ROB 430/599: Deep Learning for Robot Perception and Manipulation (DeepRob)

Lecture 3: Linear Classifier

01/14/2026

ROBOTICS

Today

- Feedback and Recap (5min)
- Linear Classifiers
  - Interpreting a linear classifier - three viewpoints (15min)
  - Softmax: Cross-Entropy Loss (25min)
  - Multi-class SVM loss (25min)
- Summary and Takeaways (5min)

Aha Slides
(In-class participation)

https://ahaslides.com/YAFWJ

Q0: Feedback/Questions so far?

Recap: Image Classification
- A Core Computer Vision/Robot Perception Task

Input: image

Output: assign image to one of a fixed set of categories

Chocolate Pretzels

Granola Bar

Potato Chips

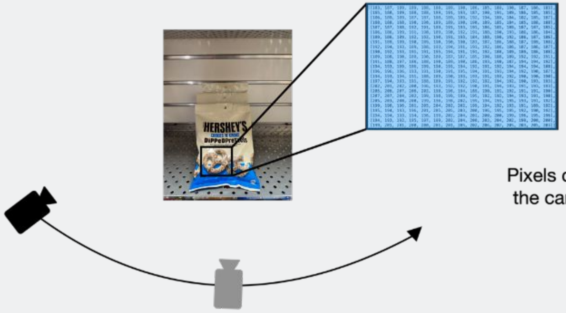
Water Bottle

Popcorn

ROBOTICS

Recap: Image Classification Challenges

Viewpoint Variation & Semantic Gap



Pixels change when
the camera moves

Illumination Changes

Milk Chocolate | White Chocolate | Cookies N' Creme | Peanut Butter | Ambiguous Category

Intraclass Variation

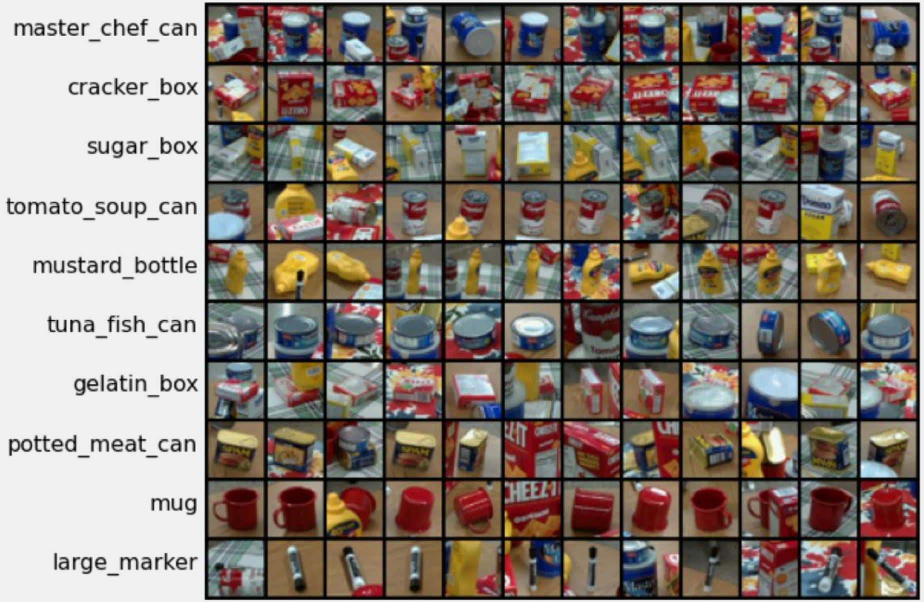
Recap:
Machine (Deep) Learning - A Data-Driven Approach

1. Collect a dataset of images and labels
2. Use Machine Learning to train a classifier
3. Evaluate the classifier on new images

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

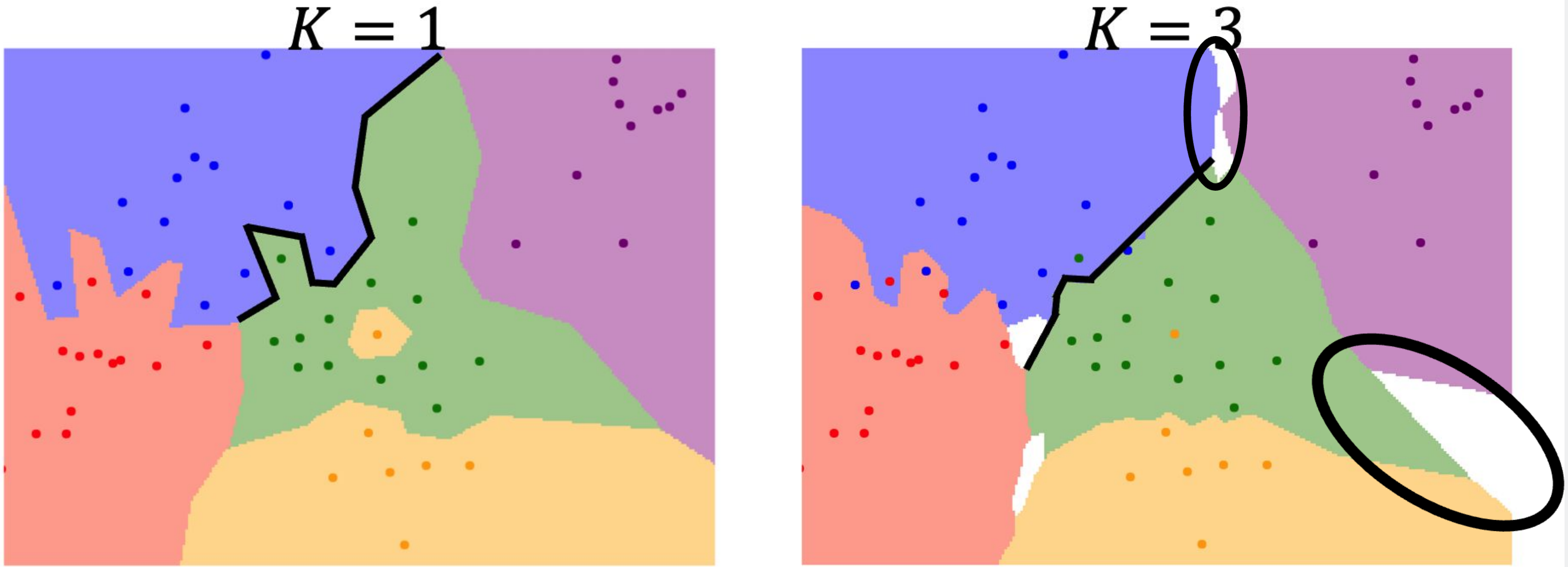
```
def predict(model, test_images):
  # Use model to predict labels
  return test_labels
```

Example training set



ROBOTICS

Recap: KNN parameters, train/val/test



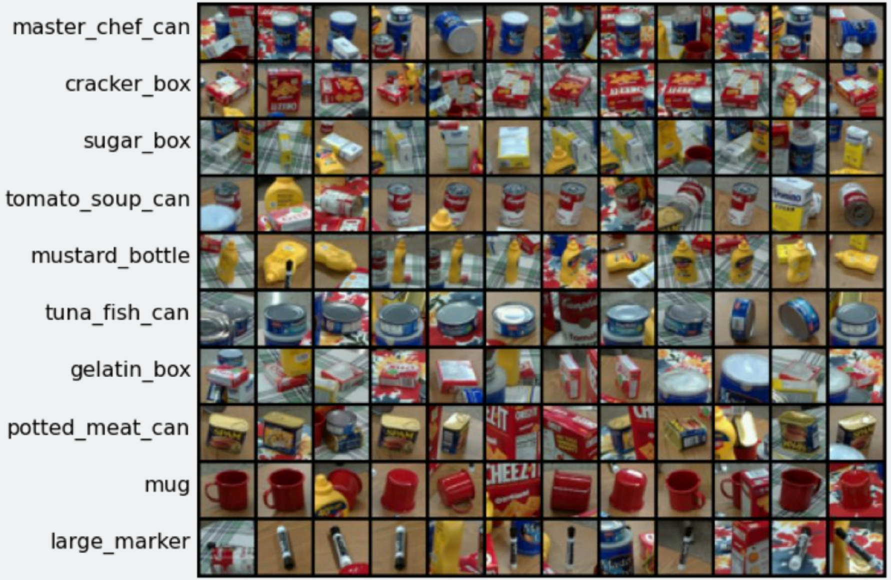
Using more neighbors helps smooth out rough decision boundaries

Linear Classifiers

Linear Classifier
- Building Block of Neural Networks

Recall: PROPS dataset

Progress **R**obot **O**bject **P**erception **S**amples **D**ataset



Chen et al., “ProgressLabeller: Visual Data Stream Annotation for Training Object-Centric 3D Perception”, IROS, 2022.

