CS 4803 / 7643: Deep Learning

Topics:

- Linear Classifiers
- Loss Functions

Dhruv Batra

Georgia Tech

Administrativia

- Notes on class webpage
  - https://www.cc.gatech.edu/classes/AY2019/cs7643_fall/

- HW0 Reminder
  - Due: 09/05

Recap from last time

Image Classification: A core task in Computer Vision

This image by Nikita is licensed under CC-BY 2.0

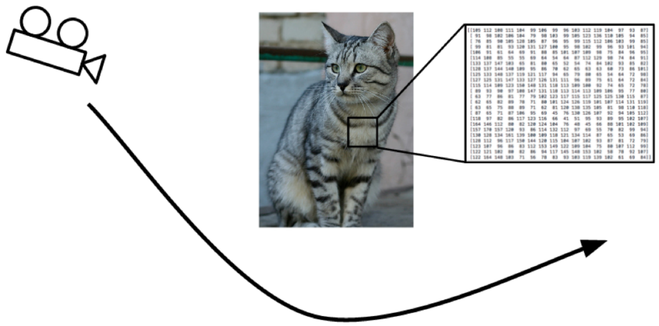
(assume given set of discrete labels)
{dog, cat, truck, plane, ...}

→ cat

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Challenges of recognition

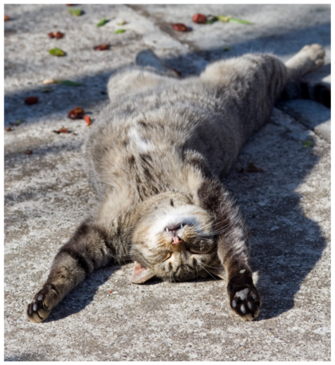
Viewpoint



Illumination

This image is CC0 1.0 public domain

Deformation



This image by Umberto Salvagnin is licensed under CC-BY 2.0

Occlusion

This image by jonsson is licensed under CC-BY 2.0

Clutter

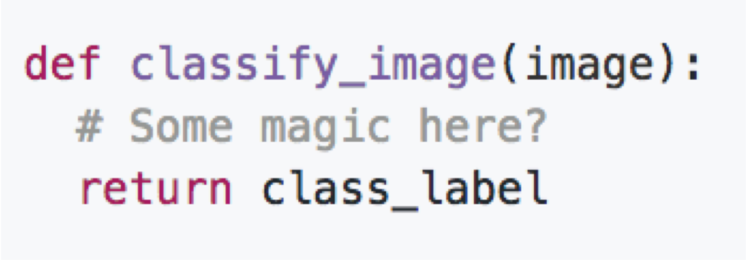
This image is CC0 1.0 public domain

Intraclass Variation

This image is CC0 1.0 public domain

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

An image classifier



```python
def classify_image(image):
  # Some magic here?
  return class_label
```

Unlike e.g. sorting a list of numbers,

no obvious way to hard-code the algorithm for recognizing a cat, or other classes.

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Supervised Learning

- Input: x (images, text, emails…)
- Output: y (spam or non-spam…)

- (Unknown) Target Function
  - f: X → Y (the "true" mapping / reality)

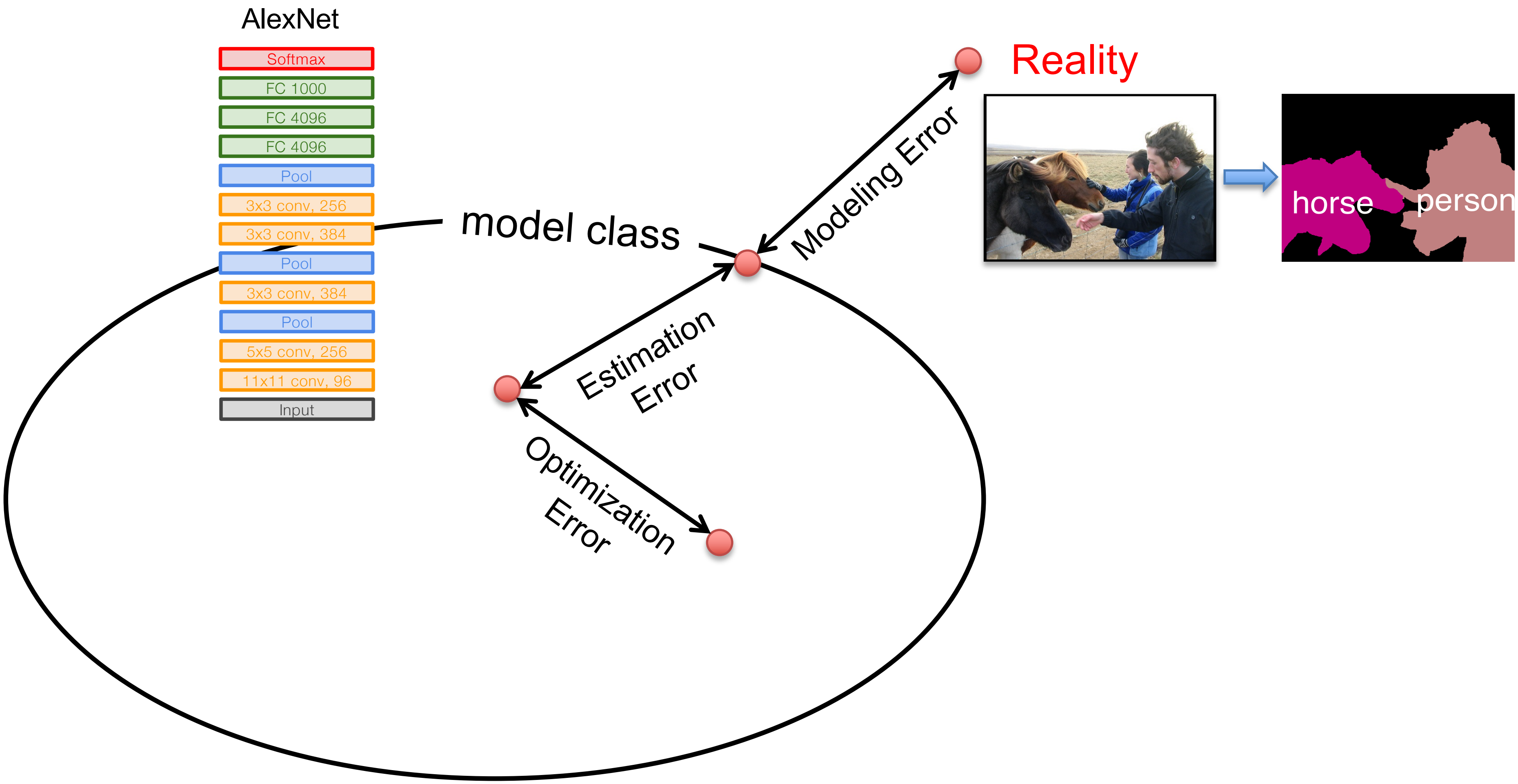
- Data
  - $(x_1,y_1), (x_2,y_2), \ldots, (x_N,y_N)$

- Model / Hypothesis Class
  - {h: X → Y}
  - e.g. $y = h(x) = \text{sign}(w^T x)$

- Loss Function
  - How good is a model wrt my data D?

- Learning = Search in hypothesis space
  - Find best h in model class.

Error Decomposition



AlexNet

Softmax

FC 1000

FC 4096

FC 4096

Pool

3x3 conv, 256

3x3 conv, 384

Pool

3x3 conv, 384

Pool

5x5 conv, 256

11x11 conv, 96

Input

model class

Reality

Modeling Error

Estimation Error

Optimization Error

horse

person

Error Decomposition

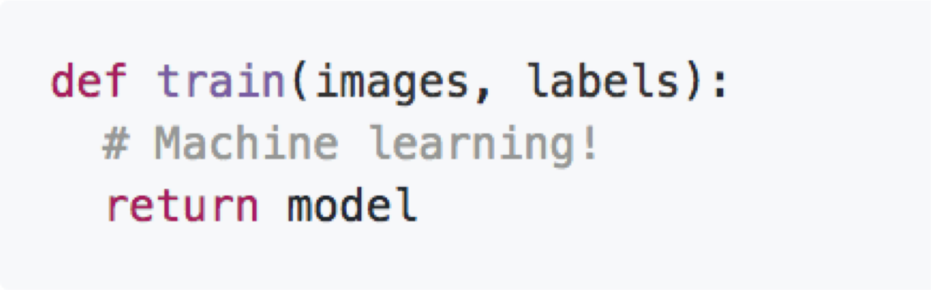
- Approximation/Modeling Error
  - You approximated reality with model

- Estimation Error
  - You tried to learn model with finite data

- Optimization Error
  - You were lazy and couldn’t/didn’t optimize to completion

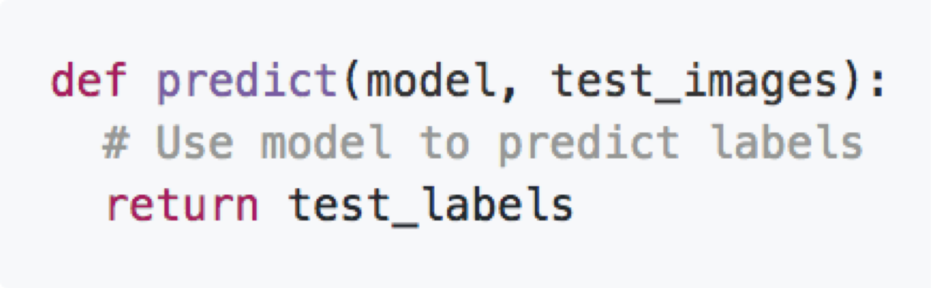
- Bayes Error
  - Reality just sucks

First classifier: **Nearest Neighbor**



```python
def train(images, labels):
  # Machine learning!
  return model
```

Memorize all
data and labels

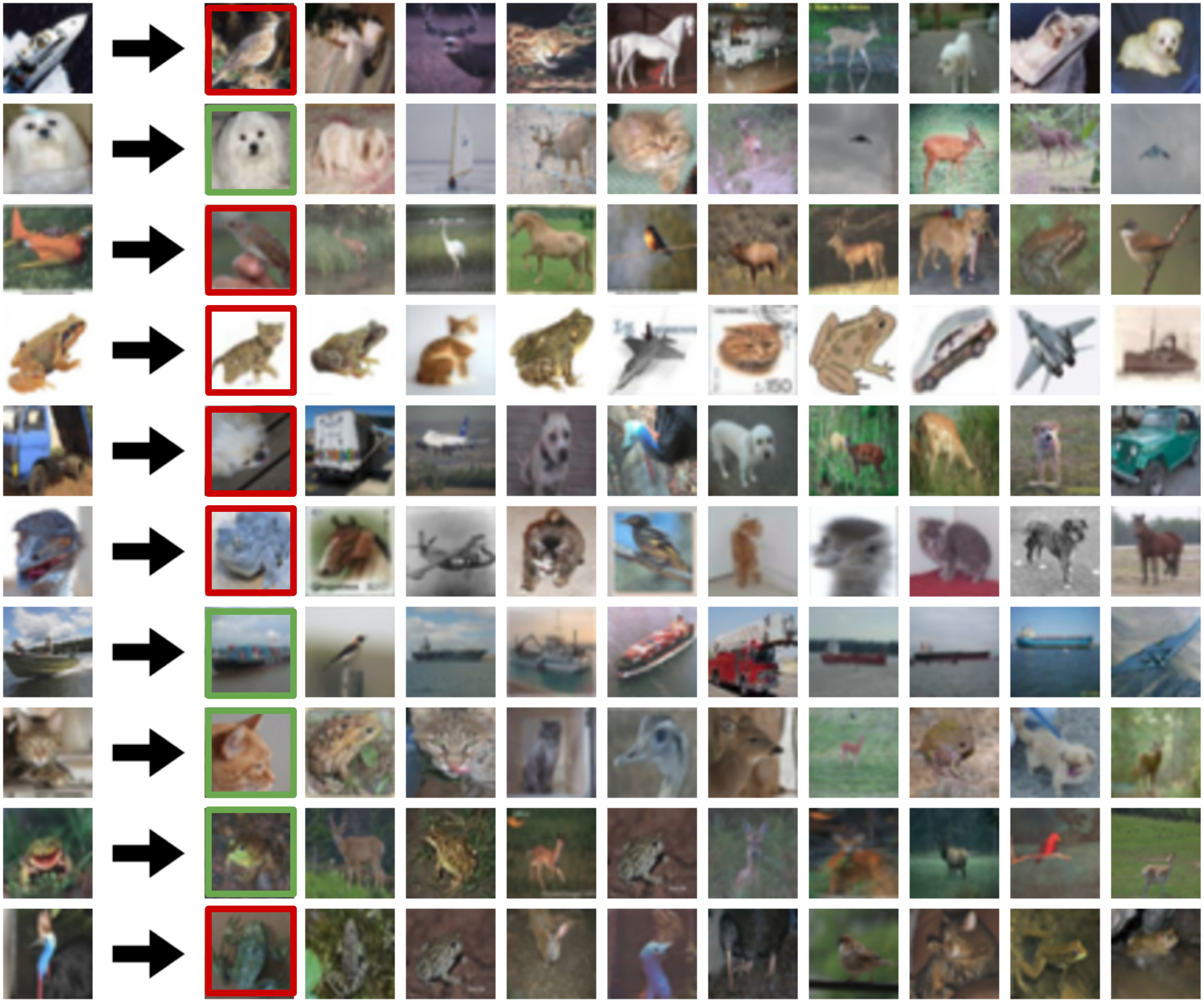


```python
def predict(model, test_images):
  # Use model to predict labels
  return test_labels
```

Predict the label
of the most similar
training image

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Nearest Neighbours



Instance/Memory-based Learning

Four things make a memory based learner:

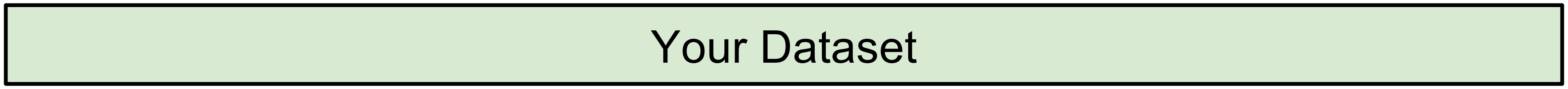
- *A distance metric*
- *How many nearby neighbors to look at?*
- *A weighting function (optional)*
- *How to fit with the local points?*

Slide Credit: Carlos Guestrin

Parametric vs Non-Parametric Models

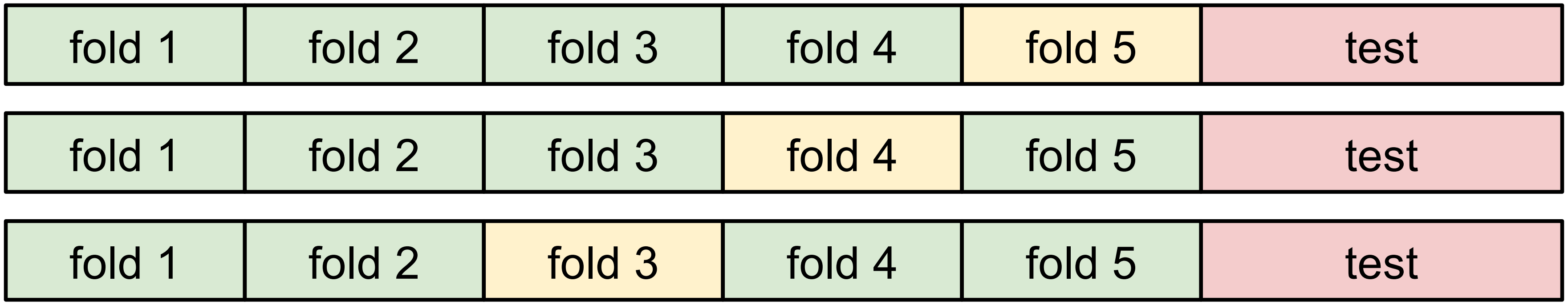
- Does the capacity (size of hypothesis class) grow with size of training data?
  – Yes = Non-Parametric Models
  – No = Parametric Models

