CS 360: Machine Learning

Prof. Sara Mathieson

Fall 2019

HAVERFORD COLLEGE

Admin

- **Midterm 2 due today at 6pm!**

- Office hours today: informal, I'm around this afternoon

- **Final project presentations:**
  - Wednesday Dec 18: 1-4pm (block out the entire time, but we may not need all of it)
  - Option to present last day of class (email me)

Outline for November 26

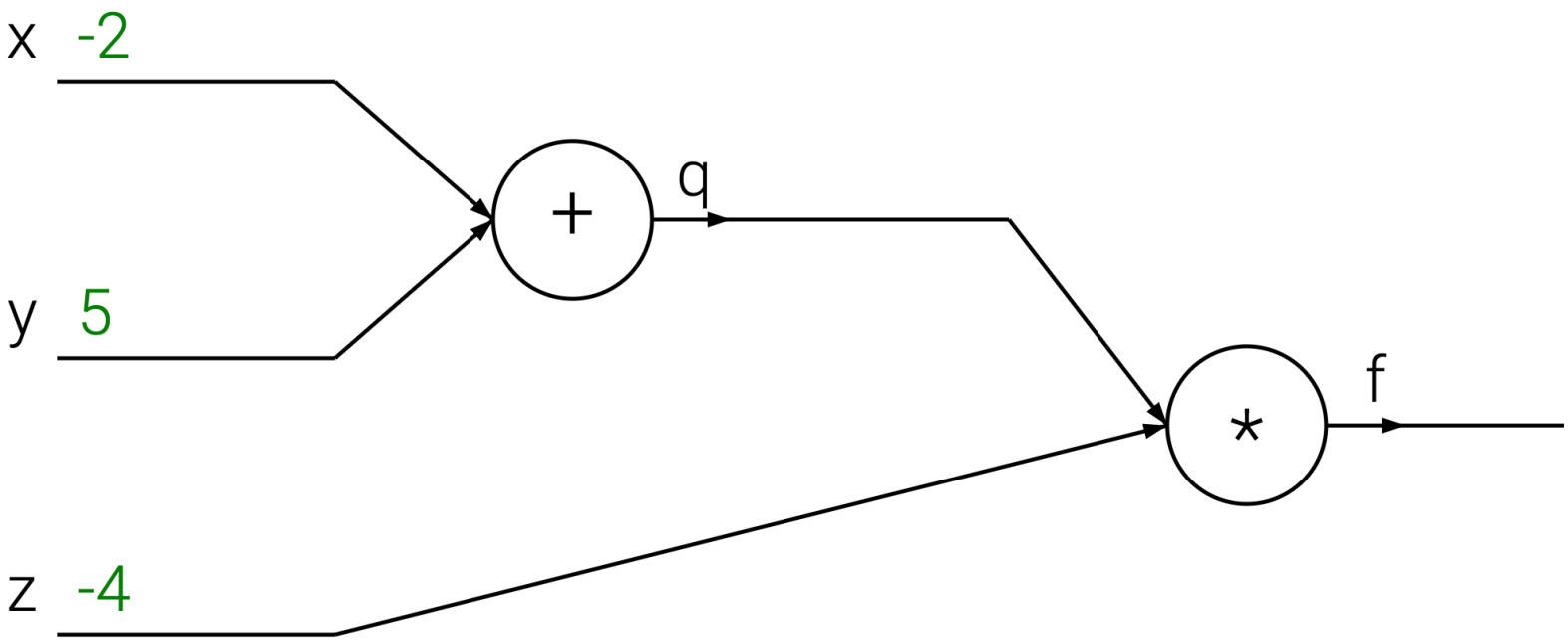
- Backpropogation
- Begin: unsupervised learning
- K-means
- Next week
  - Gaussian Mixture Models (GMM)
  - Principal Component Analysis (PCA)