10 classes
32x32 RGB images
50k training images (5k per class)
10k test images (1k per class)

Parametric Approach

Image

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

→ $f(x, W)$ →

10 numbers giving class scores

↑

W
parameters or weights

Parametric Approach

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

Image

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

$f(\mathbf{x}, \mathbf{W})$ → **10** numbers giving class scores

W
parameters or weights

Parametric Approach

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

(10,) (10, 3072) (3072,)

Image

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

→ f(**x**,**W**) →

10 numbers giving class scores

↑

W
parameters
or weights

Parametric Approach

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

$f(x,W)$: (10,) — W: (10, 3072) — x: (3072,) — b: (10,)

Image

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

→ f(x,W) → **10** numbers giving class scores

↑

W
parameters
or weights

Example for 2x2 Image, 3 classes (crackers/mug/sugar)

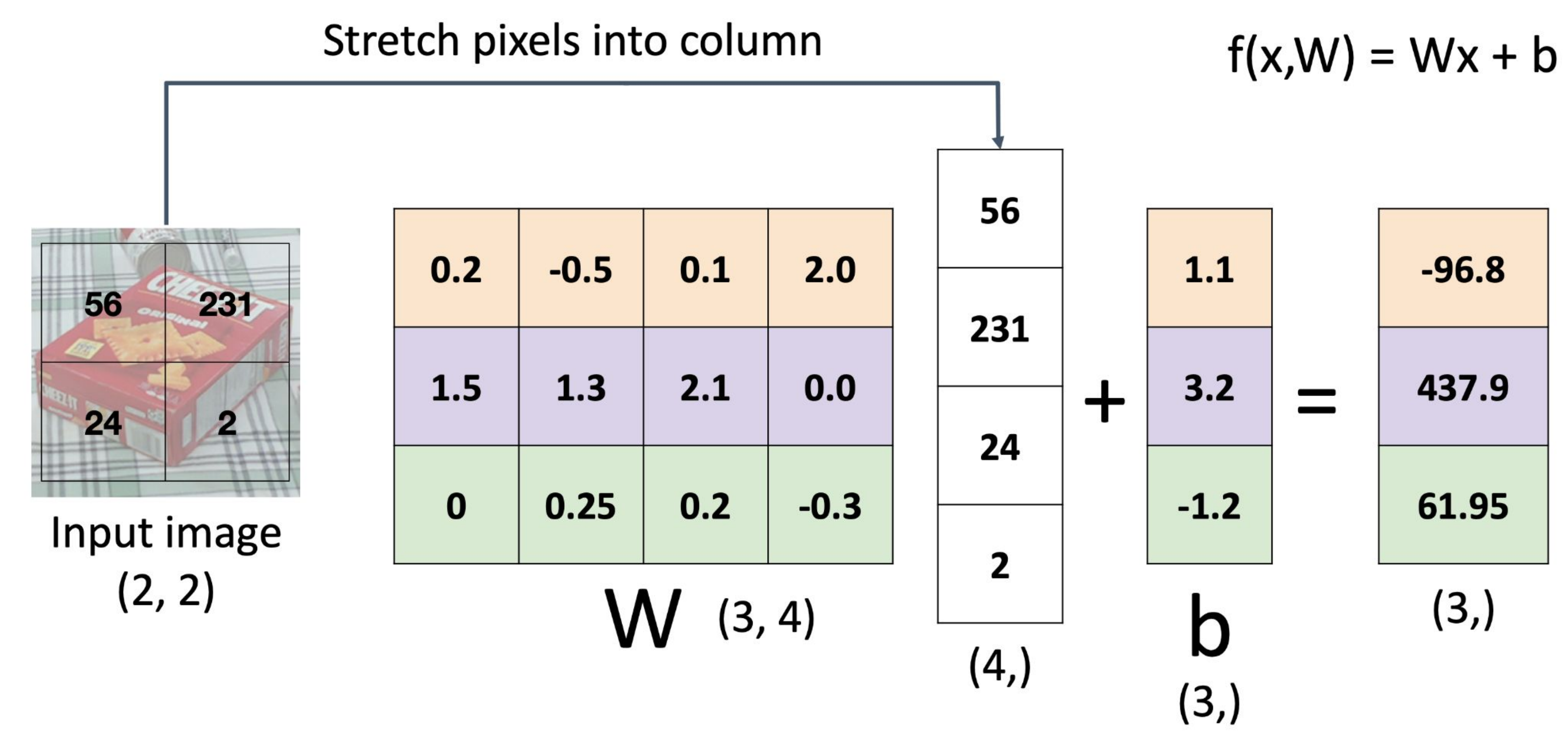
Stretch pixels into column

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

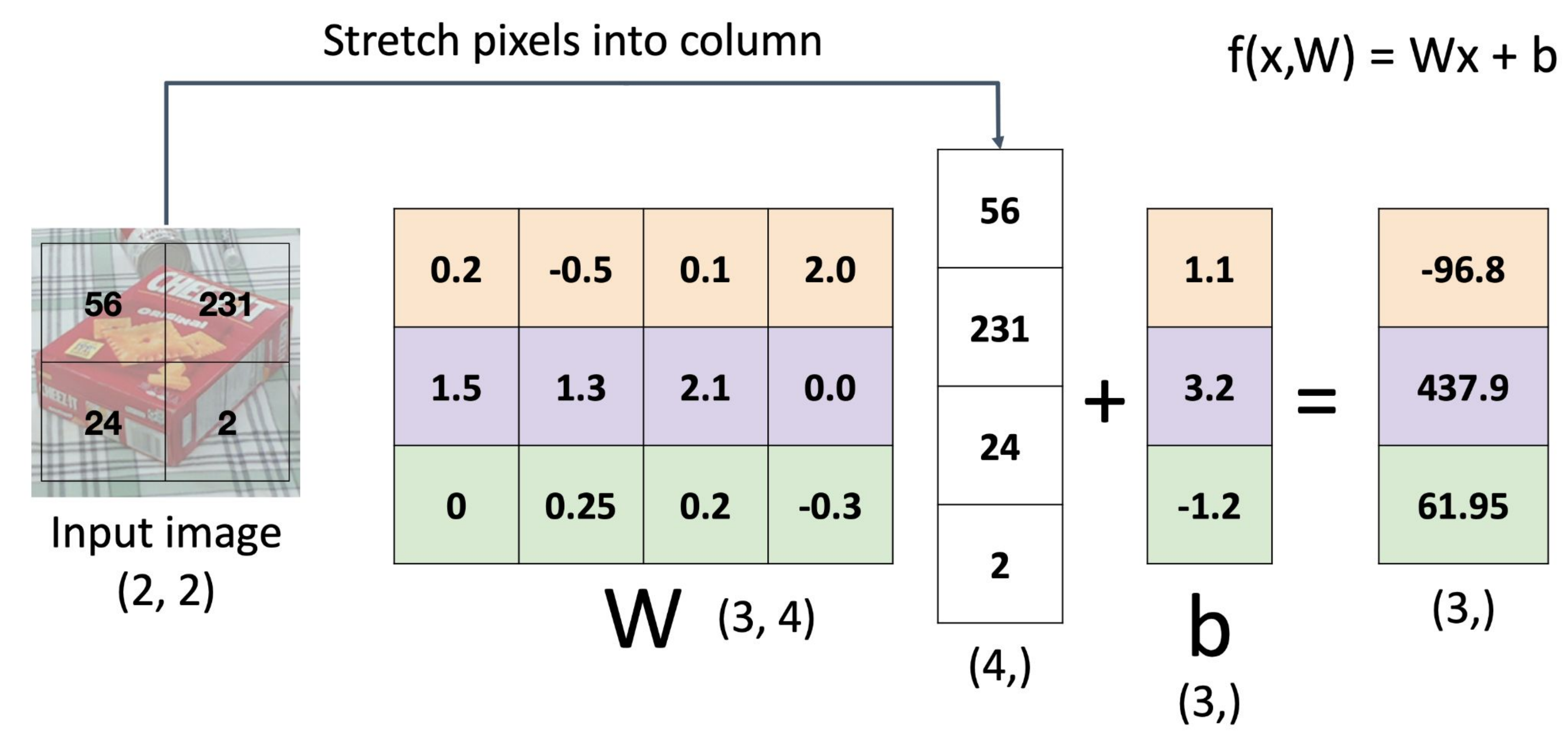
Input image
(2, 2)

(4,)

