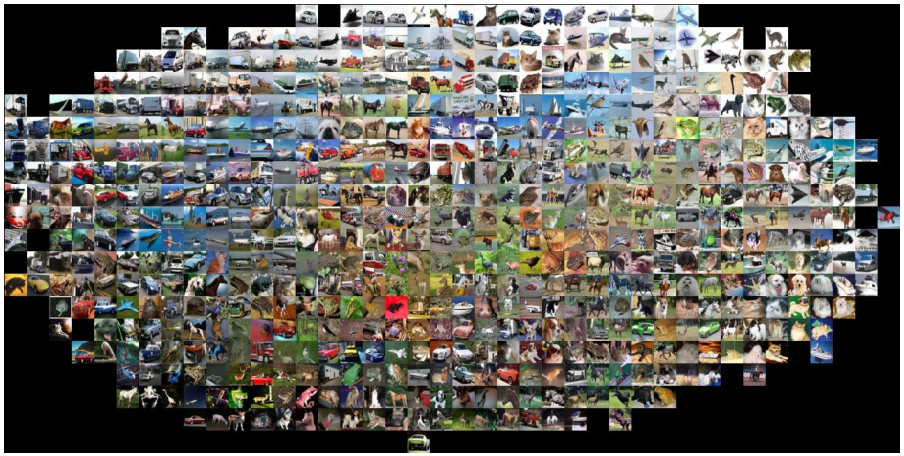
CS4670/5670: Computer Vision

Kavita Bala

Lecture 27: Recognition Basics



Slides from Andrej Karpathy and Fei-Fei Li
http://vision.stanford.edu/teaching/cs231n/

Announcements

- PA 3 Artifact voting

  Vote by Tuesday night

Today

- Image classification pipeline
- Training, validation, testing
- Score function and loss function

- Building up to CNNs for learning
  - 5-6 lectures on deep learning

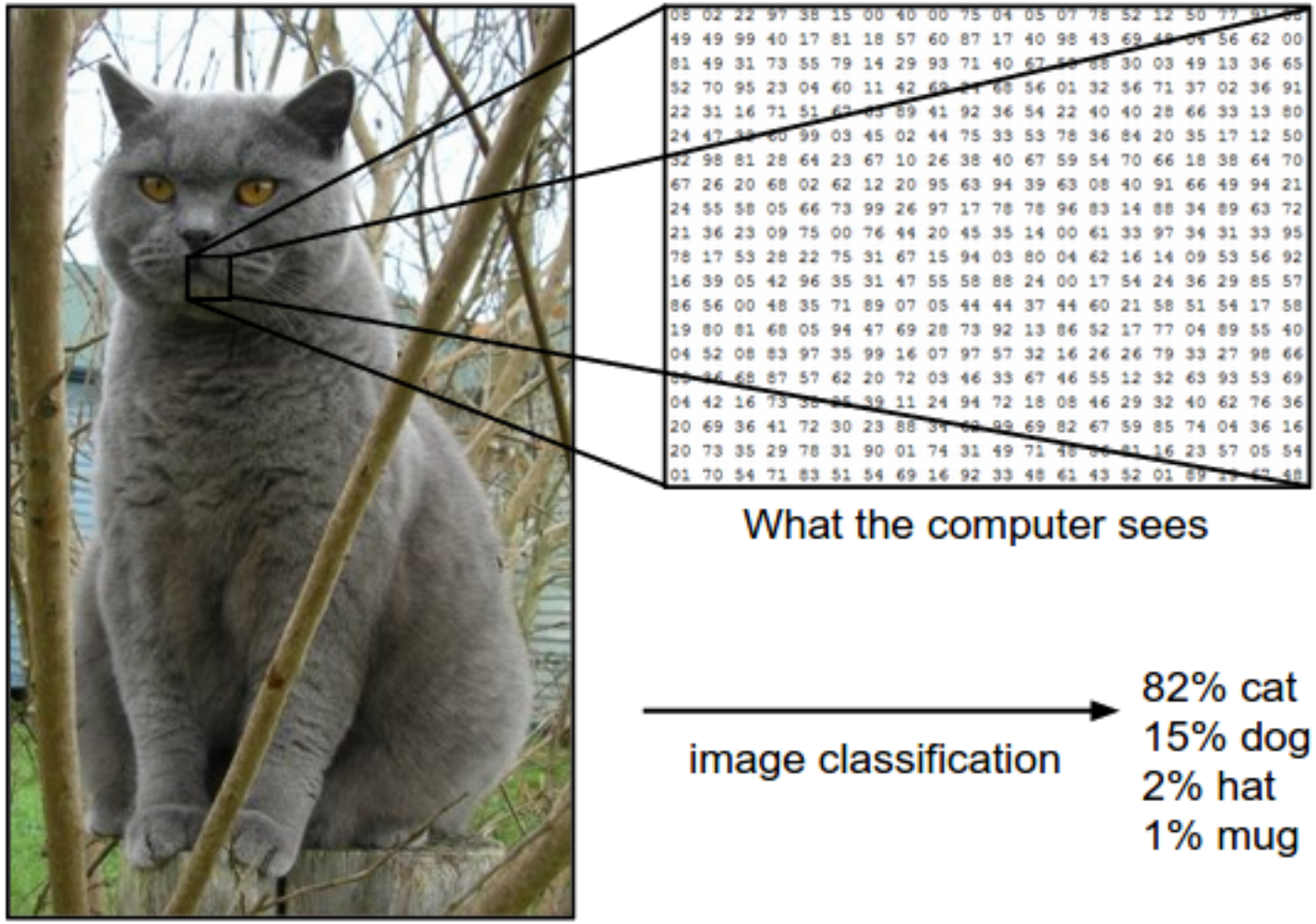
Image Classification

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

cat

Slides from Andrej Karpathy and Fei-Fei Li
http://vision.stanford.edu/teaching/cs231n/