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: Example 1

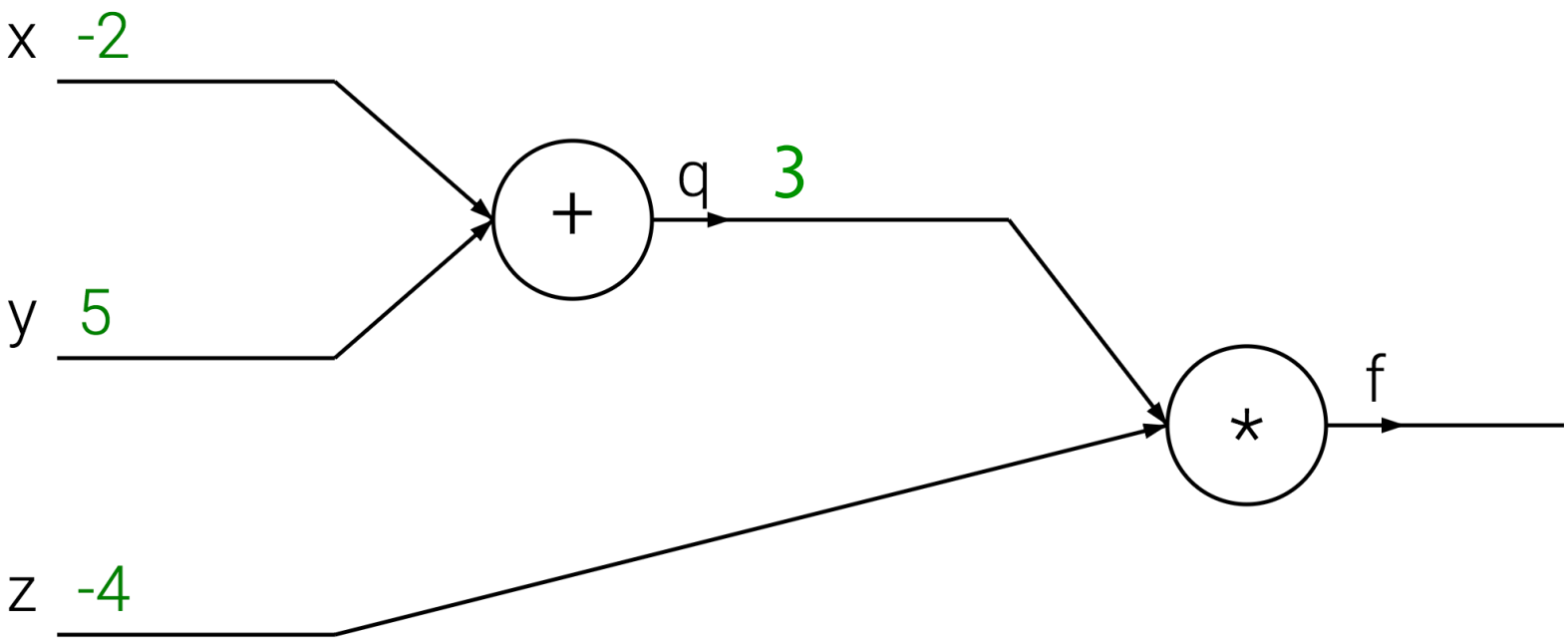
Forward pass: compute values



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

Backpropagation: Example 1

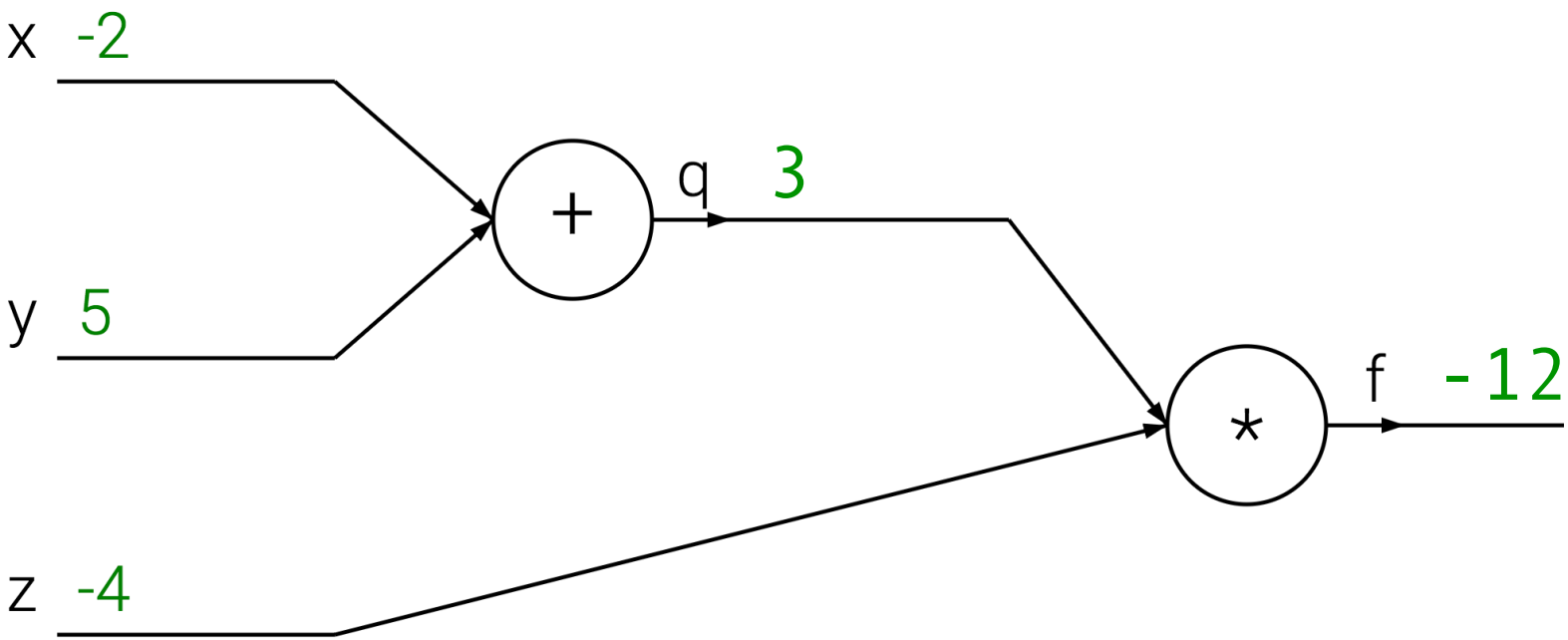
Forward pass: compute values



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

Backpropagation: Example 1

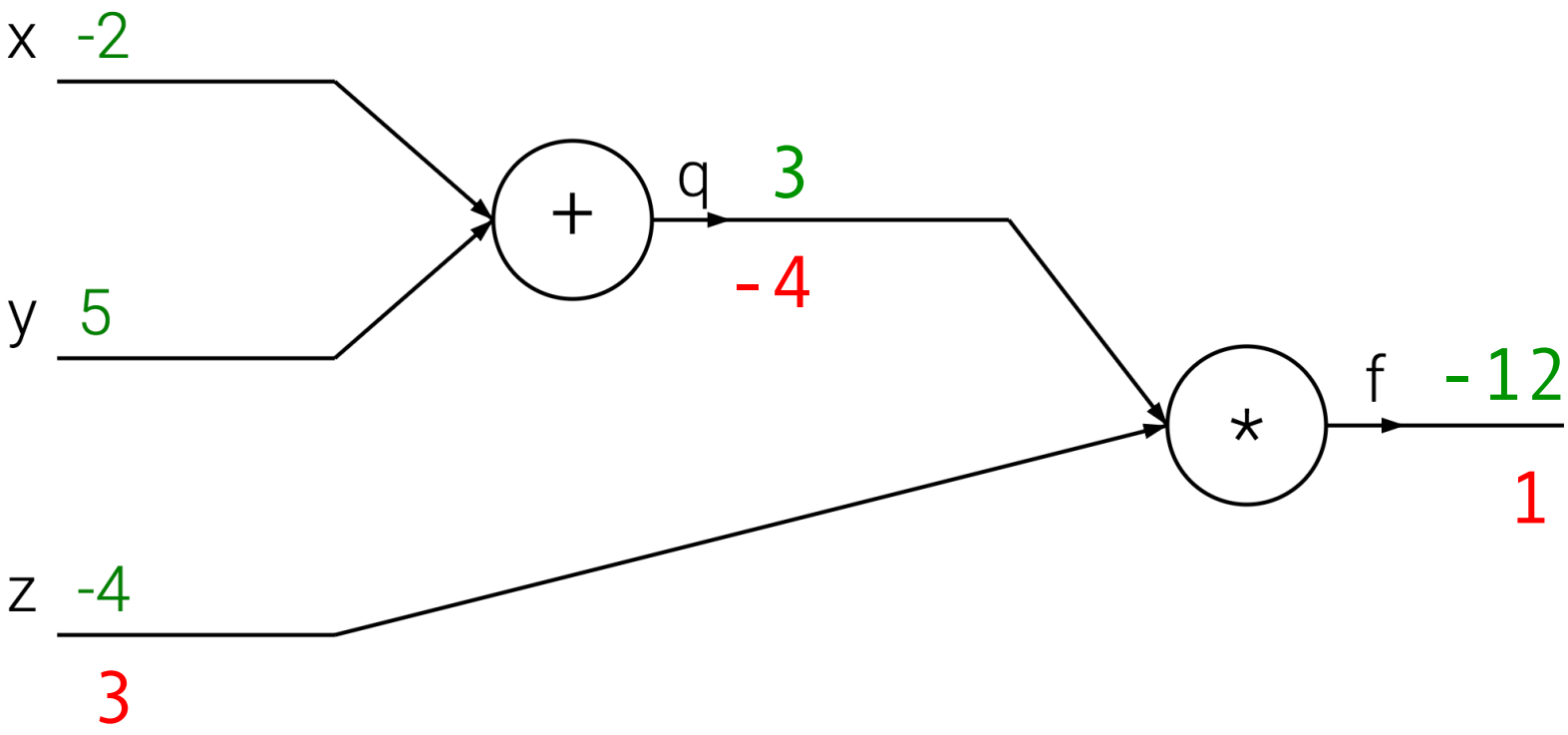
Forward pass: compute values



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

Backpropagation: Example 1

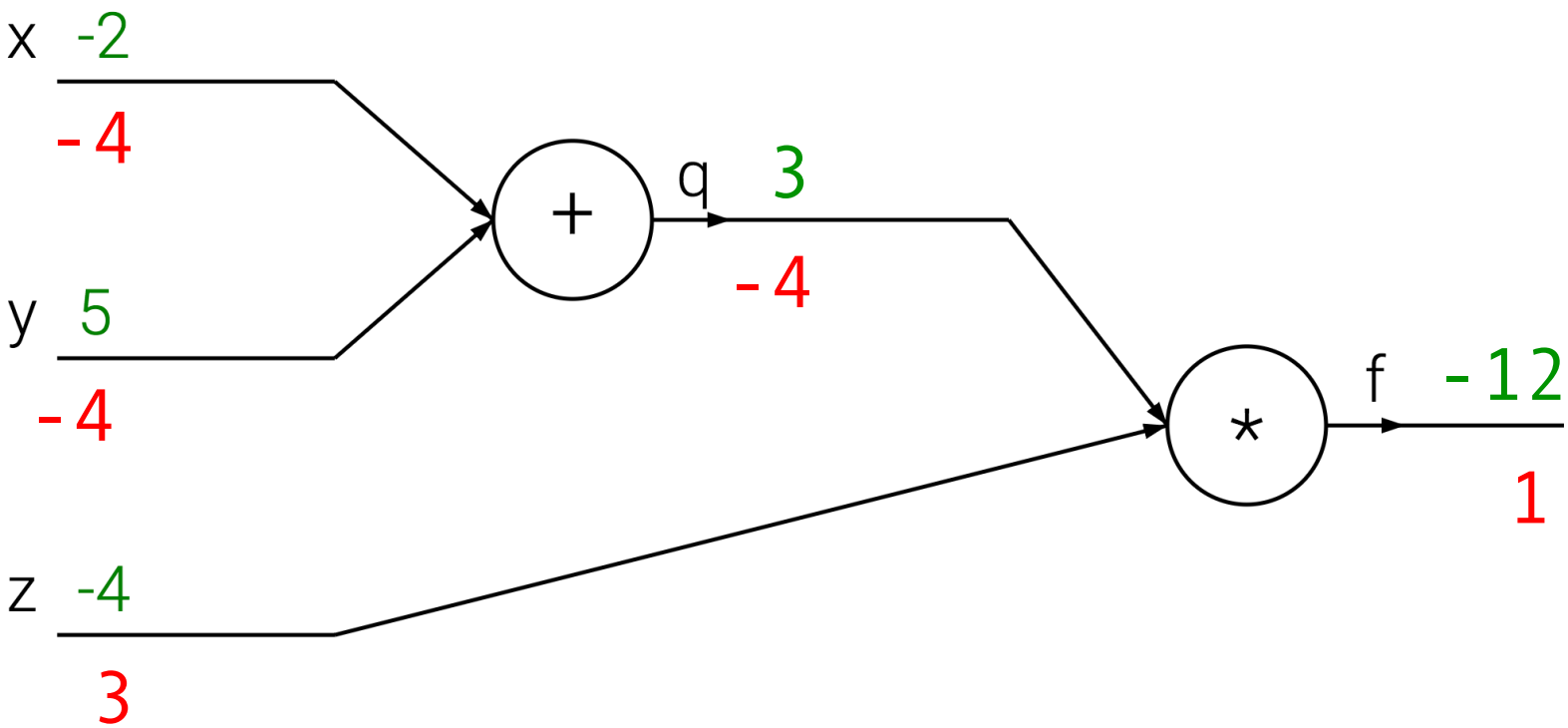
Backward pass: compute local gradients