Hyperparameters



Your Dataset

Idea #4: **Cross-Validation**: Split data into **folds**,
try each fold as validation and average the results



fold 1	fold 2	fold 3	fold 4	fold 5	test
fold 1	fold 2	fold 3	fold 4	fold 5	test
fold 1	fold 2	fold 3	fold 4	fold 5	test

Useful for small datasets, but not used too frequently in deep learning

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Problems with Instance-Based Learning

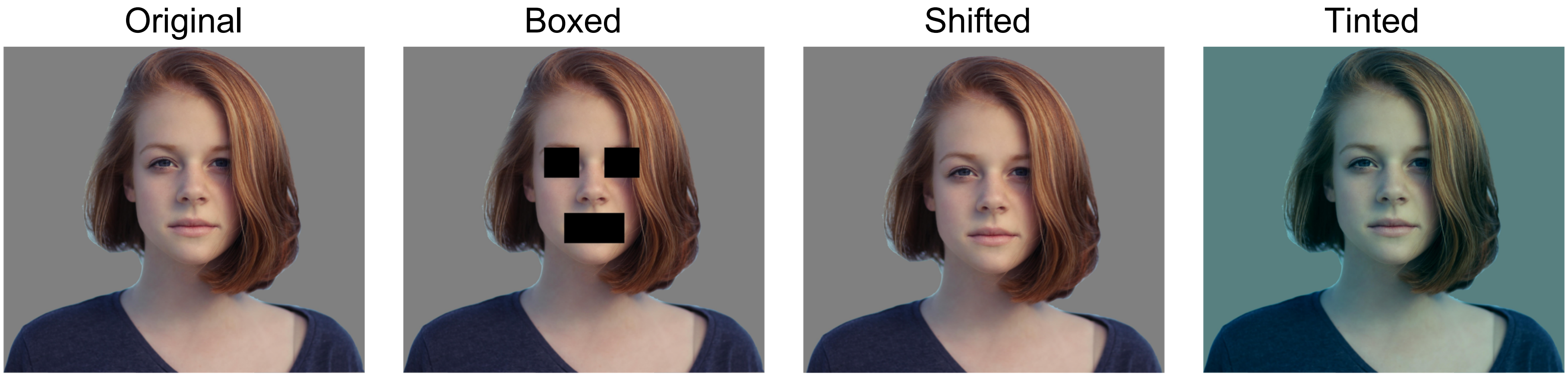
- Expensive
  - No Learning: most real work done during testing
  - For every test sample, must search through all dataset – very slow!
  - Must use tricks like approximate nearest neighbour search

- Doesn't work well when large number of irrelevant features
  - Distances overwhelmed by noisy features

- Curse of Dimensionality
  - Distances become meaningless in high dimensions
  - (See proof in next lecture)

k-Nearest Neighbor on images **never used.**

- Very slow at test time
- Distance metrics on pixels are not informative



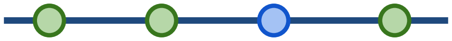
(all 3 images have same L2 distance to the one on the left)

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

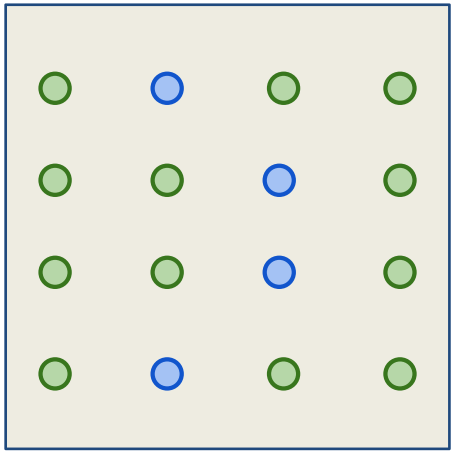
k-Nearest Neighbor on images **never used.**

- Curse of dimensionality

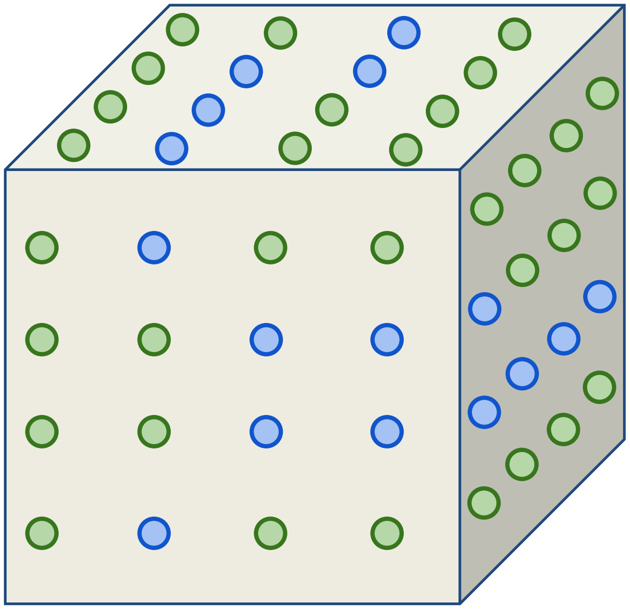
Dimensions = 1
Points = 4



Dimensions = 2
Points = 4^2



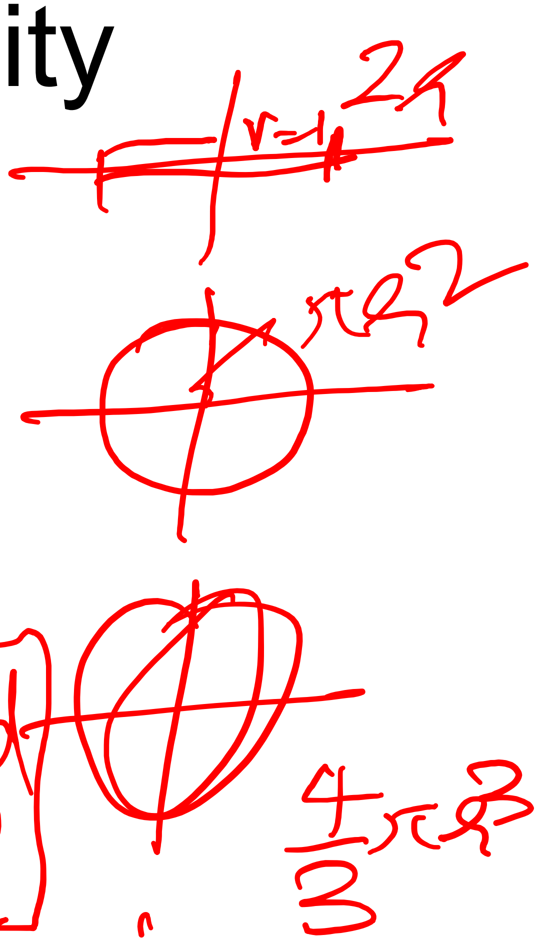
Dimensions = 3
Points = 4^3

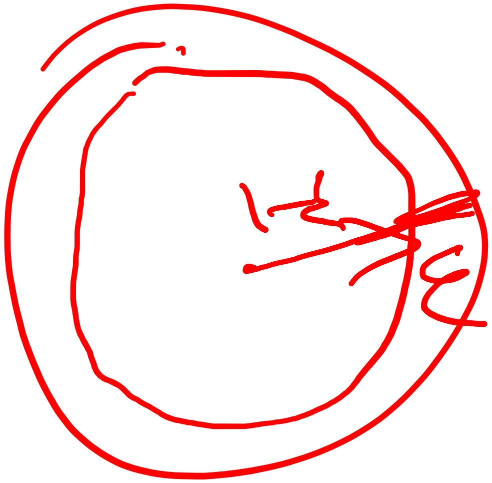


Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Curse of Dimensionality

- Consider: Sphere of radius 1 in d-dims
- Consider: an outer ε-shell in this sphere
- What is $\boxed{\frac{\textit{shell volume}}{\textit{sphere volume}}}$?

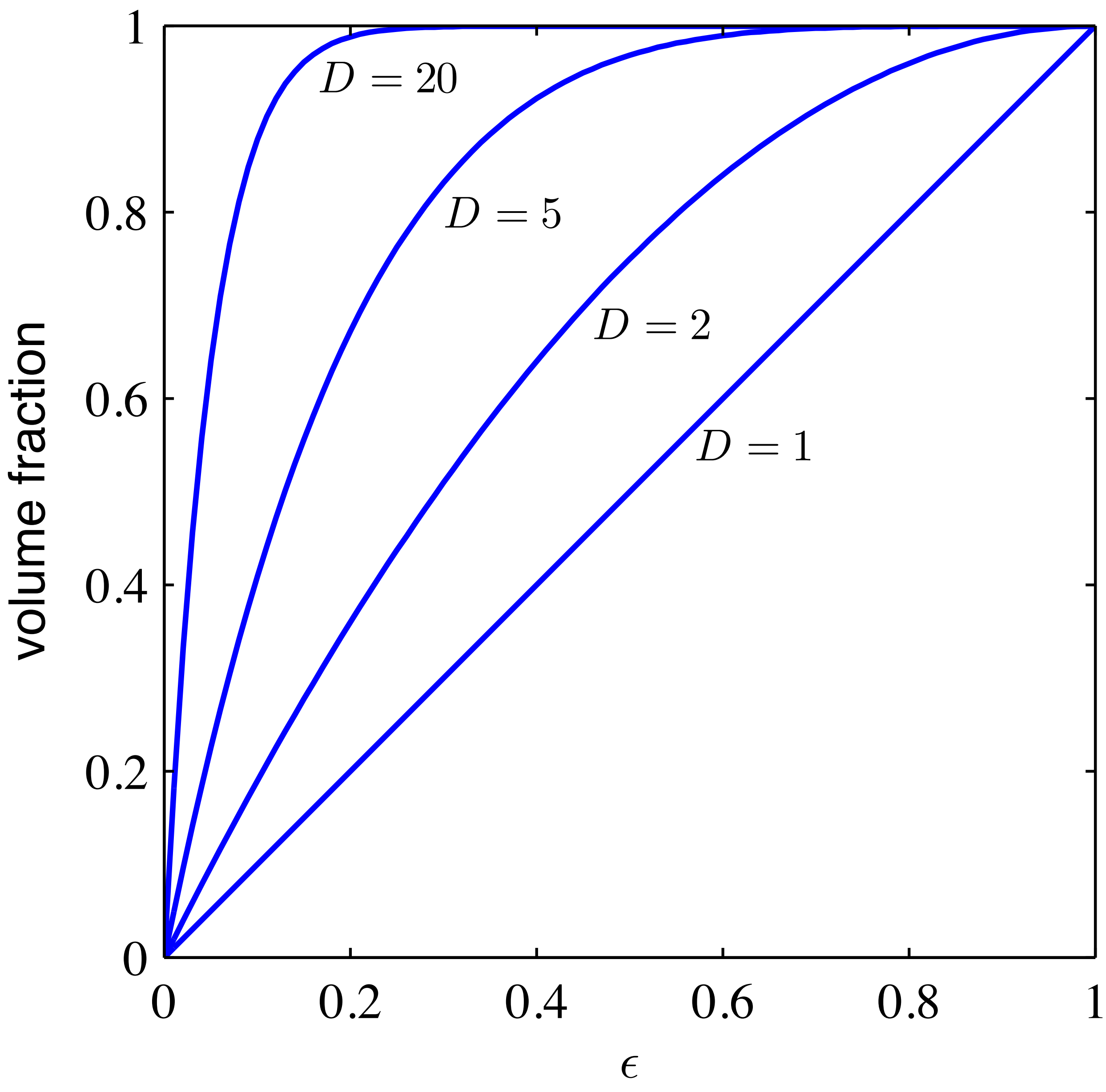




$$\frac{\boxed{r^d - (1-\varepsilon)^d}}{1^d}$$

$$1 - (1-\varepsilon)^d \quad ; \quad \propto r^d$$

Curse of Dimensionality



Plan for Today

- Linear Classifiers
  - Linear scoring functions
- Loss Functions
  - Multi-class hinge loss
  - Softmax cross-entropy loss

Linear Classification

Neural Network

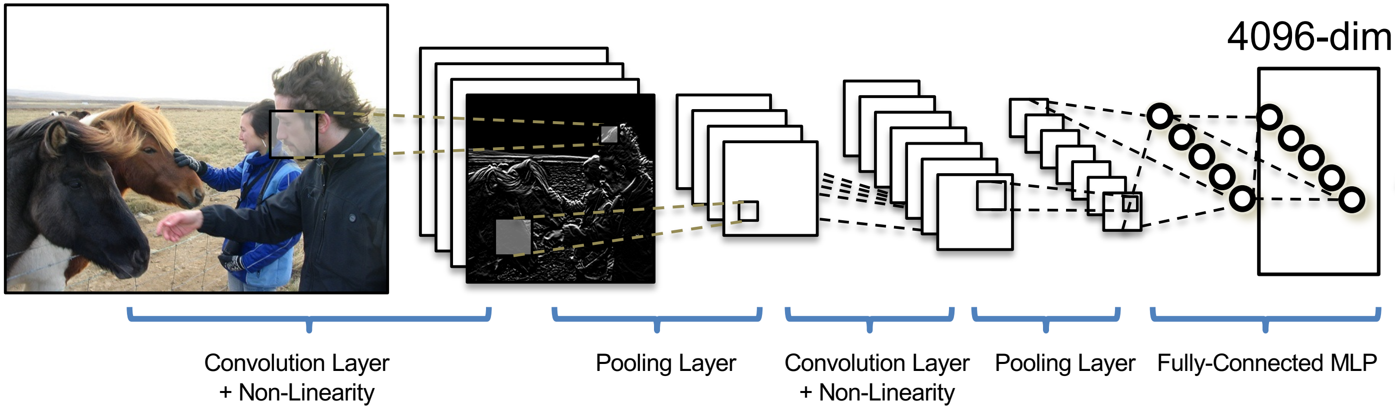
Linear classifiers

This image is CC0 1.0 public domain

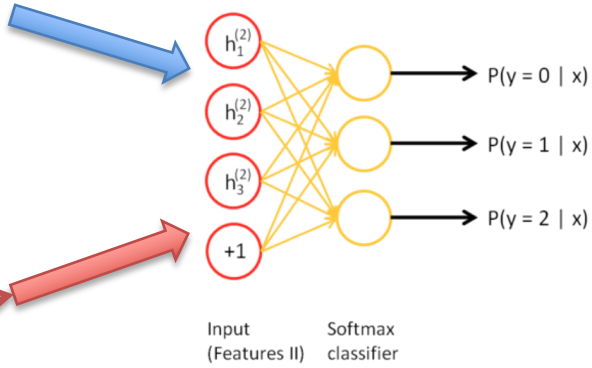
Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Visual Question Answering

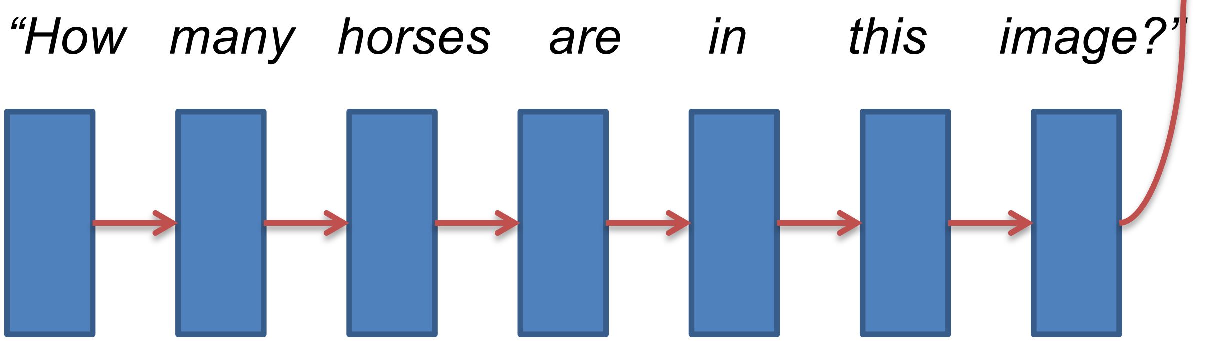
Image Embedding (VGGNet)



