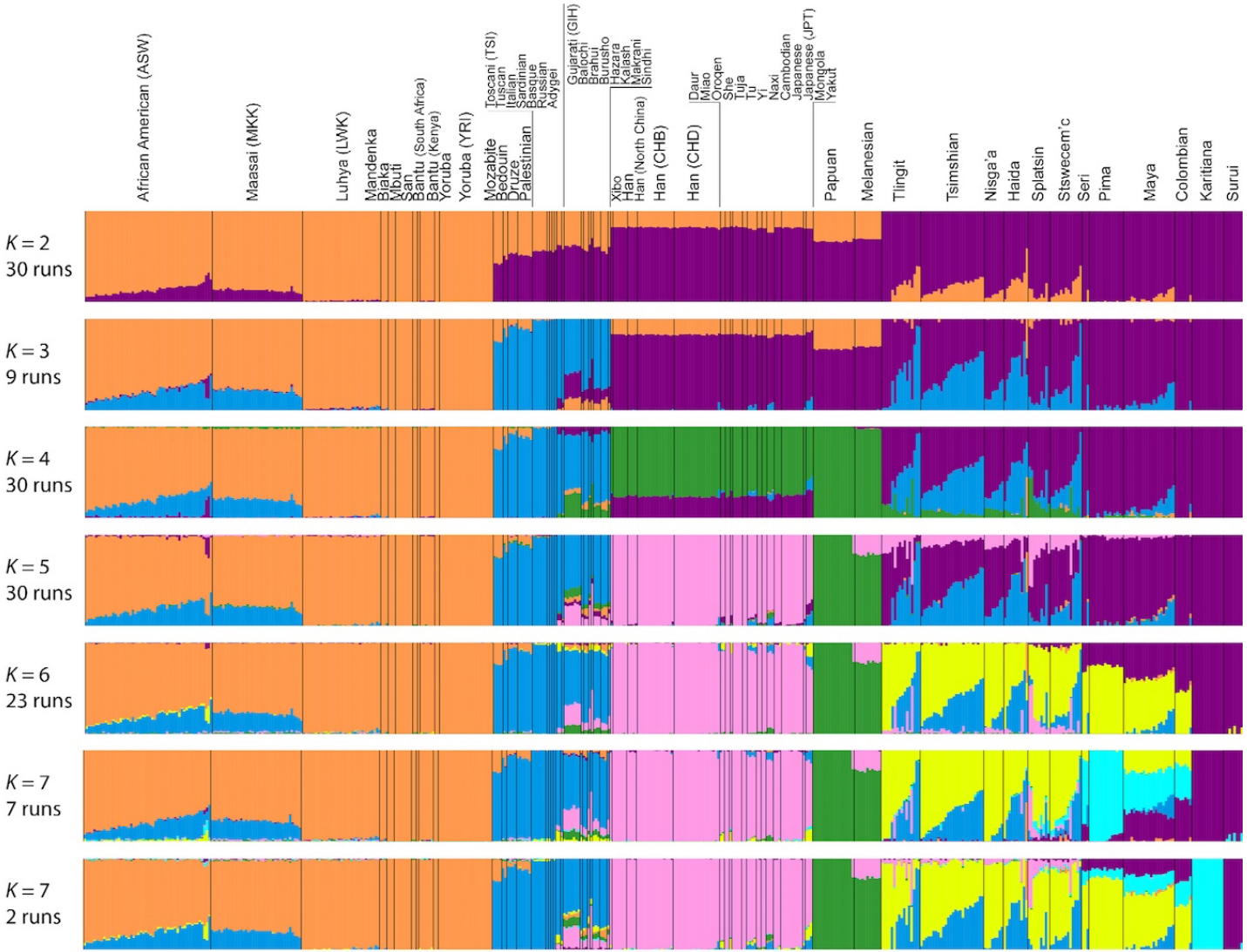