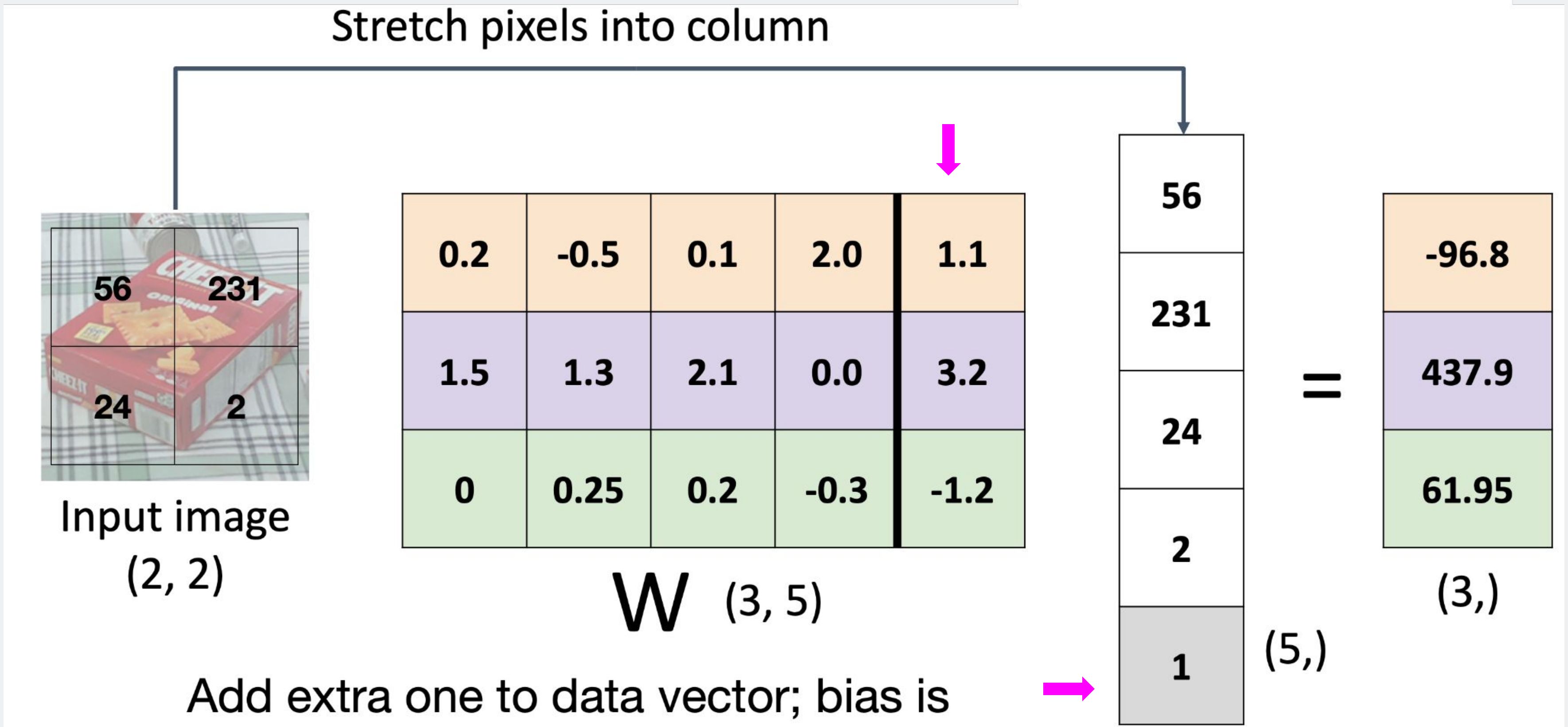
Example for 2x2 Image, 3 classes (crackers/mug/sugar)



① Algebraic Viewpoint



Linear Classifier - Bias Trick

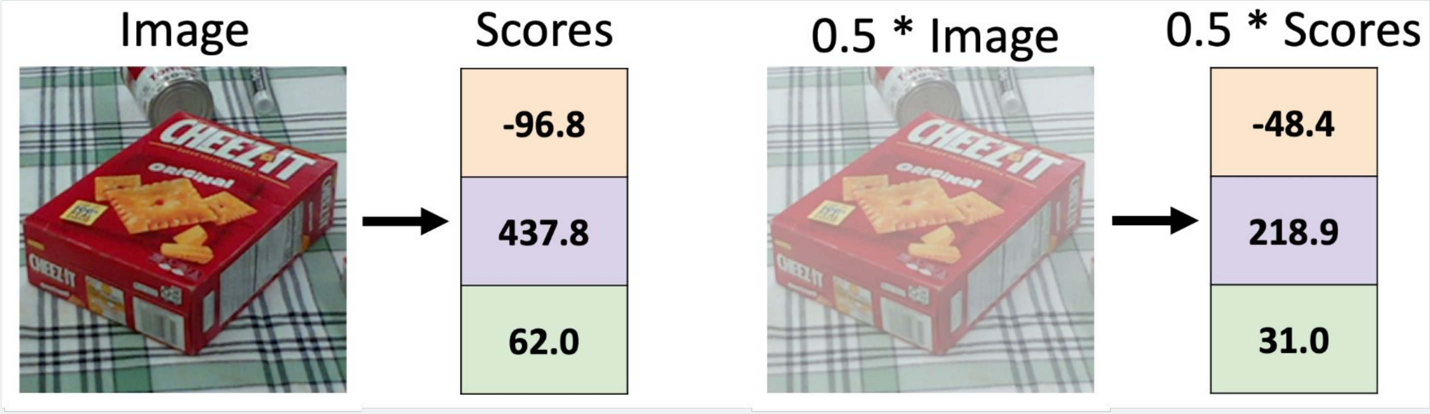


Add extra one to data vector; bias is absorbed into last column of weight matrix

Linear Classifier - Predictions are Linear

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

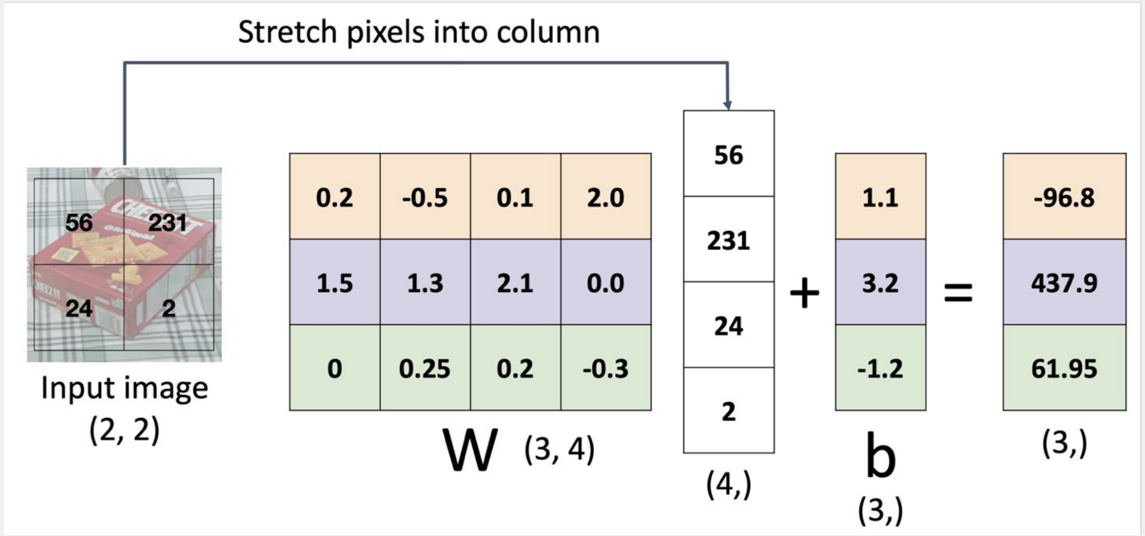
$f(cx, W) = W(cx) = c * f(x, W)$



Interpreting Linear Classifier

Algebraic Viewpoint

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

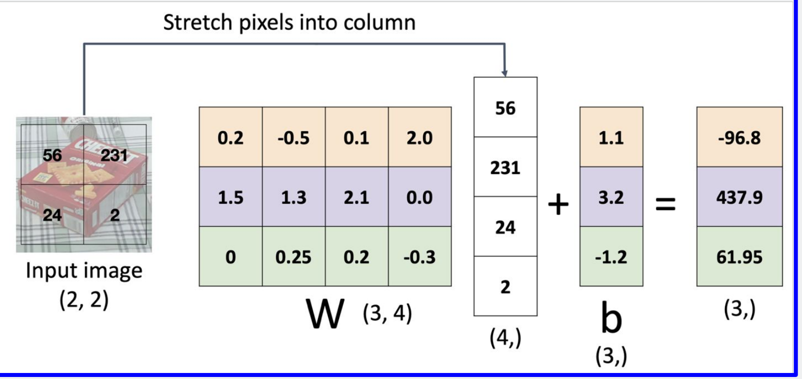


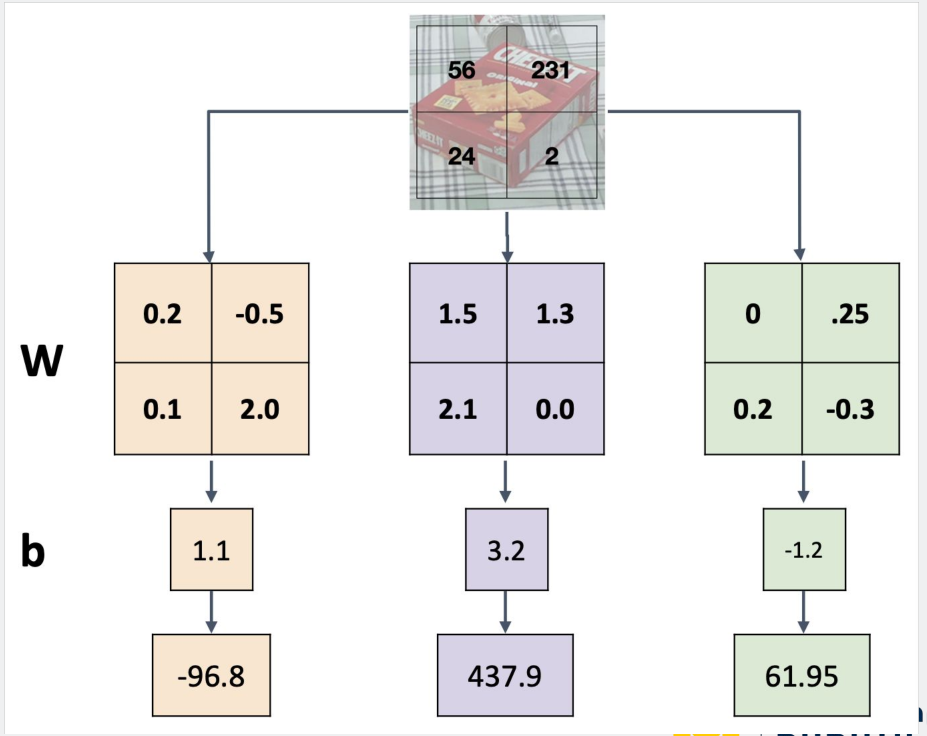
Interpreting Linear Classifier

Instead of stretching pixels into columns, we can equivalently stretch rows of W into images!