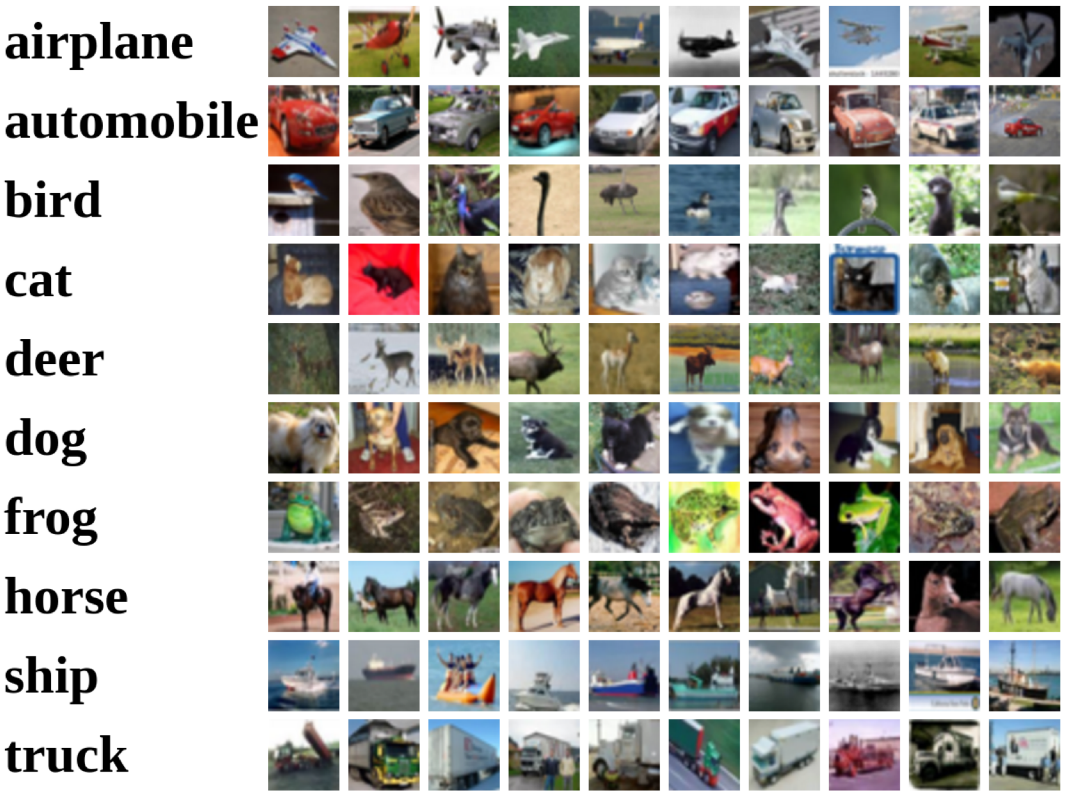
Neural Network
Softmax
over top K answers



Question Embedding (LSTM)



Recall CIFAR10

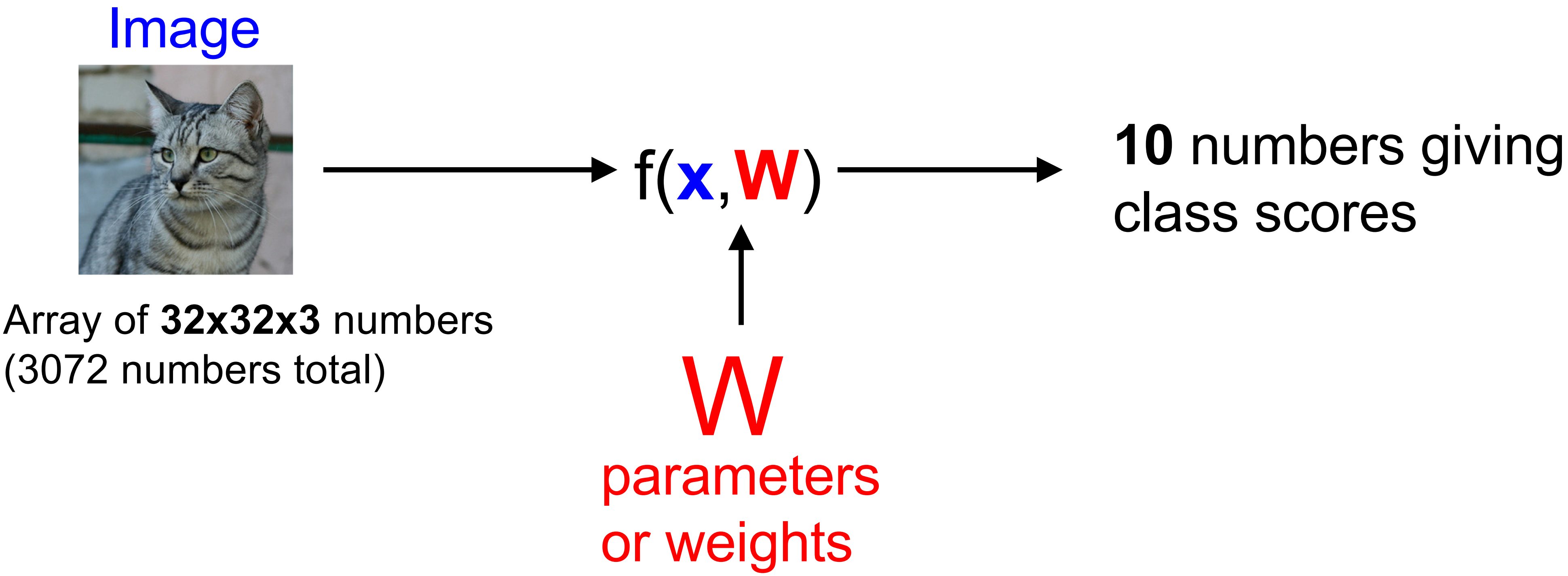


50,000 training images
each image is **32x32x3**

10,000 test images.

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Parametric Approach



Image

Array of **32x32x3** numbers
(3072 numbers total)

$f(\mathbf{x},\mathbf{W})$

10 numbers giving
class scores

W

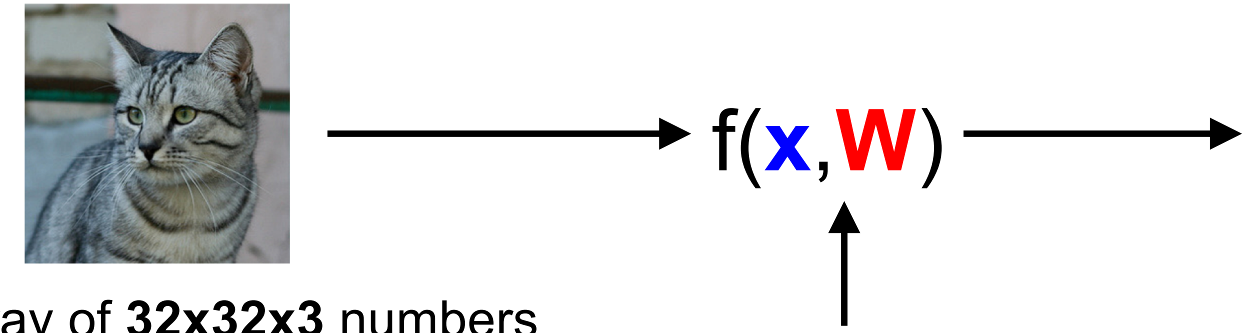
parameters
or weights

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Parametric Approach: Linear Classifier

$$f(x,W) = Wx$$

Image



Array of **32x32x3** numbers
(3072 numbers total)

$f(\mathbf{x},\mathbf{W})$

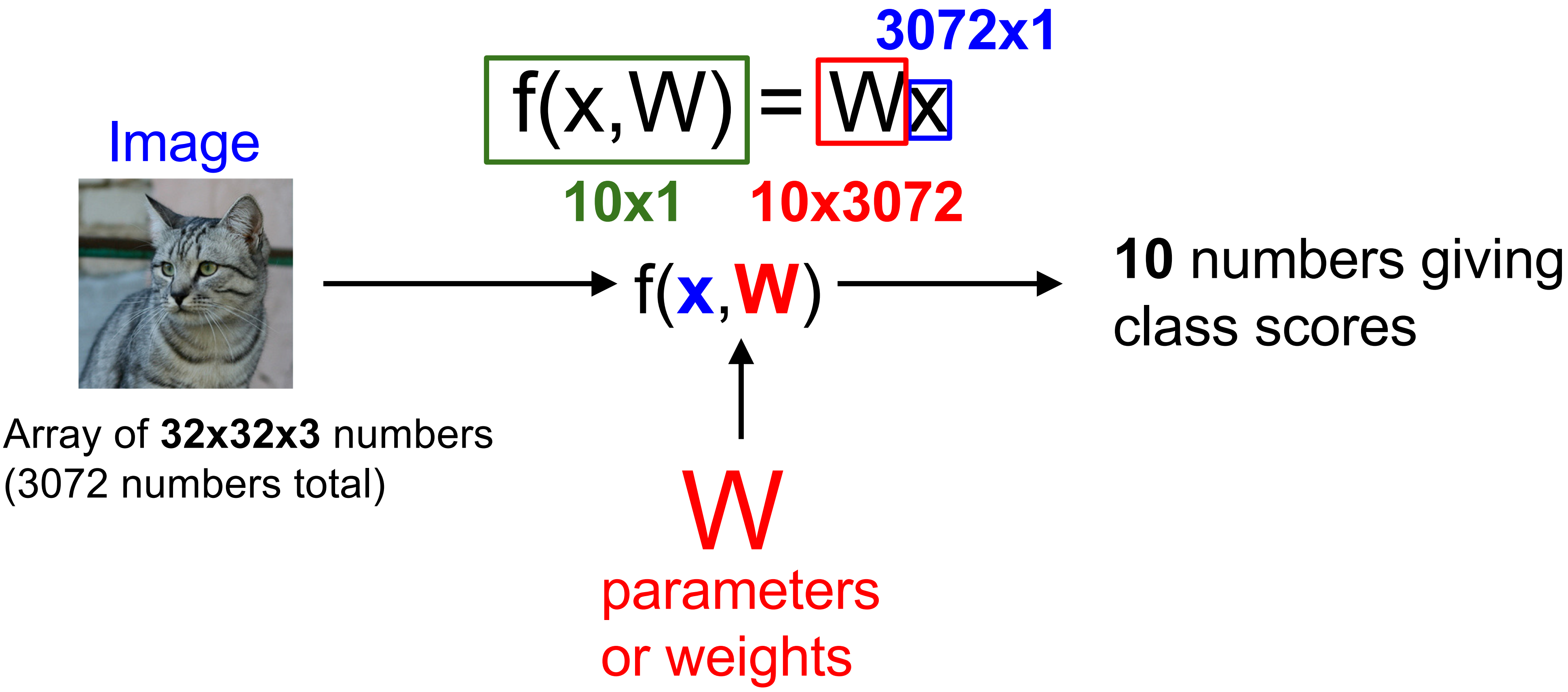
10 numbers giving
class scores

W

parameters
or weights

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Parametric Approach: Linear Classifier



3072x1

$$\boxed{f(x,W)} = \boxed{W}\boxed{x}$$

10x1 10x3072

Image

Array of **32x32x3** numbers
(3072 numbers total)

f(**x**,**W**)

10 numbers giving
class scores

W
parameters
or weights

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Parametric Approach: Linear Classifier

$$\underset{\textbf{10x1}}{\boxed{f(x,W)}} = \underset{\textbf{10x3072}}{\boxed{W}}\overset{\textbf{3072x1}}{\boxed{x}} + \boxed{b} \quad \textbf{10x1}$$

Image

Array of **32x32x3** numbers
(3072 numbers total)

$\longrightarrow$ f(**x**,**W**) $\longrightarrow$ **10** numbers giving class scores

