JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

EN.601.482/682 Deep Learning

Basics Part I:

Image Features, Regression, and Classification

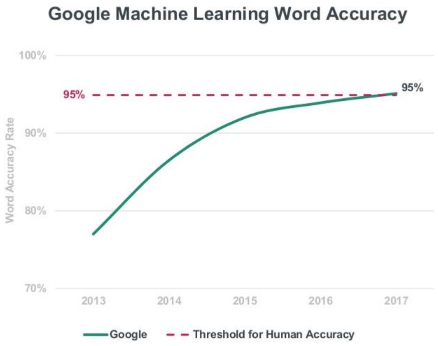
Mathias Unberath, PhD
Assistant Professor
Dept of Computer Science
Johns Hopkins University

Reminder

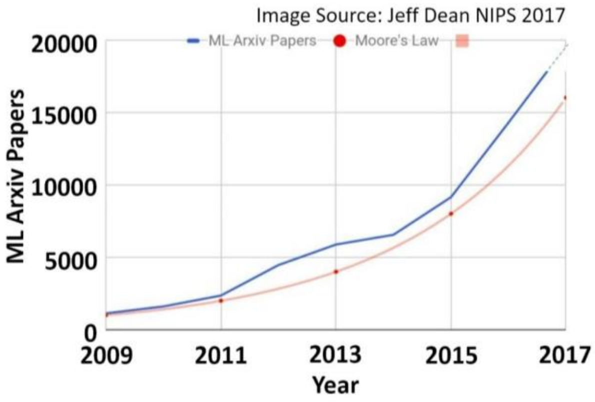
- Sign up on **Piazza** (access code dlF23)
- Read the **syllabus**
- **Homework assignment 1 will be released today** (due next Wednesday)

Reminder

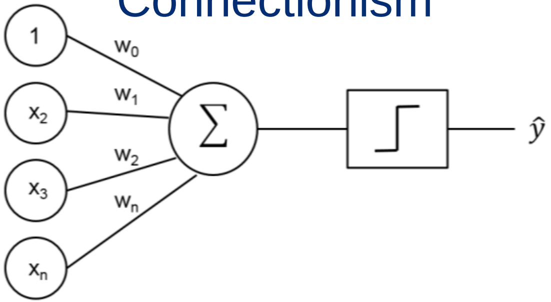
Neural networks are taking / have taken over!

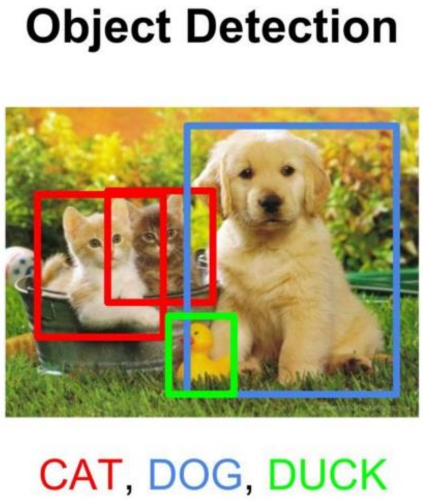


Language processing



Connectionism





Pixar movie scene of a dark skull wizard fighting against Kermit the frog as a gladiator, incredible render, Presto

Reminder

- What is the difference between unsupervised and supervised learning?

- What is the difference between classification and regression?

Today’s Lecture

Linear Classification

- Problem Statement
- Image Features
- The SVM Loss

Logistic Regression

- The Softmax Function
- A Little Bit on Maximum Likelihood

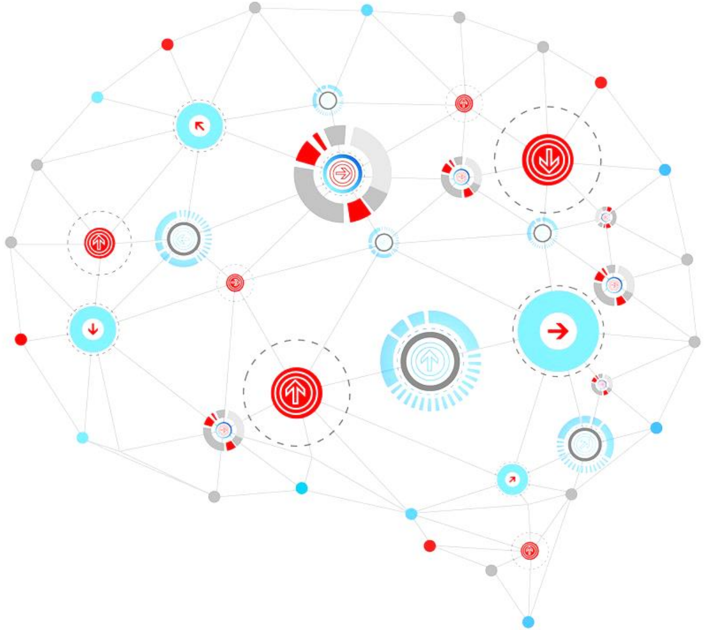
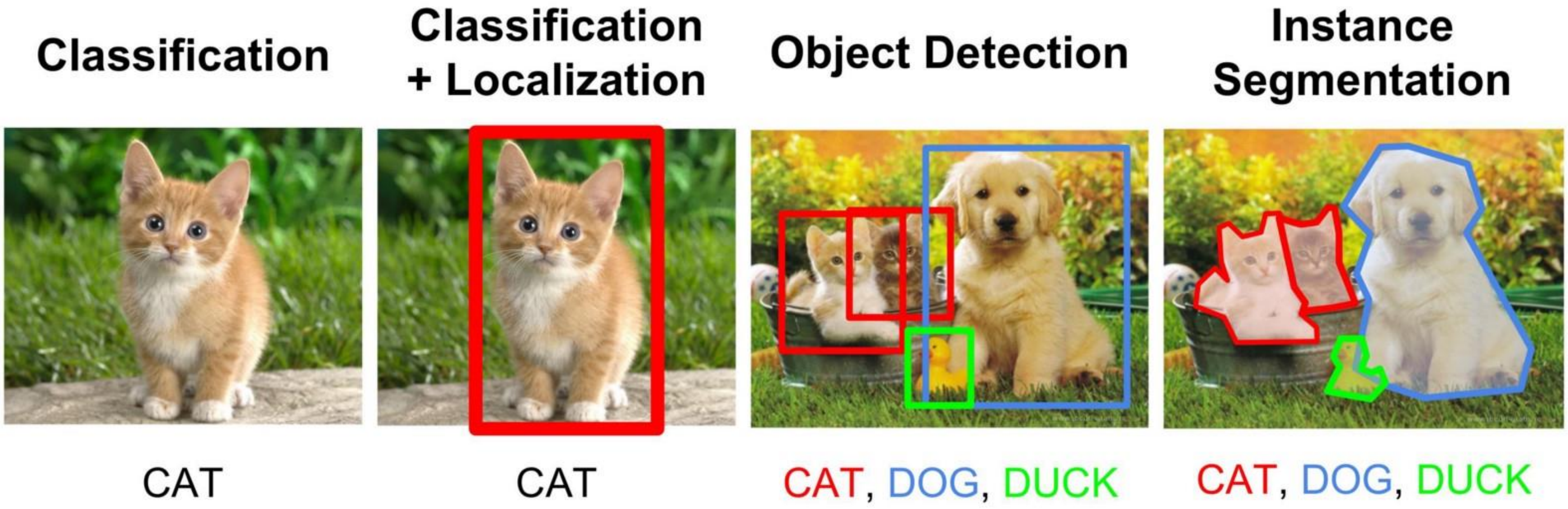


Image Features, Regression, and Classification

Problem Statement

From Classification to Instance Segmentation



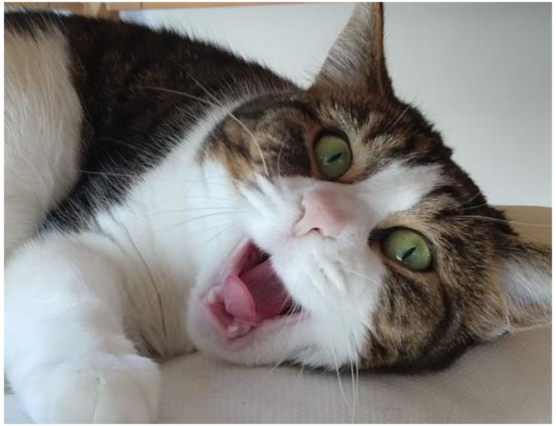
Single object

Multiple objects

Ouaknine, A. (2018) Review of DL for Object Detection Blog Post

Image Classification

Assume set of discrete labels is known, e.g. {dog, cat, car, plane, …}
Classification describes a mapping of image onto label

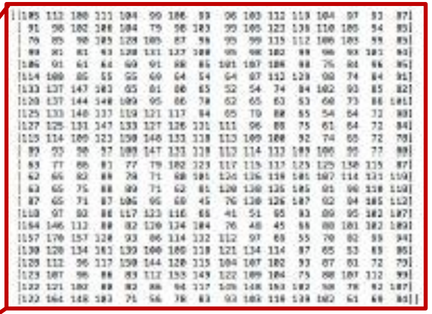


This is my cat.