Algebraic Viewpoint

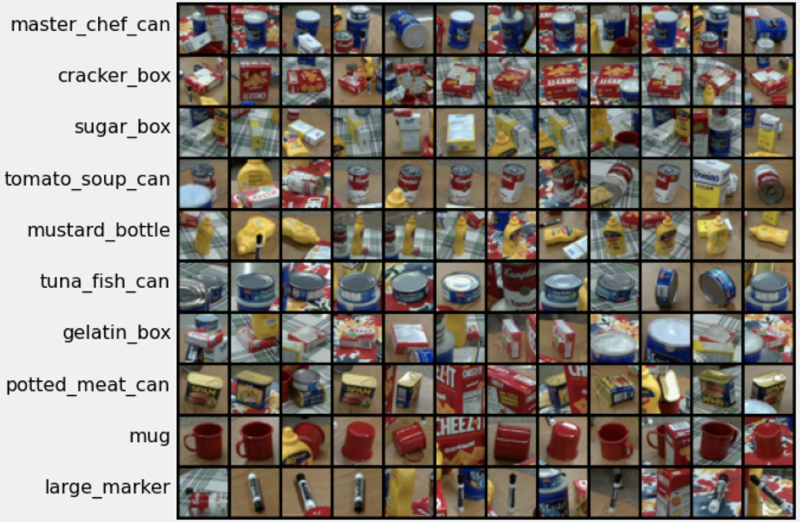
$$f(x,W) = Wx + b$$



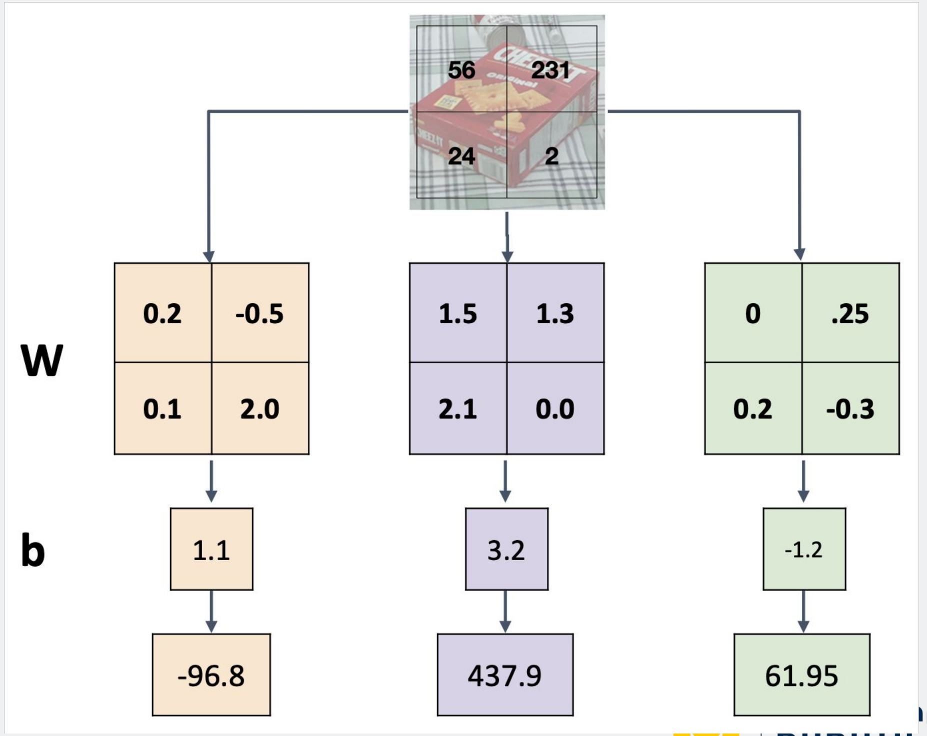


Interpreting Linear Classifier

(PROPS dataset)



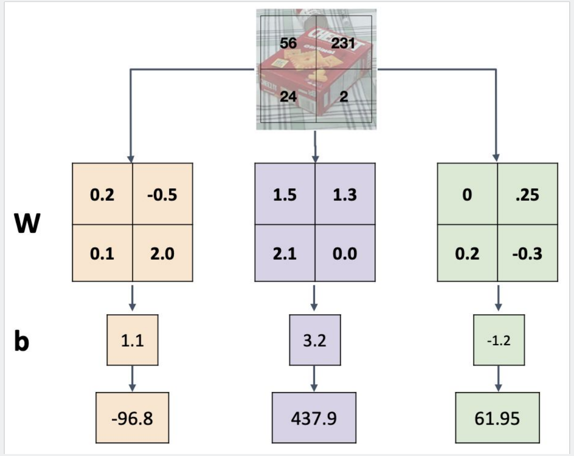
Instead of stretching pixels into columns, we can equivalently stretch rows of W into images!



Interpreting Linear Classifier

(PROPS dataset)

Instead of stretching pixels into columns, we can equivalently stretch rows of W into images!

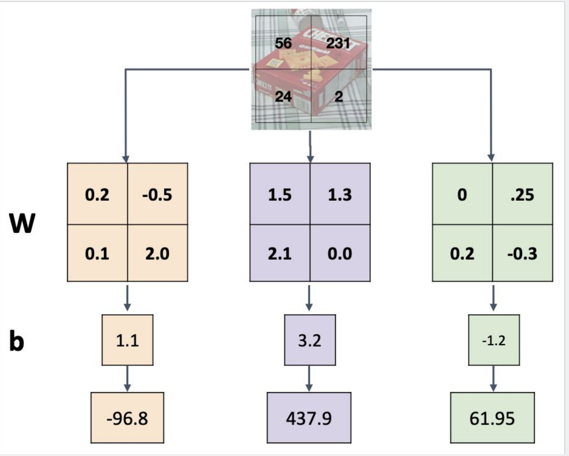


ROBOTICS

Interpreting a Linear Classifier
- ② Visual Viewpoint

Instead of stretching pixels into columns, we can equivalently stretch rows of W into images!

Linear classifier has one "template" per category

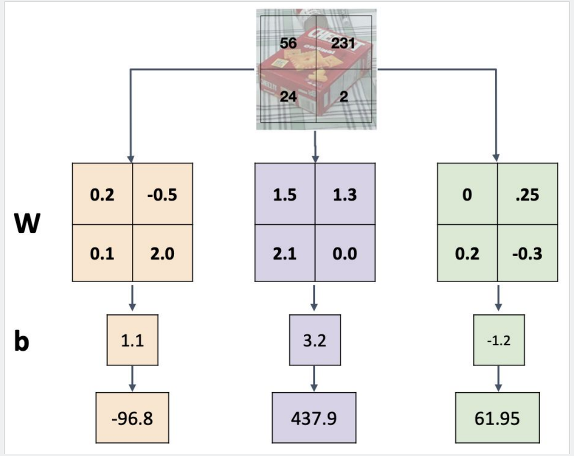


You can visualize W as a "template" pattern image

Interpreting a Linear Classifier
- Visual Viewpoint

Instead of stretching pixels into columns, we can equivalently stretch rows of W into images!

Linear classifier has one "template" per category

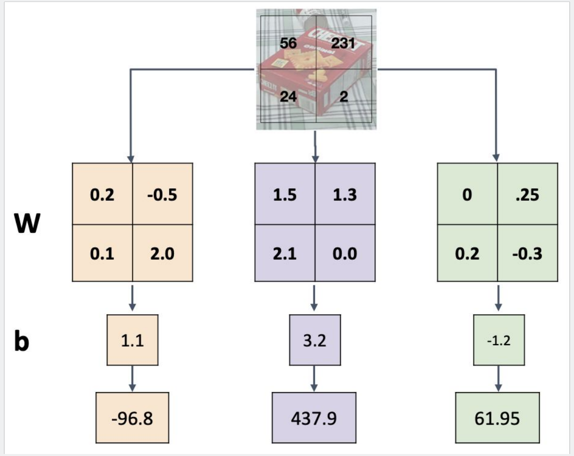


M | ROBOTICS

Interpreting a Linear Classifier
- Visual Viewpoint

Instead of stretching pixels into columns, we can equivalently stretch rows of W into images!

