CS4501: Introduction to Computer Vision

Introduction to Machine Learning

So far in this class

- Image Formation: Cameras, Projections, Light, Human Vision
- Image Processing: Basic Operations, Filtering, Convolutions, Mean Filter, Sobel Filter, Gaussian Filter
- Finding Edges – Canny , Corners -- Harris, Lines -- Hough Transform
- Finding Interest Points: Blob Filter, SIFT Features (also ORB + others)
- Feature Matching and Dense Stereo Matching / Homographies
- Camera Calibration / Estimating the Camera Matrix
- Epipolar Geometry: Essential Matrix, Fundamental Matrix

Today's Class

- Introduction to Machine Learning
  - Unsupervised Learning: Clustering (e.g. k-means clustering)
  - Supervised Learning: Classification (e.g. k-nearest neighbors, softmax classifier)

Machine Learning

- **Machine learning** is the subfield of computer science that gives "computers the ability to learn without being explicitly programmed."

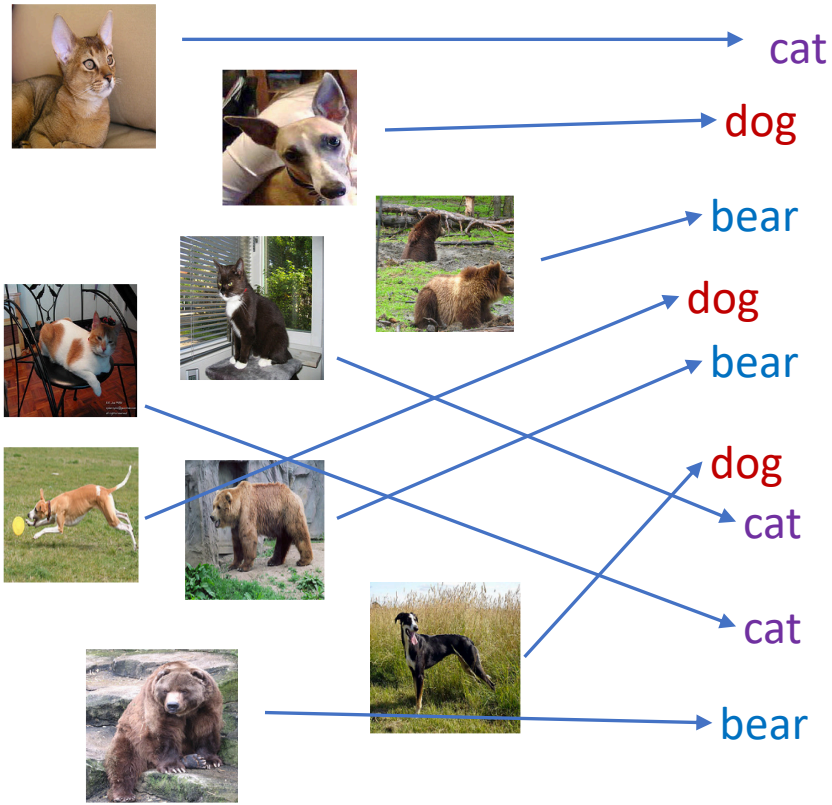
  - term coined by Arthur Samuel 1959 while at IBM

- The study of algorithms that can learn from data.

- In contrast to previous Artificial Intelligence systems based on Logic, e.g. "Expert Systems"