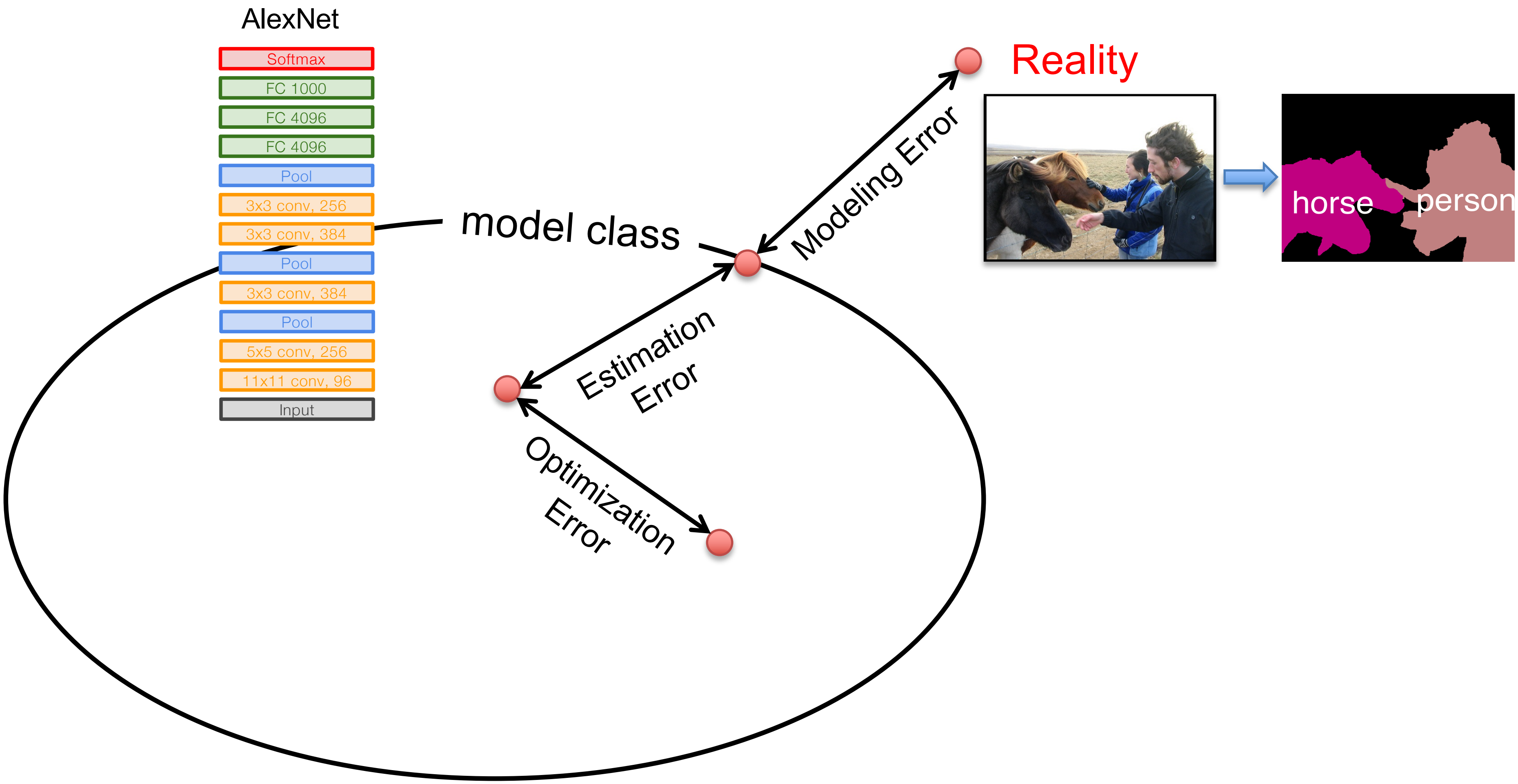
$\uparrow$

W

parameters
or weights

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Error Decomposition



AlexNet

Softmax

FC 1000

FC 4096

FC 4096

Pool

3x3 conv, 256

3x3 conv, 384

Pool

3x3 conv, 384

Pool

5x5 conv, 256

11x11 conv, 96

Input

model class

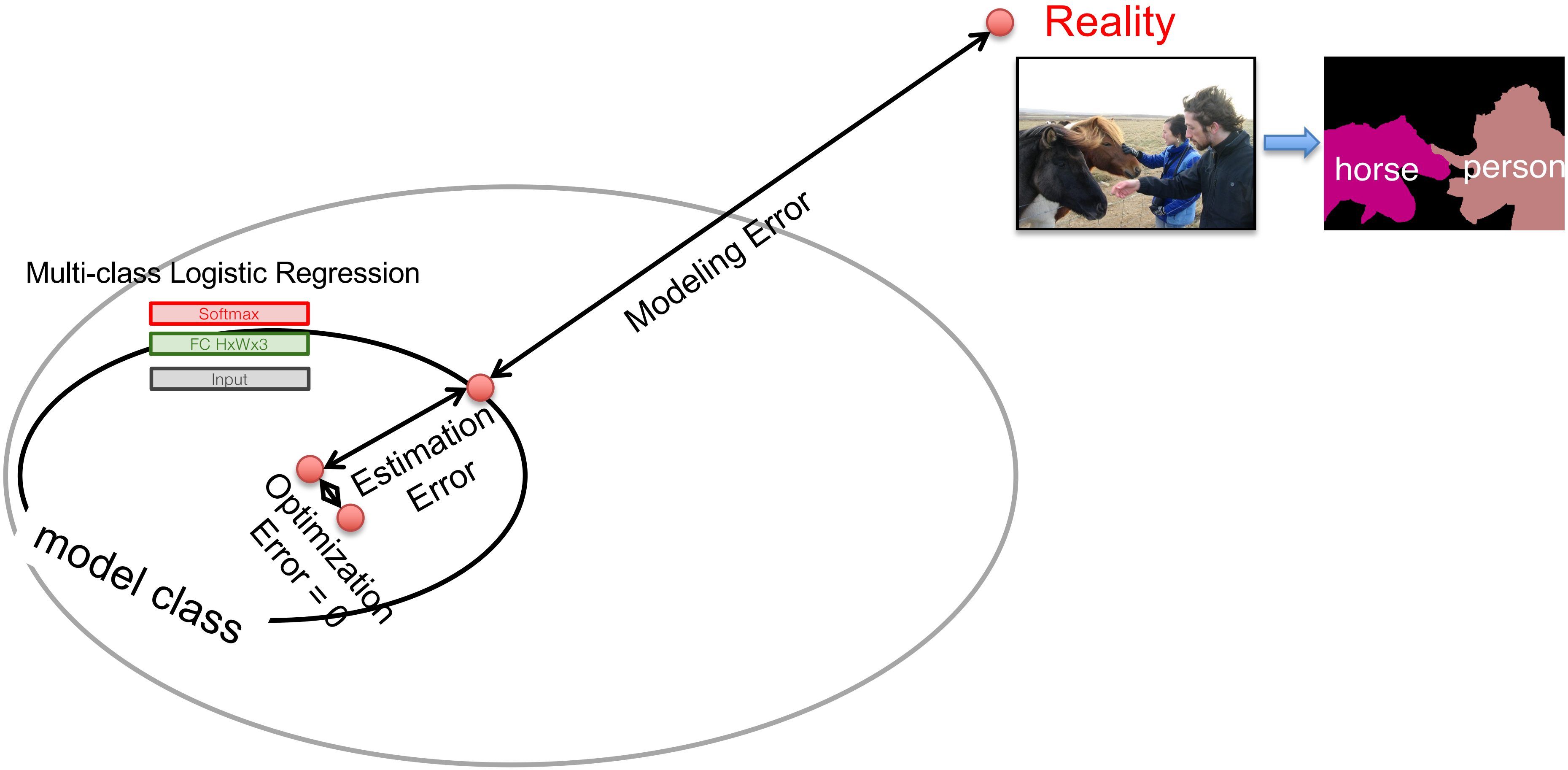
Reality

Modeling Error

Estimation Error

Optimization Error

Error Decomposition



Reality

horse

person

Modeling Error

Multi-class Logistic Regression

Softmax

FC HxWx3

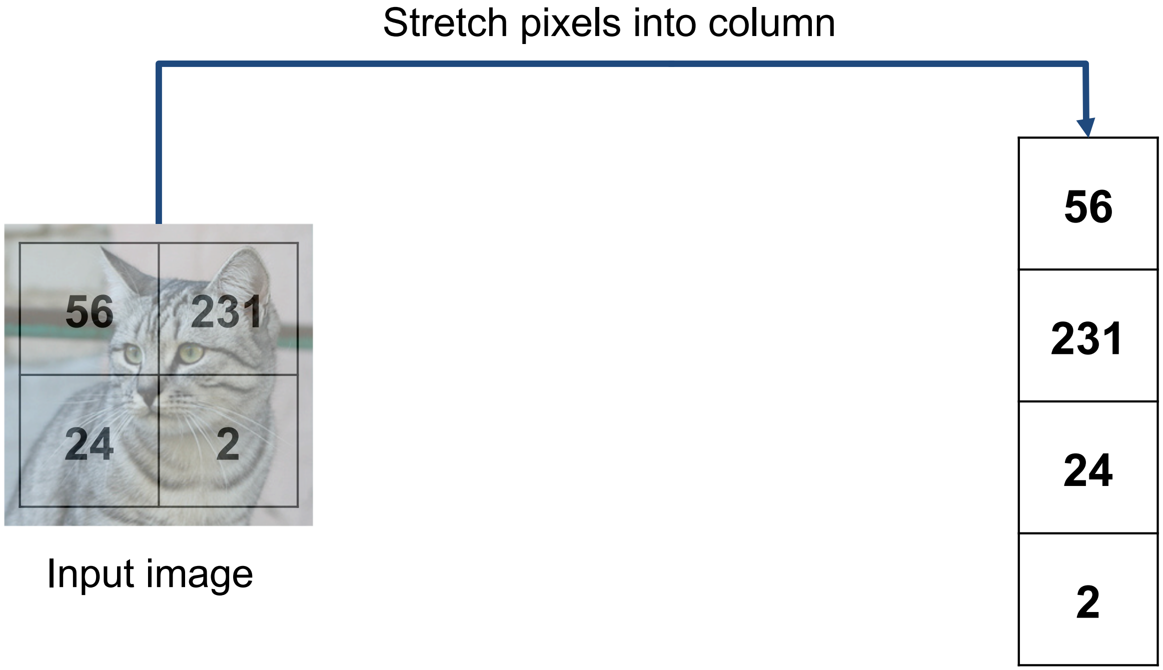
Input

Estimation Error

Optimization Error = 0

model class

Example with an image with 4 pixels, and 3 classes (cat/dog/ship)



Stretch pixels into column

56 | 231
24 | 2

Input image

56

231

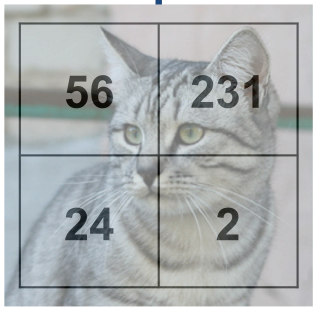
24

2

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Example with an image with 4 pixels, and 3 classes (cat/dog/ship)

Stretch pixels into column



Input image

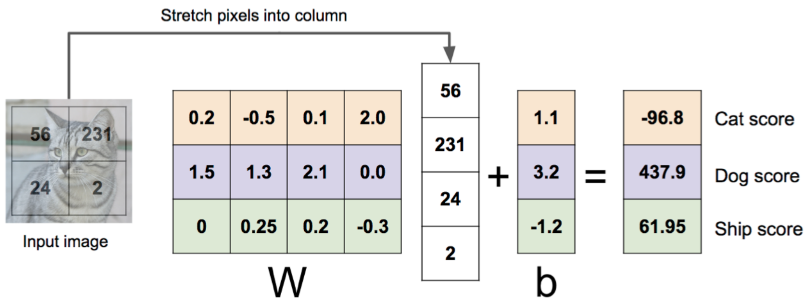
$$\underbrace{\begin{bmatrix} 0.2 & -0.5 & 0.1 & 2.0 \\ 1.5 & 1.3 & 2.1 & 0.0 \\ 0 & 0.25 & 0.2 & -0.3 \end{bmatrix}}_{W} \begin{bmatrix} 56 \\ 231 \\ 24 \\ 2 \end{bmatrix} + \underbrace{\begin{bmatrix} 1.1 \\ 3.2 \\ -1.2 \end{bmatrix}}_{b} = \begin{array}{cl} \begin{bmatrix} -96.8 \\ 437.9 \\ 61.95 \end{bmatrix} & \begin{array}{l} \text{Cat score} \\ \text{Dog score} \\ \text{Ship score} \end{array} \end{array}$$

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Example with an image with 4 pixels, and 3 classes (cat/dog/ship)

Algebraic Viewpoint

f(x,W) = Wx



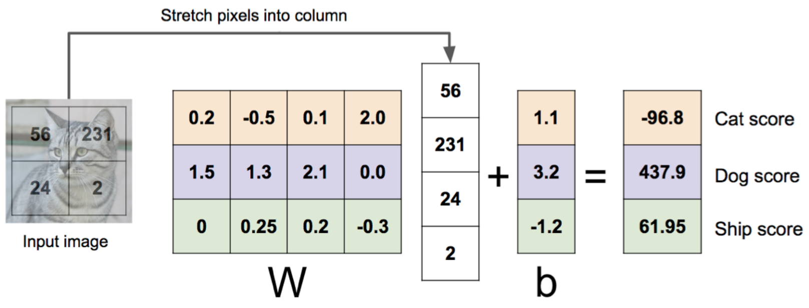
$$\begin{bmatrix} 0.2 & -0.5 & 0.1 & 2.0 \\ 1.5 & 1.3 & 2.1 & 0.0 \\ 0 & 0.25 & 0.2 & -0.3 \end{bmatrix} \begin{bmatrix} 56 \\ 231 \\ 24 \\ 2 \end{bmatrix} + \begin{bmatrix} 1.1 \\ 3.2 \\ -1.2 \end{bmatrix} = \begin{bmatrix} -96.8 \\ 437.9 \\ 61.95 \end{bmatrix}$$

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

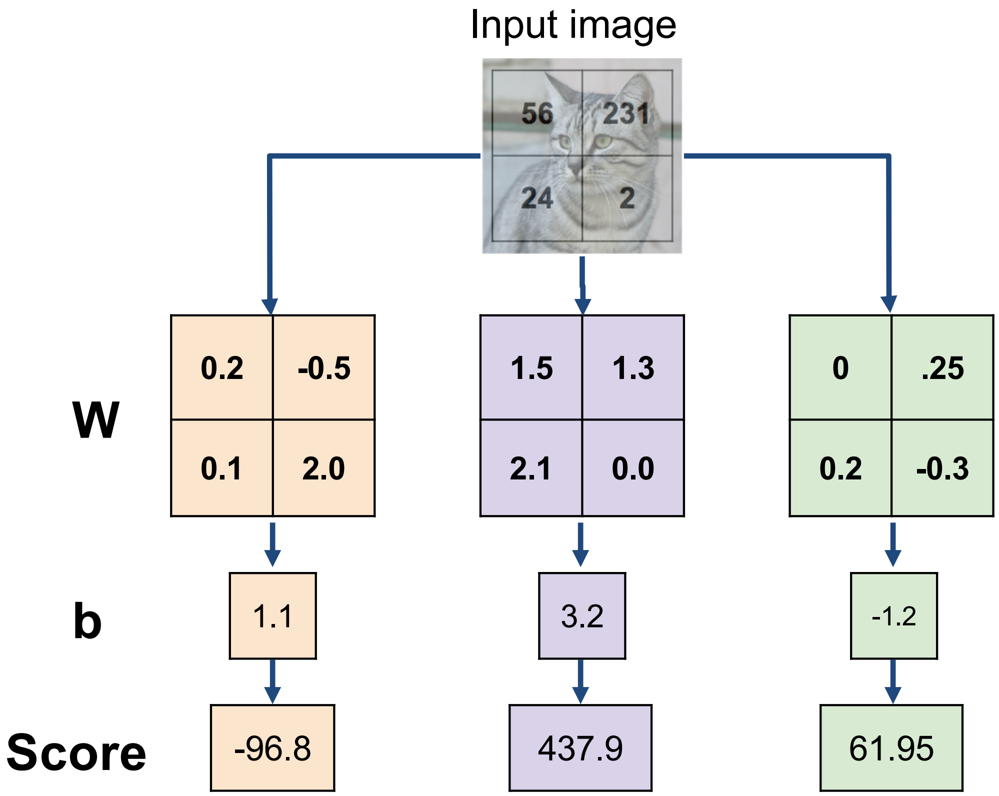
Example with an image with 4 pixels, and 3 classes (cat/dog/ship)

Algebraic Viewpoint

f(x,W) = Wx

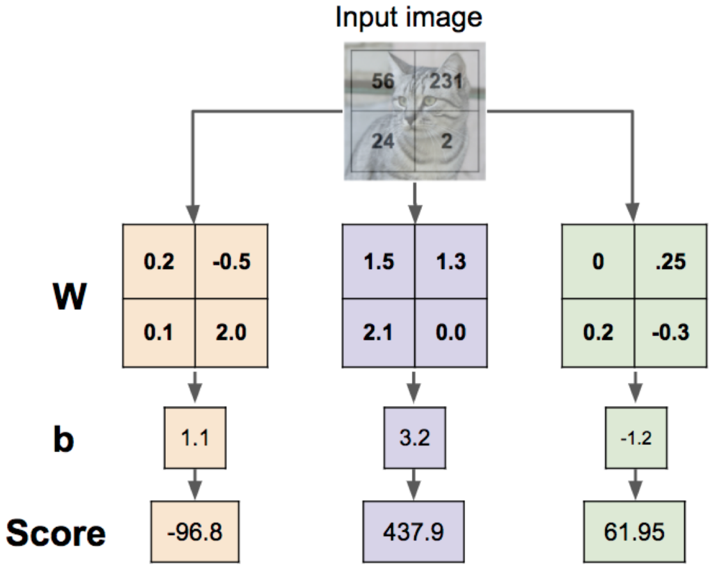


$$\underbrace{\begin{bmatrix} 0.2 & -0.5 & 0.1 & 2.0 \\ 1.5 & 1.3 & 2.1 & 0.0 \\ 0 & 0.25 & 0.2 & -0.3 \end{bmatrix}}_{W} \begin{bmatrix} 56 \\ 231 \\ 24 \\ 2 \end{bmatrix} + \underbrace{\begin{bmatrix} 1.1 \\ 3.2 \\ -1.2 \end{bmatrix}}_{b} = \begin{bmatrix} -96.8 \\ 437.9 \\ 61.95 \end{bmatrix}$$