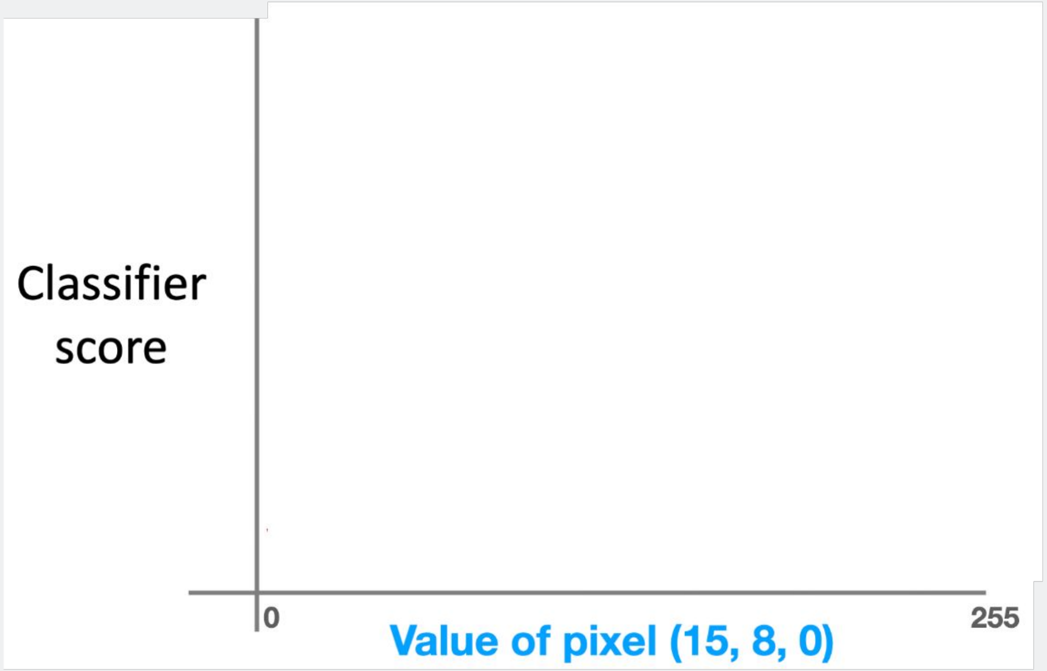
Linear classifier has one "template" per category



***Note:** A single template **cannot** capture multiple modes of the data
e.g., Rotation

M | ROBOTICS

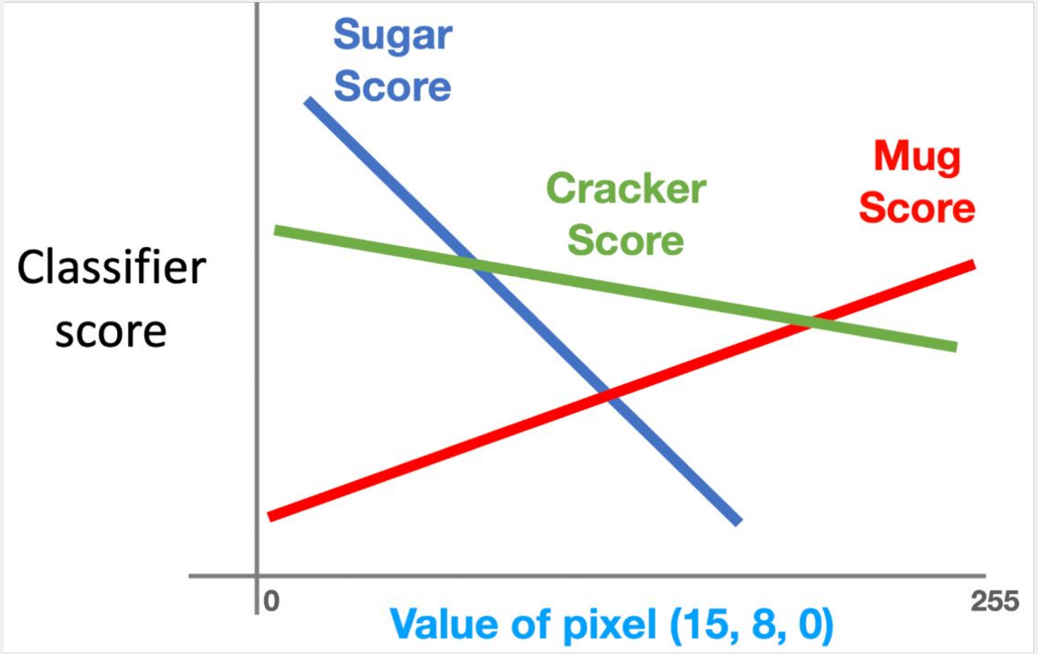
Interpreting a Linear Classifier
- ③ Geometric Viewpoint



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

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

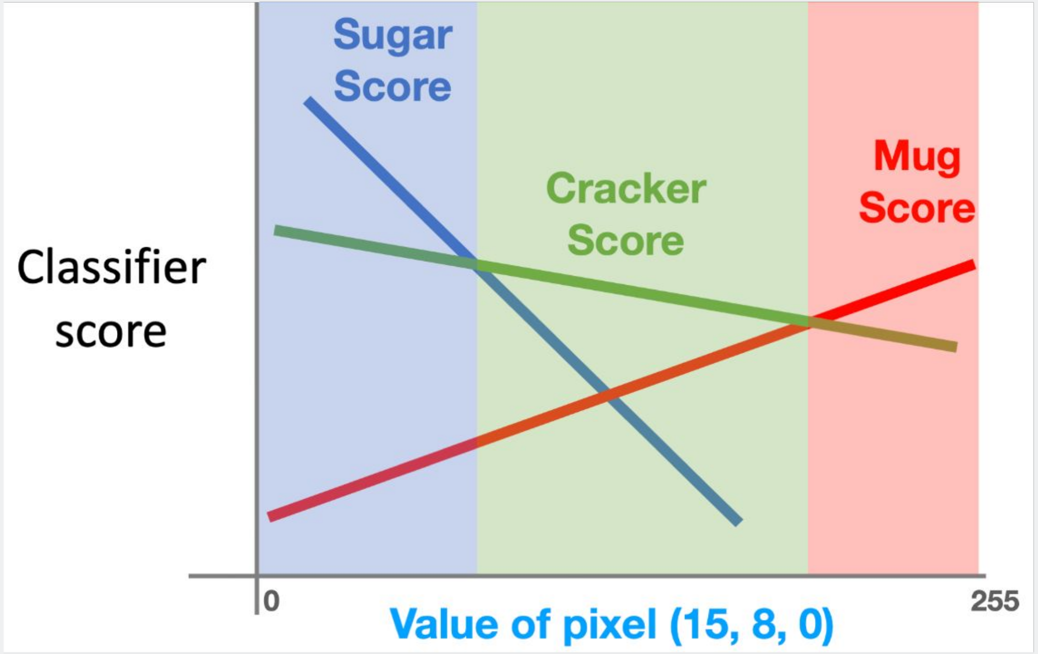
Interpreting a Linear Classifier
- Geometric Viewpoint



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

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

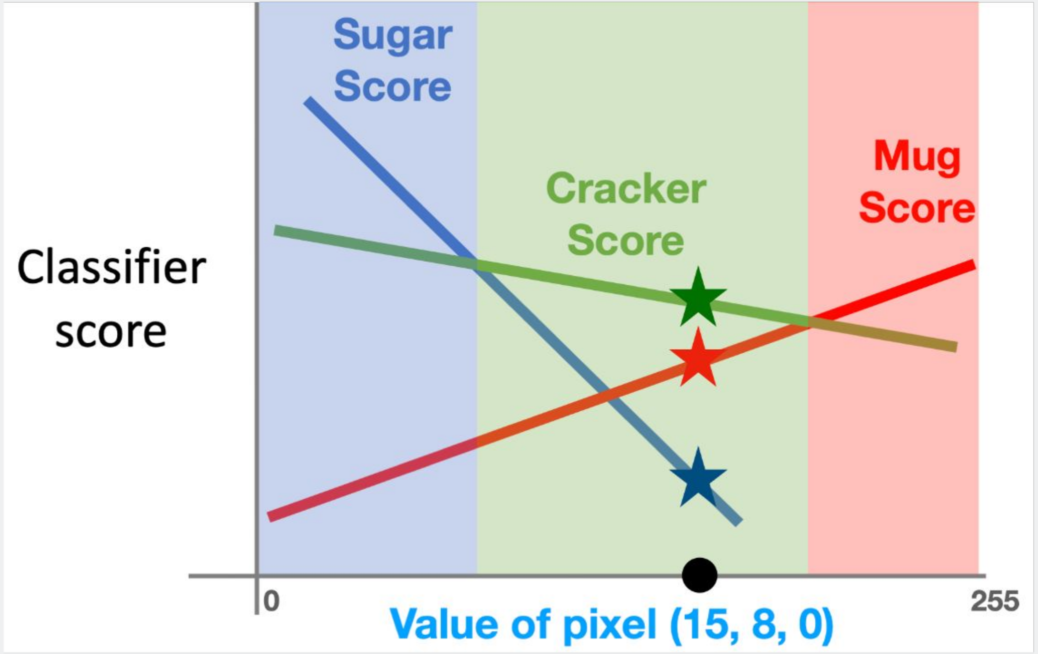
Interpreting a Linear Classifier
- Geometric Viewpoint