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

Backpropagation: Example 1

Backward pass: compute local gradients



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

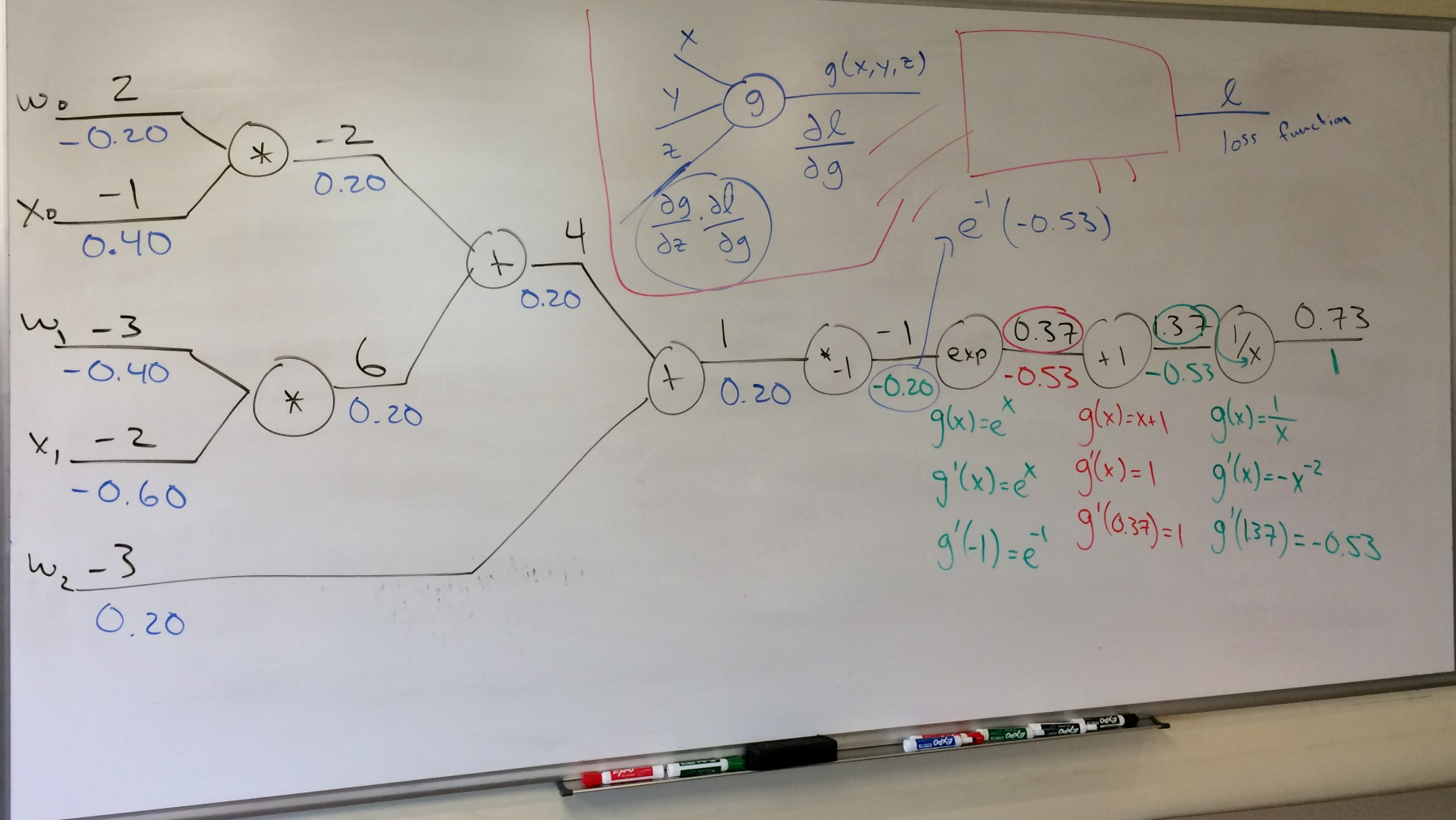
Backpropagation

Q3, Q5, Q6

Idea: want to know how output (loss) depends on input (weights)

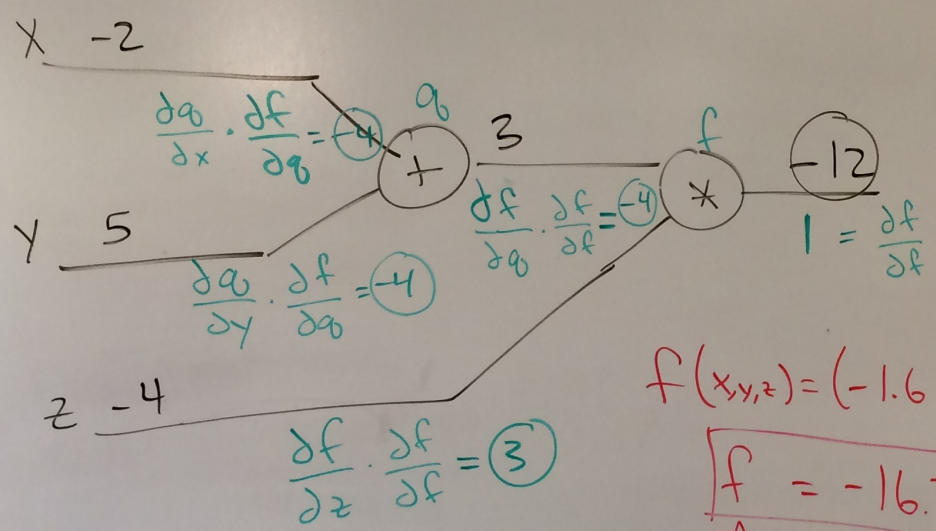
"loss" $\longrightarrow$ $f(x,y,z) = (\underbrace{x+y}_{q})z = qz$

$$\frac{\partial f}{\partial x} = \overbrace{\underbrace{\frac{\partial f}{\partial q}}_{z} \cdot \underbrace{\frac{\partial q}{\partial x}}_{1}}^{\text{chain rule}}$$