Supervised Learning vs Unsupervised Learning

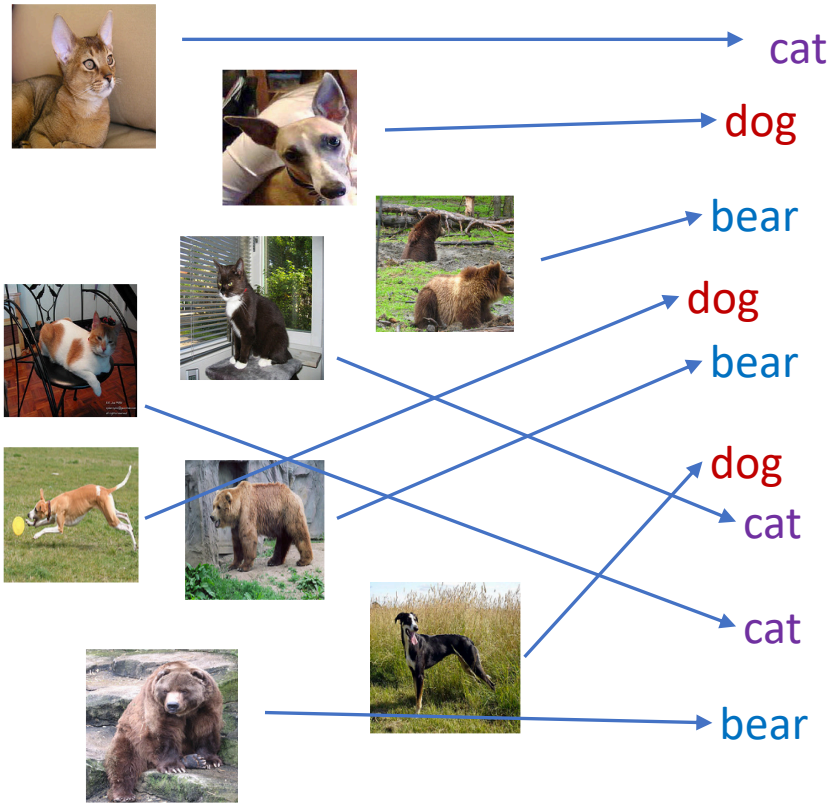
$x \rightarrow y$



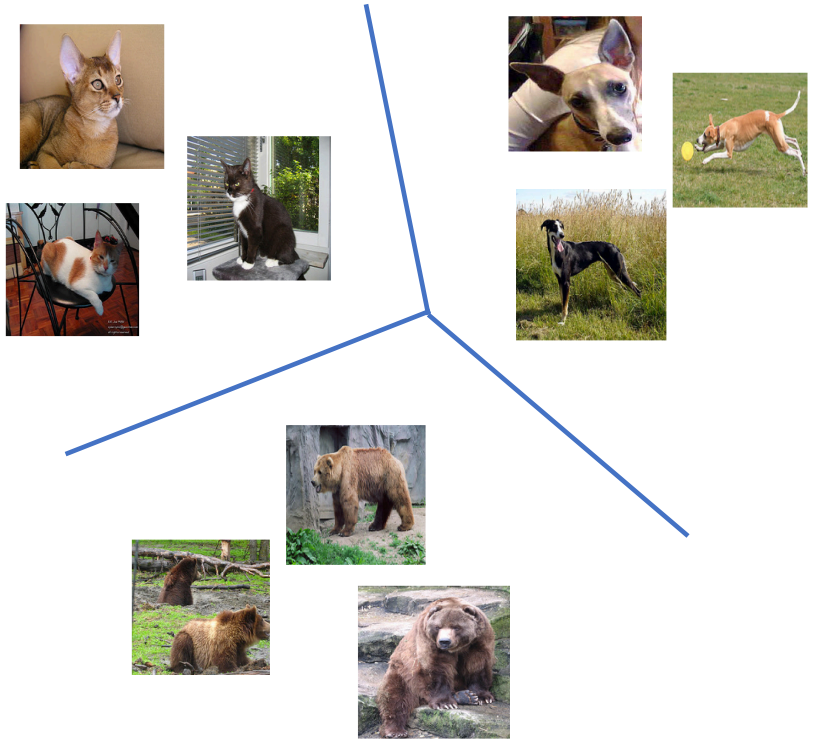
x

Supervised Learning vs Unsupervised Learning

$x \rightarrow y$

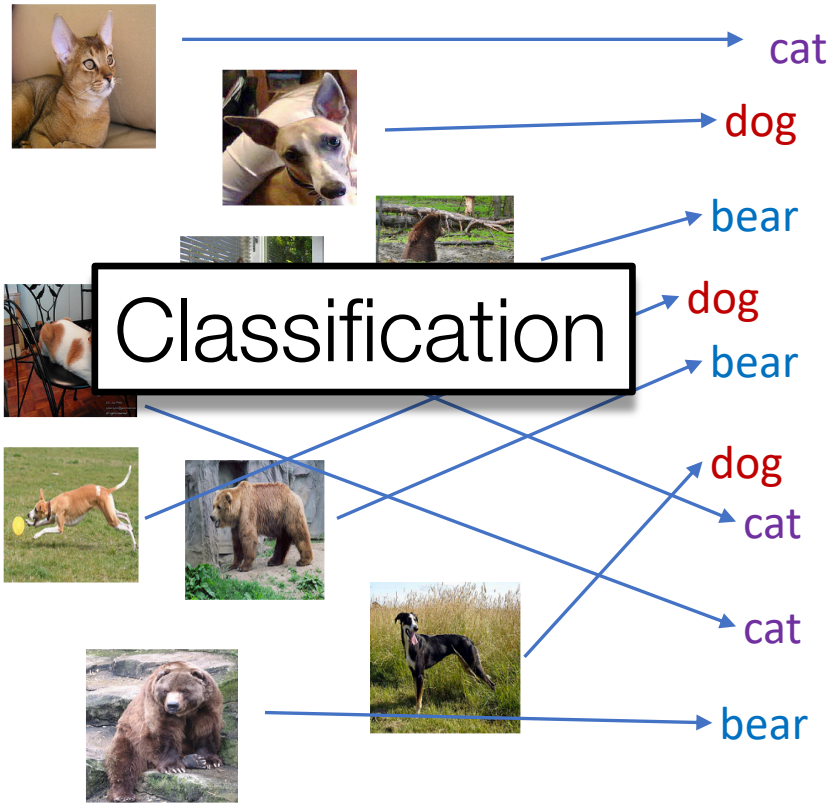


x



Supervised Learning vs Unsupervised Learning

$x \rightarrow y$



x

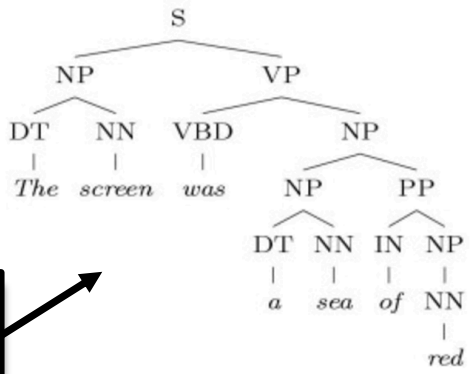
Supervised Learning Examples

Classification → cat

Face Detection →

The screen was a sea of red

Language Parsing →

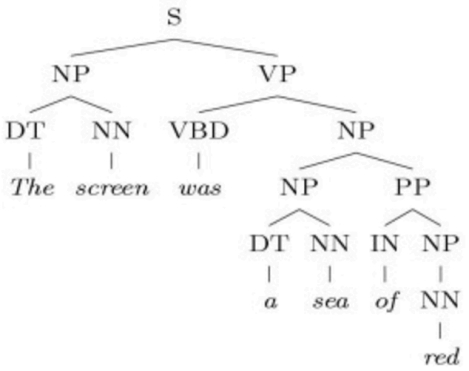


Structured Prediction

Supervised Learning Examples:

cat $= f($ $)$

$= f($ $)$



$= f($ The screen was a sea of red $)$

Estimate the function f from input/output examples

How to choose a problem with Machine Learning?

- Does the problem make sense?
  - Is the problem ill-posed? Are there problematic underlying assumptions?

How to choose a problem with Machine Learning?

- Does the problem make sense?
  - Is the problem ill-posed? Are there problematic underlying assumptions?

- Is the problem solvable?
  - Do the outputs can be reasonably resolved from the provided inputs? Is there a good reason to believe the task would work?

Automated Inference on Criminality using Face Images

https://arxiv.org/abs/1611.04135v1

(a) Three samples in criminal ID photo set S_c.

(b) Three samples in non-criminal ID photo set S_n

Figure 1. Sample ID photos in our data set.

How to choose a problem with Machine Learning?

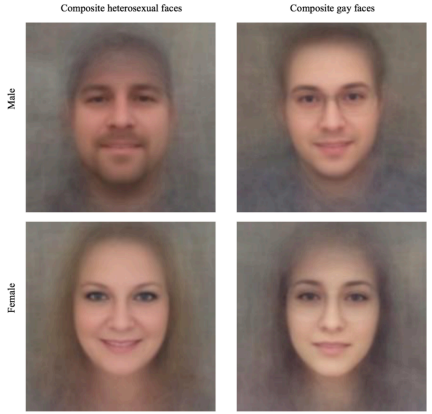
- Does the problem make sense?
  - Is the problem ill-posed? Are there problematic underlying assumptions?

- Is the problem solvable?
  - Do the outputs can be reasonably resolved from the provided inputs?

DEEP NEURAL NETWORKS CAN DETECT SEXUAL ORIENTATION FROM FACES

https://www.theregister.co.uk/2019/03/05/ai_gaydar/

https://medium.com/@blaisea/do-algorithms-reveal-sexual-orientation-or-just-expose-our-stereotypes-d998fafdf477



How to choose a problem with Machine Learning?

- Does a half-baked solution produce more harms than benefits?

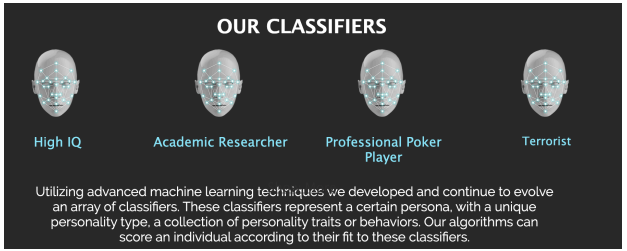
Falls in the previous category too

AI Is Now Analyzing Candidates' Facial Expressions During Video Job Interviews

https://www.inc.com/minda-zetlin/ai-is-now-analyzing-candidates-facial-expressions-during-video-job-interviews.html

Falls in the previous category too

FACEPTION
Facial Personality Analytics



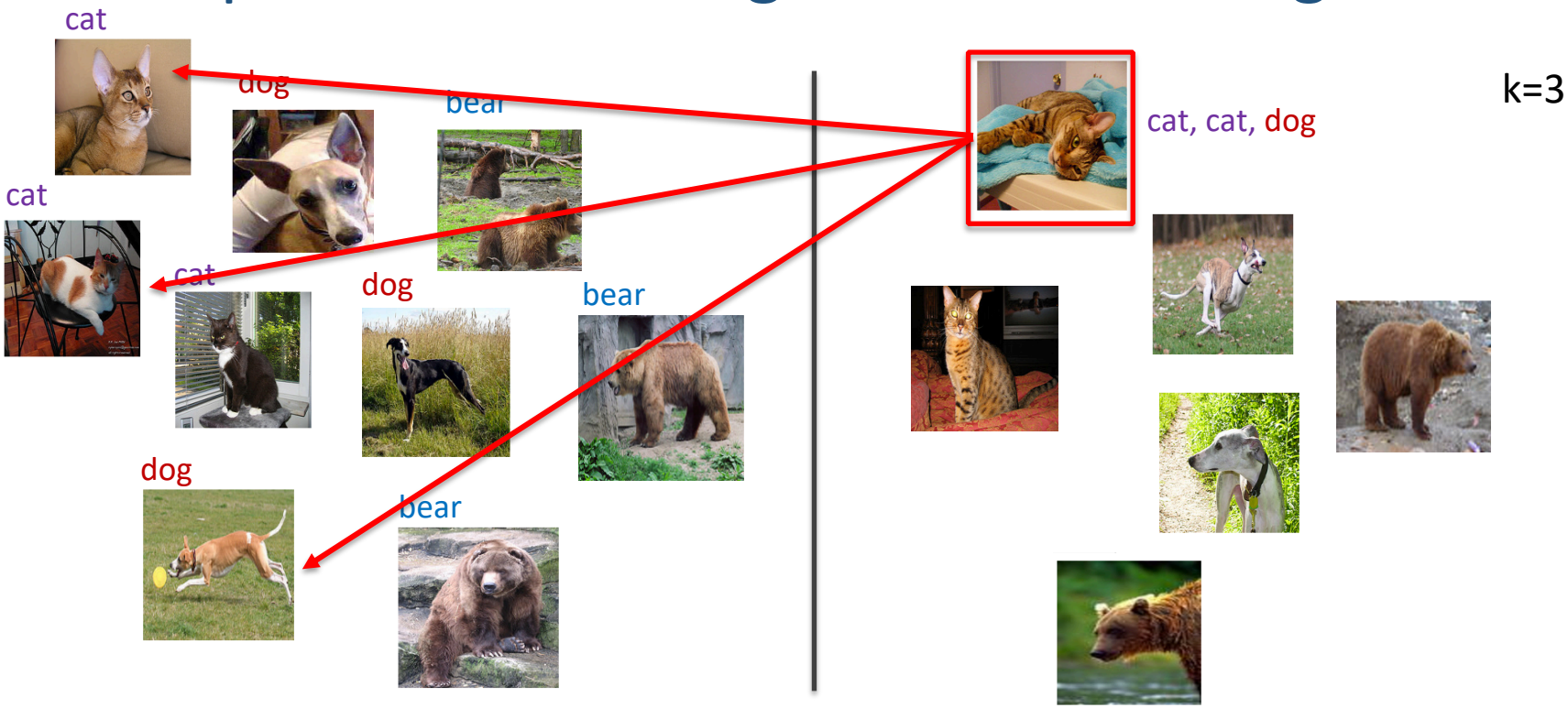
How to choose a problem with Machine Learning?

- Should we solve the problem? Is this a problem?
  - Does the solution produce more harms than benefits even if working accurately?
    - Several examples mentioned earlier, detecting criminality from faces, sexual orientation, job aptitude, etc. But also…
    - Intelligent weapons, massive surveillance, etc.