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

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

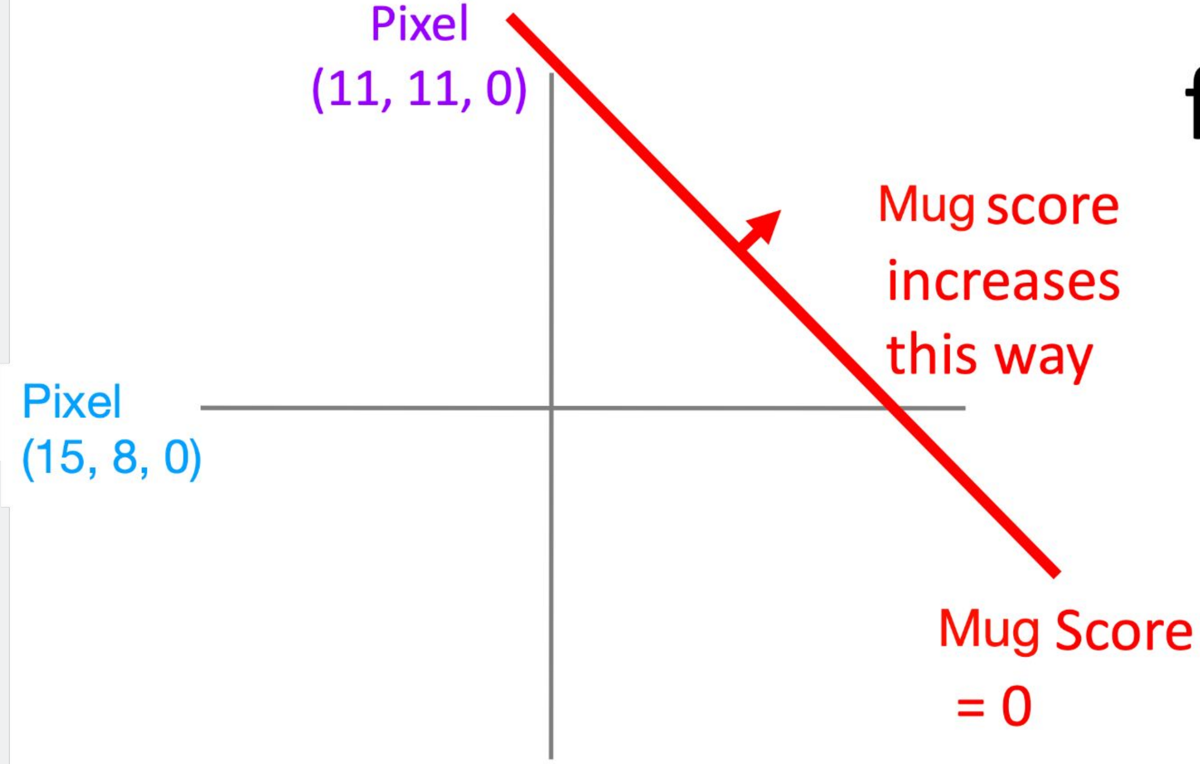
Interpreting a Linear Classifier
- Geometric Viewpoint



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

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

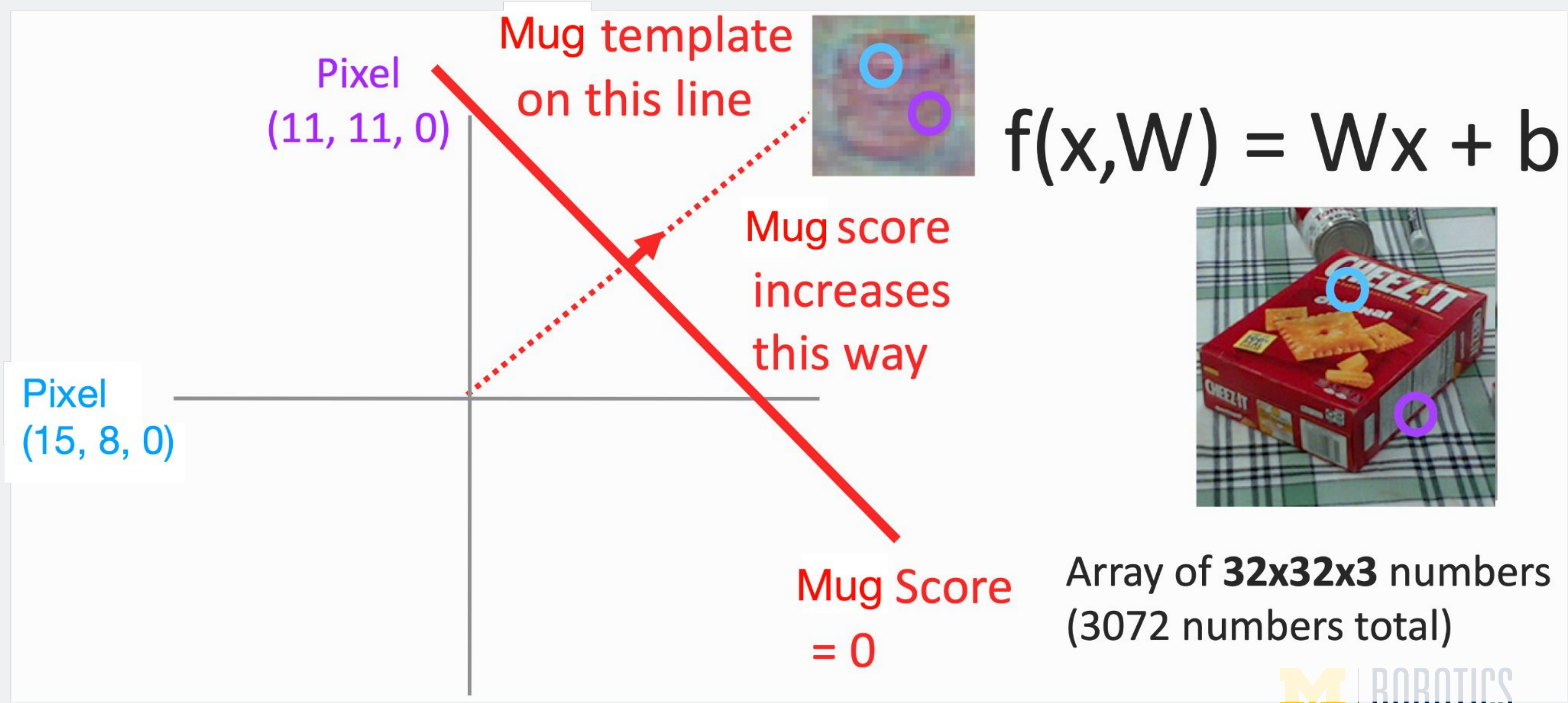
Interpreting a Linear Classifier
- Geometric Viewpoint