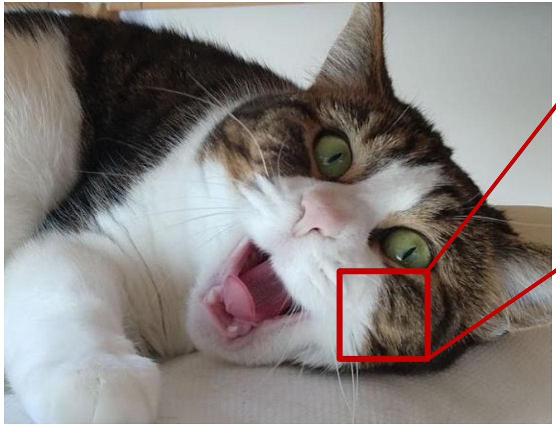
Image Classification

Why is this challenging?

Semantic gap



What the computer sees



What the human sees

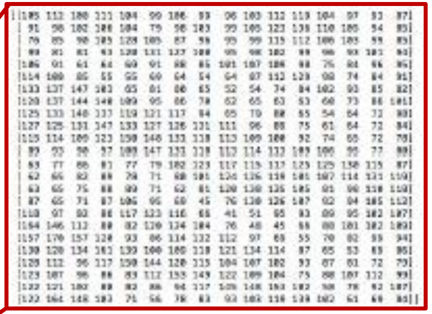
An image is essentially a grid of numbers (often, but not always) between [0,255].

Very common:
CV: 640 x 480 x 3 (RGB channels, 8 bit)
X-ray: 1240 x 960 x 1 (Grayscale, 14 bit)

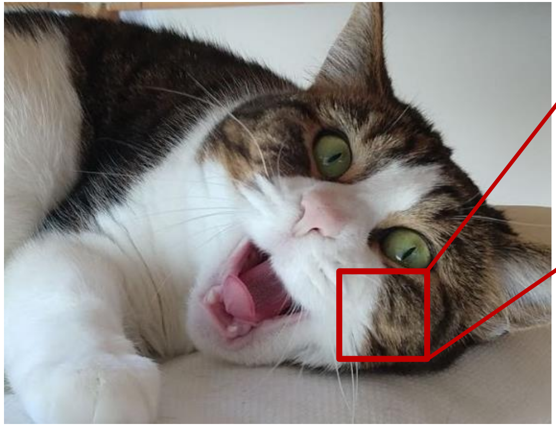
Image Classification

Why is this challenging?

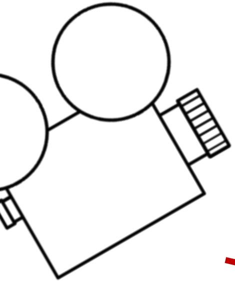
Viewpoint

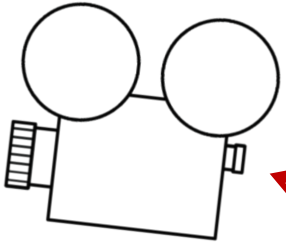


What the computer sees



What the human sees



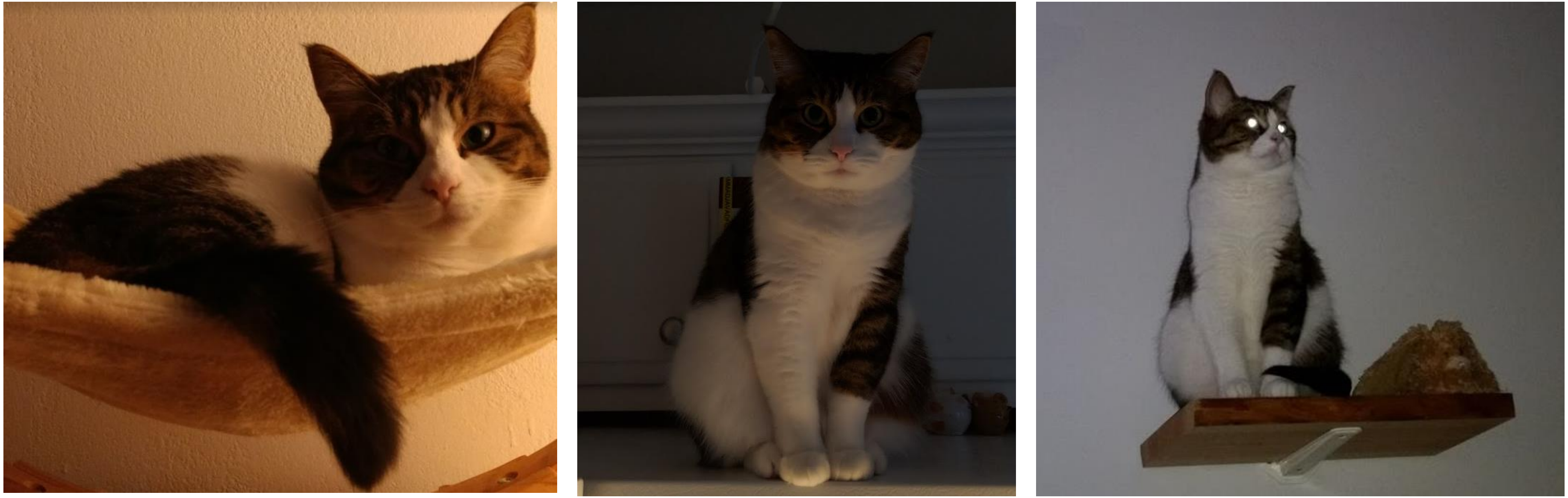


When camera moves,
all pixel values change!

Image Classification

Why is this challenging?

Lighting conditions

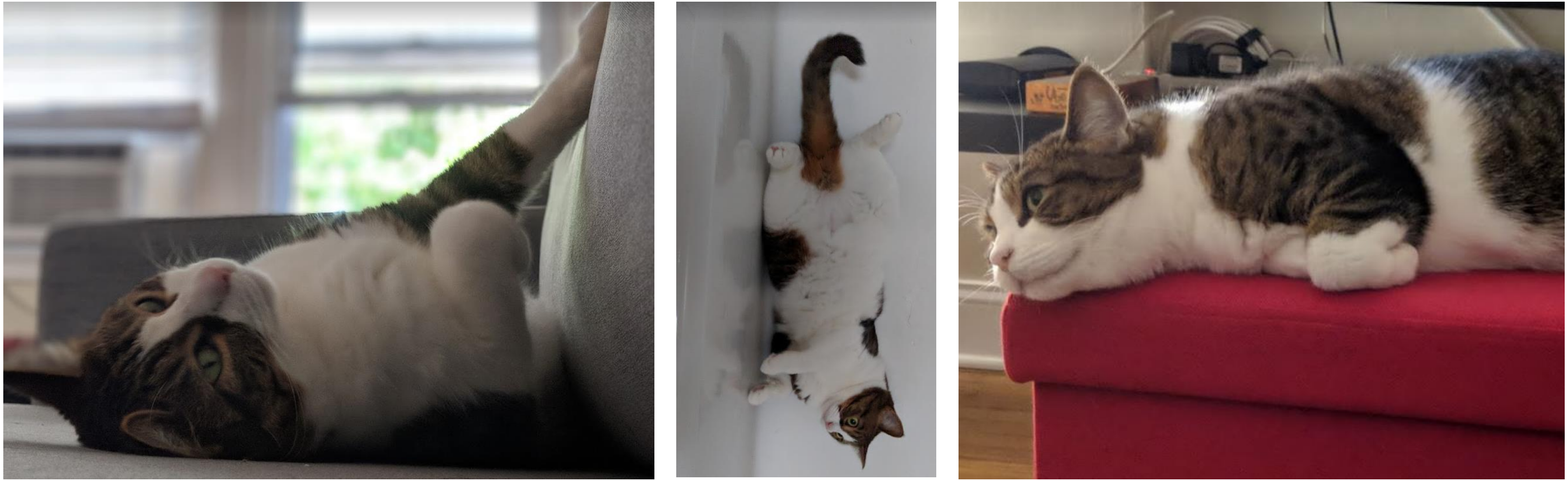


Still my cat.

Image Classification

Why is this challenging?

Deformation



Still my cat.

Image Classification

Why is this challenging?

Occlusion

Still my cat.

Image Classification

Why is this challenging?

Background clutter

These images are CC0 1.0 public domain.

Image Classification

Why is this challenging?

Intraclass variation

This image is CC0 1.0 public domain.

An Example Classifier

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

There is no obvious way to hard-code such classification algorithm.

The Machine Learning Approach

1. Curate a (large) dataset of images with corresponding annotation (i.e. labels)
2. Train a classifier using machine learning
3. Evaluate the classifier on unseen images

Example training set

The Machine Learning Approach

1. Curate a (large) dataset of images with corresponding annotation (i.e. labels)
2. Train a classifier using machine learning
3. Evaluate the classifier on unseen images

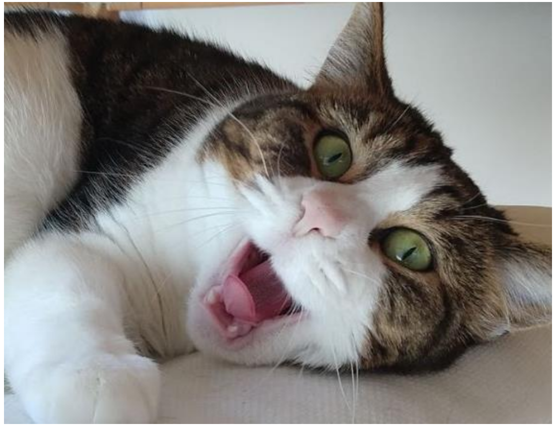
Example dataset: **CIFAR10**
10 classes
50,000 training images
10,000 testing images

Each image: 32 x 32 x 3

Krizhevsky, A. (2009) Learning Multiple Layers of Features from Tiny Images

Image Classification

If there are multiple classes, e.g. {cat, car, frog}
→ Response for every label



5.1	cat	1
2.3	car	0
-0.7	frog	0

One-hot encoding

This is my cat.