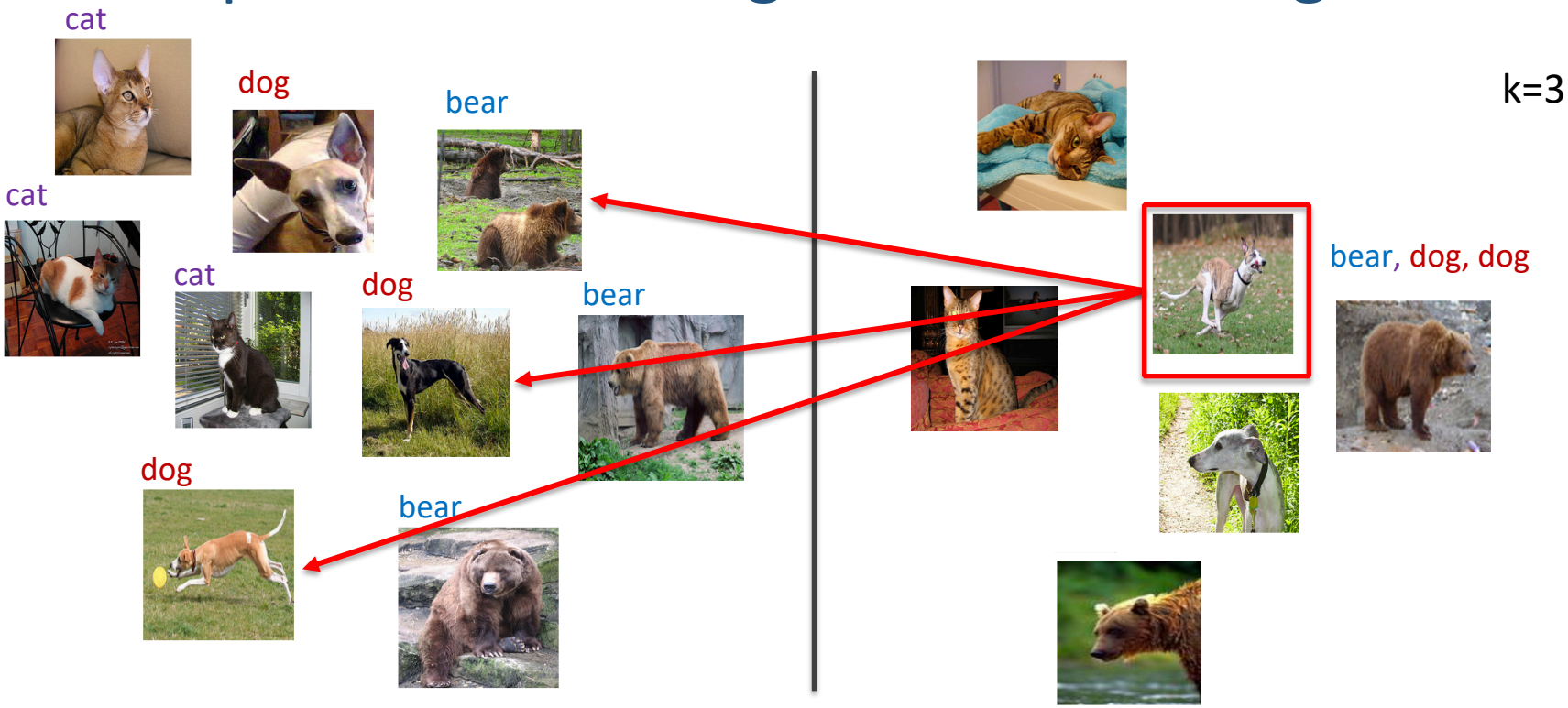
Back to Supervised vs Unsupervised

- Supervised Learning
- Unsupervised Learning

Supervised Learning – k-Nearest Neighbors



Supervised Learning – k-Nearest Neighbors



Supervised Learning – k-Nearest Neighbors

- How do we choose the right K?
- How do we choose the right features?
- How do we choose the right distance metric?

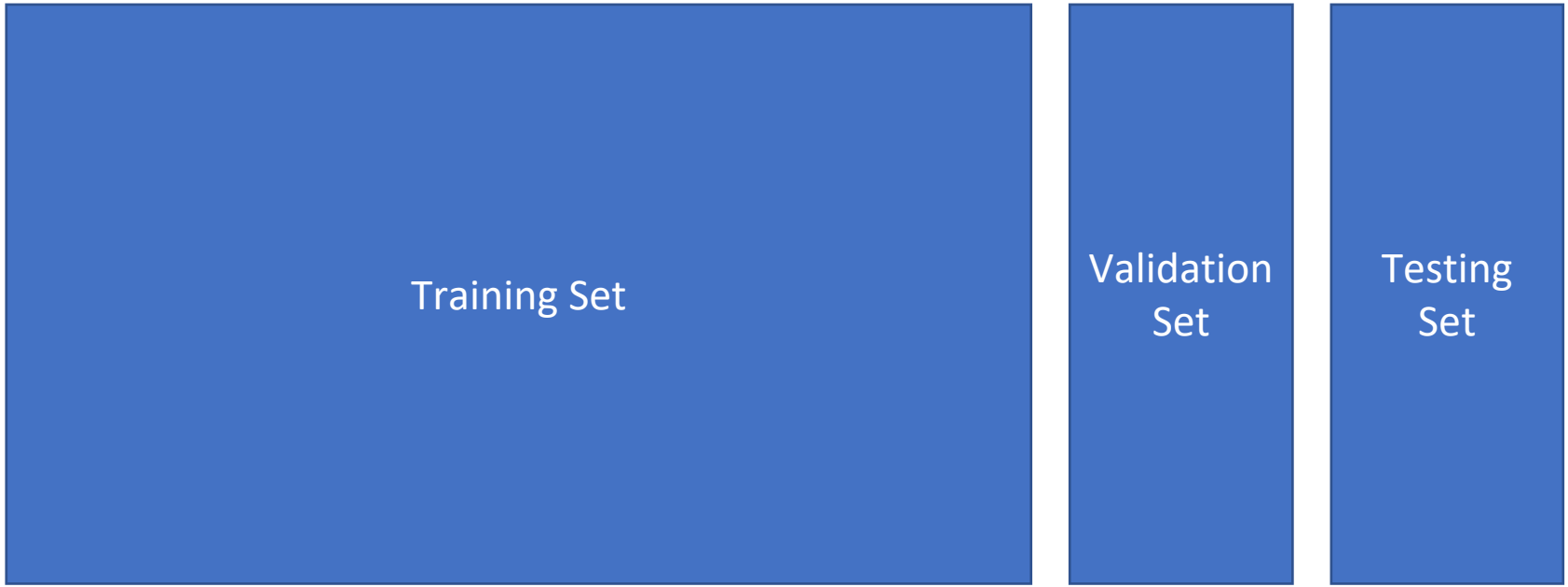
Supervised Learning – k-Nearest Neighbors

- How do we choose the right K?
- How do we choose the right features?
- How do we choose the right distance metric?

Answer: Just choose the one combination that works best!
BUT not on the test data.

Instead split the training data into a ”Training set” and a ”Validation set” (also called ”Development set”)

Training, Validation (Dev), Test Sets



Training, Validation (Dev), Test Sets

Used during development

Training, Validation (Dev), Test Sets

Only to be used for evaluating the model at the very end of development and any changes to the model after running it on the test set, could be influenced by what you saw happened on the test set, which would invalidate any future evaluation.