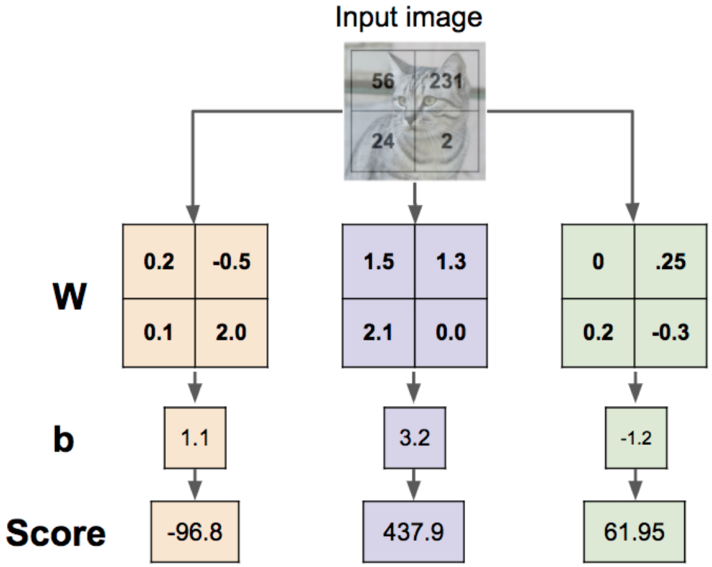
Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Interpreting a Linear Classifier



Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Interpreting a Linear Classifier: Visual Viewpoint



Input image

56 231
24 2

W

0.2 -0.5
0.1 2.0

1.5 1.3
2.1 0.0

0 .25
0.2 -0.3

b

1.1

3.2

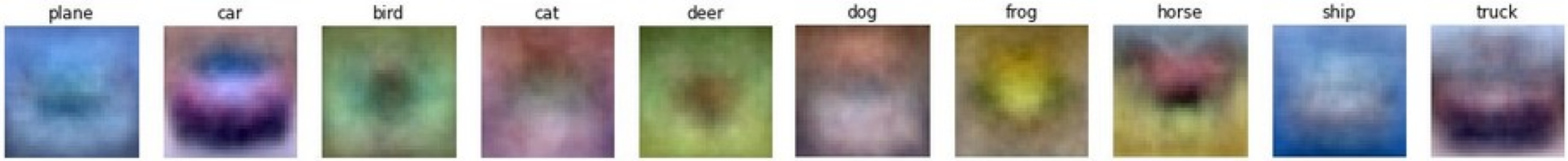
-1.2

Score

-96.8

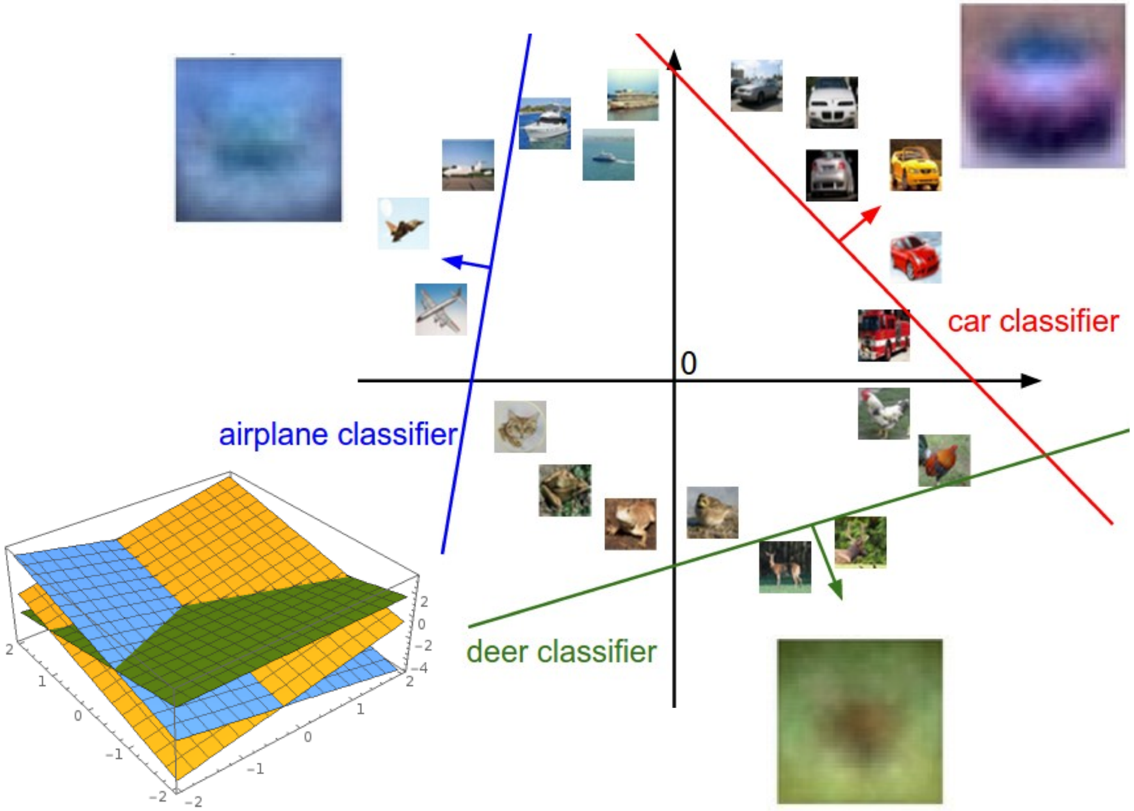
437.9

61.95



Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Interpreting a Linear Classifier: Geometric Viewpoint



$$f(x,W) = Wx + b$$

Array of **32x32x3** numbers
(3072 numbers total)

Plot created using Wolfram Cloud

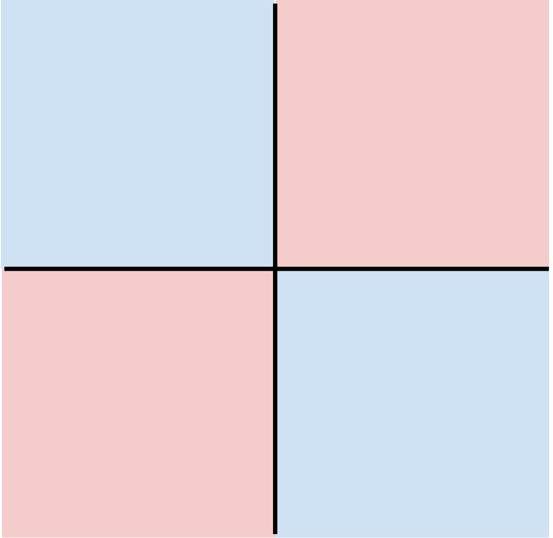
Cat image by Nikita is licensed under CC-BY 2.0

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Hard cases for a linear classifier

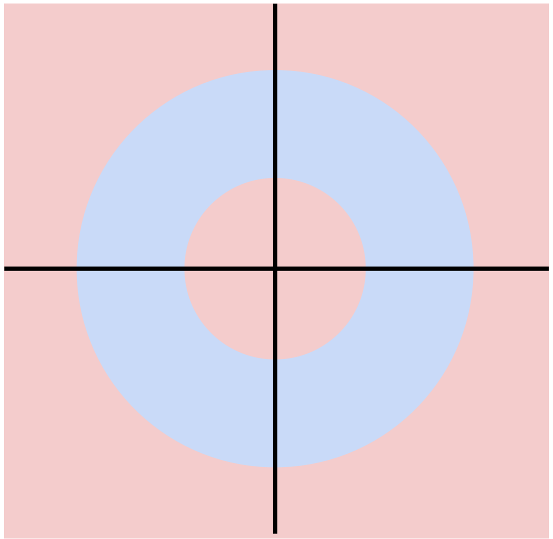
Class 1:
First and third quadrants

Class 2:
Second and fourth quadrants



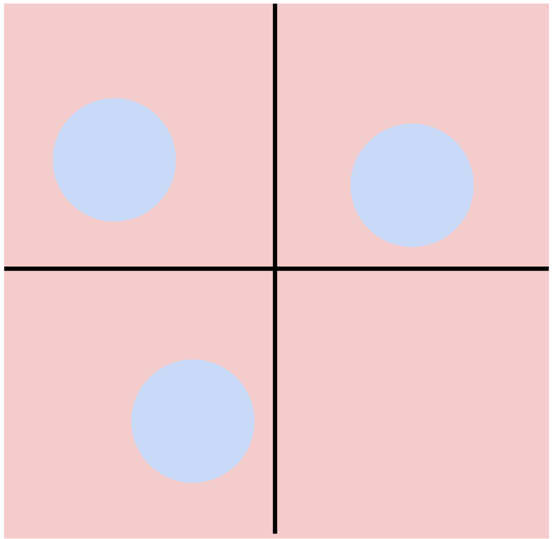
Class 1:
1 <= L2 norm <= 2

Class 2:
Everything else



Class 1:
Three modes

Class 2:
Everything else

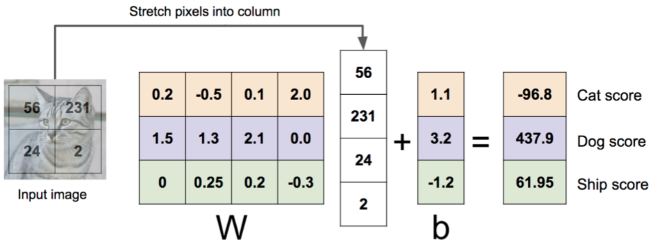


Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Linear Classifier: Three Viewpoints

Algebraic Viewpoint

$f(x,W) = Wx$

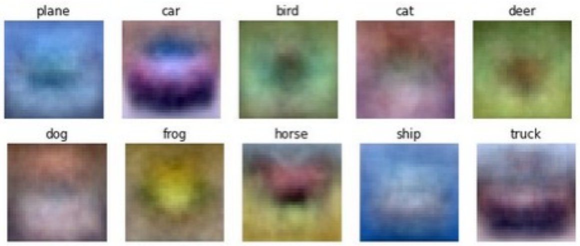


Stretch pixels into column

$$\underbrace{\begin{bmatrix} 0.2 & -0.5 & 0.1 & 2.0 \\ 1.5 & 1.3 & 2.1 & 0.0 \\ 0 & 0.25 & 0.2 & -0.3 \end{bmatrix}}_{W} \begin{bmatrix} 56 \\ 231 \\ 24 \\ 2 \end{bmatrix} + \underbrace{\begin{bmatrix} 1.1 \\ 3.2 \\ -1.2 \end{bmatrix}}_{b} = \begin{bmatrix} -96.8 \\ 437.9 \\ 61.95 \end{bmatrix} \begin{matrix} \text{Cat score} \\ \text{Dog score} \\ \text{Ship score} \end{matrix}$$

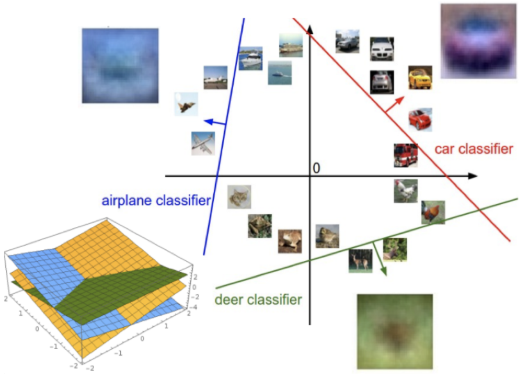
Visual Viewpoint

One template
per class



Geometric Viewpoint

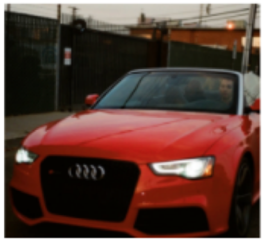
Hyperplanes
cutting up space



Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

So far: Defined a (linear) <u>score function</u>

f(x,W) = Wx + b

			
airplane	-3.45	-0.51	3.42
automobile	-8.87	**6.04**	4.64
bird	0.09	5.31	2.65
cat	**2.9**	-4.22	5.1
deer	4.48	-4.19	2.64
dog	8.02	3.58	5.55
frog	3.78	4.49	**-4.34**
horse	1.06	-4.37	-1.5
ship	-0.36	-2.09	-4.79
truck	-0.72	-2.93	6.14

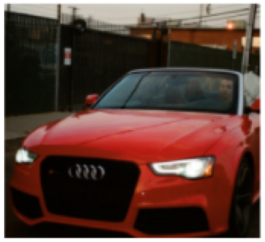
Cat image by Nikita is licensed under CC-BY 2.0; Car image is CC0 1.0 public domain; Frog image is in the public domain

Example class scores for 3 images for some W:

How can we tell whether this W is good or bad?

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

So far: Defined a (linear) score function

			
airplane	-3.45	-0.51	3.42
automobile	-8.87	**6.04**	4.64
bird	0.09	5.31	2.65
cat	**2.9**	-4.22	5.1
deer	4.48	-4.19	2.64
dog	8.02	3.58	5.55
frog	3.78	4.49	**-4.34**
horse	1.06	-4.37	-1.5
ship	-0.36	-2.09	-4.79
truck	-0.72	-2.93	6.14

Cat image by Nikita is licensed under CC-BY 2.0; Car image is CC0 1.0 public domain; Frog image is in the public domain

TODO:

1. Define a **loss function** that quantifies our unhappiness with the scores across the training data.

2. Come up with a way of efficiently finding the parameters that minimize the loss function. **(optimization)**

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Supervised Learning

- Input: x (images, text, emails…)
- Output: y (spam or non-spam…)

- (Unknown) Target Function
  - f: X → Y (the “true” mapping / reality)

- Data
  - (x_1,y_1), (x_2,y_2), …, (x_N,y_N)

- Model / Hypothesis Class
  - {h: X → Y}
  - e.g. $y = h(x) = \text{sign}(w^T x)$

- Loss Function
  - How good is a model wrt my data D?

- Learning = Search in hypothesis space
  - Find best h in model class.

Loss Functions

Suppose: 3 training examples, 3 classes.
With some W the scores $f(x,W) = Wx$ are:

cat	**3.2**	1.3	2.2
car	5.1	**4.9**	2.5
frog	-1.7	2.0	**-3.1**

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Suppose: 3 training examples, 3 classes.
With some W the scores $f(x,W) = Wx$ are:

cat	**3.2**	1.3	2.2
car	5.1	**4.9**	2.5
frog	-1.7	2.0	**-3.1**

A **loss function** tells how good our current classifier is

Given a dataset of examples

$$\{(x_i, y_i)\}_{i=1}^{N}$$

Where x_i is image and
y_i is (integer) label

Loss over the dataset is a sum of loss over examples:

$$L = \frac{1}{N} \sum_i L_i(f(x_i, W), y_i)$$

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Suppose: 3 training examples, 3 classes.
With some W the scores $f(x,W) = Wx$ are:

cat	**3.2**	1.3	2.2
car	5.1	**4.9**	2.5
frog	-1.7	2.0	**-3.1**

Multiclass SVM loss:

Given an example (x_i, y_i) where x_i is the image and where y_i is the (integer) label,

and using the shorthand for the scores vector: $s = f(x_i, W)$

the SVM loss has the form:

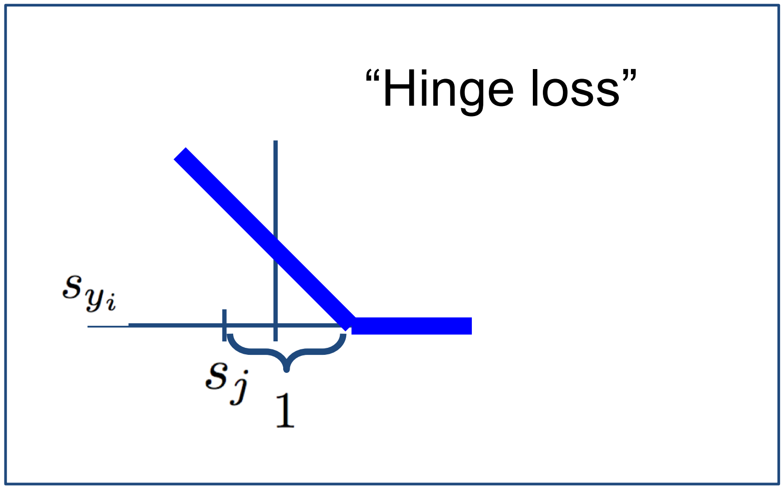
$$L_i = \sum_{\boxed{j\neq y_i}} \begin{cases} 0 & \text{if } s_{y_i} \geq \underline{s_j + 1} \\ \underline{s_j - s_{y_i} + 1} & \text{otherwise} \end{cases}$$
$$= \sum_{\underline{j\neq y_i}} \underline{\max}(0, \underline{s_j - s_{y_i} + 1})$$