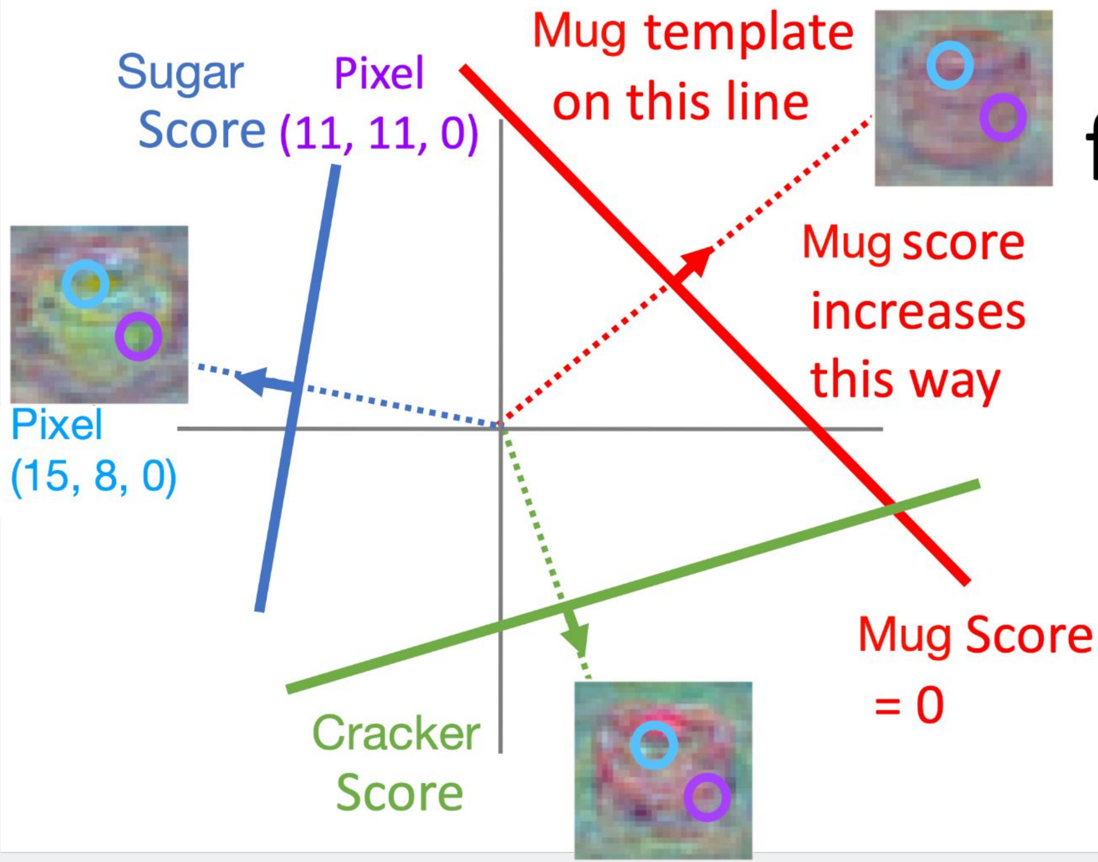
$$f(x,W) = Wx + b$$

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

Interpreting a Linear Classifier
- Geometric Viewpoint



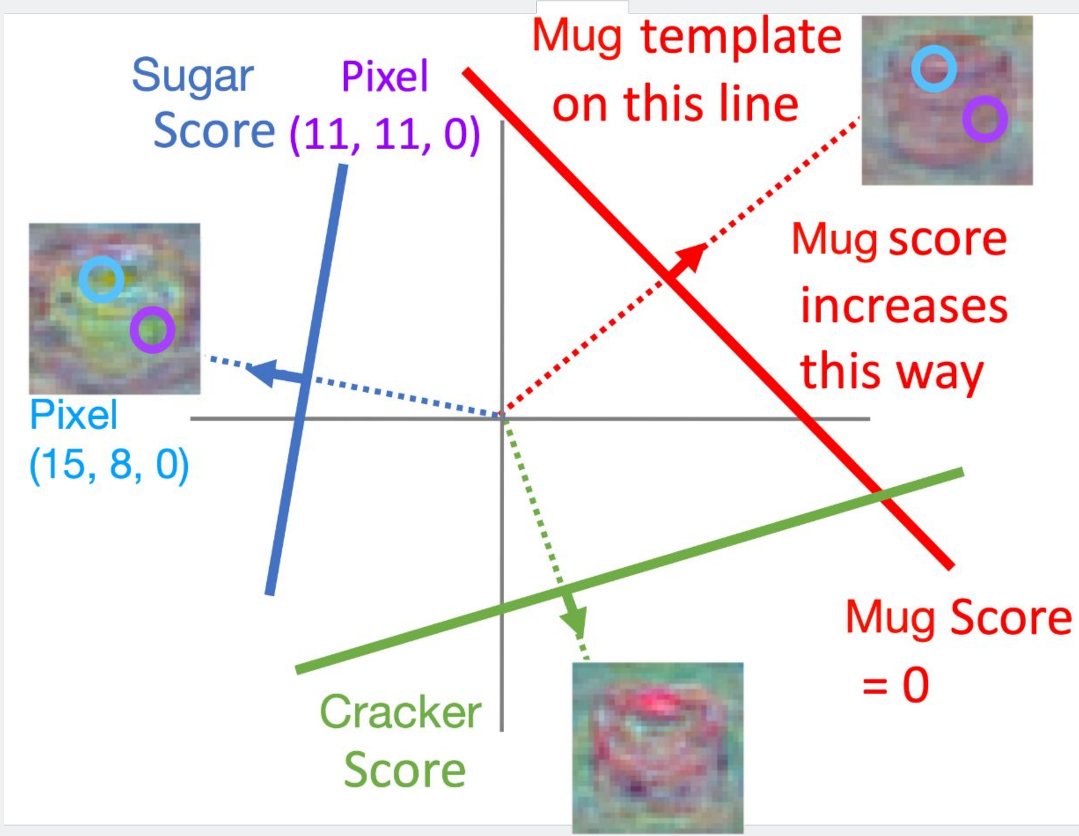
Interpreting a Linear Classifier
- Geometric Viewpoint