Unsupervised Learning – k-means clustering

k = 3

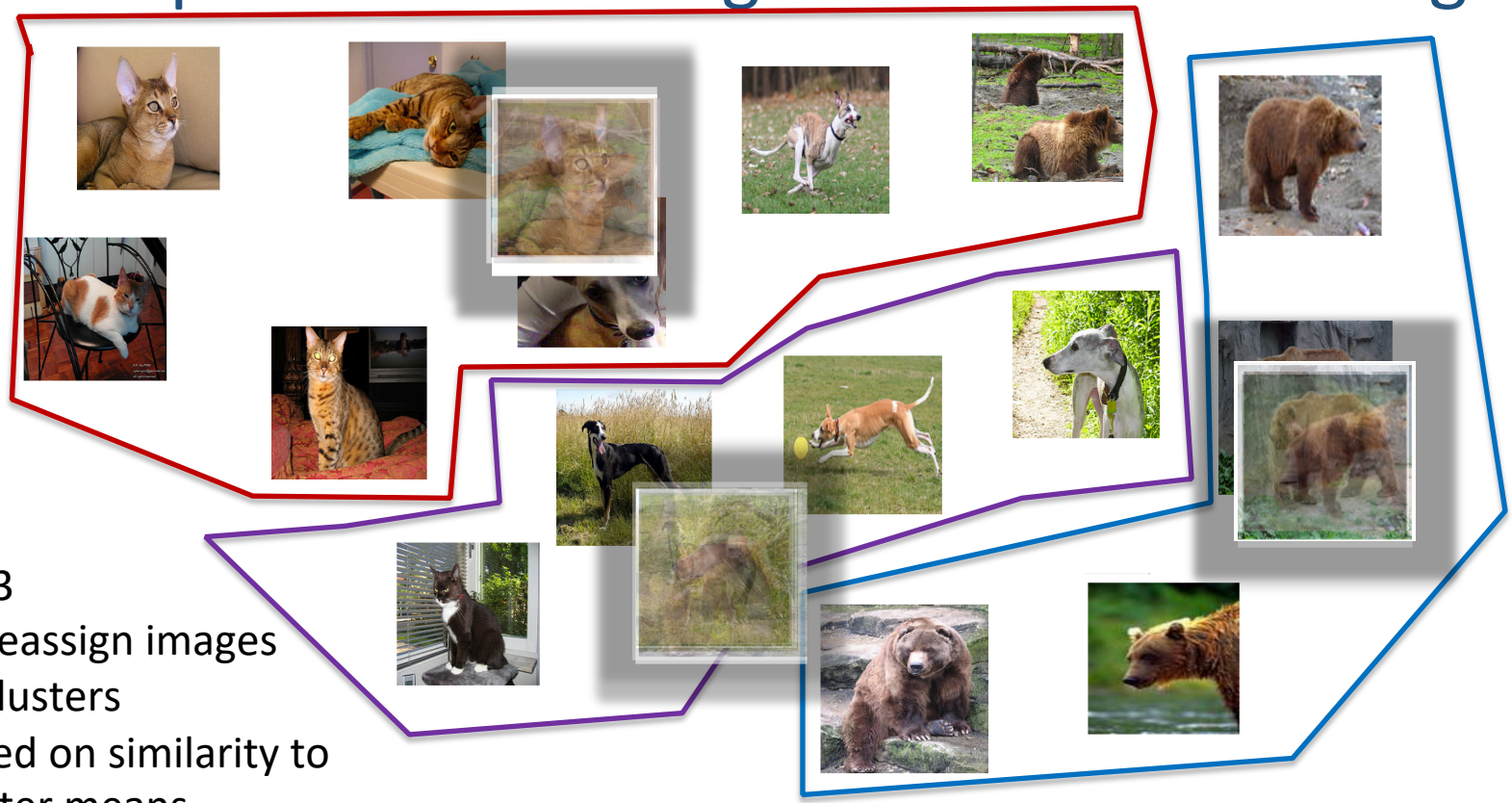
1. Initially assign all images to a random cluster

Unsupervised Learning – k-means clustering

k = 3

2. Compute the mean image (in feature space) for each cluster

Unsupervised Learning – k-means clustering



k = 3

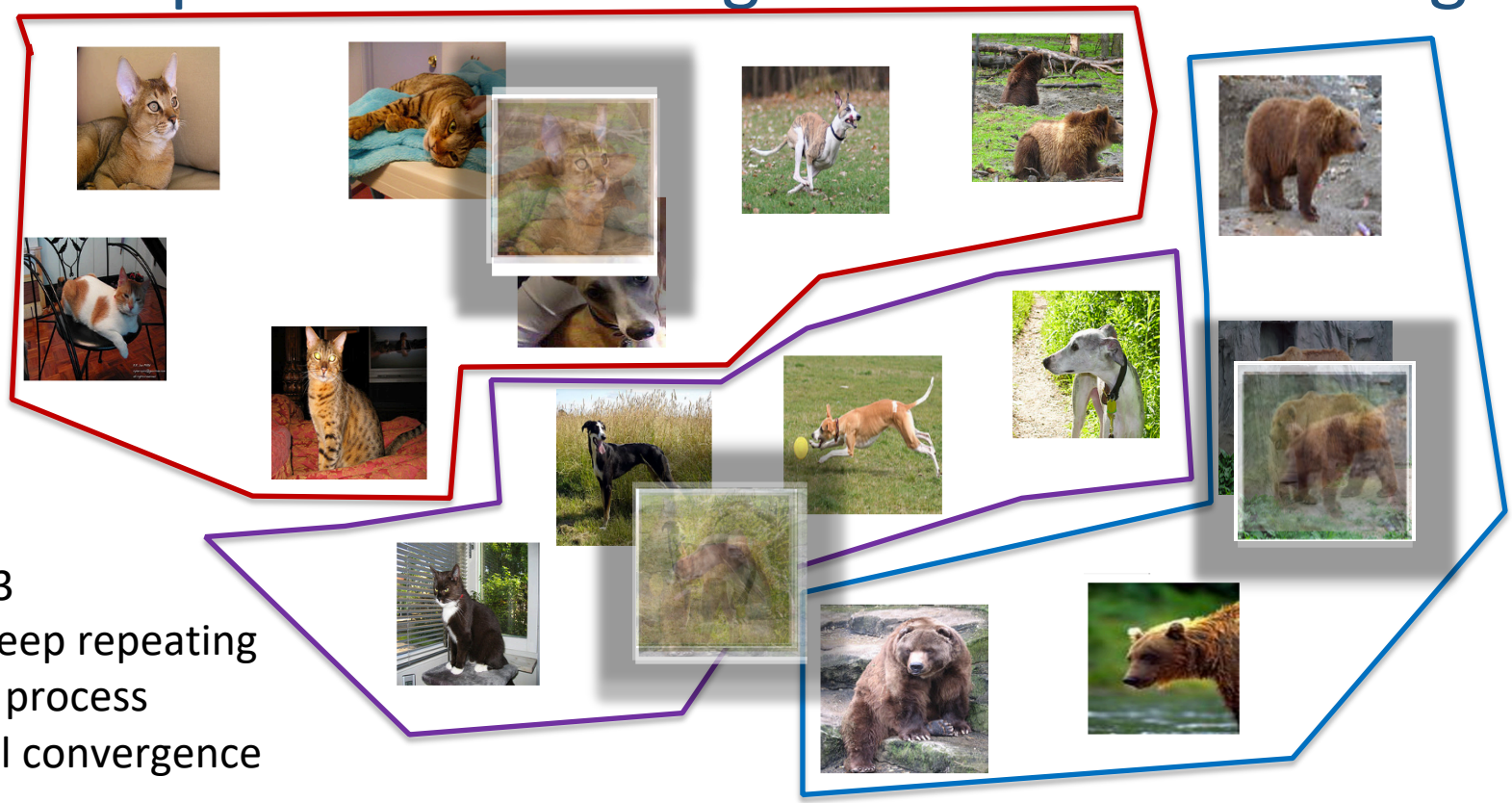
3. Reassign images to clusters based on similarity to cluster means

Unsupervised Learning – k-means clustering

k = 3

4. Keep repeating this process until convergence

Unsupervised Learning – k-means clustering



k = 3

4. Keep repeating this process until convergence

Unsupervised Learning – k-means clustering

k = 3

4. Keep repeating this process until convergence

Unsupervised Learning – k-means clustering

- How do we choose the right K?
- How do we choose the right features?
- How do we choose the right distance metric?
- How sensitive is this method with respect to the random assignment of clusters?

Answer: Just choose the one combination that works best!
BUT not on the test data.

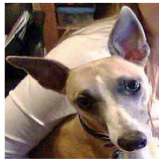
Instead split the training data into a ”Training set” and a ”Validation set” (also called ”Development set”)

Back to Supervised Learning - Classification

Training Data

cat

dog

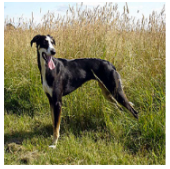


bear

cat

cat

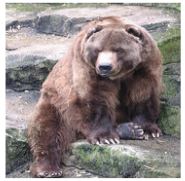
dog



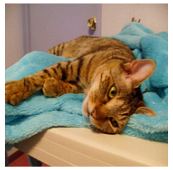
bear

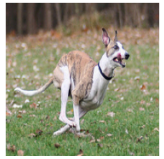
dog

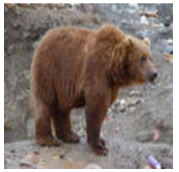
bear

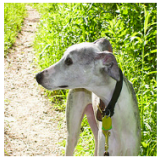


Test Data









Supervised Learning - Classification

Training Data

cat

dog

cat

.
.
.

bear

Test Data

.
.
.

Supervised Learning - Classification

Training Data

$x_1 = [$ $]$ $y_1 = [\text{cat} \quad]$

$x_2 = [$ $]$ $y_2 = [\text{dog} \quad]$

$x_3 = [$ $]$ $y_3 = [\text{cat} \quad]$

.
.
.

$x_n = [$ $]$ $y_n = [\text{bear} \quad]$

Supervised Learning - Classification

Training Data

inputs

targets / labels / ground truth

predictions

$x_1 = [x_{11} \quad x_{12} \quad x_{13} \quad x_{14}]$ $y_1 = 1$ $\hat{y}_1 = 1$

$x_2 = [x_{21} \quad x_{22} \quad x_{23} \quad x_{24}]$ $y_2 = 2$ $\hat{y}_2 = 2$

$x_3 = [x_{31} \quad x_{32} \quad x_{33} \quad x_{34}]$ $y_3 = 1$ $\hat{y}_3 = 2$

.
.
.