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Suppose: 3 training examples, 3 classes.
With some W the scores $f(x,W) = Wx$ are:

cat	**3.2**	1.3	2.2
car	5.1	**4.9**	2.5
frog	-1.7	2.0	**-3.1**

Multiclass SVM loss:



$$L_i = \sum_{j \neq y_i} \begin{cases} 0 & \text{if } s_{y_i} \geq s_j + 1 \\ s_j - s_{y_i} + 1 & \text{otherwise} \end{cases}$$

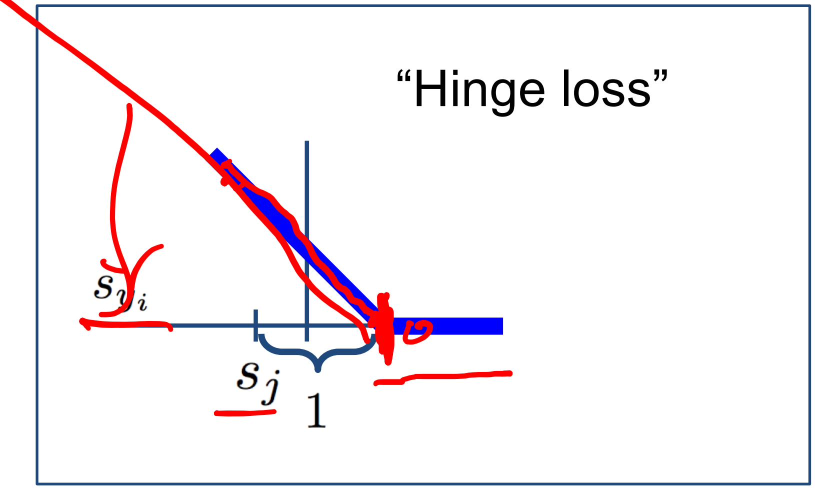
$$= \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Suppose: 3 training examples, 3 classes.
With some W the scores $f(x,W) = Wx$ are:

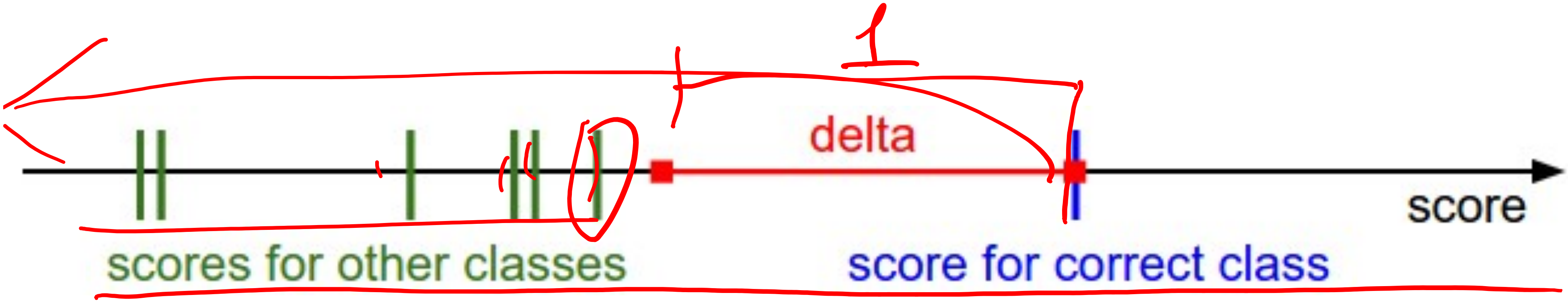
cat	**3.2**	1.3	2.2
car	5.1	**4.9**	2.5
frog	-1.7	2.0	**-3.1**

Multiclass SVM loss:



$$L_i = \sum_{j \neq y_i} \begin{cases} 0 & \text{if } s_{y_i} \geq s_j + 1 \\ s_j - s_{y_i} + 1 & \text{otherwise} \end{cases}$$

$$= \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$



Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Suppose: 3 training examples, 3 classes.
With some W the scores $f(x, W) = Wx$ are:

cat	**3.2**	1.3	2.2
car	5.1	**4.9**	2.5
frog	-1.7	2.0	**-3.1**

Multiclass SVM loss:

Given an example (x_i, y_i)
where x_i is the image and
where y_i is the (integer) label,

and using the shorthand for the
scores vector: $s = f(x_i, W)$

the SVM loss has the form:

$$\boxed{\underline{L_i} = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)}$$

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Suppose: 3 training examples, 3 classes.
With some W the scores $f(x,W) = Wx$ are:

cat	**3.2**	1.3	2.2
car	5.1	**4.9**	2.5
frog	-1.7	2.0	**-3.1**
Losses:	2.9		

Multiclass SVM loss:

Given an example (x_i, y_i)
where x_i is the image and
where y_i is the (integer) label,

and using the shorthand for the scores vector: $s = f(x_i, W)$

the SVM loss has the form:

$$\boxed{L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)}$$

= max(0, 5.1 - 3.2 + 1)
+max(0, -1.7 - 3.2 + 1)
= max(0, 2.9) + max(0, -3.9)
= 2.9 + 0
= 2.9

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Suppose: 3 training examples, 3 classes.
With some W the scores $f(x, W) = Wx$ are:

cat	**3.2**	1.3	2.2
car	5.1	**4.9**	2.5
frog	-1.7	2.0	**-3.1**
Losses:	2.9	0	

Multiclass SVM loss:

Given an example (x_i, y_i)
where x_i is the image and
where y_i is the (integer) label,

and using the shorthand for the scores vector: $s = f(x_i, W)$

the SVM loss has the form:

$$\boxed{L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)}$$

$$\begin{aligned}
&= \max(0, 1.3 - 4.9 + 1) \\
&\quad + \max(0, 2.0 - 4.9 + 1) \\
&= \max(0, -\underline{2.6}) + \max(0, -\underline{1.9}) \\
&= 0 + 0 \\
&= \underline{0}
\end{aligned}$$

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Suppose: 3 training examples, 3 classes.
With some W the scores $f(x,W) = Wx$ are:

cat	**3.2**	1.3	2.2
car	5.1	**4.9**	2.5
frog	-1.7	2.0	**-3.1**
Losses:	2.9	0	12.9

Multiclass SVM loss:

Given an example (x_i, y_i)
where x_i is the image and
where y_i is the (integer) label,

and using the shorthand for the scores vector: $s = f(x_i, W)$

the SVM loss has the form:

$$\boxed{L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)}$$

= max(0, 2.2 - (-3.1) + 1)
  +max(0, 2.5 - (-3.1) + 1)
= max(0, 6.3) + max(0, 6.6)
= 6.3 + 6.6
= 12.9

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Suppose: 3 training examples, 3 classes.
With some W the scores $f(x,W) = Wx$ are:

cat	**3.2**	1.3	2.2
car	5.1	**4.9**	2.5
frog	-1.7	2.0	**-3.1**
Losses:	2.9	0	12.9

Multiclass SVM loss:

Given an example (x_i, y_i)
where x_i is the image and
where y_i is the (integer) label,

and using the shorthand for the
scores vector: $s = f(x_i, W)$

the SVM loss has the form:

$$\boxed{L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)}$$

Loss over full dataset is average:

$$L = \frac{1}{N} \sum_{i=1}^{N} L_i$$

$$L = (2.9 + 0 + 12.9)/3$$
$$= \boxed{\mathbf{5.27}}$$

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Suppose: 3 training examples, 3 classes.
With some W the scores $f(x, W) = Wx$ are:

cat	**3.2**	$1.3 - \varepsilon$	2.2
car	5.1	**4.9** $+\varepsilon$	2.5
frog	-1.7	$2.0 + \varepsilon$	**-3.1**
Losses:	2.9	0	12.9

Multiclass SVM loss:

Given an example (x_i, y_i) where x_i is the image and where y_i is the (integer) label,

and using the shorthand for the scores vector: $s = f(x_i, W)$

the SVM loss has the form:

$$\boxed{L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)}$$

Q: What happens to loss if car image scores change a bit?

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Suppose: 3 training examples, 3 classes.
With some W the scores $f(x, W) = Wx$ are:

cat	**3.2**	1.3	2.2
car	5.1	**4.9**	2.5
frog	-1.7	2.0	**-3.1**
Losses:	2.9	0	12.9

Multiclass SVM loss:

Given an example (x_i, y_i)
where x_i is the image and
where y_i is the (integer) label,

and using the shorthand for the
scores vector: $s = f(x_i, W)$

the SVM loss has the form:

$$\boxed{L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)}$$

Q2: what is the min/max possible loss?

$0, \infty$

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Suppose: 3 training examples, 3 classes.
With some W the scores $f(x, W) = Wx$ are:

cat	**3.2**	1.3	2.2
car	5.1	**4.9**	2.5
frog	-1.7	2.0	**-3.1**
Losses:	2.9	0	12.9

Multiclass SVM loss:

Given an example (x_i, y_i)
where x_i is the image and
where y_i is the (integer) label,

and using the shorthand for the
scores vector: $s = f(x_i, W)$

the SVM loss has the form:

$$\boxed{L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)}$$

Q3: At initialization W is small so all s ≈ 0. What is the loss?

(# classes - 1)

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Suppose: 3 training examples, 3 classes.
With some W the scores $f(x,W) = Wx$ are:

cat	**3.2**	1.3	2.2
car	5.1	**4.9**	2.5
frog	-1.7	2.0	**-3.1**
Losses:	2.9	0	12.9

Multiclass SVM loss:

Given an example (x_i, y_i)
where x_i is the image and
where y_i is the (integer) label,

and using the shorthand for the
scores vector: $s = f(x_i, W)$

the SVM loss has the form:

$$\boxed{L_i = \boxed{\sum_{j \neq y_i}} \max(0, \underline{s_j} - \underline{s_{y_i}} + \underline{1})}$$

Q4: What if the sum was over all classes? (including j = y_i)

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Suppose: 3 training examples, 3 classes.
With some W the scores $f(x,W) = Wx$ are:

cat	**3.2**	1.3	2.2
car	5.1	**4.9**	2.5
frog	-1.7	2.0	**-3.1**
Losses:	2.9	0	12.9

Multiclass SVM loss:

Given an example (x_i, y_i) where x_i is the image and where y_i is the (integer) label,

and using the shorthand for the scores vector: $s = f(x_i, W)$

the SVM loss has the form:

$\frac{1}{K}$

$$\boxed{L_i = \boxed{\sum_{j \neq y_i}} \max(0, s_j - s_{y_i} + 1)}$$

Q5: What if we used mean instead of sum?

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Suppose: 3 training examples, 3 classes.
With some W the scores $f(x,W) = Wx$ are:

cat	**3.2**	1.3	2.2
car	5.1	**4.9**	2.5
frog	-1.7	2.0	**-3.1**
Losses:	2.9	0	12.9

Multiclass SVM loss:

Given an example (x_i, y_i) where x_i is the image and where y_i is the (integer) label,

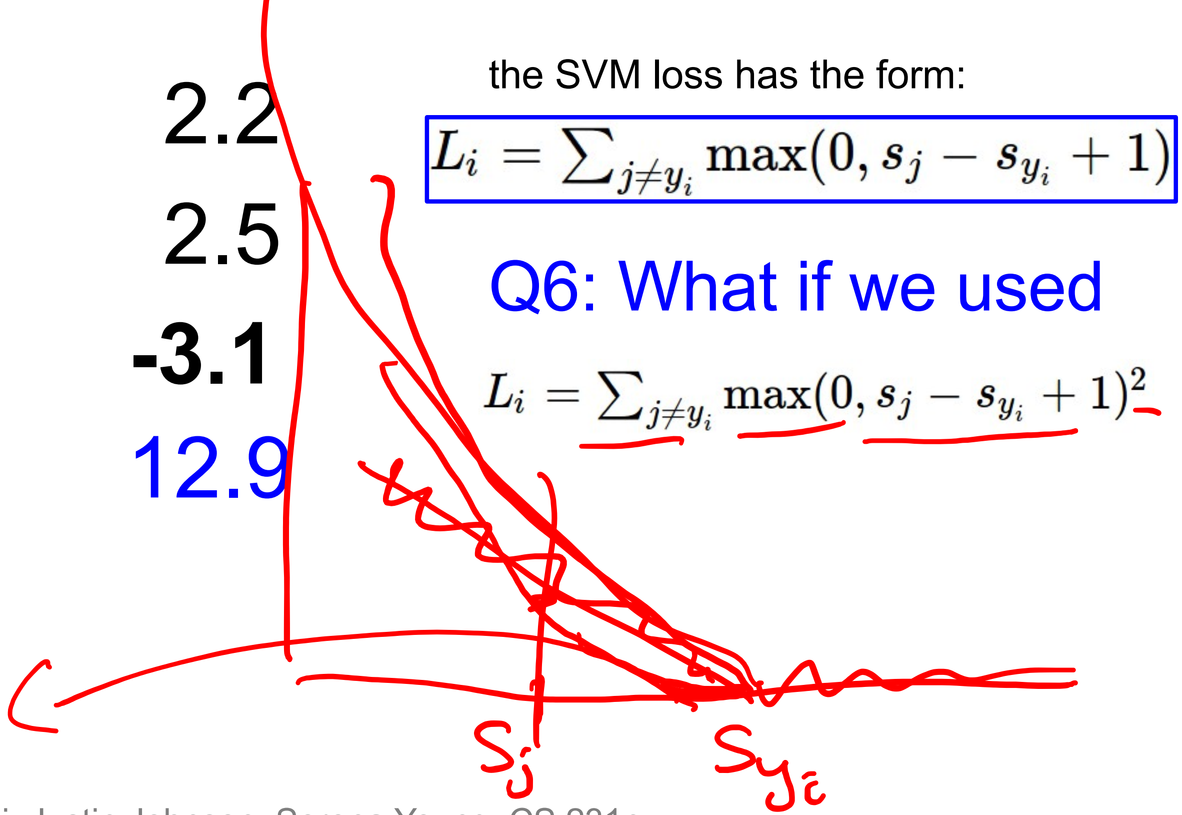
and using the shorthand for the scores vector: $s = f(x_i, W)$

the SVM loss has the form:

$$\boxed{L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)}$$

Q6: What if we used

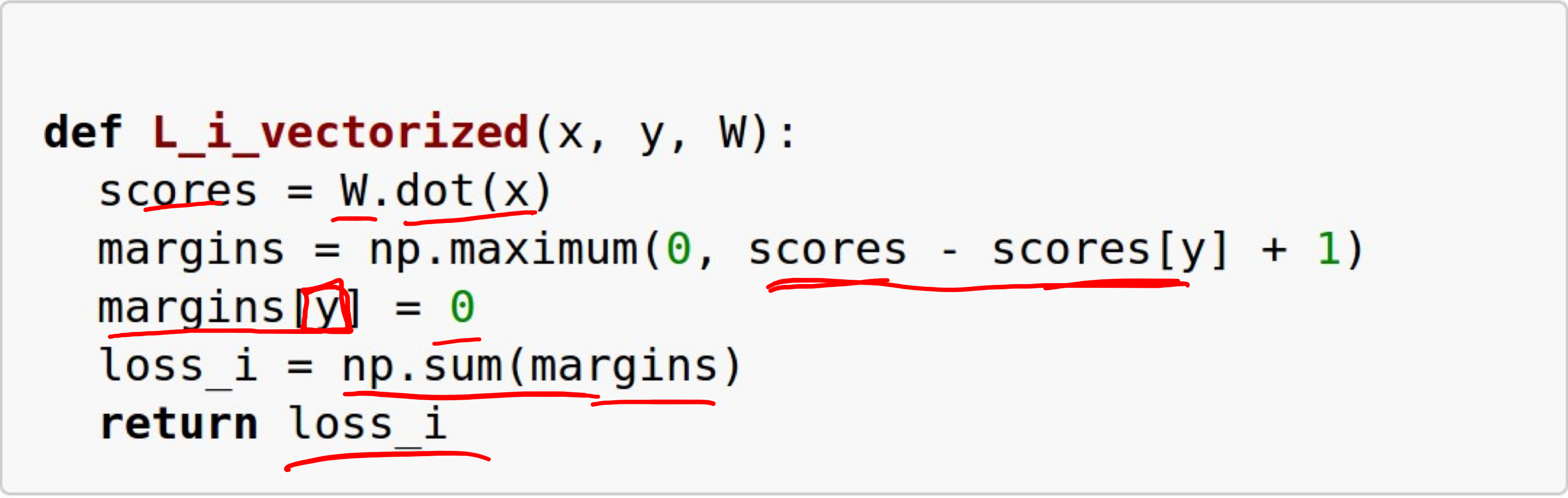
$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)^2$$



Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Multiclass SVM Loss: Example code

$$L_i = \sum_{j\neq y_i} \max(0, s_j - s_{y_i} + 1)$$



```python
def L_i_vectorized(x, y, W):
  scores = W.dot(x)
  margins = np.maximum(0, scores - scores[y] + 1)
  margins[y] = 0
  loss_i = np.sum(margins)
  return loss_i
```

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

$$\underline{f(x,W) = Wx}$$

$$\underline{L} = \boxed{\frac{1}{N}\sum_{i=1}^{N}}\boxed{\sum_{j\neq y_i}}\underline{\max}(0, \underline{f(x_i;W)_j - f(x_i;W)_{y_i}} + \underline{1})$$

E.g. Suppose that we found a W such that L = 0.
Is this W unique?