w_0 2
-0.20
x_0 -1
0.40
*
-2
0.20
w_1 -3
-0.40
x_1 -2
-0.60
*
6
0.20
+
4
0.20
w_2 -3
0.20
+
1
0.20
*-1
-1
-0.20
exp
0.37
-0.53
+1
1.37
-0.53
1/x
0.73
1
x
y
z
g
g(x,y,z)
$\frac{\partial l}{\partial g}$
$\frac{\partial g}{\partial z} \cdot \frac{\partial l}{\partial g}$
$e^{-1}(-0.53)$
l
loss function
$g(x)=e^x$
$g'(x)=e^x$
$g'(-1)=e^{-1}$
$g(x)=x+1$
$g'(x)=1$
$g'(0.37)=1$
$g(x)=\frac{1}{x}$
$g'(x)=-x^{-2}$
$g'(1.37)=-0.53$

Example 1

$f(x,y,z) = (x+y)z$ want to minimize



$f(x,y,z) = (-1.6 + 5.4)(-4.3)$

$\boxed{f = -16.34}$

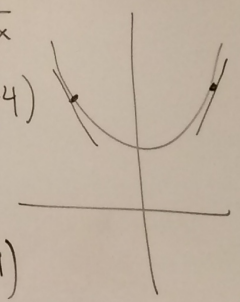
lower than before!