Parametric Approach

Input image

$\mathbf{x}$

Array of 32x32x3 numbers
(3072 numbers total)

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

$\mathbf{W}$

Parameters or weights

10 numbers for class scores

Parametric Approach

Input image

x

Array of 32x32x3 numbers
(3072 numbers total)

Goal in machine learning:
Find this f(x,W)!

$f(x,W)$

10 numbers for class scores

W

Parameters or weights

Linear Classifier

Input image

x

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

Array of 32x32x3 numbers
(3072 numbers total)

$f(x,W)$

10 numbers for class scores

W

Parameters or weights

Linear Classifier

Input image

x

Array of 32x32x3 numbers
(3072 numbers total)

$$\underbrace{\boxed{f(\mathbf{x},\mathbf{W})}}_{10 \times 1} = \underbrace{\boxed{\mathbf{W}}}_{10 \times 3072}\overset{3072 \times 1}{\boxed{\mathbf{x}}}$$

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

10 numbers for class scores

W

Parameters or weights

Linear Classifier

$$\underbrace{\boxed{f(x,W)}}_{10 \times 1} = \underbrace{\boxed{W}}_{10 \times 3072}\overbrace{\boxed{x}}^{3072 \times 1} + \boxed{b} \quad 10 \times 1$$

Bias term: constant!
→ Input independent preference

Input image
x

Array of 32x32x3 numbers
(3072 numbers total)

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

10 numbers for class scores

W
Parameters or weights

Linear Classifier

$$\underbrace{\boxed{f(x,W)}}_{10 \times 1} = \underbrace{\boxed{W}}_{10 \times 2\text{ Mio.}}\overset{2\text{ Mio.} \times 1}{\boxed{x}} + \boxed{b}\ \ 10 \times 1$$

Input image

x

800x800x3 numbers
(~2 Mio. numbers total)

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

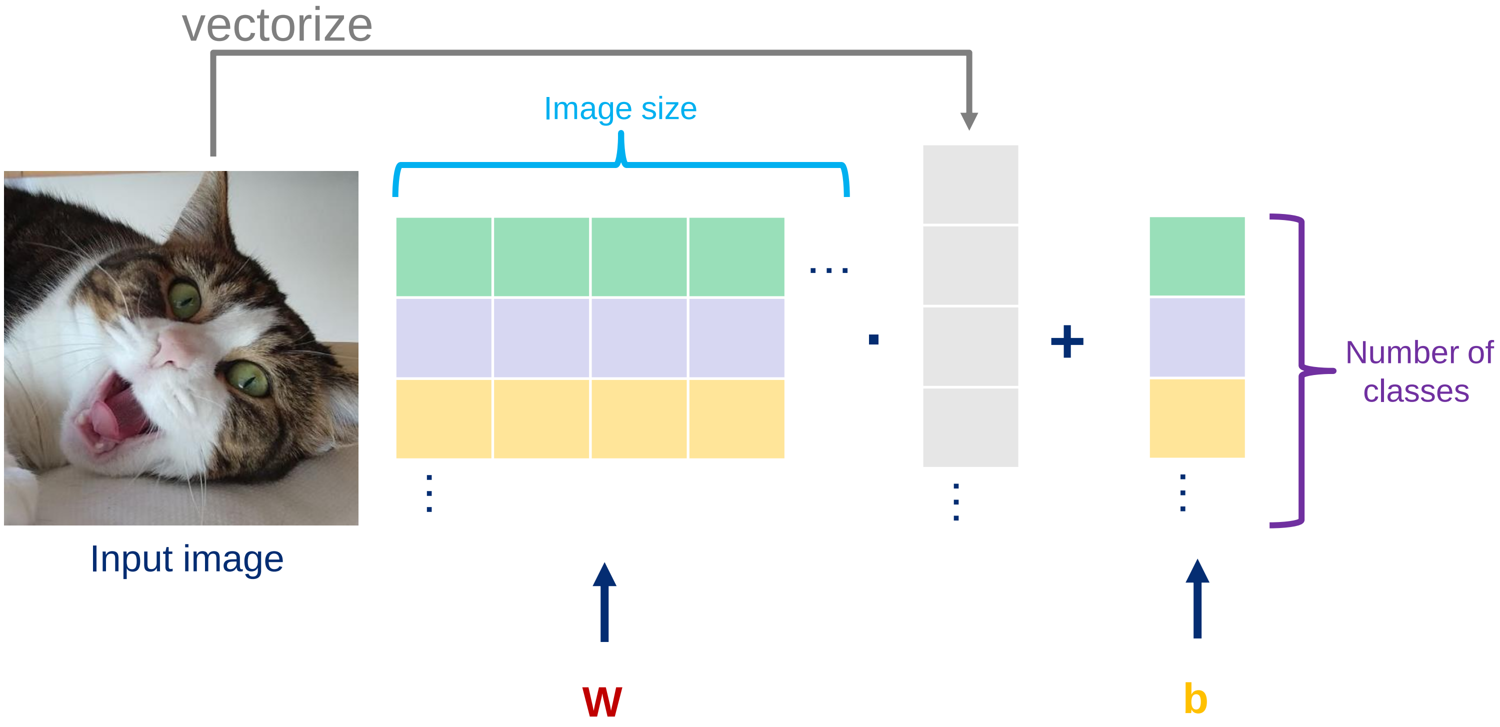
10 numbers for class scores

W

Parameters or weights

Actual 32x32x3

Interpretation of Linear Classification in this Formalism



vectorize

Image size

Number of classes

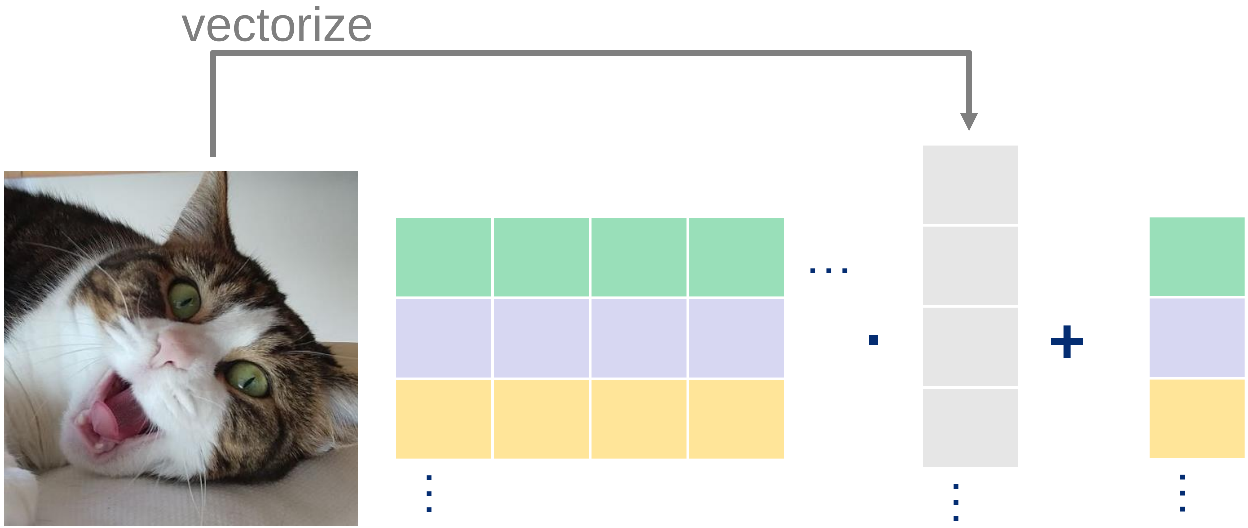
Input image

$\cdot$ +

W

b

Interpretation of Linear Classification in this Formalism

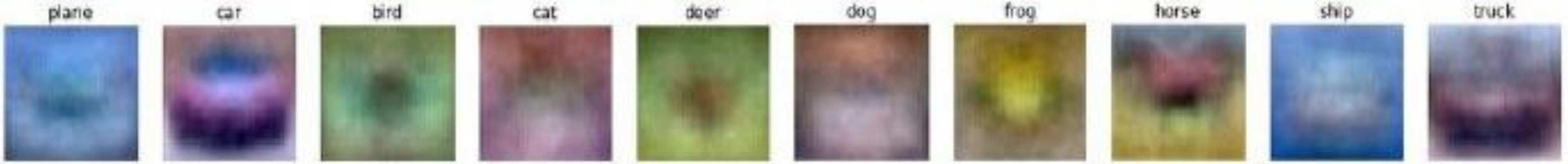


Input image

- Every row 🟩🟩🟩🟩 … of **W** acts like a “template” for the corresponding class

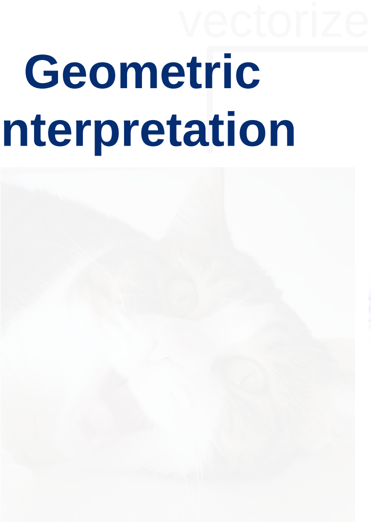
Interpretation of Linear Classification in this Formalism