Image Classification: Problem



Data-driven approach

- Collect a database of images with labels
- Use ML to train an image classifier
- Evaluate the classifier on test images

Example training set

Slides from Andrej Karpathy and Fei-Fei Li
http://vision.stanford.edu/teaching/cs231n/

Data-driven approach

- Collect a database of images with labels
- Use ML to train an image classifier
- Evaluate the classifier on test images

```
def train(train_images, train_labels):
    # build a model of images -> labels

def predict(image):
    # evaluate the model on the image
    return class_label
```

Slides from Andrej Karpathy and Fei-Fei Li
http://vision.stanford.edu/teaching/cs231n/

Train and Test

- Split dataset between training images and test images

- Be careful about inflation of results

Classifiers

- Nearest Neighbor
- kNN

- SVM
- …

Nearest Neighbor Classifier

- Train
  - Remember all training images and their labels
- Predict
  - Find the closest (most similar) training image
  - Predict its label as the true label

How to find the most similar training image? What is the distance metric?

L1 distance: $d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$

Where I_1 denotes image 1, and p denotes each pixel

test image

56	32	10	18
90	23	128	133
24	26	178	200
2	0	255	220

-

training image

10	20	24	17
8	10	89	100
12	16	178	170
4	32	233	112

=

pixel-wise absolute value differences

46	12	14	1
82	13	39	33
12	10	0	30
2	32	22	108

→ 456

Slides from Andrej Karpathy and Fei-Fei Li
http://vision.stanford.edu/teaching/cs231n/

Choice of distance metric

- Hyperparameter

L1 (Manhattan) distance

$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$

L2 (Euclidean) distance

$$d_2(I_1, I_2) = \sqrt{\sum_p \left(I_1^p - I_2^p\right)^2}$$

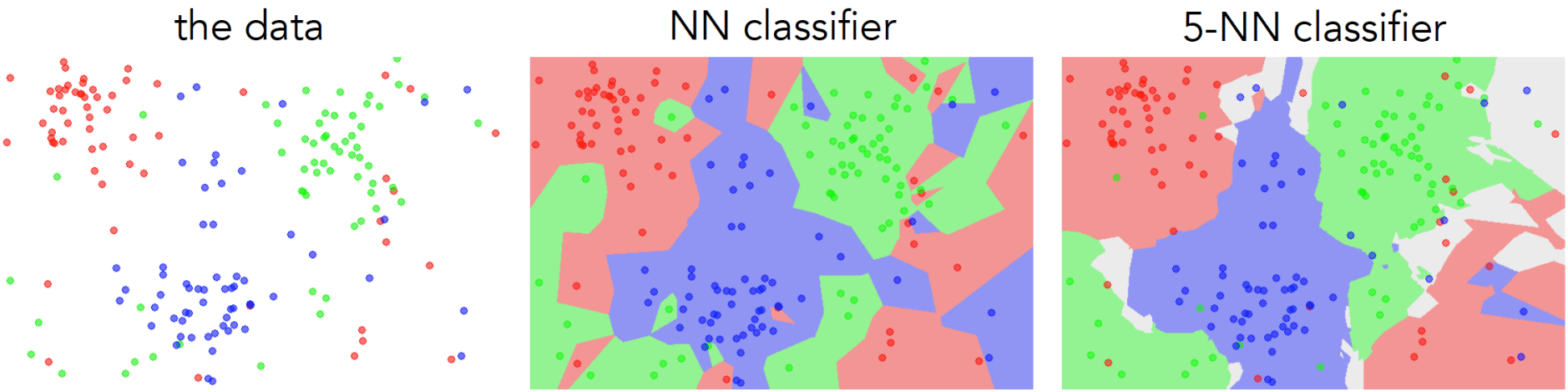
- Two most commonly used special cases of p-norm

$$||x||_p = \left(|x_1|^p + \cdots + |x_n|^p\right)^{\frac{1}{p}} \qquad p \geq 1, x \in \mathbb{R}^n$$

Slides from Andrej Karpathy and Fei-Fei Li
http://vision.stanford.edu/teaching/cs231n/

k-nearest neighbor

- Find the k closest points from training data
- Labels of the k points “vote” to classify