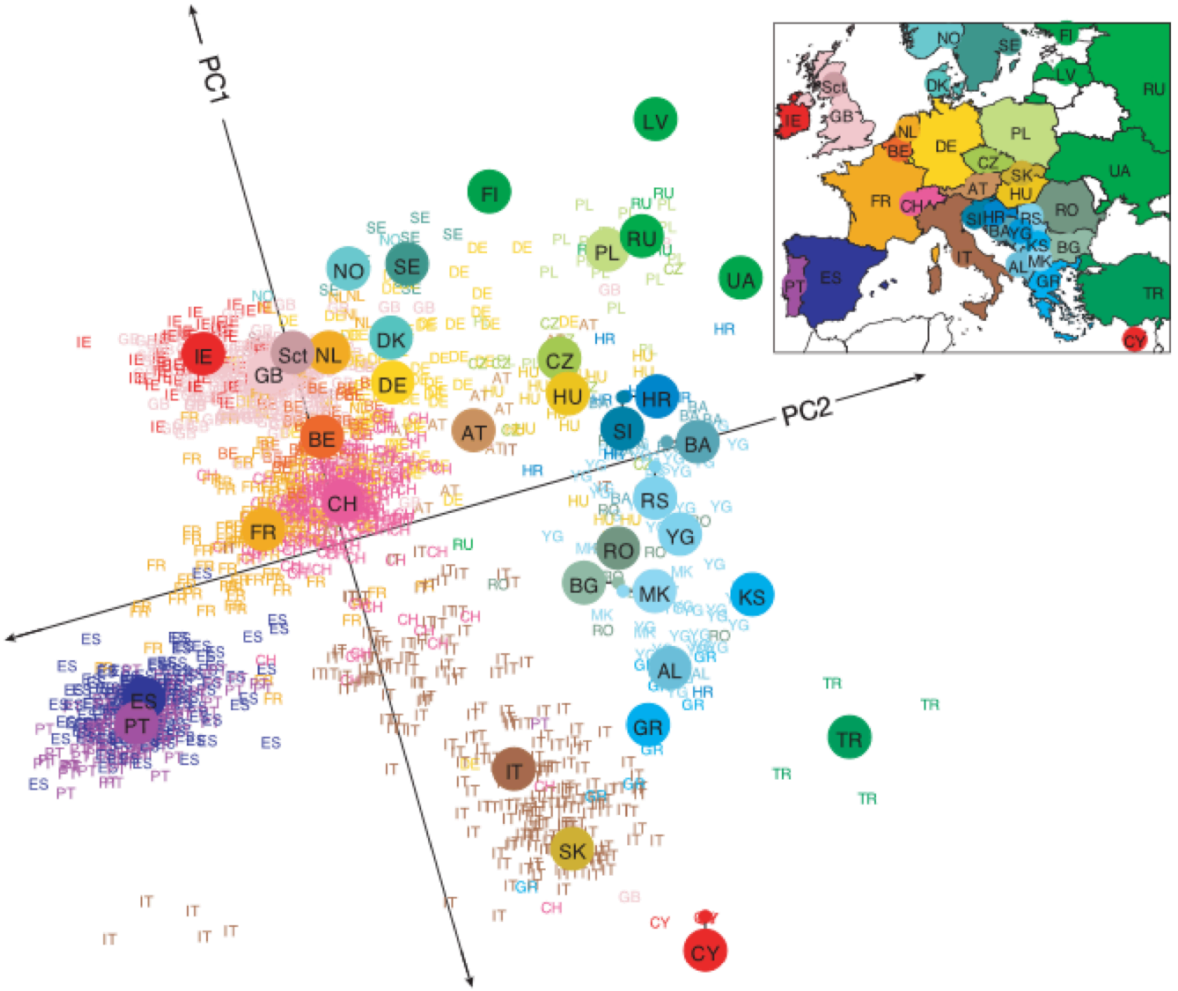