Visual Interpretation

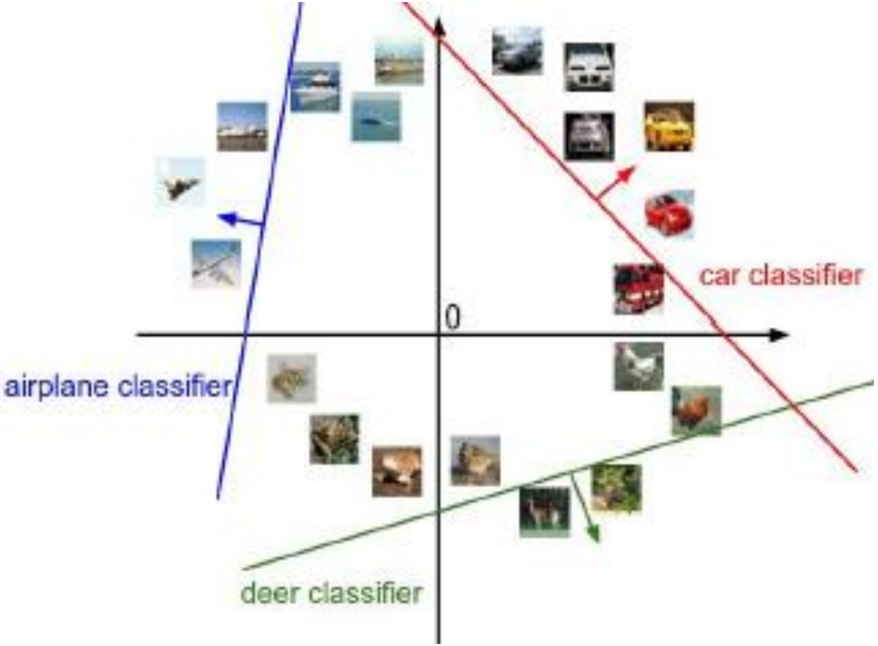


Interpretation of Linear Classification in this Formalism

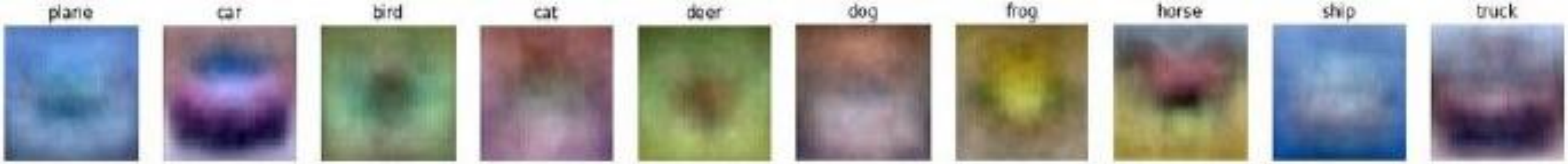
Geometric Interpretation



Input image



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



An Obvious Challenge

Template assumption on pixel level is strong!

We need some form of abstraction
→ Enter: **Features**!

Seagull

Where are we now? Linear classifier

- Algebraic formalism: f(**x**,**W**) = **W x + b**

- Visual interpretation:
  Rows of **W** form templates for classes

- Geometric interpretation:
  Instances **x** are points in high-dimensional space
  Rows of **W** define hyperplanes (linear decision boundaries)

- Challenge:
  Images exhibit lots of variation! Template assumption difficult to justify.
  → Higher-level representation of images

Image Features, Regression, and Classification

Image Features

Image Features

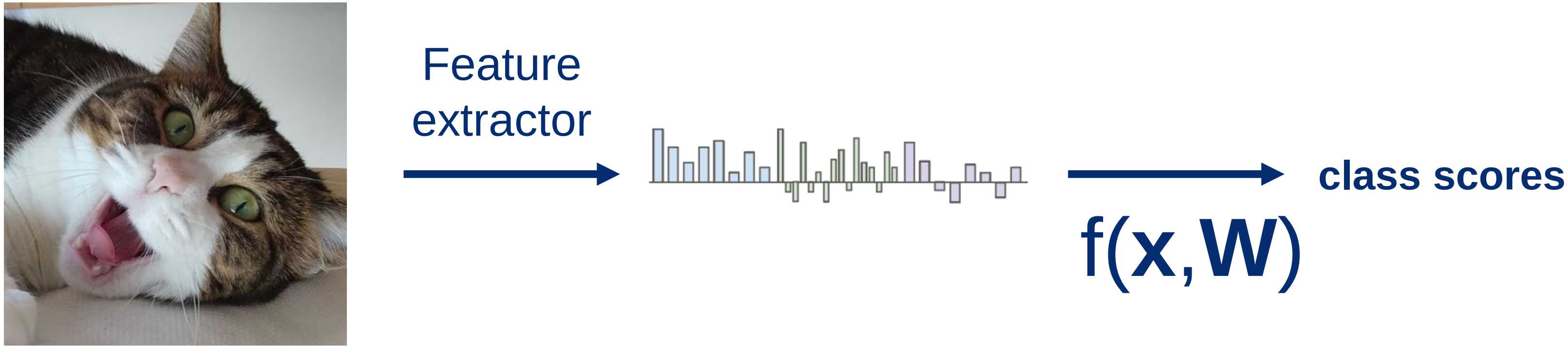
- What are the image features in this example? Pixel intensities!

class scores

$f(x,W)$

Parametric Approach

- Higher-level representation via feature extraction

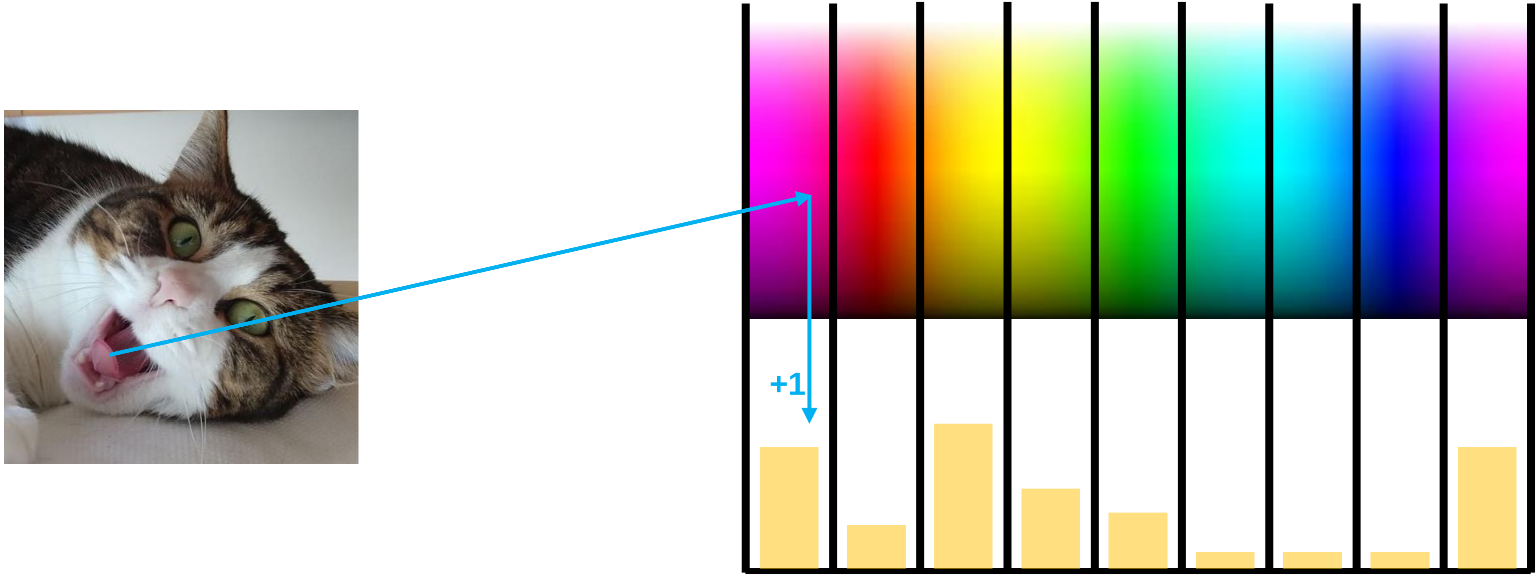


Feature extractor

class scores

$f(x,W)$

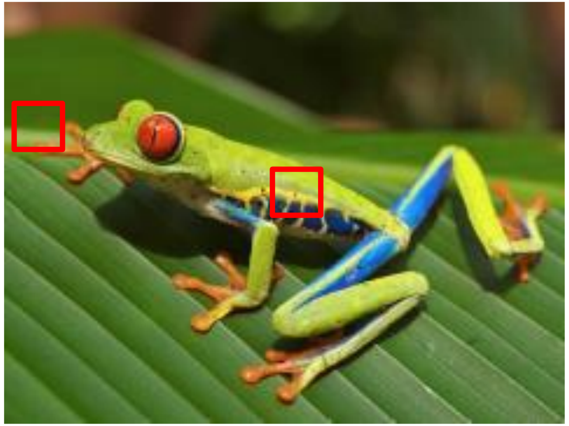
Feature Extractor: Color Histogram

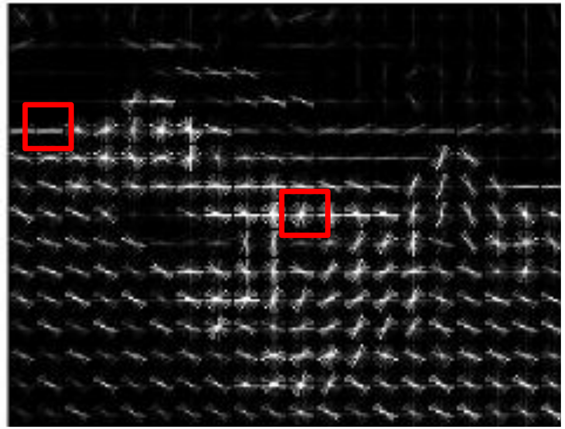


Color histogram is a *global* descriptor
Feature vector has #(hist bins) elements