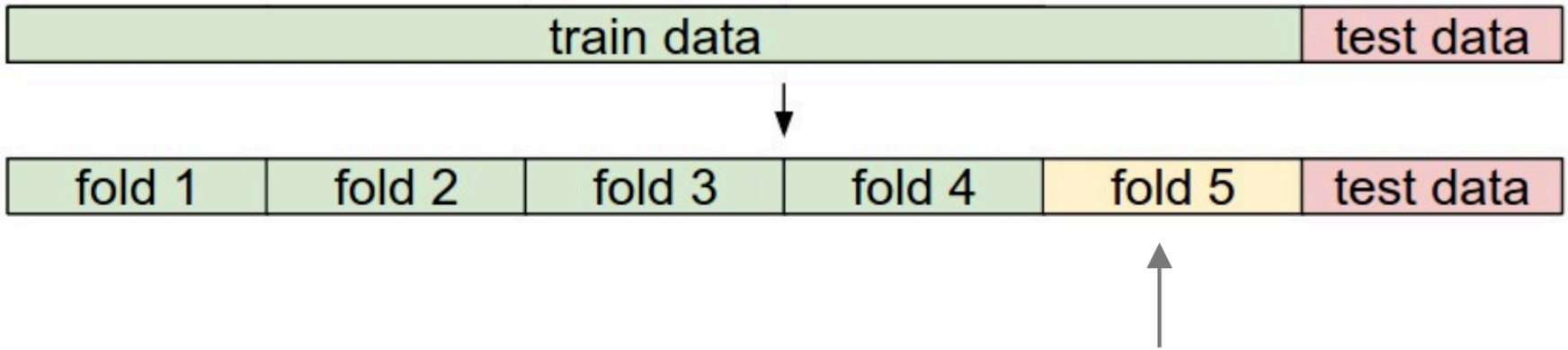


How to pick hyperparameters?

- Methodology
  - Train and test
  - Train, validate, test

- Train for original model
- Validate to find hyperparameters
- Test to understand generalizability

Validation



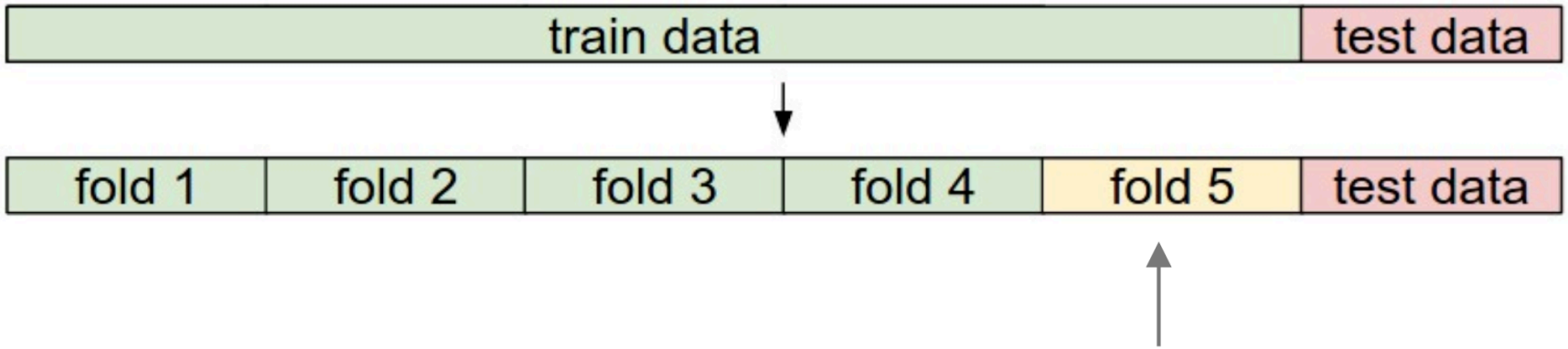
Validation data

use to tune hyperparameters

evaluate on test set ONCE at the end

Slides from Andrej Karpathy and Fei-Fei Li
http://vision.stanford.edu/teaching/cs231n/