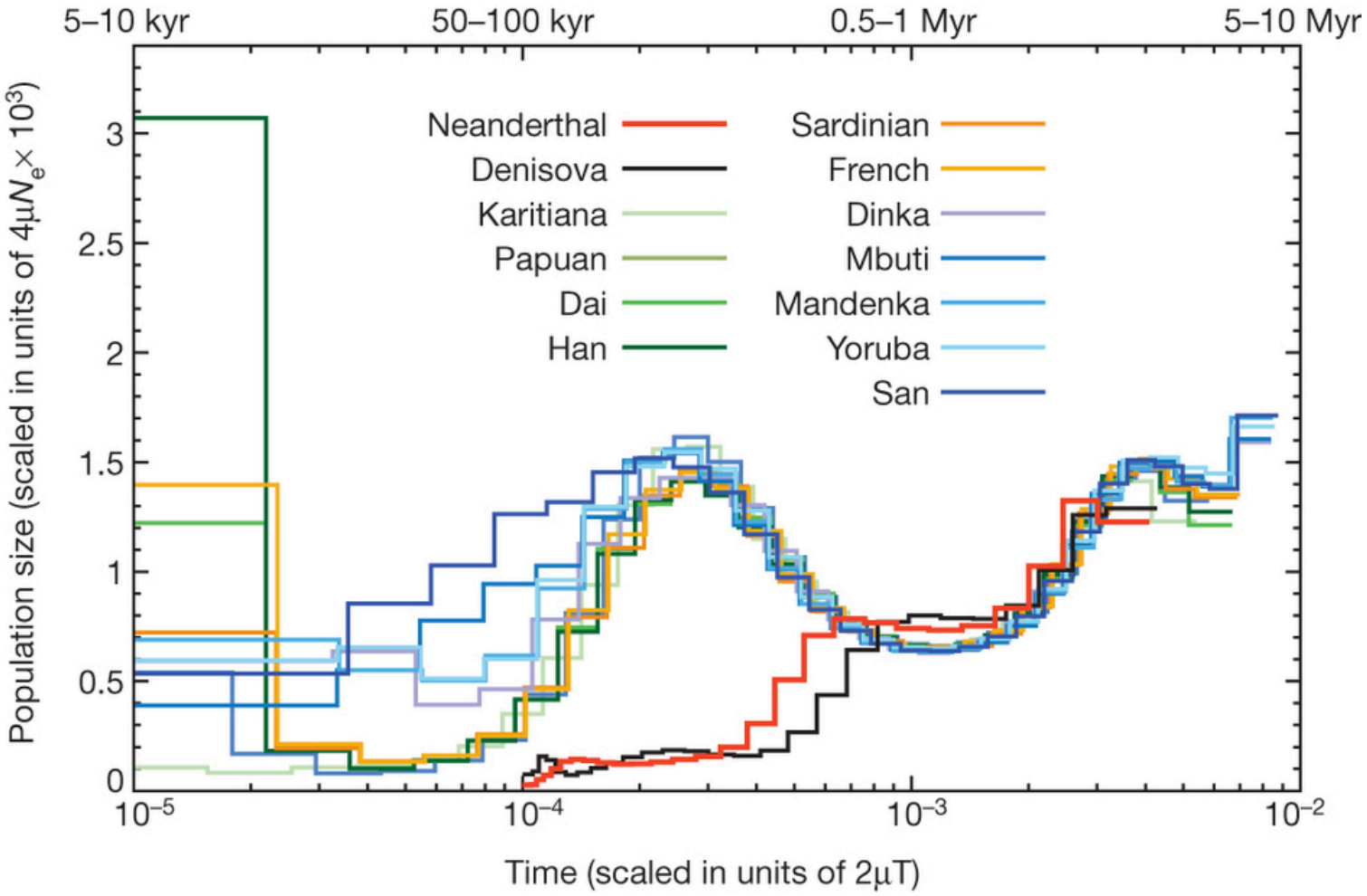

Figure: German Dziebel

# Unsupervised learning examples from biology: structured prediction




*Genes mirror geography within Europe (2008)*

# Unsupervised learning examples from biology: structured prediction




*The complete genome sequence of a Neanderthal from the Altai Mountains, Prufer et al (2014)*

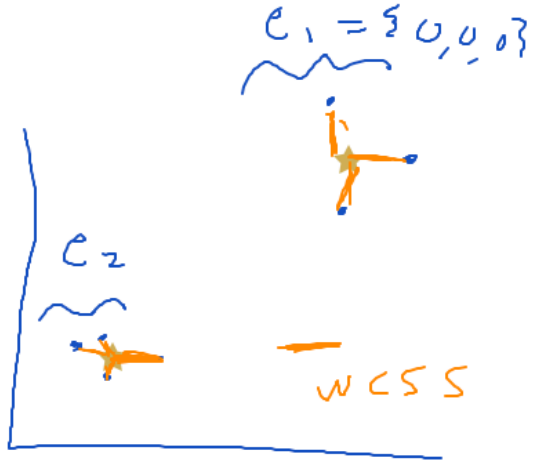
# Outline for December 1

- Training a NN: backpropagation
- Begin unsupervised learning
- K-means clustering
- Next time
  - Gaussian Mixture Models (GMM)
  - Principal Component Analysis (PCA)

# Clustering Goals

- Learn about underlying structure in the data
- Cluster new data (testing)

- Goal: minimize "within cluster sum of squares" (WCSS)




find $\{C_1, C_2, \ldots C_K\} = C$ s.t.

WCSS (cost) $\longrightarrow$ $J(C) = \sum_{k=1}^{K} \sum_{i \in C_k} \| \vec{x}_i - \vec{\mu}_k \|^2$

$k=1$: all clusters; $\vec{x}_i$: train data; $\vec{\mu}_k$: cluster mean

truly minimizing WCSS

$\Rightarrow$ NP-hard

# K-means algorithm

$n, p$

- 0) Initialization: choose means (centers) of K clusters

typically from training data

$\vec{\mu}_1^{(1)}, \vec{\mu}_2^{(1)}, \ldots \vec{\mu}_K^{(1)}$

- 1) **E-step (assignment):** for each training example, find closest mean

- 2) **M-step (update):** compute new means