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

$$f(x, W) = Wx$$

$$L = \frac{1}{N} \sum_{i=1}^{N} \sum_{j \neq y_i} \max(0, f(x_i; W)_j - f(x_i; W)_{y_i} + 1)$$

E.g. Suppose that we found a W such that L = 0.
Is this W unique?

No! 2W is also has L = 0!

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Suppose: 3 training examples, 3 classes.
With some W the scores $f(x,W) = Wx$ are:

cat	**3.2**	1.3	2.2
car	5.1	**4.9**	2.5
frog	-1.7	2.0	**-3.1**
Losses:	2.9	0	

$$\boxed{L_i = \sum_{j\neq y_i} \max(0, s_j - s_{y_i} + 1)}$$

Before:

= max(0, 1.3 - 4.9 + 1)
+max(0, 2.0 - 4.9 + 1)
= max(0, -2.6) + max(0, -1.9)
= 0 + 0
= 0

With W twice as large:

= max(0, 2.6 - 9.8 + 1)
+max(0, 4.0 - 9.8 + 1)
= max(0, -6.2) + max(0, -4.8)
= 0 + 0
= 0

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Softmax Classifier (Multinomial Logistic Regression)

Want to interpret raw classifier scores as **probabilities**

cat	**3.2**
car	5.1
frog	-1.7

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Softmax Classifier (Multinomial Logistic Regression)

Want to interpret raw classifier scores as **probabilities**

$$\boxed{s = f(x_i; W)}$$

$$P(\underline{Y = k}|\underline{X = x_i}) = \boxed{\frac{e^{s_k}}{\underline{\sum_j e^{s_j}}}}$$

Softmax Function

cat	**3.2**
car	5.1
frog	-1.7

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Softmax Classifier (Multinomial Logistic Regression)

Want to interpret raw classifier scores as **probabilities**

$$\boxed{s = f(x_i; W)}$$

$$\boxed{P(Y = k|X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}}$$

Softmax Function

Probabilities must be >= 0

		exp	
cat	**3.2**	→	**24.5**
car	5.1		164.0
frog	-1.7		0.18

unnormalized probabilities

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Softmax Classifier (Multinomial Logistic Regression)

Want to interpret raw classifier scores as **probabilities**

$$\boxed{s = f(x_i; W)}$$

$$\boxed{P(Y = k | X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}}$$

Softmax Function

		$\xrightarrow{\text{exp}}$	Probabilities must be >= 0	$\xrightarrow{\text{normalize}}$	Probabilities must sum to 1
cat	**3.2**		**24.5**		**0.13**
car	5.1		164.0		0.87
frog	-1.7		0.18		0.00
			unnormalized probabilities		probabilities

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Softmax Classifier (Multinomial Logistic Regression)

Want to interpret raw classifier scores as **probabilities**

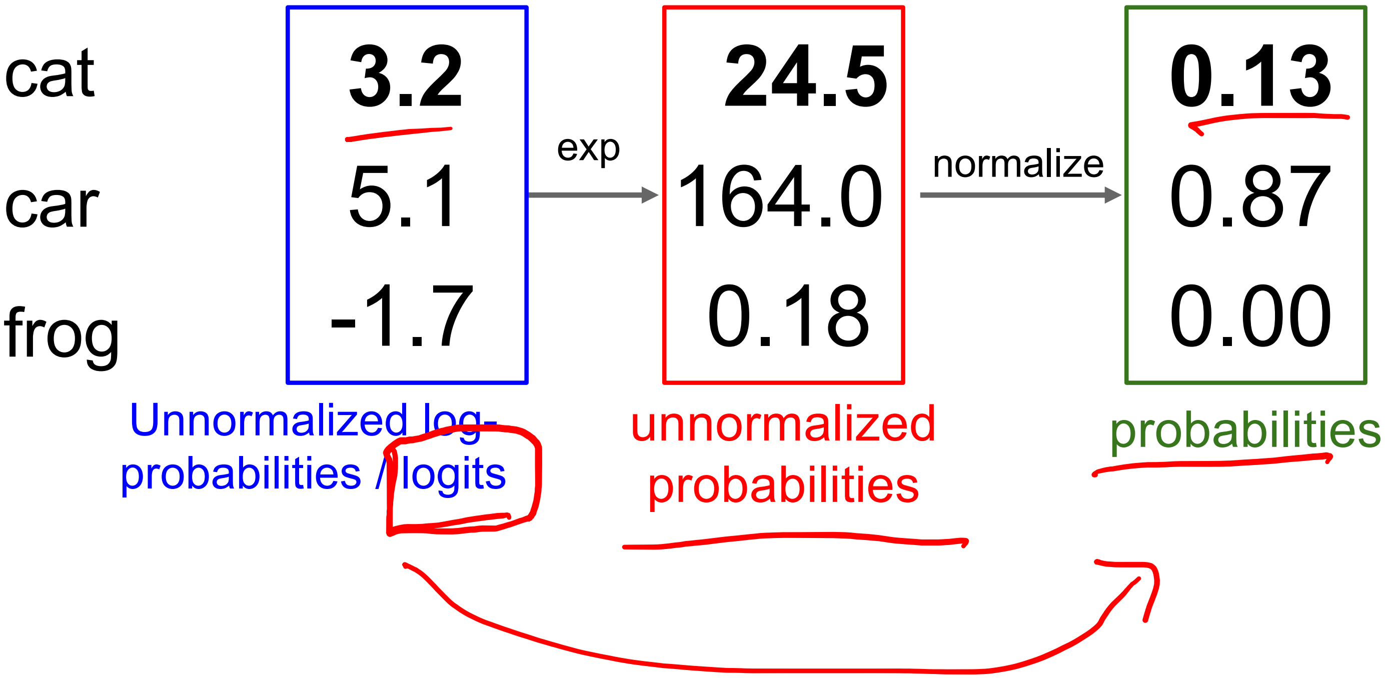
$$\boxed{s = f(x_i; W)}$$

$$\boxed{P(Y = k|X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}}$$

Softmax Function

Probabilities must be >= 0

Probabilities must sum to 1



cat	**3.2**	**24.5**	**0.13**
car	5.1	164.0	0.87
frog	-1.7	0.18	0.00

Unnormalized log-probabilities / logits

unnormalized probabilities

probabilities

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Softmax Classifier (Multinomial Logistic Regression)

Want to interpret raw classifier scores as **probabilities**

$$s = f(x_i; W)$$

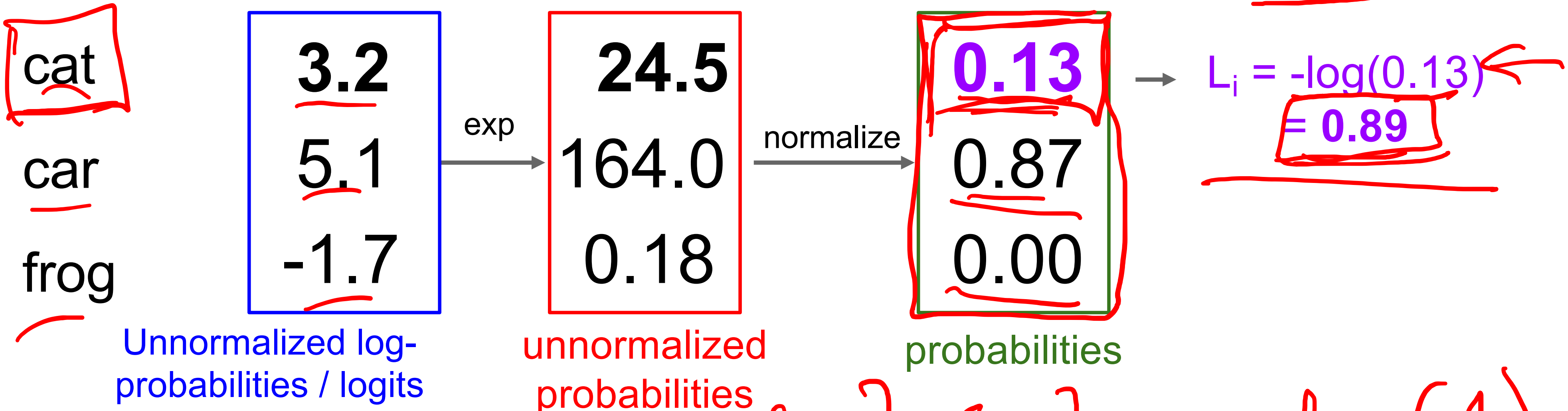
$$P(Y = k|X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}$$

Softmax Function

Probabilities must be >= 0

Probabilities must sum to 1

$$L_i = -\log P(Y = y_i|X = x_i)$$



$-\log(0)$

$$\begin{bmatrix}0\\1\\0\end{bmatrix} \quad \begin{bmatrix}1\\0\\0\end{bmatrix}$$

$-\log(1)$

$= 0$

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Softmax Classifier (Multinomial Logistic Regression)

Want to interpret raw classifier scores as **probabilities**

$$s = f(x_i; W)$$

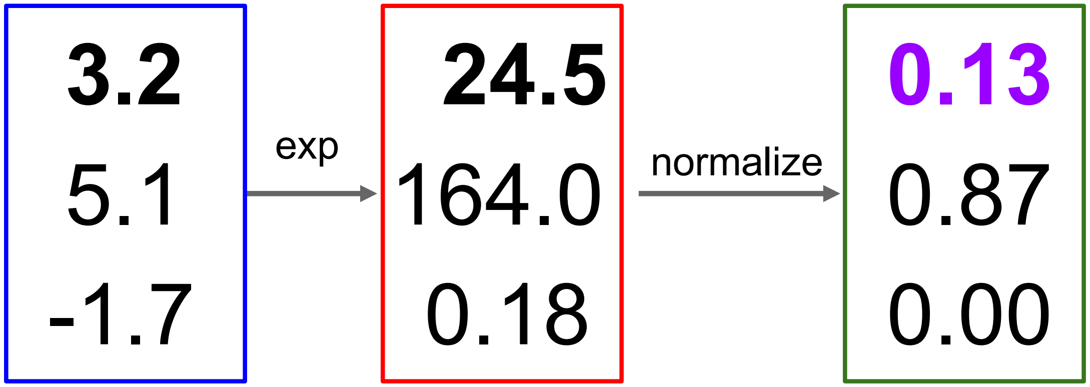
$$P(Y=k|X=x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}$$

Softmax Function

Probabilities must be >= 0

Probabilities must sum to 1

$$L_i = -\log P(Y=y_i|X=x_i)$$



		exp		normalize	
cat	**3.2**		**24.5**		**0.13**
car	5.1		164.0		0.87
frog	-1.7		0.18		0.00
	Unnormalized log-probabilities / logits		unnormalized probabilities		probabilities

→ L_i = -log(0.13) = **2.04**

Maximum Likelihood Estimation
Choose probabilities to maximize the likelihood of the observed data

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Log-Likelihood / KL-Divergence / Cross-Entropy

$D = \{(x_i, y_i)\}$ IID $\sim P^*$

$$\boxed{\hat{w}_{MLE}} = \max P(D \mid w)$$

$$= \max \log P(D \mid w)$$

$$= \max \boxed{\sum_i} \log P(y_i \mid x_i, w)$$

Log-Likelihood / KL-Divergence / Cross-Entropy

$$\underline{p^{gt}} = \overset{y_i}{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}} \qquad \underline{\hat{p}} = \begin{bmatrix} P(y=1 \mid x_i, w) \\ \vdots \\ P(y=K \mid x_i, w) \end{bmatrix}$$

$$\underline{H(p^{gt}, \hat{p})} = -\sum_y \underline{p^{gt}(y)} \log \underline{\hat{p}(y)}$$

$$= -\log \hat{P}(y_i \mid x_i, w)$$

$$\underline{D_{KL}(p^{gt}, \hat{p})} = \sum_y \underline{p^{gt}(y)} \log \frac{p^{gt}(y)}{\hat{p}(y)}$$

$$= \underline{\sum p^{gt}(y) \log p^{gt}(y)} - \sum_y p^{gt}(y) \log \hat{p}(y)$$

$$= -H(p^{gt}) + H(p^{gt}, \hat{p})$$

Log-Likelihood / KL-Divergence / Cross-Entropy

Softmax Classifier (Multinomial Logistic Regression)

Want to interpret raw classifier scores as **probabilities**

$$s = f(x_i; W)$$

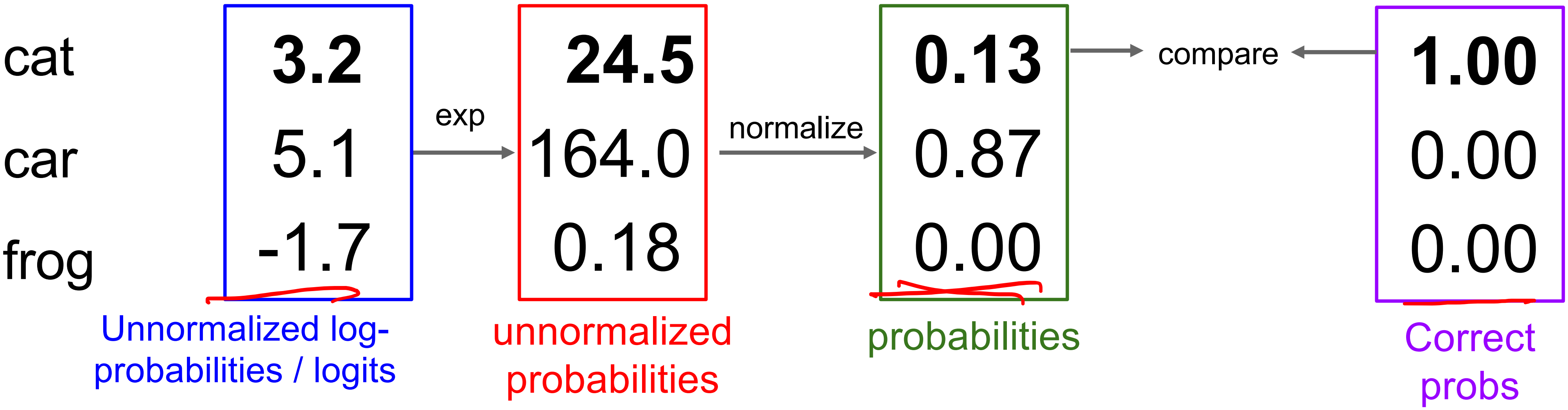
$$P(Y = k | X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}$$