Cross-validation

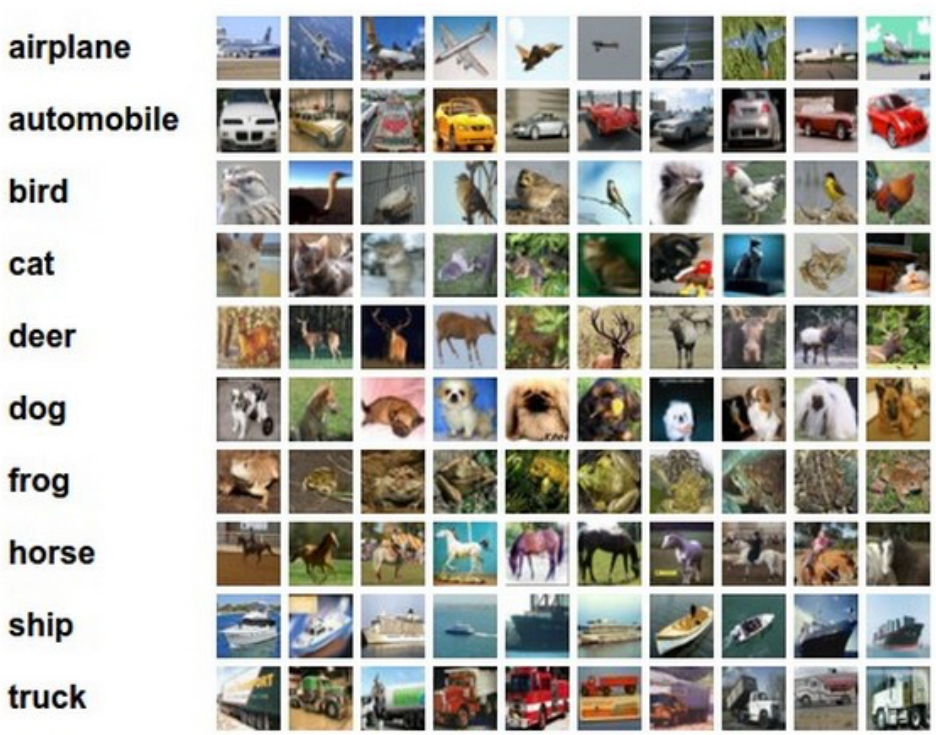


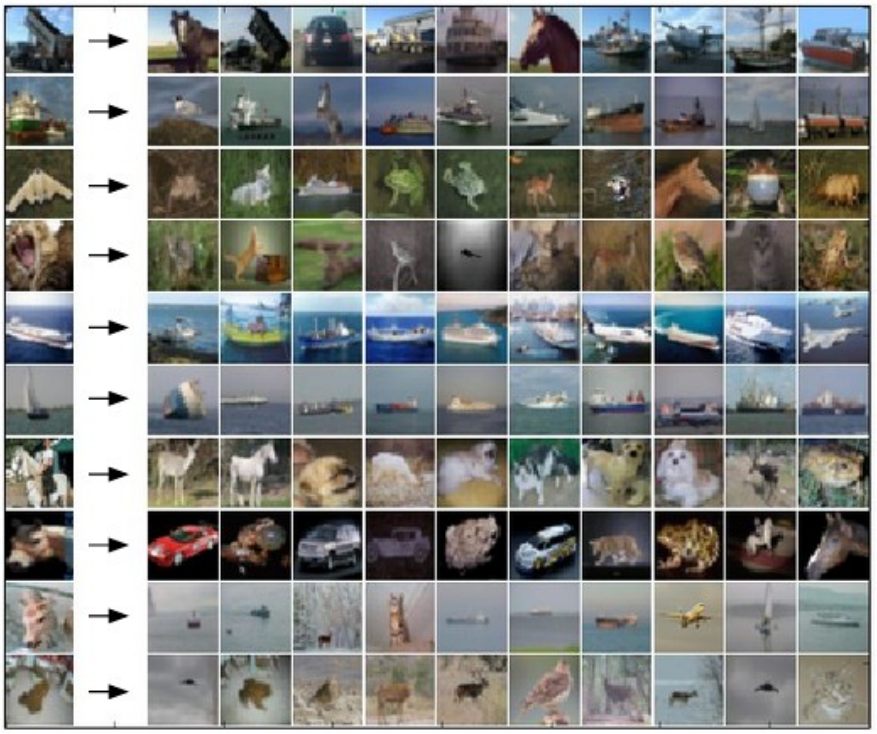
Cross-validation

cycle through the choice of which fold is the validation fold, average results.

Slides from Andrej Karpathy and Fei-Fei Li
http://vision.stanford.edu/teaching/cs231n/

CIFAR-10 and NN results

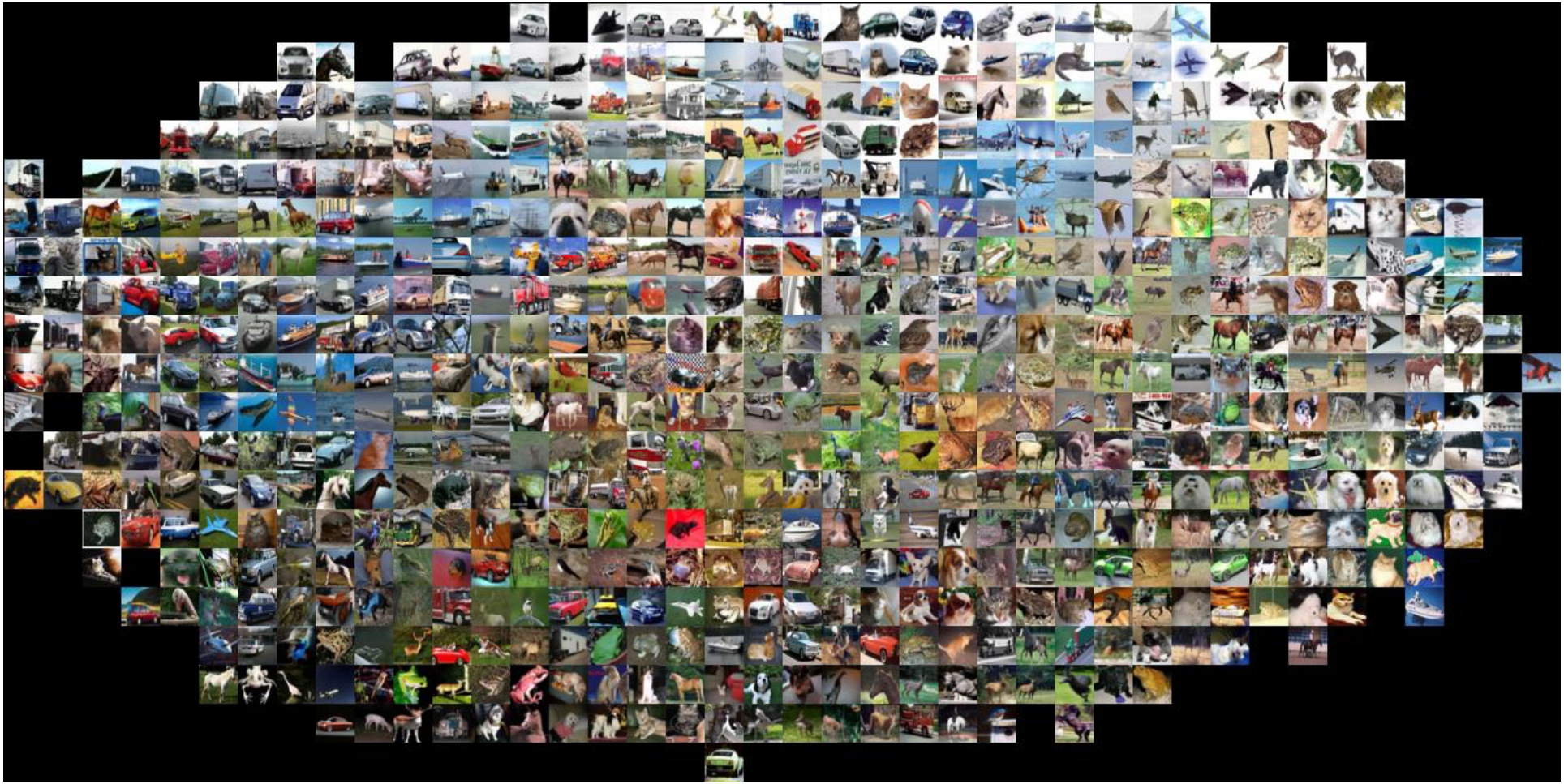




Example dataset: CIFAR-10
10 labels
50,000 training images
10,000 test images.