$x_n = [x_{n1} \quad x_{n2} \quad x_{n3} \quad x_{n4}]$ $y_n = 3$ $\hat{y}_n = 1$

We need to find a function that maps *x* and *y* for any of them.

$$\hat{y}_i = f(x_i; \theta)$$

How do we ”learn” the parameters of this function?

We choose ones that makes the following quantity small:

$$\sum_{i=1}^{n} Cost(\hat{y}_i, y_i)$$

Supervised Learning – Linear Softmax

Training Data

inputs

targets / labels / ground truth

$x_1 = [x_{11} \quad x_{12} \quad x_{13} \quad x_{14}]$ $y_1 = 1$

$x_2 = [x_{21} \quad x_{22} \quad x_{23} \quad x_{24}]$ $y_2 = 2$

$x_3 = [x_{31} \quad x_{32} \quad x_{33} \quad x_{34}]$ $y_3 = 1$

.
.
.

$x_n = [x_{n1} \quad x_{n2} \quad x_{n3} \quad x_{n4}]$ $y_n = 3$

Supervised Learning – Linear Softmax

Training Data

inputs

targets / labels / ground truth

predictions

$x_1 = [x_{11} \quad x_{12} \quad x_{13} \quad x_{14}]$ $y_1 = [1 \quad 0 \quad 0]$ $\hat{y}_1 = [0.85 \quad 0.10 \quad 0.05]$

$x_2 = [x_{21} \quad x_{22} \quad x_{23} \quad x_{24}]$ $y_2 = [0 \quad 1 \quad 0]$ $\hat{y}_2 = [0.20 \quad 0.70 \quad 0.10]$

$x_3 = [x_{31} \quad x_{32} \quad x_{33} \quad x_{34}]$ $y_3 = [1 \quad 0 \quad 0]$ $\hat{y}_3 = [0.40 \quad 0.45 \quad 0.15]$

.
.
.

$x_n = [x_{n1} \quad x_{n2} \quad x_{n3} \quad x_{n4}]$ $y_n = [0 \quad 0 \quad 1]$ $\hat{y}_n = [0.40 \quad 0.25 \quad 0.35]$

Supervised Learning – Linear Softmax

$x_i = [x_{i1} \quad x_{i2} \quad x_{i3} \quad x_{i4}]$ $\quad y_i = [1 \quad 0 \quad 0]$ $\quad \hat{y}_i = [f_c \quad f_d \quad f_b]$

$$g_c = w_{c1}x_{i1} + w_{c2}x_{i2} + w_{c3}x_{i3} + w_{c4}x_{i4} + b_c$$
$$g_d = w_{d1}x_{i1} + w_{d2}x_{i2} + w_{d3}x_{i3} + w_{d4}x_{i4} + b_d$$
$$g_b = w_{b1}x_{i1} + w_{b2}x_{i2} + w_{b3}x_{i3} + w_{b4}x_{i4} + b_b$$

$$f_c = e^{g_c}/(e^{g_c}+e^{g_d} + e^{g_b})$$
$$f_d = e^{g_d}/(e^{g_c}+e^{g_d} + e^{g_b})$$
$$f_b = e^{g_b}/(e^{g_c}+e^{g_d} + e^{g_b})$$

How do we find a good w and b?

$x_i = [x_{i1} \quad x_{i2} \quad x_{i3} \quad x_{i4}]$ $\quad y_i = [1 \quad 0 \quad 0]$ $\quad \hat{y}_i = [f_c(w,b) \quad f_d(w,b) \quad f_b(w,b)]$

We need to find w, and b that minimize the following:

$$L(w,b) = \sum_{i=1}^{n}\sum_{j=1}^{3} -y_{i,j}\log(\hat{y}_{i,j}) = \sum_{i=1}^{n} -\log(\hat{y}_{i,label}) = \sum_{i=1}^{n} -\log f_{i,label}(w,b)$$

Why?

Gradient Descent (GD)

$\lambda = 0.01$

Initialize w and b randomly

$$L(w,b) = \sum_{i=1}^{n} -\log f_{i,label}(w,b)$$

for e = 0, num_epochs **do**

Compute: $dL(w,b)/dw$ and $dL(w,b)/db$

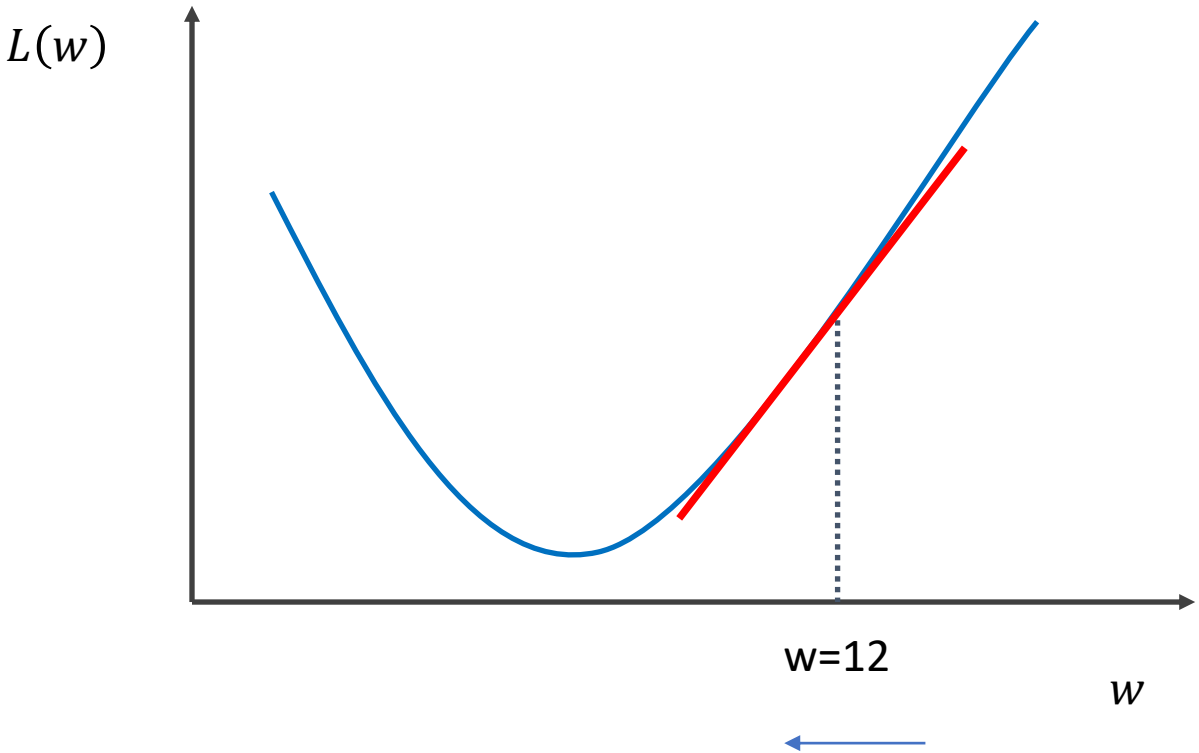
Update w: $w = w - \lambda\, dL(w,b)/dw$

Update b: $b = b - \lambda\, dL(w,b)/db$

Print: $L(w,b)$ // Useful to see if this is becoming smaller or not.

end

Gradient Descent (GD) (idea)



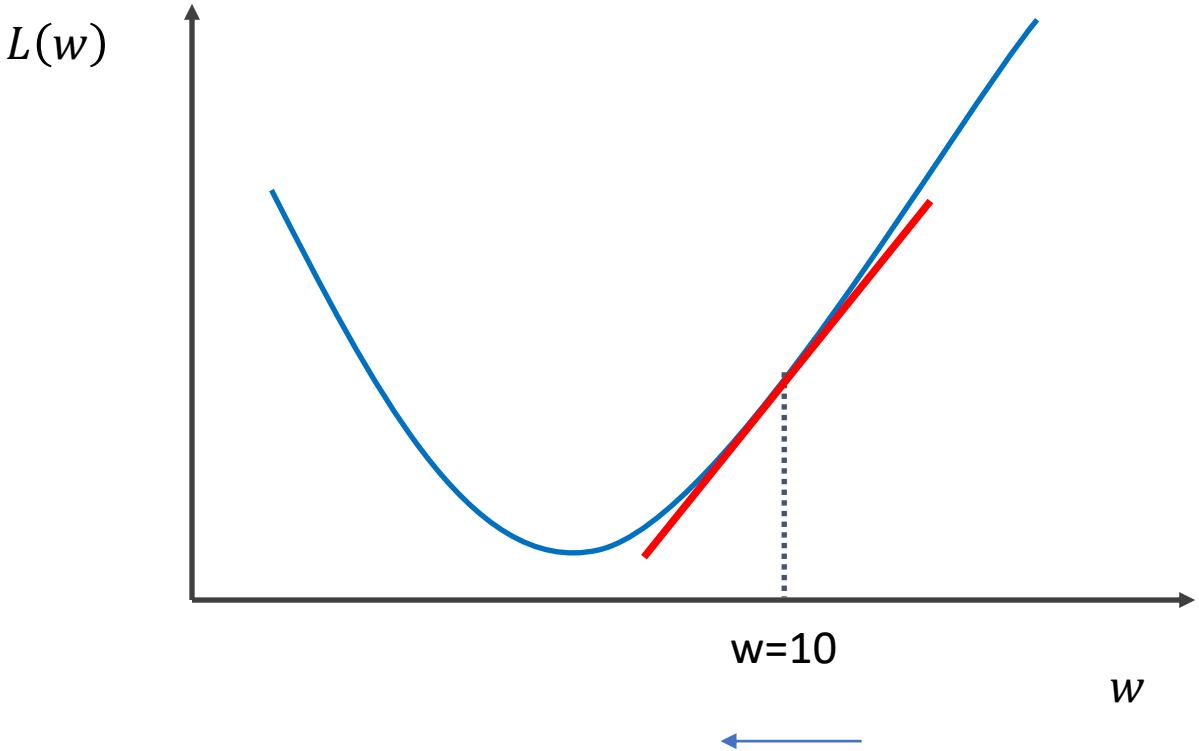
1. Start with a random value of w (e.g. w = 12)