Feature Extractor: Histogram of Oriented Gradients

- Divide into small regions
- Quantize gradient direction into bins





Example numbers:

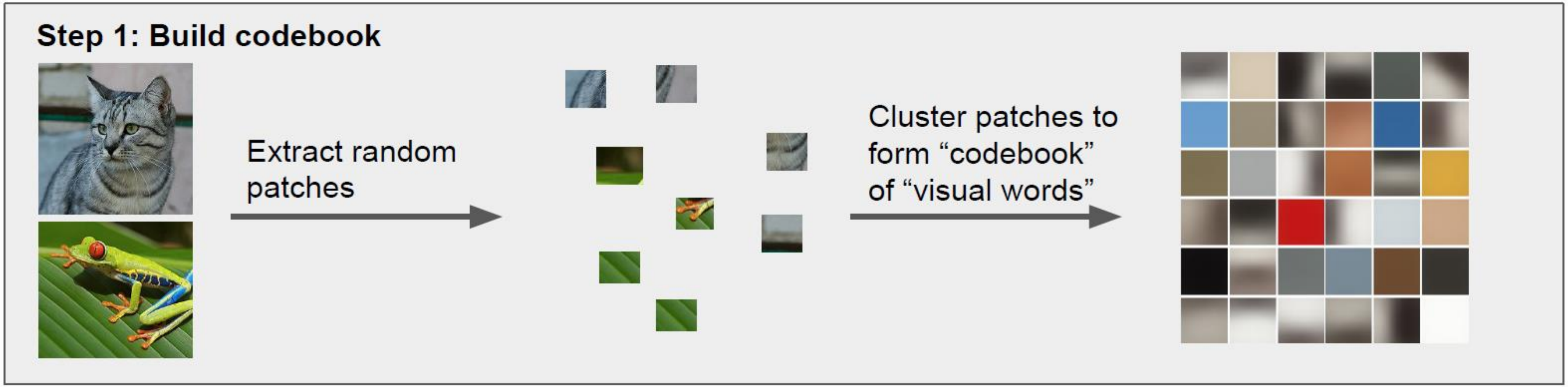
Input image 320x240; Patch size: 8x8; 9 gradient bins.

→ 40x30 patches, each patch is 9-bin histogram: Feature vector has 40*30*9 = 10,800 elements!

Dalal, N., & Triggs, B. (2005) Histograms of oriented gradients for human detection. CVPR

Images from Stanford cs231n.

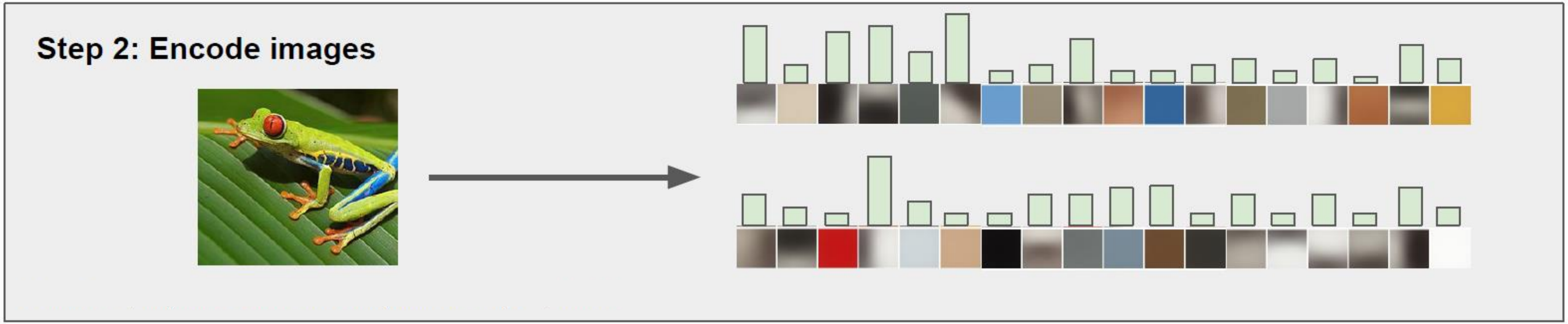
Feature Extractor: Bag of Visual Words



Step 1: Build codebook

Extract random patches

Cluster patches to form “codebook” of “visual words”



Step 2: Encode images

Fei-Fei, L., & Perona, P. (2005) A bayesian hierarchical model for learning natural scene categories. CVPR

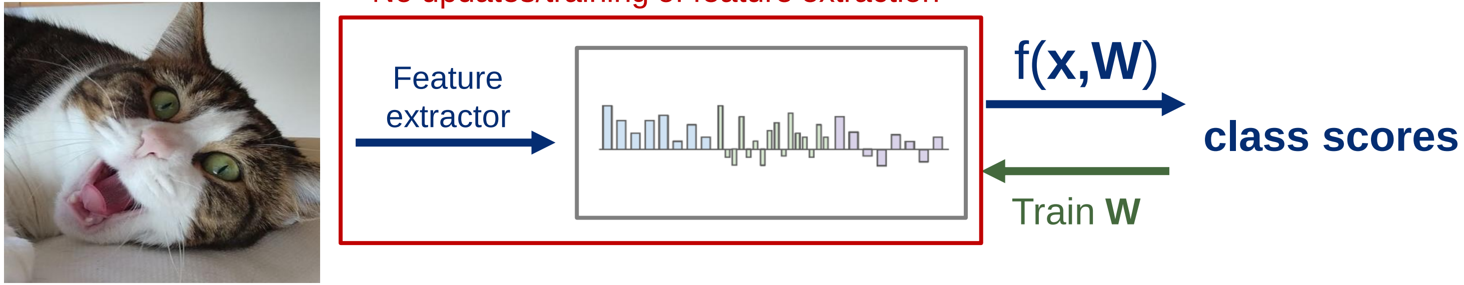
Images from Stanford cs231n.

Some Observations

- For all feature extractors, we have some design choice (hyperparameters)
  - E.g. for color histogram: Number of bins

- Feature engineering is not trivial
  → Requires strong domain knowledge

- Previously:



Fixed!
No updates/training of feature extraction

Guyon, I., & Elisseeff, A. (2003) An introduction to variable and feature selection. *Journal of machine learning research*, *3*(Mar), 1157-1182.

Some Observations

- For all feature extractors, we have some design choice (hyperparameters)
  - E.g. for color histogram: Number of bins

- Feature engineering is not trivial
  → Requires strong domain knowledge

- Teaser:

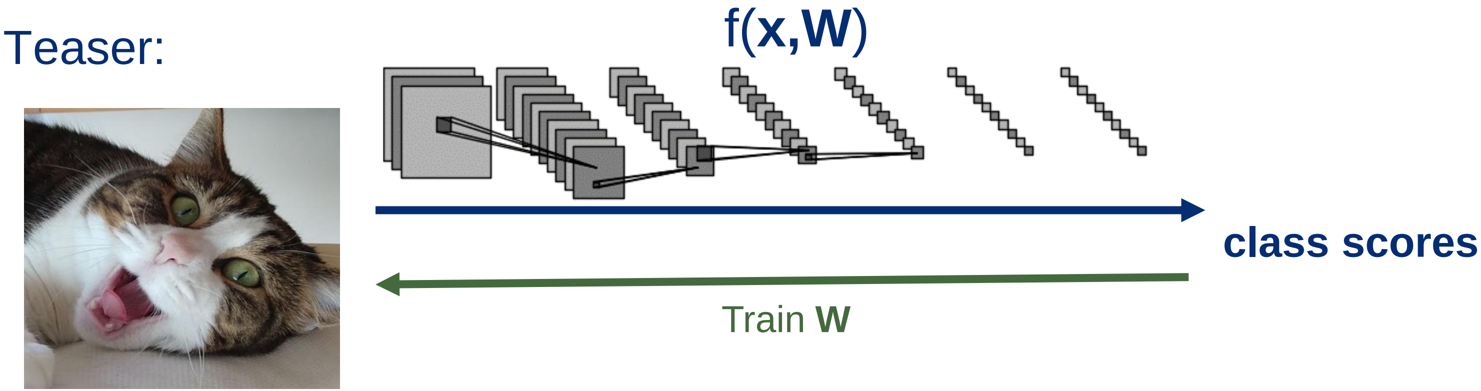


Image from DeepGrid.

Image Features, Regression, and Classification

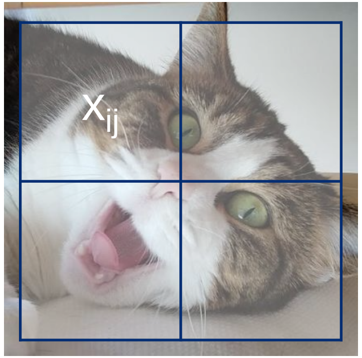
The SVM Loss

Unhappy?