Slides from Andrej Karpathy and Fei-Fei Li
http://vision.stanford.edu/teaching/cs231n/

Visualization: L2 distance



Complexity and Storage

- N training images, M testing images

- Training: O(1)
- Testing: O(MN)

- Hmm
  - Normally need the opposite
  - Slow training (ok), fast testing (necessary)

Summary

- Data-driven: Train, validate, test
  - Need labeled data
- Classifier
  - Nearest neighbor, kNN (approximate NN, ANN)

Score function

class scores

Slides from Andrej Karpathy and Fei-Fei Li
http://vision.stanford.edu/teaching/cs231n/

Linear Classifier

define a **score function**

data (image)

$$f(x_i, W, b) = Wx_i + b$$

class scores

"weights"

"bias vector"

"parameters"

Linear Classifier

(assume CIFAR-10 example so
32 x 32 x 3 images, 10 classes)

data (image)
[3072 x 1]

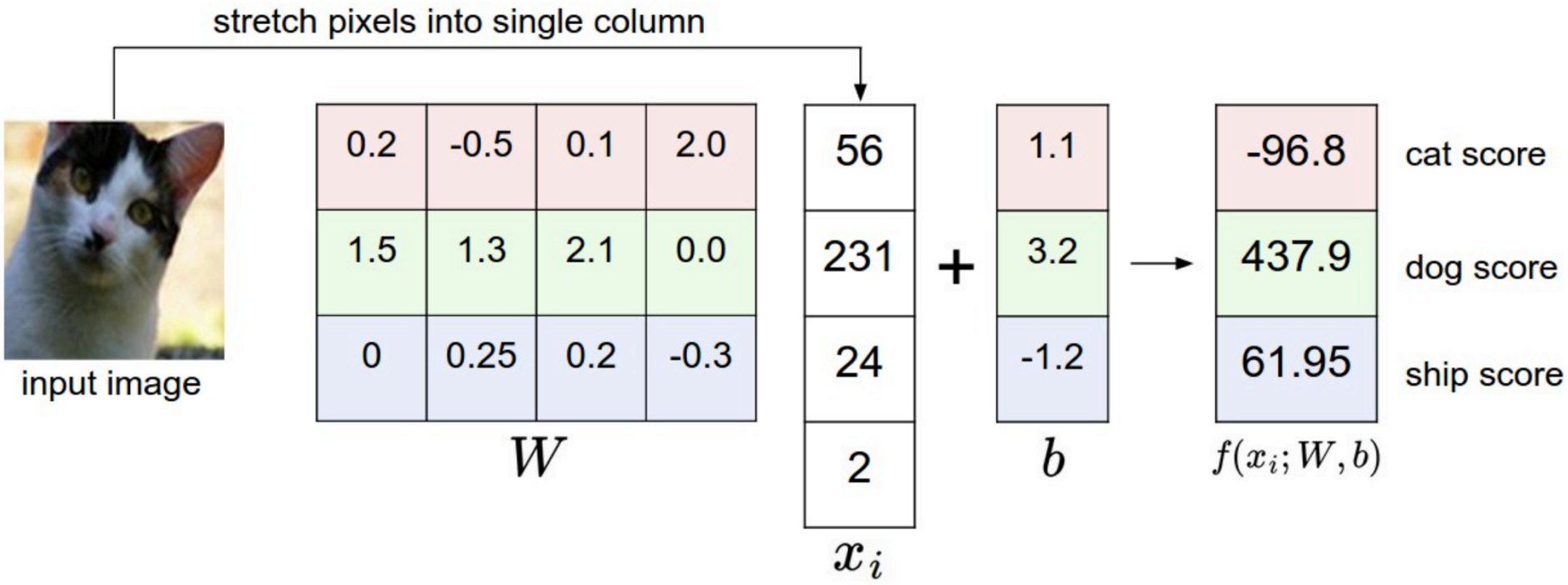
$$f(x_i, W, b) = Wx_i + b$$

class scores
[10 x 1]