update using gradient descent

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

$x \leftarrow -2 - 0.1(-4)$

$\boxed{x \leftarrow -1.6}$



$y \leftarrow 5 - 0.1(-4)$

$\boxed{y \leftarrow 5.4}$

$z \leftarrow -4 - 0.1(3)$

$\boxed{z \leftarrow -4.3}$

Handout 19

3

$\frac{\partial f}{\partial a}$

$a = 1 - 0.1(-24)$

$\boxed{a = 3.4}$

$b = 7 - 0.1(8)$

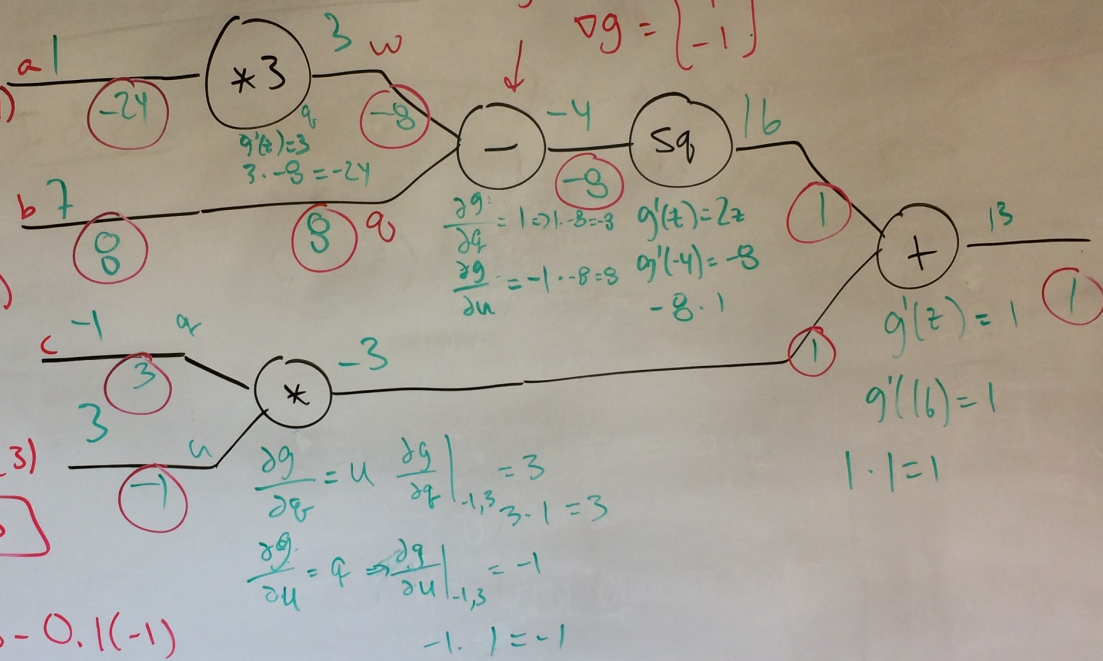
$\boxed{b = 6.2}$

$c = -1 - 0.1(3)$

$\boxed{c = -1.3}$

$d = 3 - 0.1(-1)$

$\boxed{d = 3.1}$



$g(w,q) = w - q$

$\nabla g = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

$g'(z) = 3$

$3 \cdot -8 = -24$

$\frac{\partial g}{\partial q} = 1 \Rightarrow 1 \cdot -8 = -8$

$\frac{\partial g}{\partial u} = -1 \cdot -8 = 8$

$g'(z) = 2z$

$g'(-4) = -8$

$-8 \cdot 1$

$g'(z) = 1$

$g'(16) = 1$

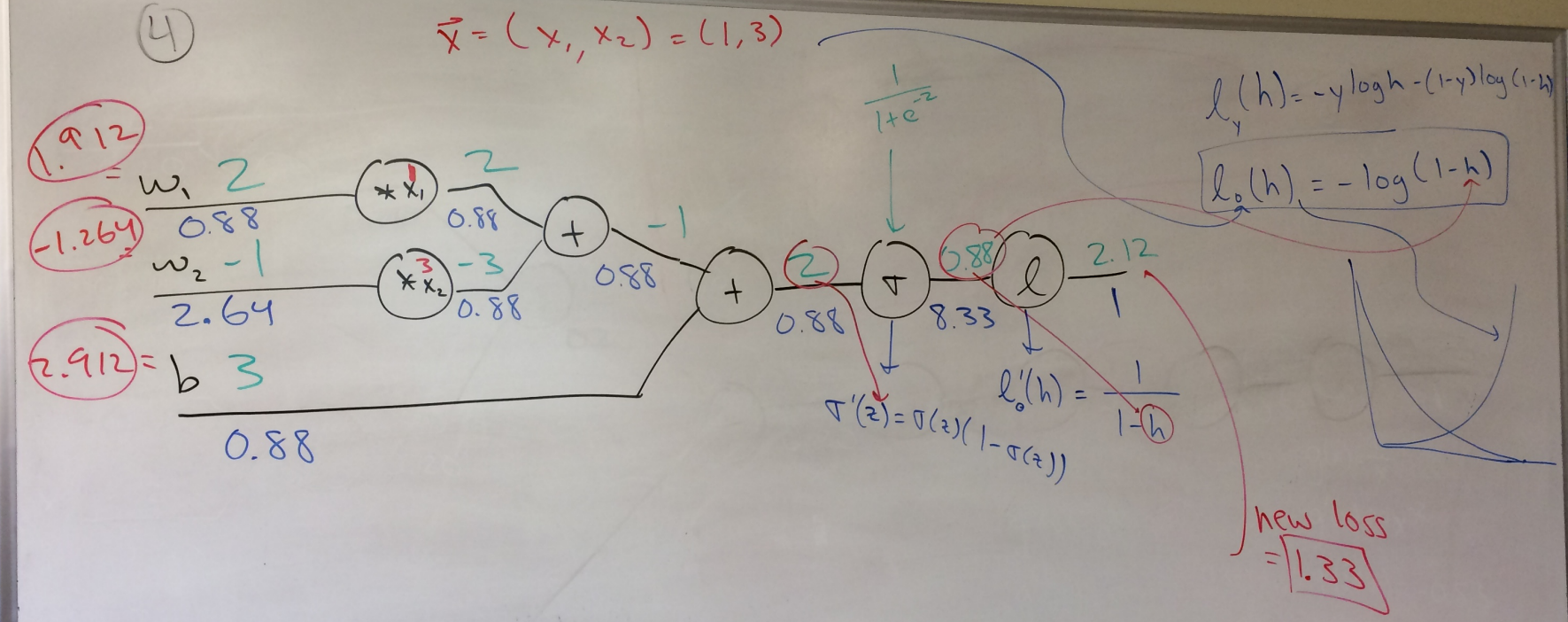
$1 \cdot 1 = 1$

$\frac{\partial g}{\partial q} = u \quad \left.\frac{\partial g}{\partial q}\right|_{-1,3} = 3$