Input **features**

$\mathbf{x}$

$$f(\mathbf{x},\mathbf{W}) = \mathbf{W}\,\mathbf{x} + \mathbf{b}$$



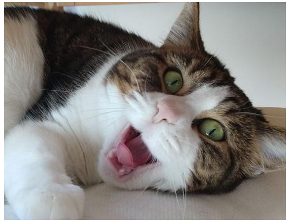
$$\xrightarrow{\quad} f(\mathbf{x},\mathbf{W}) \xrightarrow{\quad}$$

3.2	cat
4.9	car
-3.1	bird

- Given some **W**, how to express our (un)happiness with the current prediction?
→ Loss functions
- How to minimize our unhappiness? Update **W** → Optimization

The SVM Loss

- 3 training examples and 3 classes: cat, car, bird
- W has been determined, the scores are:

			
Cat	**3.2**	1.3	2.2
Car	5.1	**4.9**	2.5
Bird	-1.7	2.0	-**3.1**

The SVM Loss

- 3 training examples and 3 classes: cat, car, bird
- W has been determined, the scores are:

Cat	**3.2**	1.3	2.2
Car	5.1	**4.9**	2.5
Bird	-1.7	2.0	**-3.1**

Loss function:
Quantifies classifier performance

Given a dataset $\{(x_i, y_i)\}_{i=1}^{N}$ where

x_i the image (or its feature representation)

y_i the corresponding label (integer)

Loss over dataset = sum over all examples

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

The SVM Loss

- 3 training examples and 3 classes: cat, car, bird
- W has been determined, the scores are:

Cat	**3.2**	1.3	2.2
Car	5.1	**4.9**	2.5
Bird	-1.7	2.0	**-3.1**

Given a dataset $\{(x_i, y_i)\}_{i=1}^N$ where
x_i the image (or its feature representation)
y_i the corresponding label (integer)

Using: $s = f(x_i, W)$

The Support Vector Machine (SVM) loss:

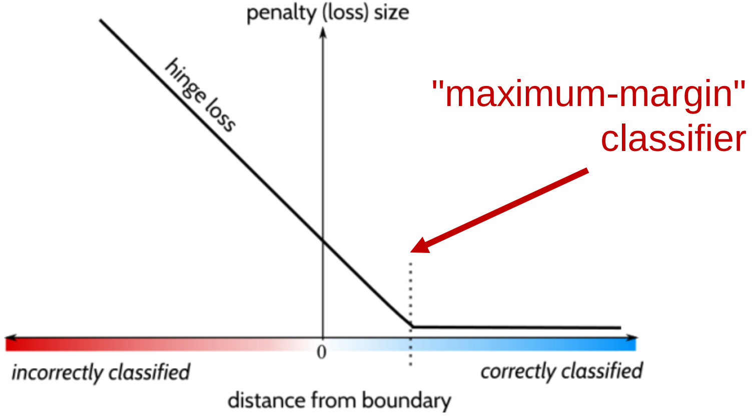
$$L_i = \sum_{j \neq y_i} \begin{cases} 0 & \text{if } s_{y_i} \geq s_j + 1 \\ s_j - s_{y_i} + 1 & \text{otherwise} \end{cases}$$

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

The SVM Loss

- 3 training examples and 3 classes: cat, car, bird
- W has been determined, the scores are:

Cat	**3.2**	1.3	2.2
Car	5.1	**4.9**	2.5
Bird	-1.7	2.0	**-3.1**



"maximum-margin" classifier

The Support Vector Machine (SVM) loss:

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

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

The SVM Loss

- 3 training examples and 3 classes: cat, car, bird
- W has been determined, the scores are:

Cat	**3.2**	1.3	2.2
Car	5.1	**4.9**	2.5
Bird	-1.7	2.0	**-3.1**
Loss	**2.9**		

The Support Vector Machine (SVM) loss:

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

Example: Cat

L_i = 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

The SVM Loss

- 3 training examples and 3 classes: cat, car, bird
- W has been determined, the scores are:

Cat	**3.2**	1.3	2.2
Car	5.1	**4.9**	2.5
Bird	-1.7	2.0	**-3.1**
Loss	**2.9**	**0**	

The Support Vector Machine (SVM) loss:

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

Example: Car

L_i = 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

The SVM Loss

- 3 training examples and 3 classes: cat, car, bird
- W has been determined, the scores are:

Cat	**3.2**	1.3	2.2
Car	5.1	**4.9**	2.5
Bird	-1.7	2.0	**-3.1**
Loss	**2.9**	**0**	**12.9**

The Support Vector Machine (SVM) loss:

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

Example: Bird

$L_i = \max(0, 2.2 – (–3.1) + 1)$

$\quad + \max(0, 2.5 – (–3.1) + 1)$

$\quad = \max(0, 6.3) + \max(0, 6.6)$

$\quad = 6.3 + 6.6$

$\quad = 12.9$

The SVM Loss

- 3 training examples and 3 classes: cat, car, bird
- W has been determined, the scores are:

Cat	**3.2**	1.3	2.2
Car	5.1	**4.9**	2.5
Bird	-1.7	2.0	**-3.1**
Loss	**2.9**	**0**	**12.9**

The Support Vector Machine (SVM) loss:

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

Overall loss:

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

L = 1/3 * (2.9 + 0 + 12.9) = **5.27**

The SVM Loss

- 3 training examples and 3 classes: cat, car, bird
- W has been determined, the scores are:

Cat	**3.2**	1.3	2.2
Car	5.1	**4.9**	2.5
Bird	-1.7	2.0	**-3.1**
Loss	**2.9**	**0**	**12.9**

The Support Vector Machine (SVM) loss:

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

Overall loss:

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

Q1: What happens to L if car score changes a bit?

The SVM Loss

- 3 training examples and 3 classes: cat, car, bird
- W has been determined, the scores are:

Cat	**3.2**	1.3	2.2
Car	5.1	**4.9**	2.5
Bird	-1.7	2.0	**-3.1**
Loss	**2.9**	**0**	**12.9**

The Support Vector Machine (SVM) loss:

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

Overall loss:

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

Q2: What is the min/max for L?

The SVM Loss

- 3 training examples and 3 classes: cat, car, bird
- W has been determined, the scores are:

Cat	**3.2**	1.3	2.2
Car	5.1	**4.9**	2.5
Bird	-1.7	2.0	**-3.1**
Loss	**2.9**	**0**	**12.9**

The Support Vector Machine (SVM) loss:

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

Overall loss:

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

Q3: If W is small and all s ≈ 0, what is L?

The SVM Loss

- 3 training examples and 3 classes: cat, car, bird
- W has been determined, the scores are:

Cat	**3.2**	1.3	2.2
Car	5.1	**4.9**	2.5
Bird	-1.7	2.0	**-3.1**
Loss	**2.9**	**0**	**12.9**

The Support Vector Machine (SVM) loss:

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

True label for cat (one-hot encoding).

1.00
0.00
0.00

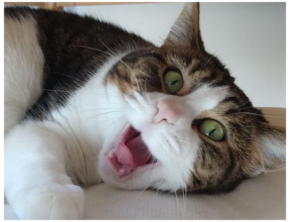
Q4: Why not use the L2 norm?

Image Features, Regression, and Classification

Logistic Regression

Multinomial Logistic Regression

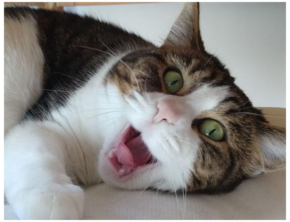
- Predictions as per linear regression are unbounded
- For classification, we would like to interpret scores as **probabilities**



Cat	**3.2**
Car	5.1
Bird	-1.7

Multinomial Logistic Regression

- Again: $s = f(x_i, W)$
- Then: $\boxed{P(Y = k|X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}}$ Softmax function

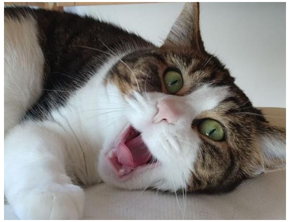


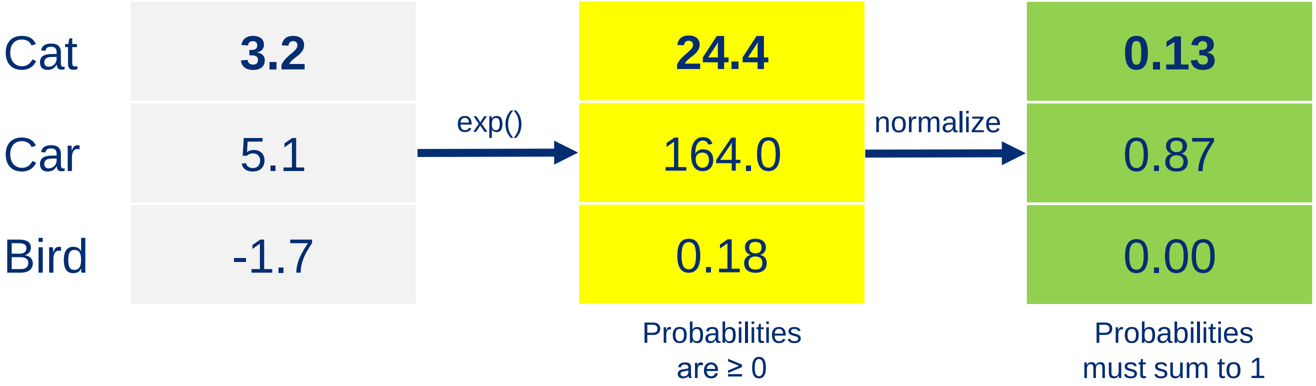
Cat	**3.2**		**24.4**
Car	5.1	$\xrightarrow{\text{exp()}}$	164.0
Bird	-1.7		0.18