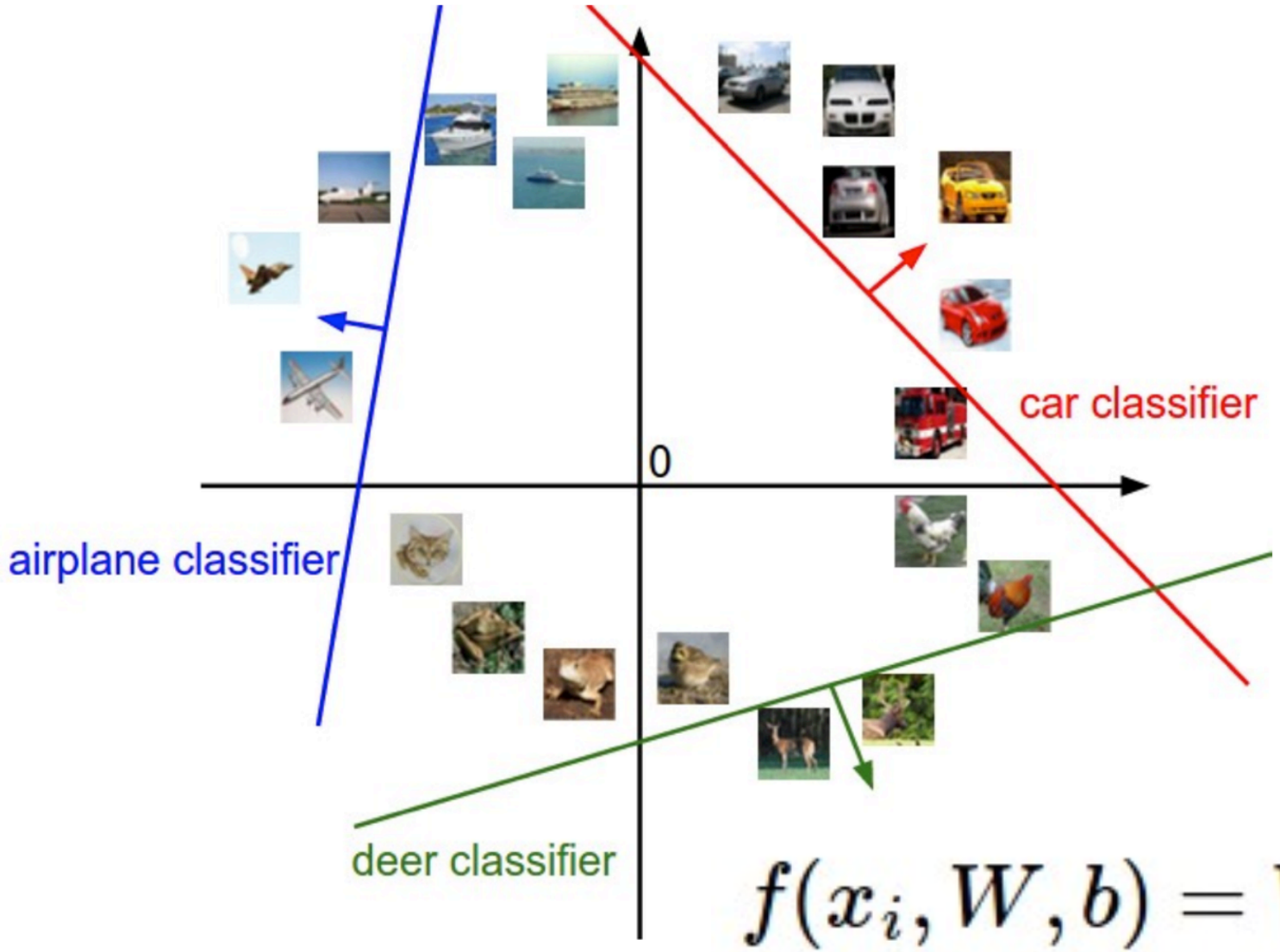
weights
[10 x 3072]

bias vector
[10 x 1]