Softmax Function

Probabilities must be >= 0

Probabilities must sum to 1

$$L_i = -\log P(Y = y_i | X = x_i)$$



	Unnormalized log-probabilities / logits	unnormalized probabilities	probabilities	Correct probs
cat	**3.2**	**24.5**	**0.13**	**1.00**
car	5.1	164.0	0.87	0.00
frog	-1.7	0.18	0.00	0.00

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Softmax Classifier (Multinomial Logistic Regression)

Want to interpret raw classifier scores as **probabilities**

$$\boxed{s = f(x_i; W)}$$

$$\boxed{P(Y = k|X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}}$$ Softmax Function

$$L_i = -\log P(Y = y_i|X = x_i)$$

	Unnormalized log-probabilities / logits	unnormalized probabilities	probabilities	Correct probs
		Probabilities must be >= 0	Probabilities must sum to 1	
cat	**3.2**	**24.5**	$\underline{\textbf{0.13}}$	**1.00**
car	5.1	164.0	0.87	0.00
frog	-1.7	0.18	0.00	0.00

exp → normalize → ; probabilities → compare ← Correct probs

Kullback–Leibler divergence

$$\boxed{D_{KL}(\underline{P}\|\underline{Q}) =}$$

$$\underline{\sum_y P(y) \log \frac{P(y)}{Q(y)}}$$

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

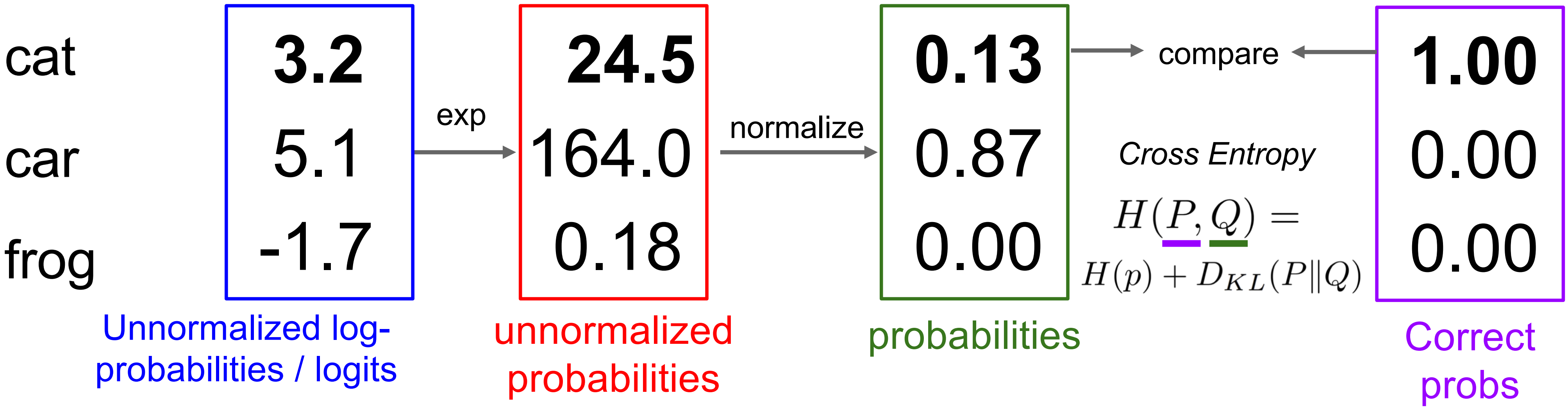
Softmax Classifier (Multinomial Logistic Regression)

Want to interpret raw classifier scores as **probabilities**

$$\boxed{s = f(x_i; W)}$$

$$\boxed{P(Y = k|X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}}$$ Softmax Function

$$L_i = -\log P(Y = y_i|X = x_i)$$



Probabilities must be >= 0

Probabilities must sum to 1

	Unnormalized log-probabilities / logits	$\xrightarrow{\text{exp}}$	unnormalized probabilities	$\xrightarrow{\text{normalize}}$	probabilities	compare	Correct probs
cat	**3.2**		**24.5**		**0.13**	→ compare ←	**1.00**
car	5.1		164.0		0.87		0.00
frog	-1.7		0.18		0.00		0.00

Cross Entropy

$$H(\underline{P}, \underline{Q}) = H(p) + D_{KL}(P\|Q)$$

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Softmax Classifier (Multinomial Logistic Regression)

Want to interpret raw classifier scores as **probabilities**

$$\boxed{s = f(x_i; W)}$$

$$\boxed{P(Y = k|X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}}$$

Softmax Function

Maximize probability of correct class

$$L_i = -\log P(Y = y_i|X = x_i)$$

Putting it all together:

$$\underline{L_i} = -\underline{\log(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}})}$$

cat	**3.2**
car	5.1
frog	-1.7

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Softmax Classifier (Multinomial Logistic Regression)

Want to interpret raw classifier scores as **probabilities**

$$\boxed{s = f(x_i; W)}$$

$$\boxed{P(Y = k | X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}}$$

Softmax Function

Maximize probability of correct class

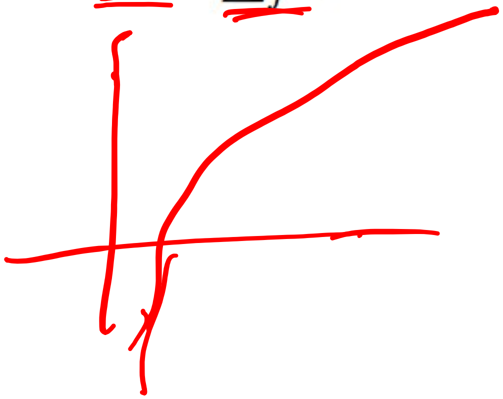
$$L_i = -\log P(Y = y_i | X = x_i)$$

Putting it all together:

$$L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$

cat	**3.2**
car	5.1
frog	-1.7

Q: What is the min/max possible loss L_i?



$-\log(0)$

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Softmax Classifier (Multinomial Logistic Regression)

cat	**3.2**
car	5.1
frog	-1.7

Want to interpret raw classifier scores as **probabilities**

$$\boxed{s = f(x_i; W)}$$

$$\boxed{P(Y = k|X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}}$$

Softmax Function

Maximize probability of correct class

$$L_i = -\log P(Y = y_i|X = x_i)$$

Putting it all together:

$$L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$

Q: What is the min/max possible loss L_i?

A: min 0, max infinity

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Softmax Classifier (Multinomial Logistic Regression)

Want to interpret raw classifier scores as **probabilities**

$$\boxed{s = f(x_i; W)}$$

$$\boxed{P(Y = k|X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}}$$ Softmax Function

Maximize probability of correct class

$$L_i = -\log P(Y = y_i|X = x_i)$$

Putting it all together:

$$L_i = -\log(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}) \leftarrow$$

cat	0	**3.2**
car	0	5.1
frog	0	-1.7

Q2: At initialization all s will be approximately equal; what is the loss?

$$-\log(\frac{1}{3}) = \log(3)$$

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Softmax Classifier (Multinomial Logistic Regression)

Want to interpret raw classifier scores as **probabilities**

$$\boxed{s = f(x_i; W)}$$

$$\boxed{P(Y = k | X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}}$$

Softmax Function

Maximize probability of correct class

$$L_i = -\log P(Y = y_i | X = x_i)$$

Putting it all together:

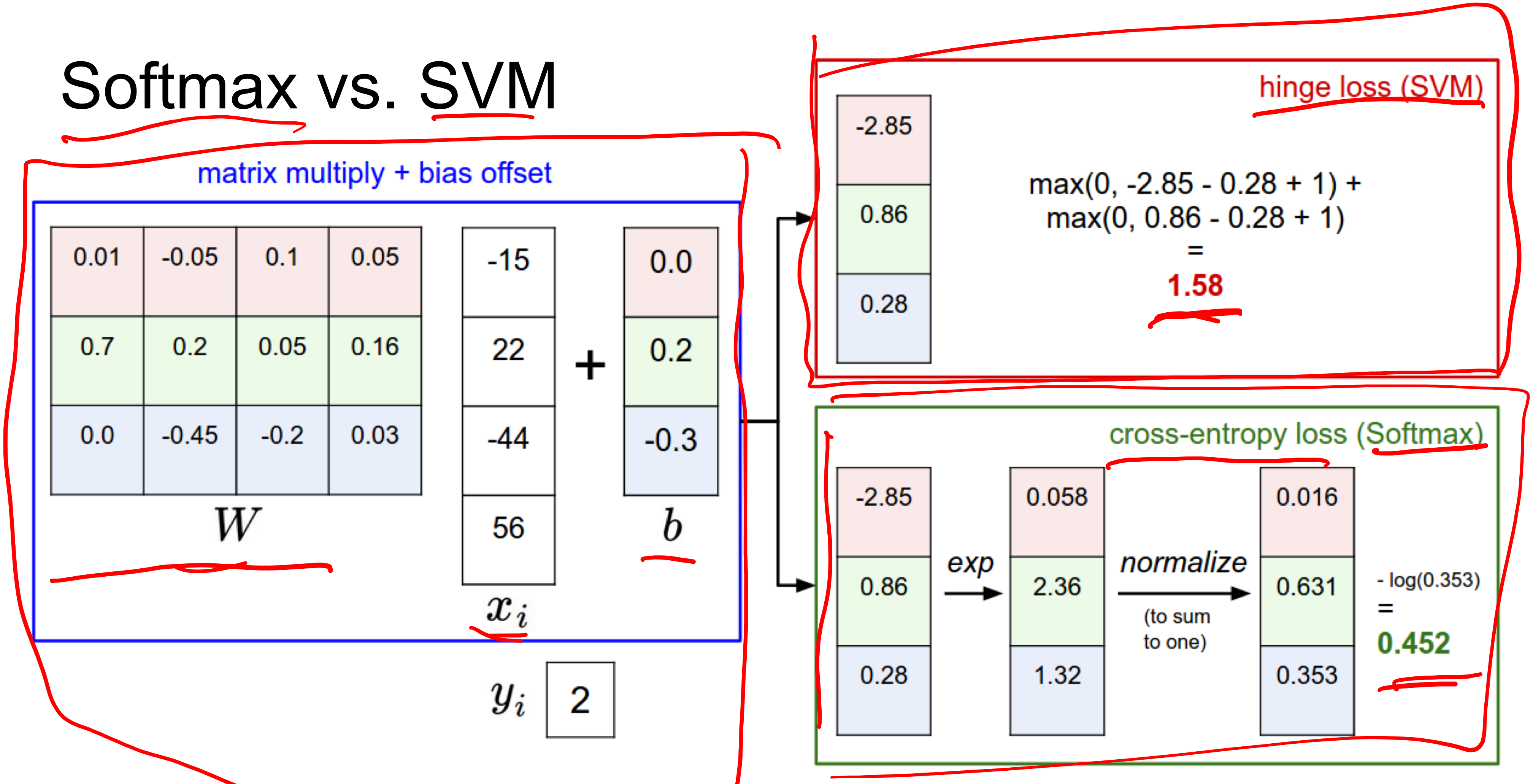
$$L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$

cat	**3.2**
car	5.1
frog	-1.7

Q2: At initialization all s will be approximately equal; what is the loss?
A: log(C), eg log(10) ≈ 2.3

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Softmax vs. SVM



matrix multiply + bias offset

$$\underset{W}{\begin{bmatrix}0.01 & -0.05 & 0.1 & 0.05\\ 0.7 & 0.2 & 0.05 & 0.16\\ 0.0 & -0.45 & -0.2 & 0.03\end{bmatrix}} \underset{x_i}{\begin{bmatrix}-15\\ 22\\ -44\\ 56\end{bmatrix}} + \underset{b}{\begin{bmatrix}0.0\\ 0.2\\ -0.3\end{bmatrix}}$$

y_i $\boxed{2}$

hinge loss (SVM)

$$\begin{bmatrix}-2.85\\ 0.86\\ 0.28\end{bmatrix}$$

max(0, -2.85 - 0.28 + 1) +
max(0, 0.86 - 0.28 + 1)
=
1.58

cross-entropy loss (Softmax)

$$\begin{bmatrix}-2.85\\ 0.86\\ 0.28\end{bmatrix} \xrightarrow{exp} \begin{bmatrix}0.058\\ 2.36\\ 1.32\end{bmatrix} \xrightarrow[\text{(to sum to one)}]{normalize} \begin{bmatrix}0.016\\ 0.631\\ 0.353\end{bmatrix}$$

- log(0.353)
=
0.452

Model

Objective

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Softmax vs. SVM

$$L_i = -\log(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}})$$

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Softmax vs. SVM

$$L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

assume scores:

[10, -2, 3]

[10, 9, 9]

[10, -100, -100]

and $\boxed{y_i = 0}$

Q: Suppose I take a datapoint and I jiggle a bit (changing its score slightly). What happens to the loss in both cases?

Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Recap

- We have some dataset of (x,y)
- We have a **score function:** $s = f(x; W) \overset{\text{e.g.}}{=} Wx$
- We have a **loss function**:

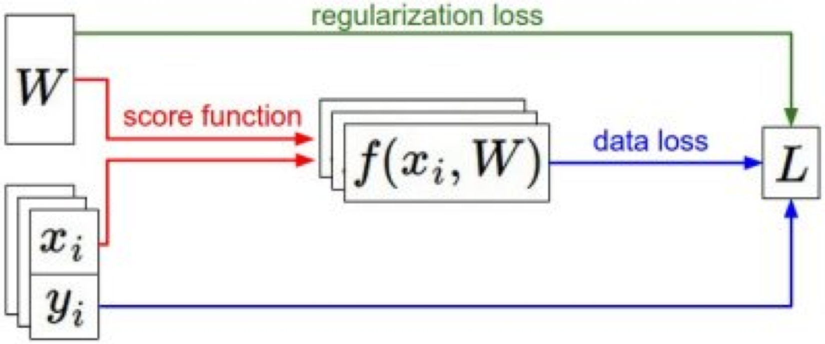
Softmax

$$L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$

SVM

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

$$L = \frac{1}{N} \sum_{i=1}^{N} L_i + R(W)$$ Full loss



Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

Recap

How do we find the best W?

- We have some dataset of (x,y)
- We have a **score function:** $s = f(x; W) \overset{\text{e.g.}}{=} Wx$
- We have a **loss function**:

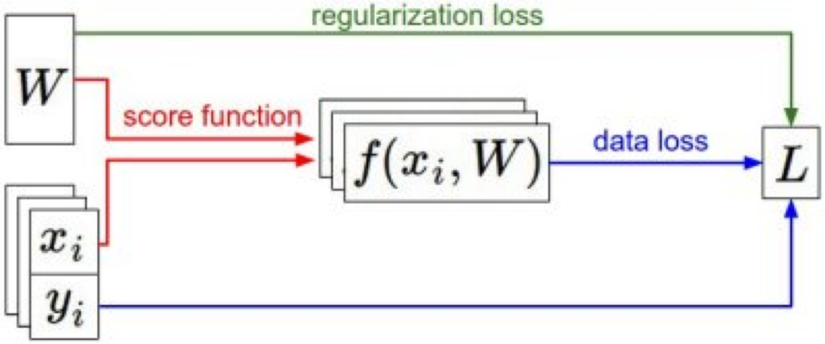
Softmax

$$L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$

SVM

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

$$L = \frac{1}{N} \sum_{i=1}^{N} L_i + R(W)$$ Full loss



Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n