$3 \cdot 1 = 3$

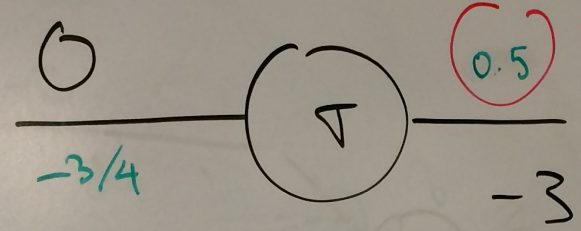
$\frac{\partial g}{\partial u} = q \Rightarrow \left.\frac{\partial g}{\partial u}\right|_{-1,3} = -1$

$-1 \cdot 1 = -1$

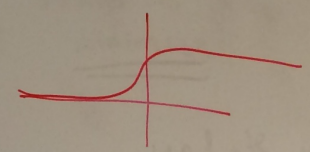

4
$\vec{x} = (x_1, x_2) = (1,3)$
1.912 = w_1 2
0.88
-1.264 w_2 -1
2.64
2.912 = b 3
0.88
$*x_1$
$*x_2$
2
0.88
-3
0.88
+
-1
0.88
+
2
0.88
$\frac{1}{1+e^{-z}}$
σ
0.88
8.33
ℓ
2.12
1
$\sigma'(z) = \sigma(z)(1-\sigma(z))$
$\ell_0'(h) = \frac{1}{1-h}$
$\ell_y(h) = -y\log h - (1-y)\log(1-h)$
$\ell_0(h) = -\log(1-h)$
new loss = 1.33

Handout 19

(5.)

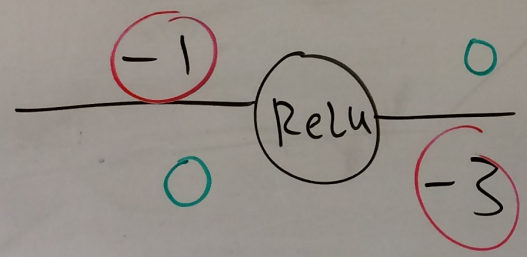


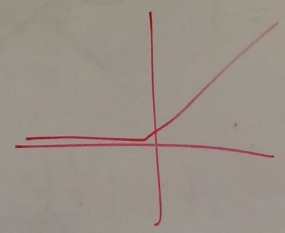
$$\frac{1}{1+e^{-0}} = \frac{1}{2}$$



$$\sigma'(0) = \underbrace{\sigma(0)}_{\frac{1}{2}}\underbrace{(1-\sigma(0))}_{\frac{1}{2}} = \frac{1}{4}$$

(6.)





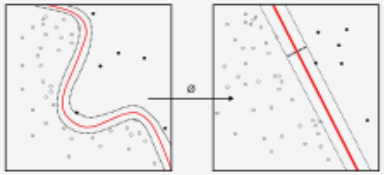
Handout 19

Outline for November 26

- Backpropogation
- Begin: unsupervised learning
- K-means
- Next week
  - Gaussian Mixture Models (GMM)
  - Principal Component Analysis (PCA)

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

Machine learning and data mining



Problems [show]

Supervised learning [hide]
(classification · regression)

Decision trees · Ensembles (Bagging, Boosting, Random forest) · *k*-NN · Linear regression · Naive Bayes · Neural networks · Logistic regression · Perceptron · Relevance vector machine (RVM) · Support vector machine (SVM)

Clustering [hide]

BIRCH · **Hierarchical** · *k*-means · Expectation-maximization (EM) · DBSCAN · OPTICS · Mean-shift

Dimensionality reduction [hide]

Factor analysis · CCA · ICA · LDA · NMF · PCA · t-SNE

Structured prediction [hide]

Graphical models (Bayes net, CRF, HMM)

Anomaly detection [hide]

k-NN · Local outlier factor

Neural nets [hide]

Autoencoder · Deep learning · Multilayer perceptron · RNN · Restricted Boltzmann machine · SOM · Convolutional neural network

Reinforcement Learning [hide]

Q-Learning · SARSA · Temporal Difference (TD)

Theory [show]

Machine learning venues [show]