Computing scores



Geometric Interpretation



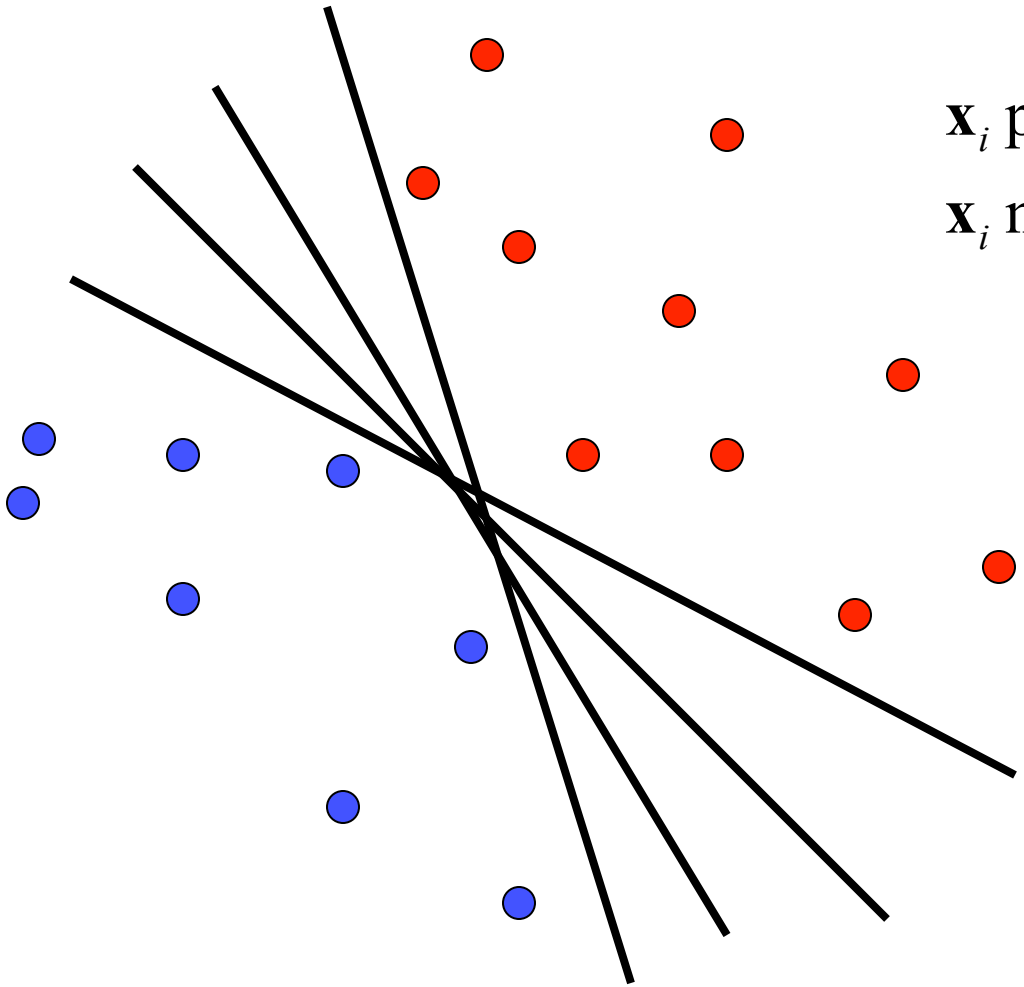
$$f(x_i, W, b) = Wx_i + b$$

Interpretation: Template matching

$$f(x_i, W, b) = Wx_i + b$$

Linear classifiers

- Find linear function (*hyperplane*) to separate positive and negative examples



$$\mathbf{x}_i \text{ positive}: \quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 0$$

$$\mathbf{x}_i \text{ negative}: \quad \mathbf{x}_i \cdot \mathbf{w} + b < 0$$

Which hyperplane is best?

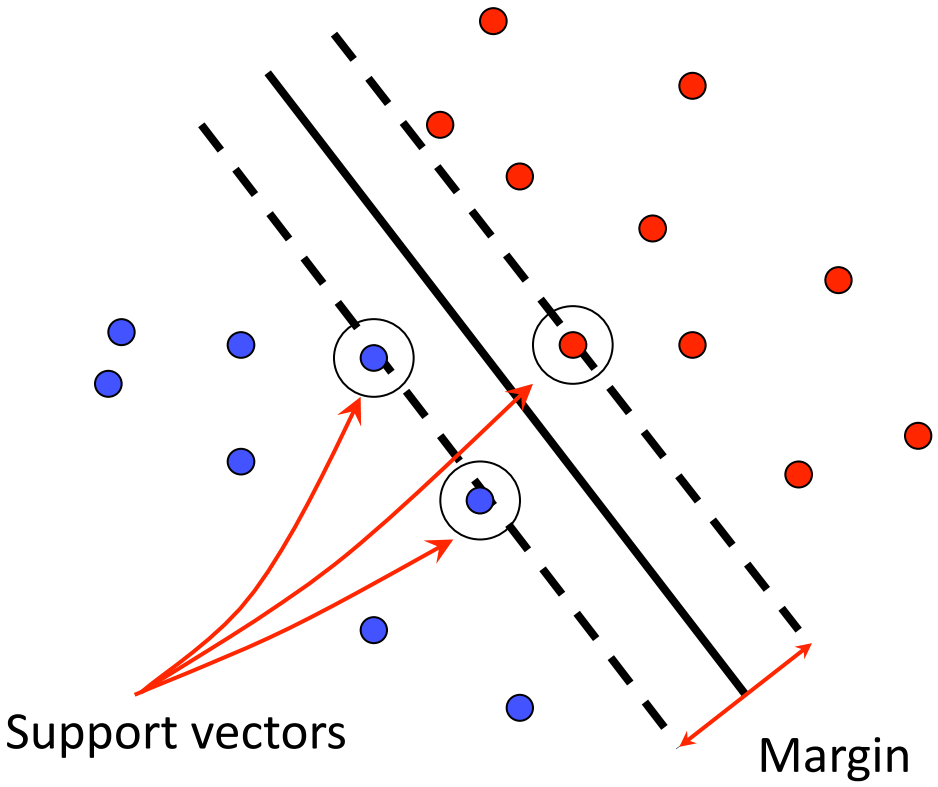
Support vector machines

- Find hyperplane that maximizes the *margin* between the positive and negative examples

C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998

Support vector machines

- Find hyperplane that maximizes the *margin* between the positive and negative examples