2. Compute the gradient (derivative) of L(w) at point w = 12. (e.g. dL/dw = 6)

3. Recompute w as:

w = w – lambda * (dL / dw)

Gradient Descent (GD) (idea)

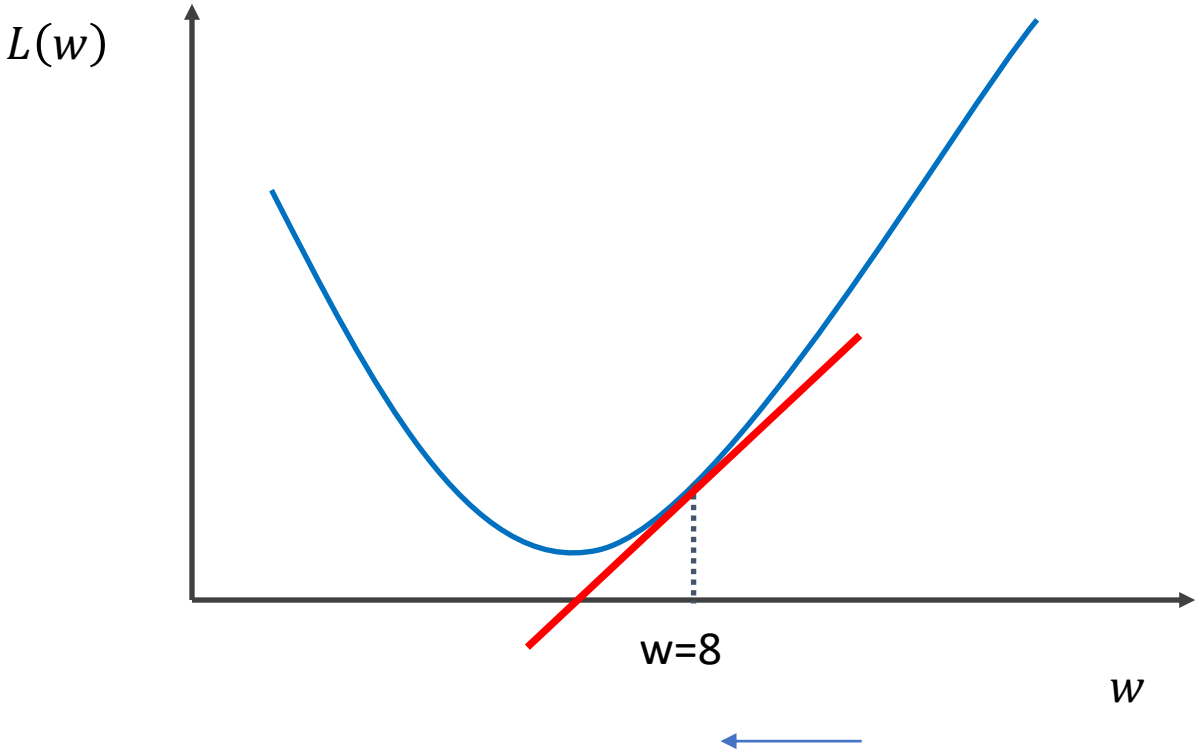


2. Compute the gradient (derivative) of L(w) at point w = 12. (e.g. dL/dw = 6)

3. Recompute w as:

w = w – lambda * (dL / dw)

Gradient Descent (GD) (idea)



2. Compute the gradient (derivative) of L(w) at point w = 12. (e.g. dL/dw = 6)

3. Recompute w as:

w = w – lambda * (dL / dw)

Our function L(w)

$$L(w) = 3 + (1 - w)^2$$

Our function L(w)

$$L(w) = 3 + (1 - w)^2$$

$$L(W, b) = \sum_{i=1}^{n} -\log f_{i,label}(W, b)$$

Our function L(w)

$$L(w) = 3 + (1 - w)^2$$

$$\mathrm{L}(w_1, w_2, .., w_{12}) = -logsoftmax\big(g(w_1, w_2, .., w_{12}, x_1)_{label_1}\big)$$
$$-logsoftmax\big(g(w_1, w_2, .., w_{12}, x_2)_{label_2}\big)$$
$$\dots$$
$$-logsoftmax\big(g(w_1, w_2, .., w_{12}, x_n)_{label_n}\big)$$

Gradient Descent (GD)

$\lambda = 0.01$

Initialize w and b randomly

for e = 0, num_epochs **do**

expensive

$$L(w,b) = \sum_{i=1}^{n} -\log f_{i,label}(w,b)$$

Compute: $dL(w,b)/dw$ and $dL(w,b)/db$

Update w: $w = w - \lambda\, dL(w,b)/dw$

Update b: $b = b - \lambda\, dL(w,b)/db$

Print: $L(w,b)$ // Useful to see if this is becoming smaller or not.

end

(mini-batch) Stochastic Gradient Descent (SGD)

$\lambda = 0.01$

Initialize w and b randomly

$$l(w,b) = \sum_{i \in B} -\log f_{i,label}(w,b)$$

for e = 0, num_epochs **do**

for b = 0, num_batches **do**

Compute: $dl(w,b)/dw$ and $dl(w,b)/db$

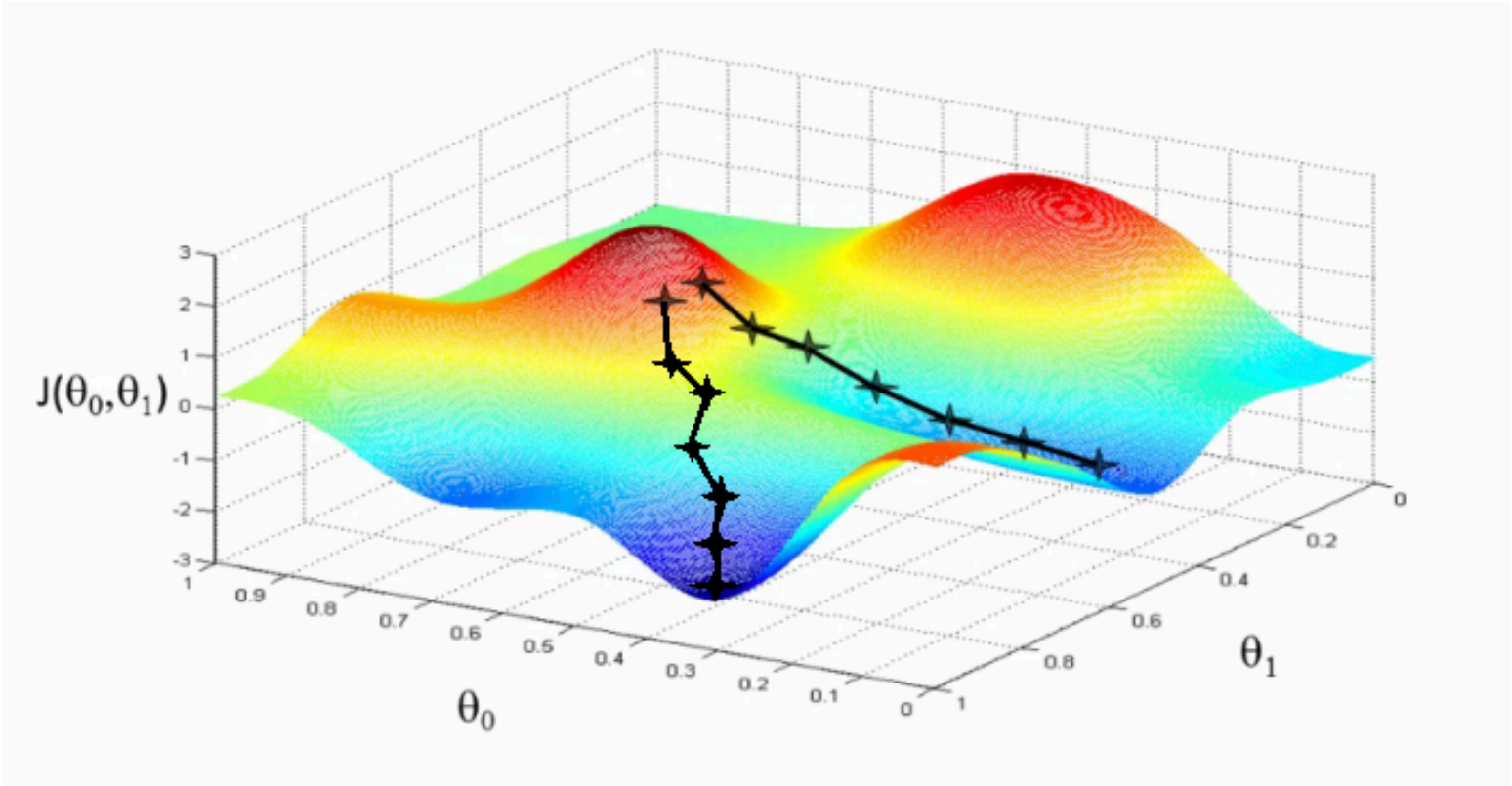
Update w: $w = w - \lambda\, dl(w,b)/dw$