Machine learning portal

V · T · E

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

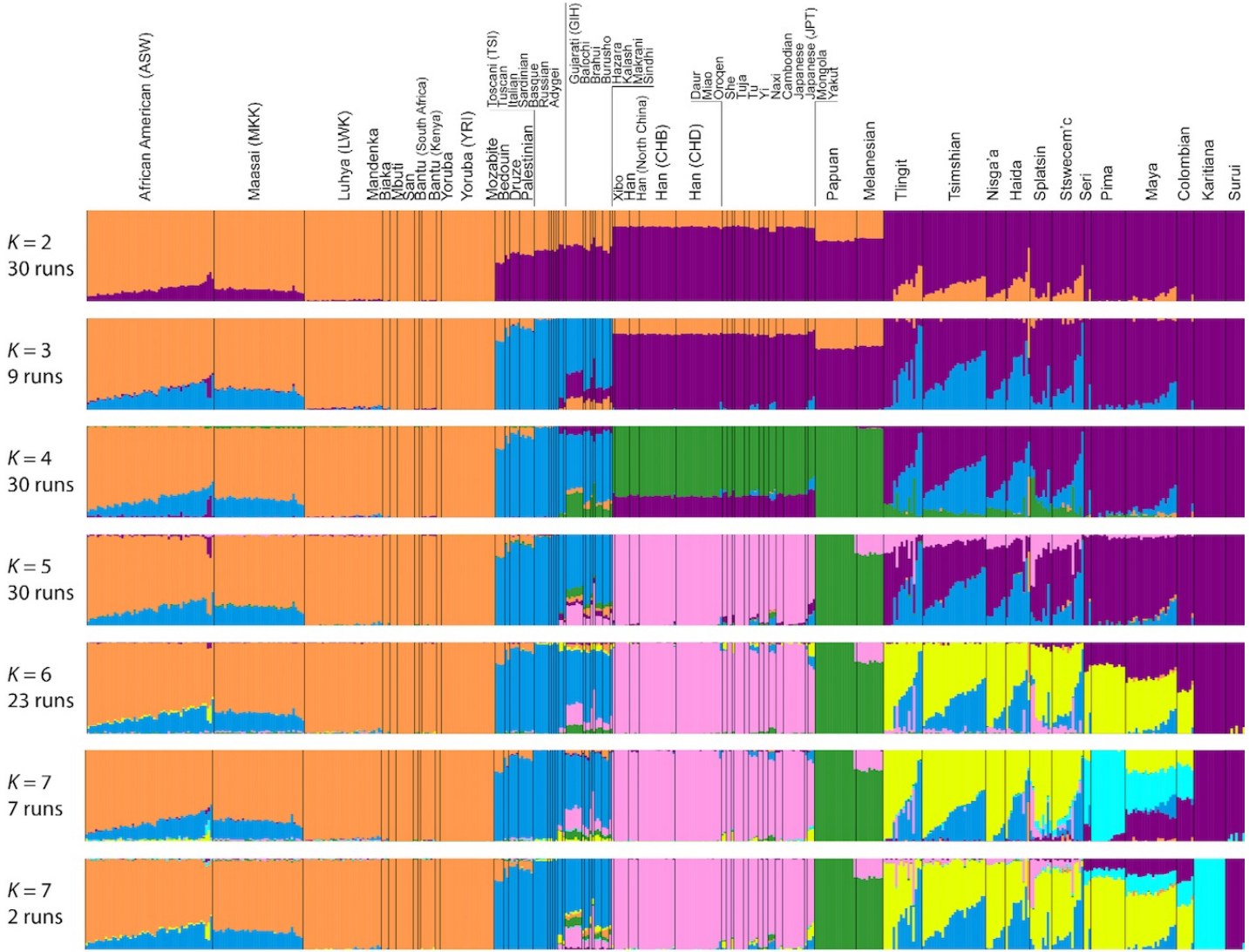
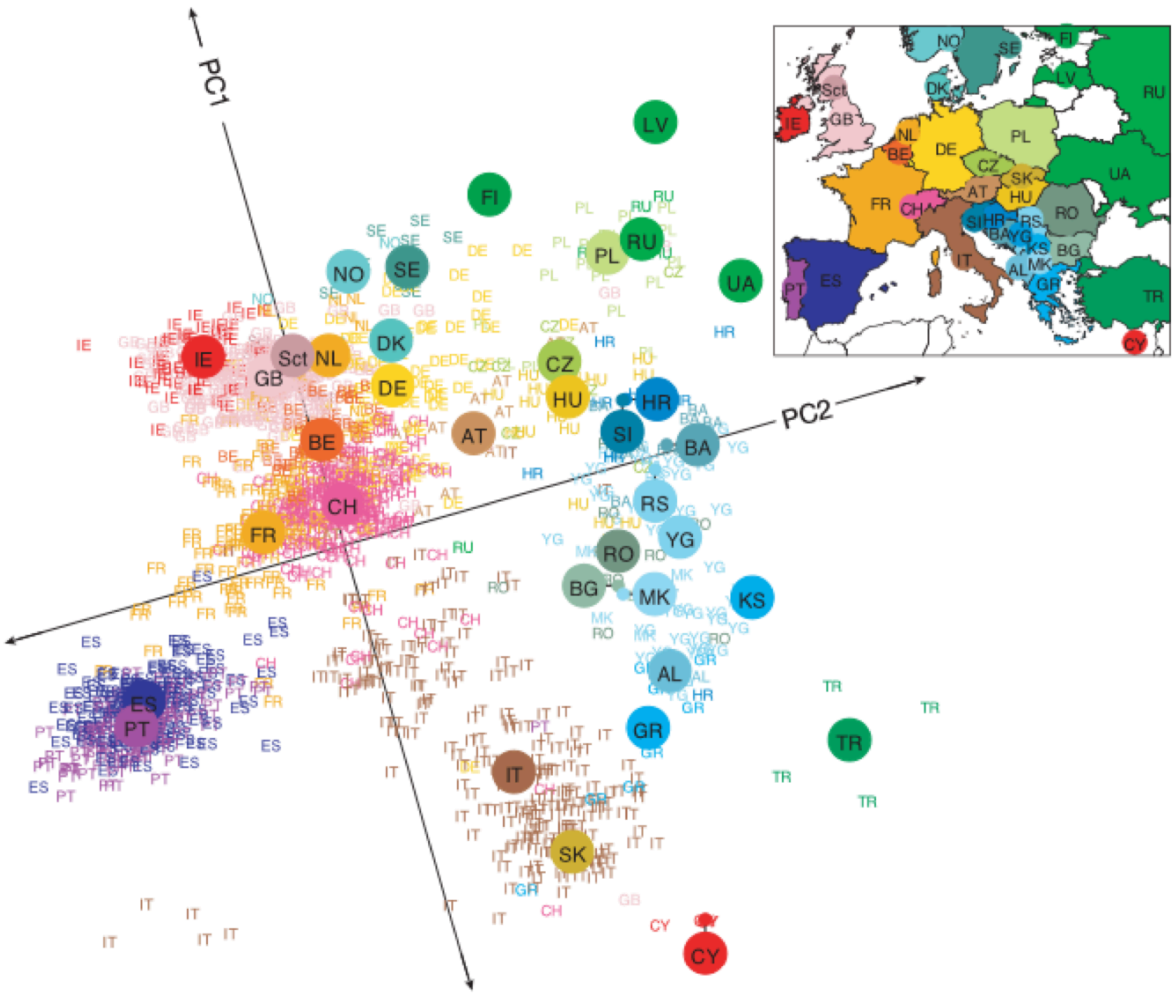