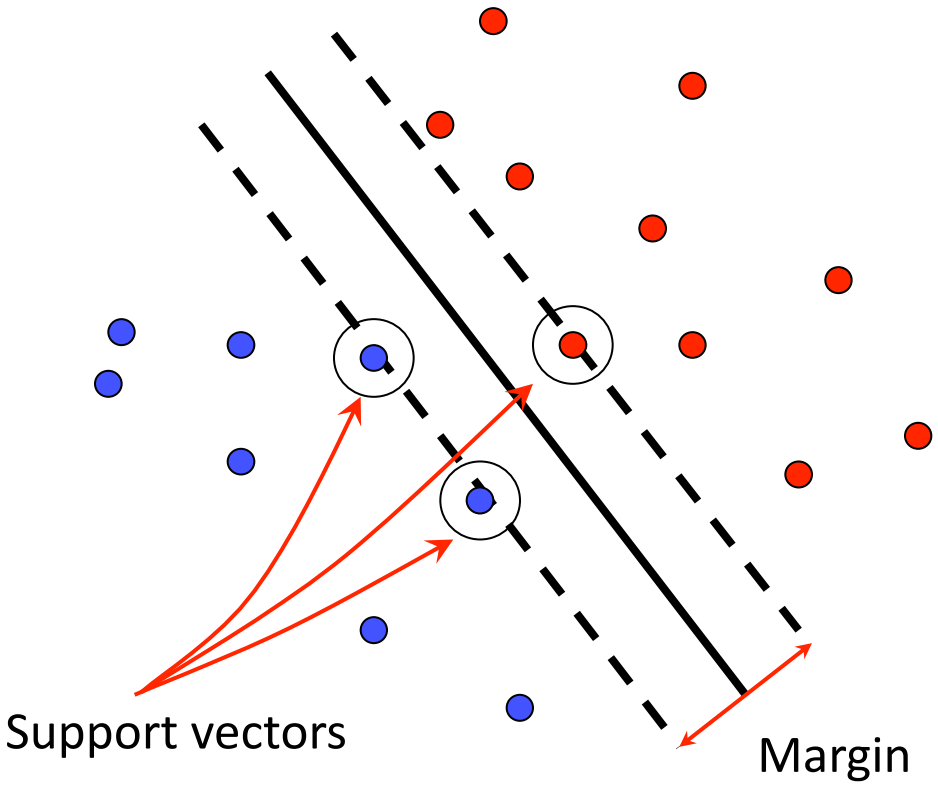
$$\mathbf{x}_i \text{ positive } (y_i = 1): \quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 1$$

$$\mathbf{x}_i \text{ negative } (y_i = -1): \quad \mathbf{x}_i \cdot \mathbf{w} + b \leq -1$$

For support, vectors, $\mathbf{x}_i \cdot \mathbf{w} + b = \pm 1$

Support vector machines

- Find hyperplane that maximizes the *margin* between the positive and negative examples



$\mathbf{x}_i$ positive $(y_i = 1)$: $\mathbf{x}_i \cdot \mathbf{w} + b \geq 1$

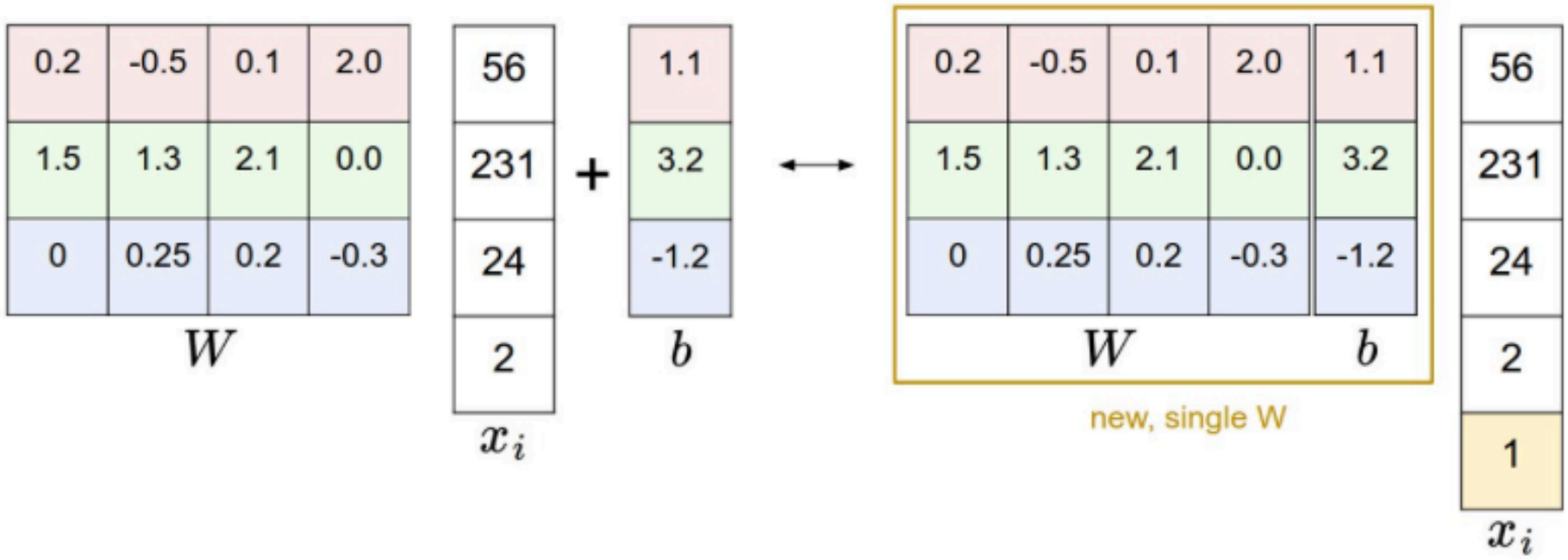
$\mathbf{x}_i$ negative $(y_i = -1)$: $\mathbf{x}_i \cdot \mathbf{w} + b \leq -1$

For support, vectors, $\mathbf{x}_i \cdot \mathbf{w} + b = \pm 1$

Distance between point and hyperplane: $\dfrac{|\mathbf{x}_i \cdot \mathbf{w} + b|}{\|\mathbf{w}\|}$

Therefore, the margin is $2 / \|\mathbf{w}\|$

Bias Trick


0.2 -0.5 0.1 2.0
1.5 1.3 2.1 0.0
0 0.25 0.2 -0.3
W
56
231
24
2
x_i
+
1.1
3.2
-1.2
b
0.2 -0.5 0.1 2.0 1.1
1.5 1.3 2.1 0.0 3.2
0 0.25 0.2 -0.3 -1.2
W b
new, single W
56
231
24
2
1
x_i