Update b: $b = b - \lambda\, dl(w,b)/db$

Print: $l(w,b)$ // Useful to see if this is becoming smaller or not.

end

end



Source: Andrew Ng

(mini-batch) Stochastic Gradient Descent (SGD)

$\lambda = 0.01$

Initialize w and b randomly

$$l(w,b) = \sum_{i \in B} -\log f_{i,label}(w,b)$$

for e = 0, num_epochs **do**

for b = 0, num_batches **do**

Compute: $dl(w,b)/dw$ and $dl(w,b)/db$ for |B| = 1

Update w: $w = w - \lambda\, dl(w,b)/dw$

Update b: $b = b - \lambda\, dl(w,b)/db$

Print: $l(w,b)$ // Useful to see if this is becoming smaller or not.

end

end

Computing Analytic Gradients

This is what we have:

$$\ell(W, b) = -\log(\hat{y}_{label}(W, b)) = -\log\left(\frac{\exp(a_{label}(W, b))}{\sum_{k=1}^{10} \exp(a_k(W, b))}\right)$$

Computing Analytic Gradients

This is what we have:

$$\ell(W,b) = -\log(\hat{y}_{label}(W,b)) = -\log\left(\frac{\exp(a_{label}(W,b))}{\sum_{k=1}^{10}\exp(a_k(W,b))}\right)$$

$$\ell = -\log\left(\frac{\exp(a_{label})}{\sum_{k=1}^{10}\exp(a_k)}\right)$$

Reminder: $a_i = (w_{i,1}x_1 + w_{i,2} + w_{i,3} + w_{i,4}) + b_i$

Computing Analytic Gradients

This is what we have:

$$\ell = -\log\left(\frac{\exp(a_{label})}{\sum_{k=1}^{10}\exp(a_k)}\right)$$

Computing Analytic Gradients

This is what we have:

$$\ell = -\log\left(\frac{\exp(a_{label})}{\sum_{k=1}^{10}\exp(a_k)}\right)$$

This is what we need:

$\frac{\partial \ell}{\partial w_{ij}}$ for each w_{ij} $\frac{\partial \ell}{\partial b_i}$ for each b_i

Computing Analytic Gradients

This is what we have:

$$\ell = -\log\left(\frac{\exp(a_{label})}{\sum_{k=1}^{10}\exp(a_k)}\right)$$

Step 1: Chain Rule of Calculus

$$\frac{\partial \ell}{\partial w_{ij}} = \frac{\partial \ell}{\partial a_i}\frac{\partial a_i}{\partial w_{ij}} \qquad \frac{\partial \ell}{\partial b_i} = \frac{\partial \ell}{\partial a_i}\frac{\partial a_i}{\partial b_i}$$

Computing Analytic Gradients

This is what we have:

$$\ell = -\log\left(\frac{\exp(a_{label})}{\sum_{k=1}^{10}\exp(a_k)}\right)$$

Step 1: Chain Rule of Calculus

Let's do these first

$$\frac{\partial \ell}{\partial w_{ij}} = \frac{\partial \ell}{\partial a_i}\boxed{\frac{\partial a_i}{\partial w_{ij}}}$$

$$\frac{\partial \ell}{\partial b_i} = \frac{\partial \ell}{\partial a_i}\boxed{\frac{\partial a_i}{\partial b_i}}$$

Computing Analytic Gradients

$$\frac{\partial a_i}{\partial w_{ij}}$$

$$\frac{\partial a_i}{\partial b_i}$$

$$a_i = (w_{i,1}x_1 + w_{i,2} + w_{i,3} + w_{i,4}) + b_i$$

$$\frac{\partial a_i}{\partial w_{i,3}} = \frac{\partial}{\partial w_{i,3}}(w_{i,1}x_1 + w_{i,2}x_2 + w_{i,3}x_3 + w_{i,4}x_4) + b_i$$

$$\frac{\partial a_i}{\partial w_{i,3}} = x_3$$

$$\frac{\partial a_i}{\partial w_{i,j}} = x_j$$

Computing Analytic Gradients

$$\frac{\partial a_i}{\partial w_{i,j}} = x_j$$

$$\frac{\partial a_i}{\partial b_i}$$

$$a_i = (w_{i,1}x_1 + w_{i,2} + w_{i,3} + w_{i,4}) + b_i$$

$$\frac{\partial a_i}{\partial b_i} = \frac{\partial}{\partial b_i}(w_{i,1}x_1 + w_{i,2}x_2 + w_{i,3}x_3 + w_{i,4}x_4) + b_i$$

$$\frac{\partial a_i}{\partial b_i} = 1$$

Computing Analytic Gradients

$$\frac{\partial a_i}{\partial w_{i,j}} = x_j$$

$$\frac{\partial a_i}{\partial b_i} = 1$$

Computing Analytic Gradients

This is what we have:

$$\ell = -\log\left(\frac{\exp(a_{label})}{\sum_{k=1}^{10}\exp(a_k)}\right)$$

Step 1: Chain Rule of Calculus

Now let's do this one (same for both!)

$$\frac{\partial \ell}{\partial w_{ij}} = \boxed{\frac{\partial \ell}{\partial a_i}}\frac{\partial a_i}{\partial w_{ij}} \qquad \frac{\partial \ell}{\partial b_i} = \boxed{\frac{\partial \ell}{\partial a_i}}\frac{\partial a_i}{\partial b_i}$$

Computing Analytic Gradients

$$\frac{\partial \ell}{\partial a_i} = \frac{\partial}{\partial a_i}\left[-\log\left(\frac{\exp(a_{label})}{\sum_{k=1}^{10}\exp(a_k)}\right)\right]$$

$$= \frac{\partial}{\partial a_i}\left[\log\left(\sum_{k=1}^{10}\exp(a_k)\right) - a_{label}\right]$$

In our cat, dog, bear classification example: i = {0, 1, 2}

Computing Analytic Gradients

$$\frac{\partial \ell}{\partial a_i} = \frac{\partial}{\partial a_i}\left[-\log\left(\frac{\exp(a_{label})}{\sum_{k=1}^{10} \exp(a_k)} \right) \right]$$

$$= \frac{\partial}{\partial a_i}\left[\log\left(\sum_{k=1}^{10} \exp(a_k) \right) - a_{label} \right]$$

In our cat, dog, bear classification example: i = {0, 1, 2}

Let's say: label = 1

We need: $\frac{\partial \ell}{\partial a_0}$ $\frac{\partial \ell}{\partial a_1}$ $\frac{\partial \ell}{\partial a_2}$

Computing Analytic Gradients

$$= \frac{\partial}{\partial a_i}\left[\log\left(\sum_{k=1}^{10}\exp(a_k)\right) - a_{label}\right]$$

$\frac{\partial \ell}{\partial a_0}$ $\frac{\partial \ell}{\partial a_2}$

when $i \neq label$:

$$\frac{\partial \ell}{\partial a_i} = \frac{\partial}{\partial a_i}\left[\log\left(\sum_{k=1}^{10}\exp(a_k)\right) - a_{label}\right]$$

$$\frac{\partial \ell}{\partial a_i} = \frac{\partial}{\partial a_i}\log\left(\sum_{k=1}^{10}\exp(a_k)\right)$$

$$\frac{\partial \ell}{\partial a_i} = \left(\frac{1}{\sum_{k=1}^{10}\exp(a_k)}\right)\left(\frac{\partial}{\partial a_i}\sum_{k=1}^{10}\exp(a_k)\right)$$

$$\frac{\partial \ell}{\partial a_i} = \frac{\exp(a_i)}{\sum_{k=1}^{10}\exp(a_k)} = \hat{y}_i$$

Remember this slide?