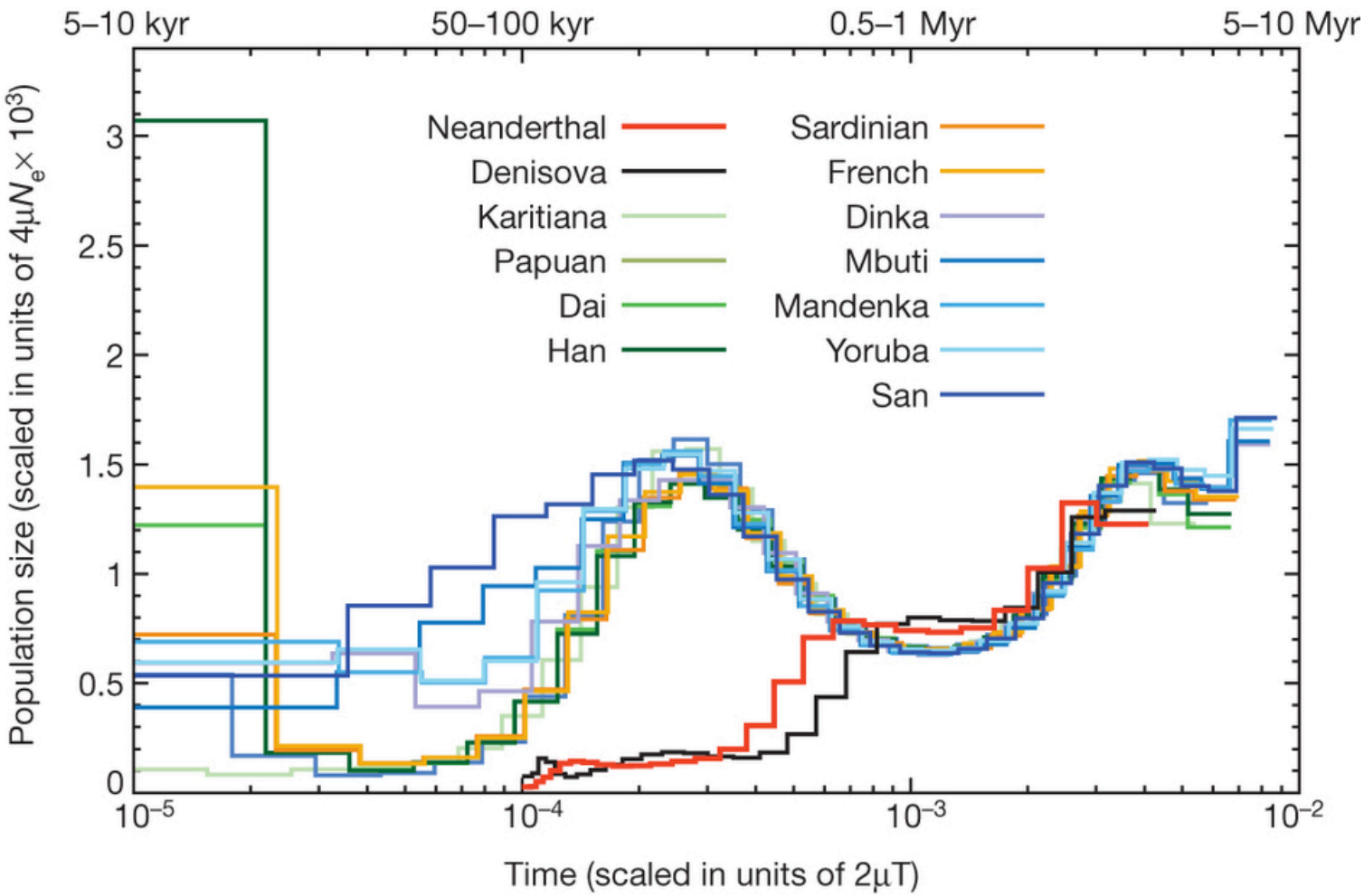
Figure: German Dziebel

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)

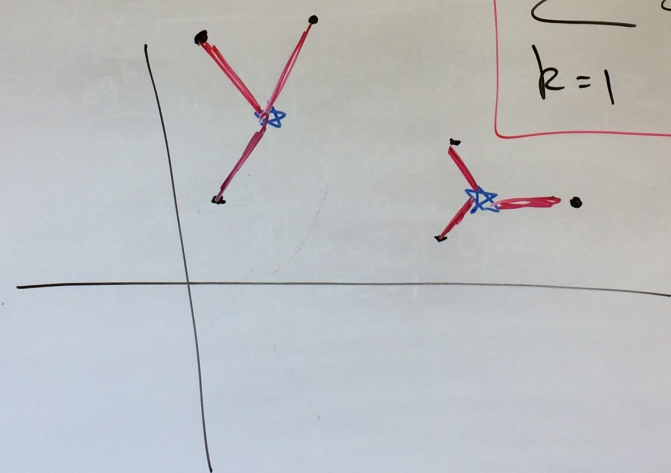
Outline for November 26

- Backpropogation
- Begin: unsupervised learning
- K-means
- Next week
  - Gaussian Mixture Models (GMM)
  - Principal Component Analysis (PCA)

Clustering Goals

* learn about the structure of the data.

* cluster new data ("test", prediction)



Goal find $\underbrace{\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K\}}_{K \text{ clusters}} = \mathcal{C}$

s.t.

minimize

within cluster sum of square (WCSS)

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

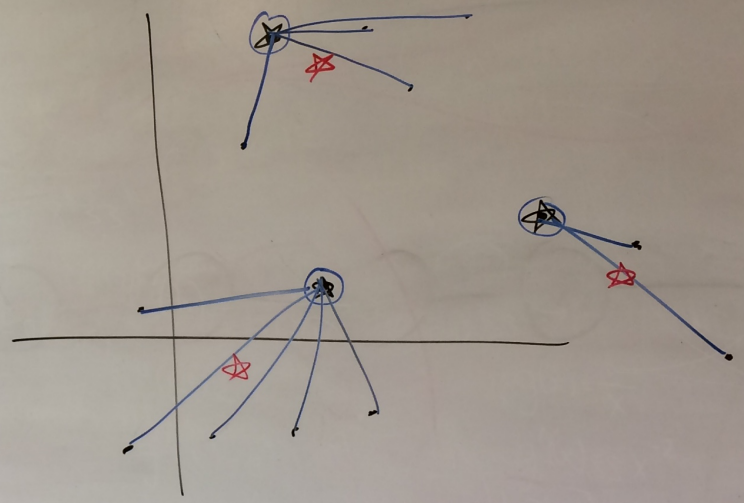
mean of cluster k

☆ cluster mean

— WCSS

K-means

$K = 3$



⓪ initialization

choose means (centers) from among the training data.

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

iterate

① $\boxed{\text{E-step}}$ Assignment: assign each datapoint to the cluster with the closest mean.

② $\boxed{\text{M-step}}$ Update: recompute each mean as the average of all cluster members (not data points)