Probabilities are ≥ 0

Multinomial Logistic Regression

- Again: $s = f(x_i, W)$
- Then: $\boxed{P(Y = k|X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}}$ Softmax function





Cat	**3.2**	**24.4**	**0.13**
Car	5.1	164.0	0.87
Bird	-1.7	0.18	0.00

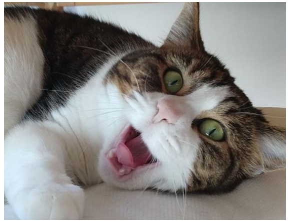
$\xrightarrow{\text{exp()}}$ $\xrightarrow{\text{normalize}}$

Probabilities are ≥ 0

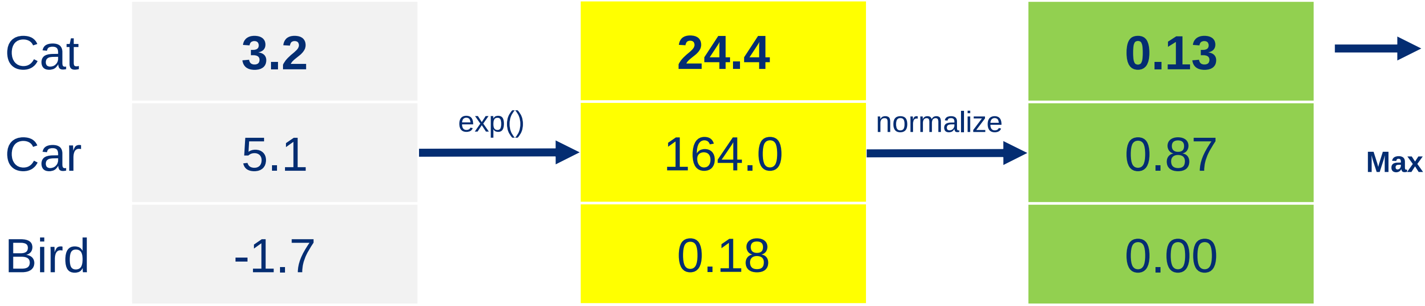
Probabilities must sum to 1

Multinomial Logistic Regression

- Again: $s = f(x_i, W)$
- Then: $\boxed{P(Y = k|X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}}$ Softmax function



Loss: $L_i = -\log P(Y = y_i|X = x_i)$ → Log-likelihood of true class



Cat	**3.2**	**24.4**	**0.13**
Car	5.1	164.0	0.87
Bird	-1.7	0.18	0.00

Probabilities are ≥ 0

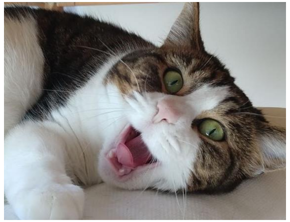
Probabilities must sum to 1

L_i = -log(0.13)
= **0.89** or **2.04**

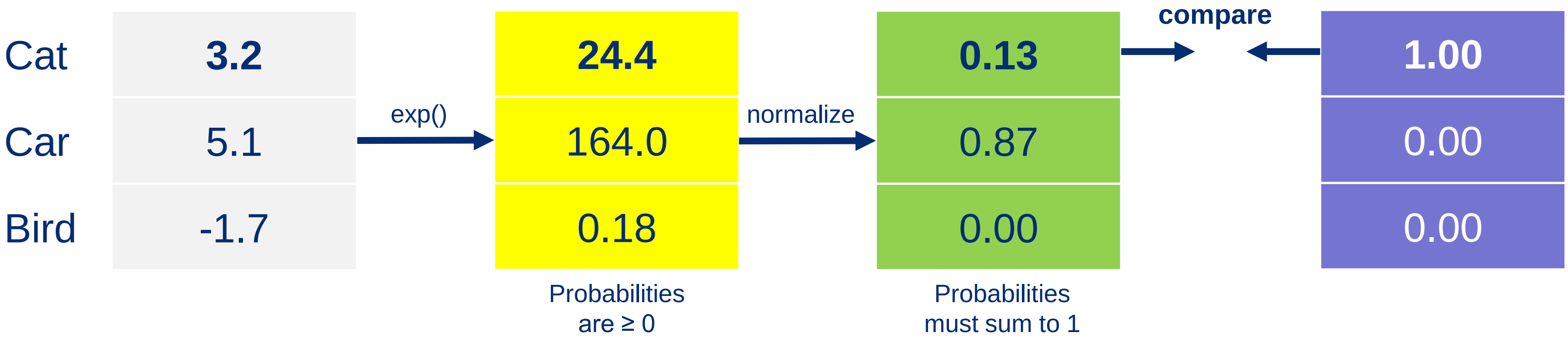
Maximum likelihood estimation
In general not trivial

Multinomial Logistic Regression

- Again: $s = f(x_i, W)$
- Then: $\boxed{P(Y = k | X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}}$ Softmax function



Loss: $D_{\mathrm{KL}}(Q, P) = \sum_k Q(k) \log \frac{Q(k)}{P(k)}$ → Kullback-Leibler Divergence



Cat	**3.2**	**24.4**	**0.13**	**1.00**
Car	5.1	164.0	0.87	0.00
Bird	-1.7	0.18	0.00	0.00

exp()

normalize

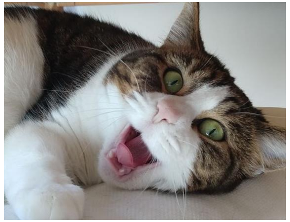
compare

Probabilities are ≥ 0

Probabilities must sum to 1

Multinomial Logistic Regression

- Again: $s = f(x_i, W)$
- Then: $\boxed{P(Y = k | X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}}$ Softmax function



Cat	**3.2**
Car	5.1
Bird	-1.7

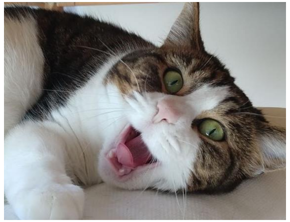
Maximize probability of correct class:

$$L_i = -\log P(Y = y_i | X = x_i) = -\log \frac{e^{s_{y_i}}}{\sum_j e^{s_j}}$$

Q1: What is min/max of this loss?

Multinomial Logistic Regression

- Again: $s = f(x_i, W)$
- Then: $\boxed{P(Y = k|X = x_i) = \frac{e^{s_k}}{\sum_j e^{s_j}}}$ Softmax function



Cat	**3.2**
Car	5.1
Bird	-1.7

Maximize probability of correct class:

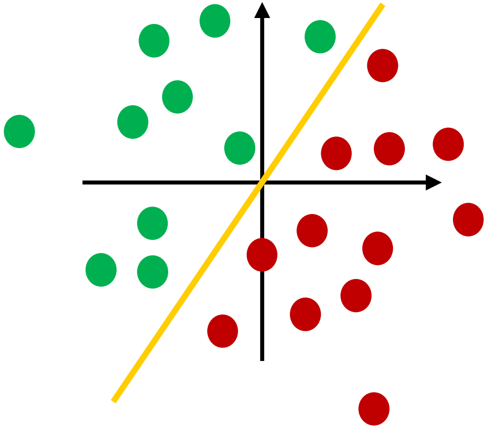
$$L_i = -\log P(Y = y_i|X = x_i) = -\log \frac{e^{s_{y_i}}}{\sum_j e^{s_j}}$$

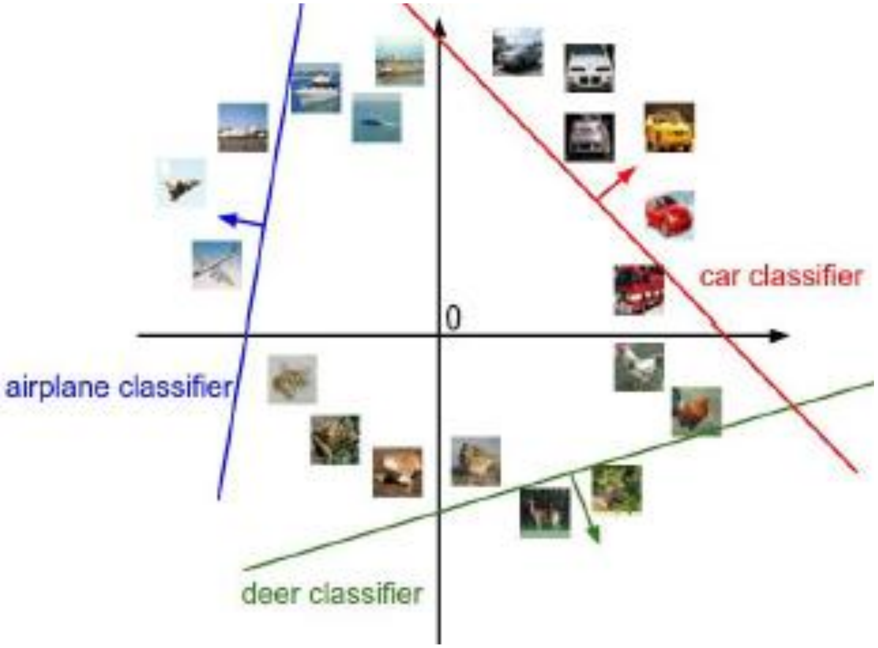
Q2: At initialization, all scores are similar and close to zero?