$x_i = [x_{i1} \quad x_{i2} \quad x_{i3} \quad x_{i4}]$ $\quad y_i = [1 \quad 0 \quad 0]$ $\quad \hat{y}_i = [f_c \quad f_d \quad f_b]$

$$g_c = w_{c1}x_{i1} + w_{c2}x_{i2} + w_{c3}x_{i3} + w_{c4}x_{i4} + b_c$$
$$g_d = w_{d1}x_{i1} + w_{d2}x_{i2} + w_{d3}x_{i3} + w_{d4}x_{i4} + b_d$$
$$g_b = w_{b1}x_{i1} + w_{b2}x_{i2} + w_{b3}x_{i3} + w_{b4}x_{i4} + b_b$$

$$f_c = e^{g_c}/(e^{g_c}+e^{g_d} + e^{g_b})$$
$$f_d = e^{g_d}/(e^{g_c}+e^{g_d} + e^{g_b})$$
$$f_b = e^{g_b}/(e^{g_c}+e^{g_d} + e^{g_b})$$

Computing Analytic Gradients

$$= \frac{\partial}{\partial a_i}\left[\log\left(\sum_{k=1}^{10}\exp(a_k)\right) - a_{label}\right]$$

$\frac{\partial \ell}{\partial a_0}$ $\frac{\partial \ell}{\partial a_2}$

when $i \neq label$:

$$\frac{\partial \ell}{\partial a_i} = \frac{\partial}{\partial a_i}\left[\log\left(\sum_{k=1}^{10}\exp(a_k)\right) - a_{label}\right]$$

$$\frac{\partial \ell}{\partial a_i} = \frac{\partial}{\partial a_i}\log\left(\sum_{k=1}^{10}\exp(a_k)\right)$$

$$\frac{\partial \ell}{\partial a_i} = \left(\frac{1}{\sum_{k=1}^{10}\exp(a_k)}\right)\left(\frac{\partial}{\partial a_i}\sum_{k=1}^{10}\exp(a_k)\right)$$

$$\frac{\partial \ell}{\partial a_i} = \frac{\exp(a_i)}{\sum_{k=1}^{10}\exp(a_k)} = \hat{y}_i$$

Computing Analytic Gradients

$$= \frac{\partial}{\partial a_i}\left[\log\left(\sum_{k=1}^{10}\exp(a_k)\right) - a_{label}\right]$$

$\frac{\partial \ell}{\partial a_1}$

when $i = label$:

$$\frac{\partial \ell}{\partial a_{label}} = \frac{\partial}{\partial a_{label}}\left[\log\left(\sum_{k=1}^{10}\exp(a_k) - a_{label}\right)\right]$$

$$\frac{\partial \ell}{\partial a_{label}} = \frac{\partial}{\partial a_{label}}\log\left(\sum_{k=1}^{10}\exp(a_k)\right) - 1$$

$$\frac{\partial \ell}{\partial a_{label}} = \left(\frac{1}{\sum_{k=1}^{10}\exp(a_k)}\right)\left(\frac{\partial}{\partial a_{label}}\sum_{k=1}^{10}\exp(a_k)\right) - 1$$

$$\frac{\partial \ell}{\partial a_{label}} = \frac{\exp(a_{label})}{\sum_{k=1}^{10}\exp(a_k)} - 1 \qquad = \hat{y}_i - 1$$

Computing Analytic Gradients

label = 1

$$\frac{\partial \ell}{\partial a_0} = \hat{y}_0 \qquad \frac{\partial \ell}{\partial a_1} = \hat{y}_1 - 1 \qquad \frac{\partial \ell}{\partial a_1} = \hat{y}_2$$

$$\frac{\partial \ell}{\partial a} = \begin{bmatrix} \frac{\partial \ell}{\partial a_0} \\ \frac{\partial \ell}{\partial a_1} \\ \frac{\partial \ell}{\partial a_2} \end{bmatrix} = \begin{bmatrix} \hat{y}_0 \\ \hat{y}_1 - 1 \\ \hat{y}_2 \end{bmatrix} = \begin{bmatrix} \hat{y}_0 \\ \hat{y}_1 \\ \hat{y}_2 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \hat{y} - y$$

$$\frac{\partial \ell}{\partial a_i} = \hat{y}_i - y_i$$

Computing Analytic Gradients

$$\frac{\partial \ell}{\partial w_{ij}} = \frac{\partial \ell}{\partial a_i} \frac{\partial a_i}{\partial w_{ij}}$$

$$\frac{\partial \ell}{\partial b_i} = \frac{\partial \ell}{\partial a_i} \frac{\partial a_i}{\partial b_i}$$

$$\frac{\partial a_i}{\partial w_{i,j}} = x_j$$

$$\frac{\partial a_i}{\partial b_i} = 1$$

$$\frac{\partial \ell}{\partial a_i} = \hat{y}_i - y_i$$

$$\boxed{\frac{\partial \ell}{\partial w_{i,j}} = (\hat{y}_i - y_i)x_j}$$

$$\boxed{\frac{\partial \ell}{\partial b_i} = (\hat{y}_i - y_i)}$$

Supervised Learning –Softmax Classifier

$$\hat{y}_i = [f_c \quad f_d \quad f_b]$$

Extract features

$$x_i = [x_{i1} \quad x_{i2} \quad x_{i3} \quad x_{i4}]$$

Run features through classifier

$$g_c = w_{c1}x_{i1} + w_{c2}x_{i2} + w_{c3}x_{i3} + w_{c4}x_{i4} + b_c$$

$$g_d = w_{d1}x_{i1} + w_{d2}x_{i2} + w_{d3}x_{i3} + w_{d4}x_{i4} + b_d$$

$$g_b = w_{b1}x_{i1} + w_{b2}x_{i2} + w_{b3}x_{i3} + w_{b4}x_{i4} + b_b$$

$$f_c = e^{g_c}/(e^{g_c}+e^{g_d} + e^{g_b})$$

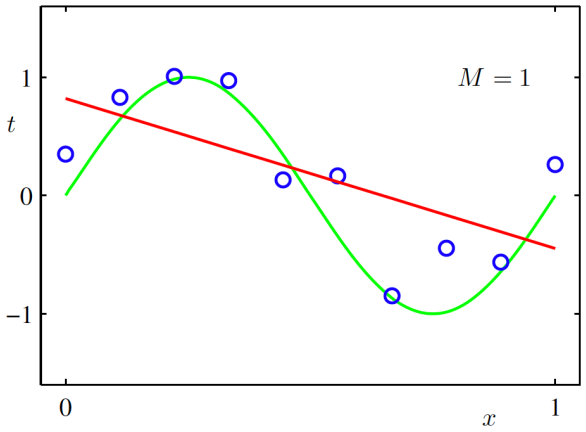
$$f_d = e^{g_d}/(e^{g_c}+e^{g_d} + e^{g_b})$$

$$f_b = e^{g_b}/(e^{g_c}+e^{g_d} + e^{g_b})$$

Get predictions

Overfitting

f is linear

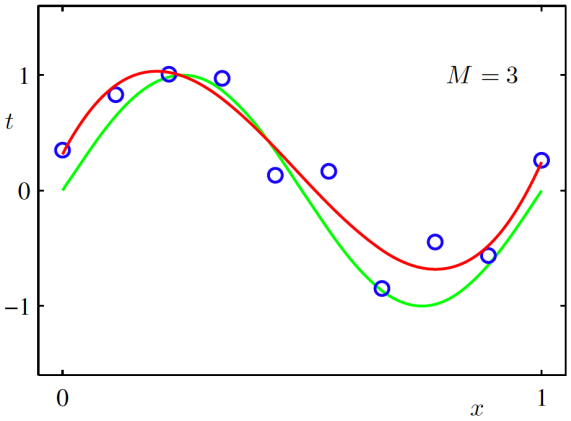


$Loss(w)$ is high

Underfitting

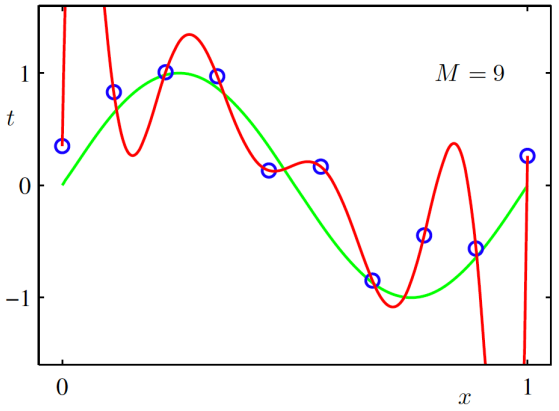
High Bias

f is cubic



$Loss(w)$ is low

f is a polynomial of degree 9



$Loss(w)$ is zero!

Overfitting

High Variance

More …

- Regularization
- Momentum updates
- Hinge Loss, Least Squares Loss, Logistic Regression Loss

Questions?