Decision Boundaries of Linear Regression

- Scores: $s = f(x_i, W) = Wx_i + b$

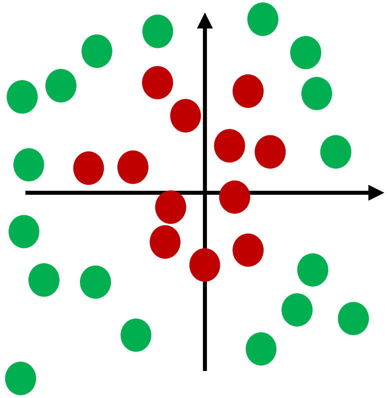
- SVM loss: “Maximum margin” at decision boundary



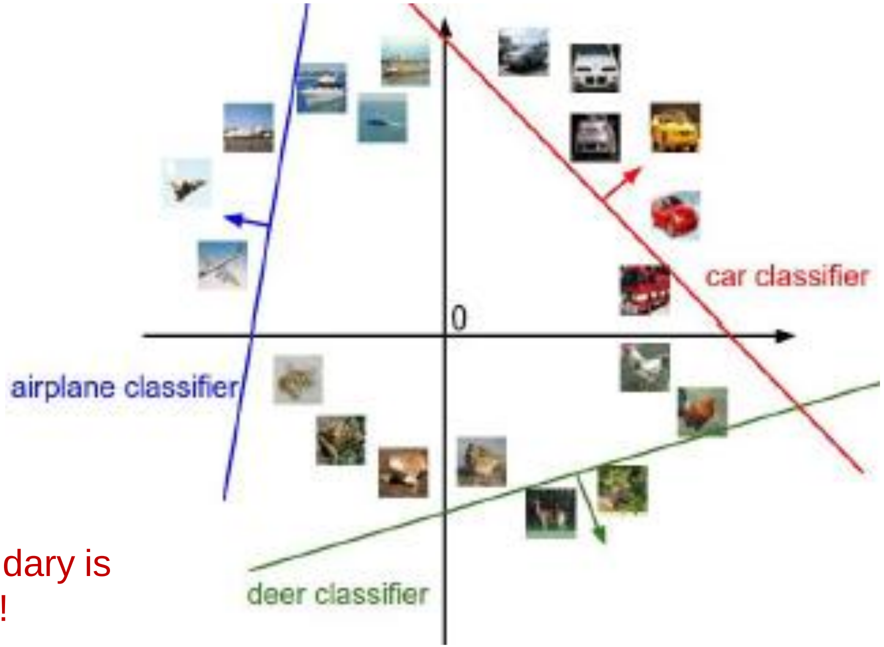


Decision Boundaries of Linear Regression

- Scores: $s = f(x_i, W) = Wx_i + b$
- SVM loss: “Maximum margin” at decision boundary



Required decision boundary is clearly non-linear!



Q: Can I solve this with Logistic Regression?

Hard Cases for Linear Classification

Class 1:
First and third quadrants

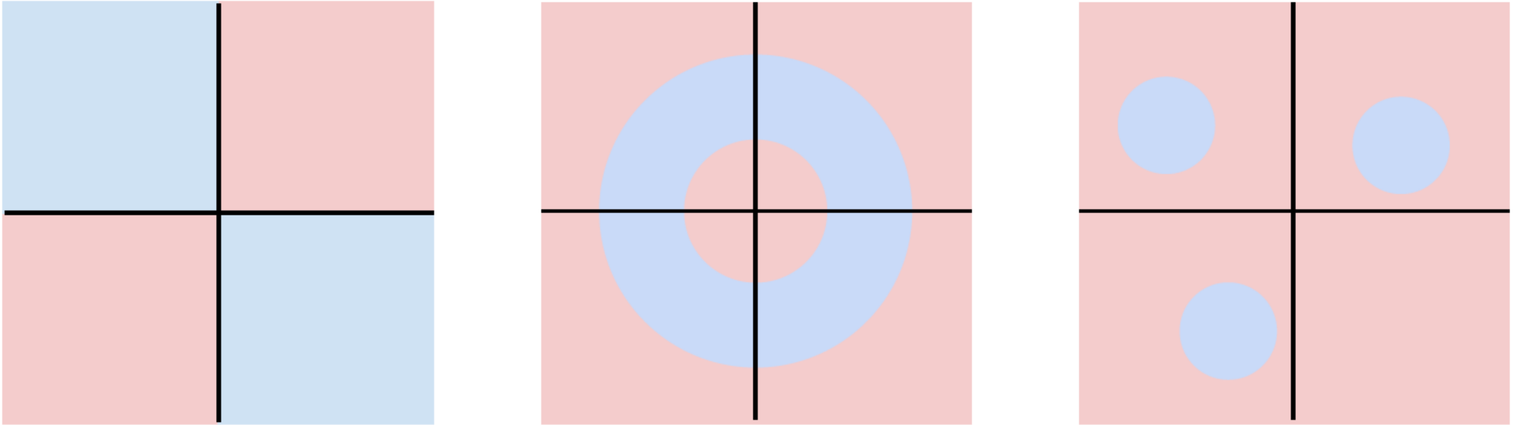
Class 2:
Second and fourth quadrants

Class 1:
1 <= L2 norm <= 2

Class 2:
Everything else

Class 1:
Three modes

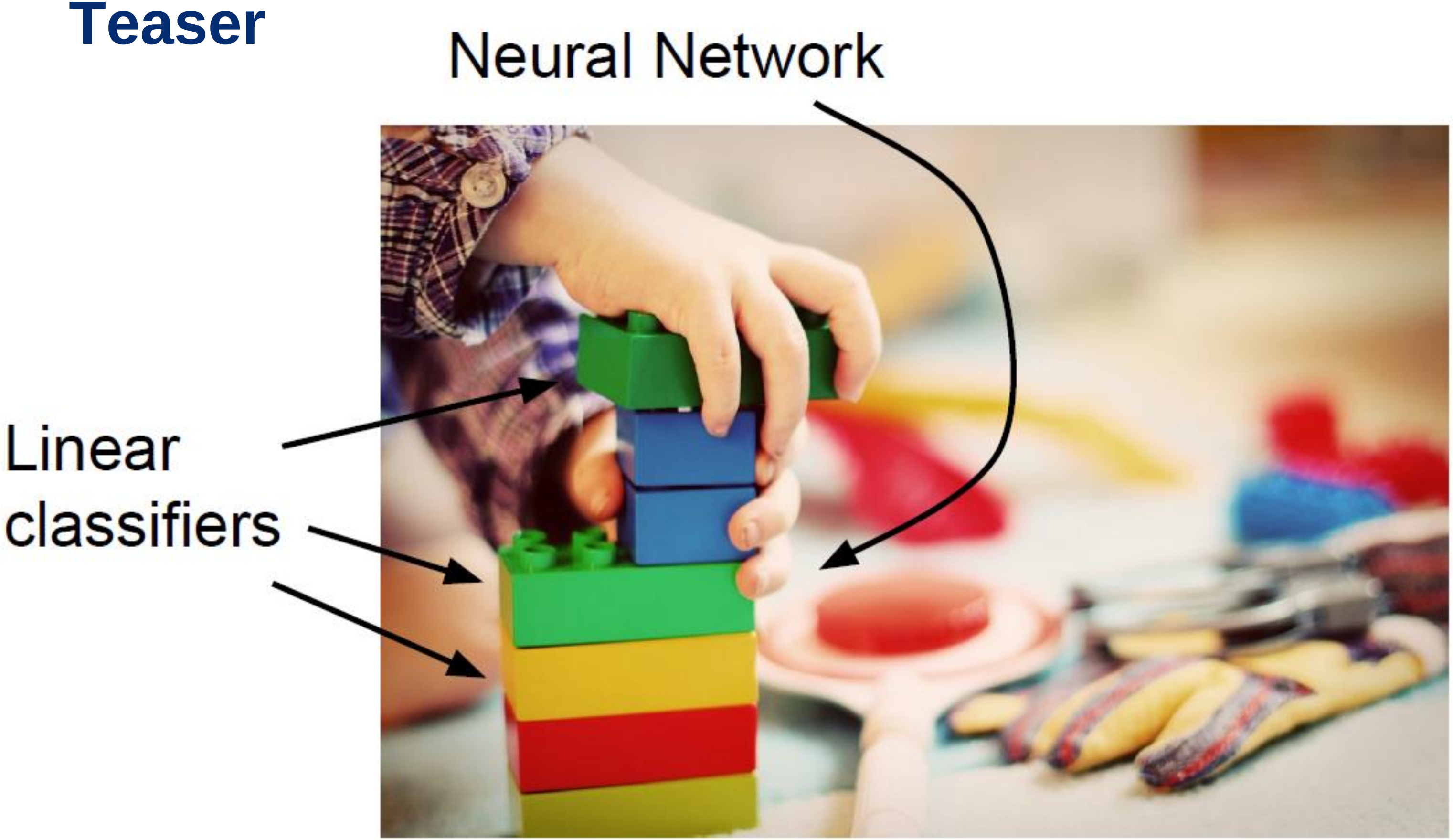
Class 2:
Everything else



- Clearly, the two classes are not linearly separable!

Images from Stanford cs231n.

Teaser



Next lecture:
- **Regularization**
- **Optimization**

Images from Stanford cs231n.

Image Features, Regression, and Classification

Questions?