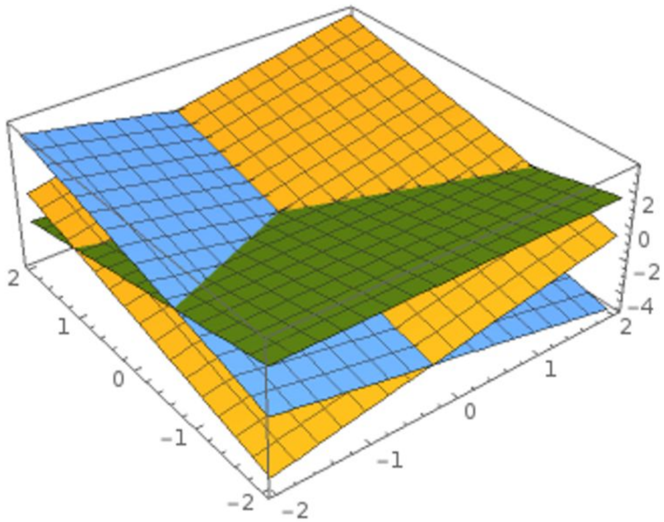


$$f(x,W) = Wx + b$$

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

Interpreting a Linear Classifier
- Geometric Viewpoint



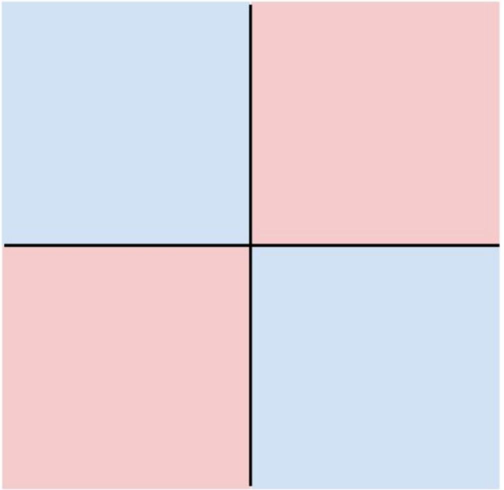


Plot created using Wolfram Cloud

Difficult Case (Examples) for a Linear Classifier

Class 1:
First and third quadrants

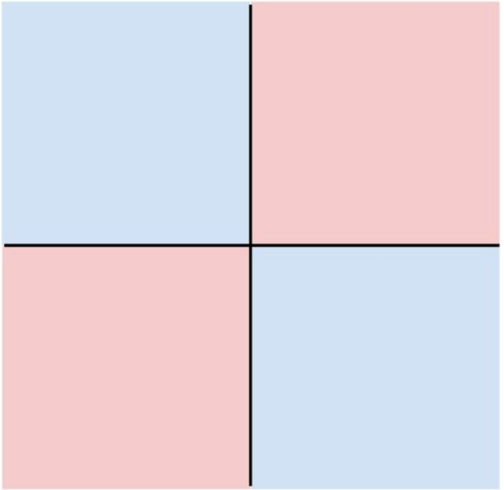
Class 2:
Second and fourth quadrants



Difficult Case (Examples) for a Linear Classifier

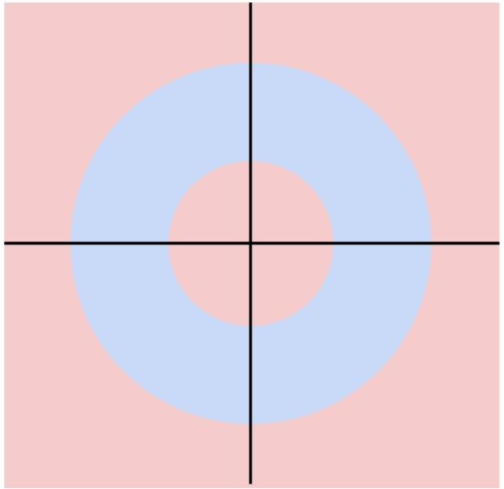
Class 1:
First and third quadrants

Class 2:
Second and fourth quadrants



Class 1:
1 <= L2 norm <= 2

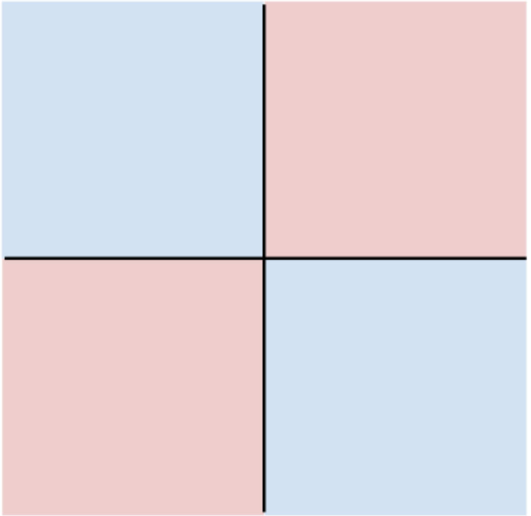
Class 2:
Everything else



Difficult Case (Examples) for a Linear Classifier

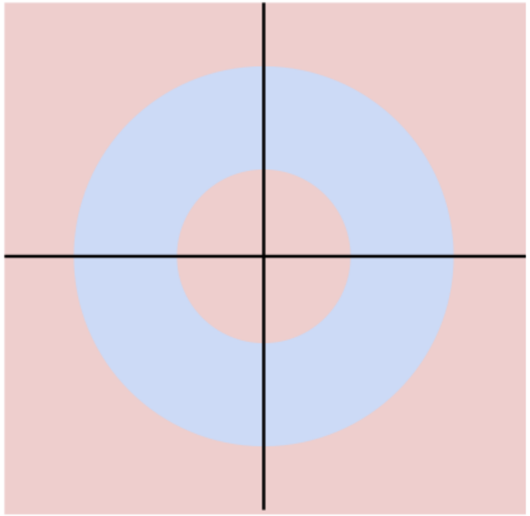
Class 1:
First and third quadrants

Class 2:
Second and fourth quadrants



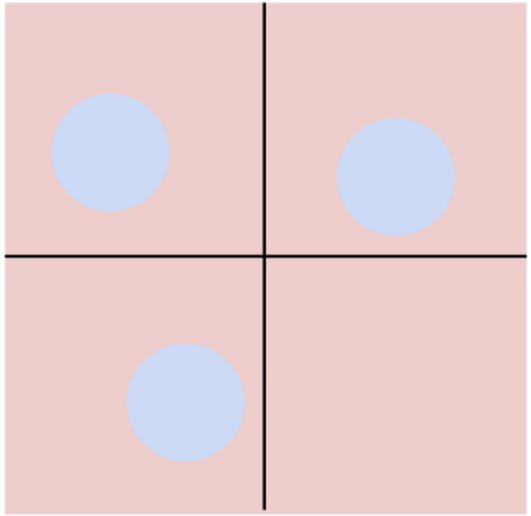
Class 1:
1 <= L2 norm <= 2

Class 2:
Everything else



Class 1:
Three modes

Class 2:
Everything else



How do we actually choose a good W?

Define a score function

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

master chef can	-3.45	-0.51	3.42
mug	-8.87	**6.04**	4.64
tomato soup can	0.09	5.31	2.65
cracker box	**2.9**	-4.22	5.1
mustard bottle	4.48	-4.19	2.64
tuna fish can	8.02	3.58	5.55
sugar box	3.78	4.49	**-4.34**
gelatin box	1.06	-4.37	-1.5
potted meat can	-0.36	-2.09	-4.79
large marker	-0.72	-2.93	6.14

Define a score function


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

1. Use a **loss function** to quantify how good a value of W is (today)

2. Find a W that minimizes the loss function (**optimization**) (next week)

maste
mug
tomat
crack
musta
tuna f
sugar
gelati

potted meat can	-0.36	-2.09	-4.79
large marker	-0.72	-2.93	6.14

Loss Function

A **loss function** measures how good our current classifier is.

Low loss = good classifier

High loss = bad classifier

Also called: **objective function, cost function, reward function, profit/utility/fitness function, etc.**

Loss Function

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

where x_i is an image and

y_i is a (discrete) label

Loss for a single example is $L_i(f(x_i, W), y_i)$

Loss for the dataset is average of per-example losses:

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

ROBOTICS

Cross Entropy Loss

Multinomial Logistic Regression

cracker	**3.2**
mug	5.1
sugar	-1.7

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

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

$$P(Y = k \mid X = x_i) = \frac{\exp(s_k)}{\sum_j \exp(s_j)}$$

Softmax function

Class k

Cross Entropy Loss: Multinomial Logistic Regression

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

$$s = f(x_i; W) \qquad P(Y = k \mid X = x_i) = \frac{\exp(s_k)}{\sum_j \exp(s_j)}$$

Softmax function

	Unnormalized log-probabilities (logits)	exp(·) → Unnormalized probabilities (Probabilities must be >=0)	normalize → Probabilities (Probabilities must sum to 1)
cracker	**3.2**	**24.5**	**0.13**
mug	5.1	164.0	0.87
sugar	-1.7	0.18	0.00

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

$$L_i = -\log(0.13) = 2.04$$

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

(EECS 445/545,
Bishop: Pattern Recognition and Machine Learning Book)

ROBOTICS

Cross Entropy Loss: Multinomial Logistic Regression

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

$$s = f(x_i; W) \qquad P(Y = k \mid X = x_i) = \frac{\exp(s_k)}{\sum_j \exp(s_j)}$$

Softmax function

	Unnormalized log-probabilities (logits)	$\exp(\cdot)$ →	Unnormalized probabilities (Probabilities must be >=0)	normalize →	Probabilities (Probabilities must sum to 1)	compare	Correct probabilities
cracker	**3.2**		**24.5**		**0.13**	→ compare ←	**1.00**
mug	5.1		164.0		0.87		0.00
sugar	-1.7		0.18		0.00		0.00

Kullback-Leibler divergence

$$D_{KL}(P \| Q) = \sum_y P(y) \log \frac{P(y)}{Q(y)}$$

Commonly used in information theories;
Compare between model probability distributions

Cross Entropy Loss: Multinomial Logistic Regression

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

$$s = f(x_i; W) \qquad P(Y = k \mid X = x_i) = \frac{\exp(s_k)}{\sum_j \exp(s_j)}$$

Softmax function

	Unnormalized log-probabilities (logits)		Unnormalized probabilities		Probabilities		Correct probabilities
cracker	**3.2**	exp(·) →	**24.5**	normalize →	**0.13**	→ compare ←	**1.00**
mug	5.1		164.0		0.87		0.00
sugar	-1.7		0.18		0.00		0.00

Probabilities must be >=0

Probabilities must sum to 1

Cross Entropy

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

Cross Entropy Loss: Multinomial Logistic Regression

cracker	**3.2**
mug	5.1
sugar	-1.7

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

$$s = f(x_i; W) \qquad P(Y = k \mid X = x_i) = \frac{\exp(s_k)}{\sum_j \exp(s_j)}$$

Softmax function

Maximize probability of correct class

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

Putting it all together

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

P1

ROBOTICS

Aha Slides
(In-class participation)

https://ahaslides.com/YAFWJ

Q1, Q2

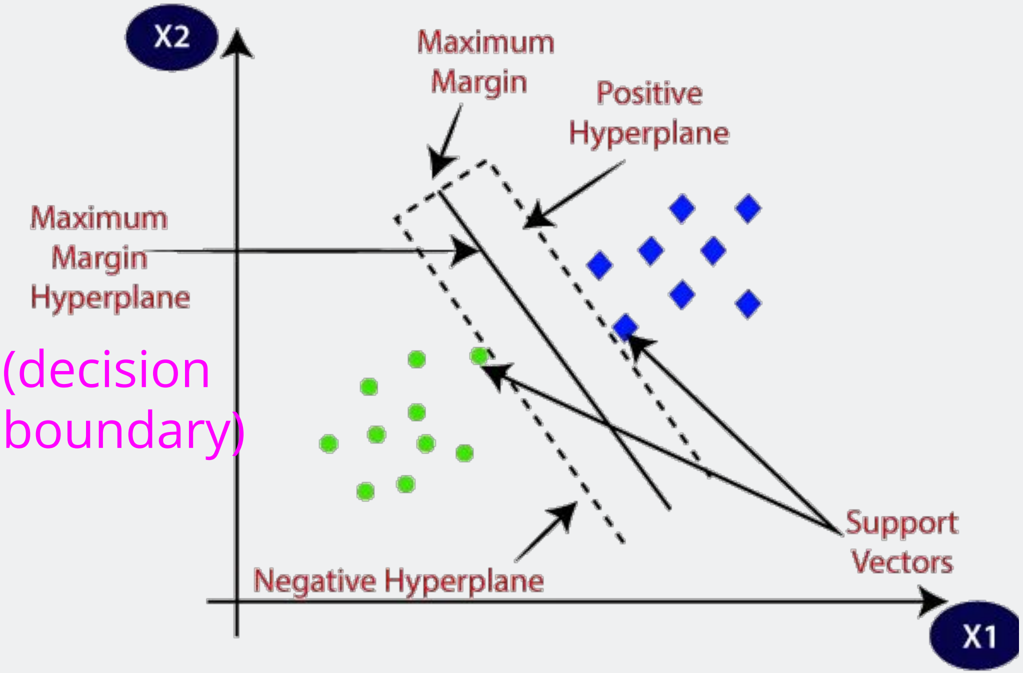
List of Questions on AhaSlides (for your record)

Q1: What is the min / max possible loss?

Q2: If all scores are small random values, what is the loss?

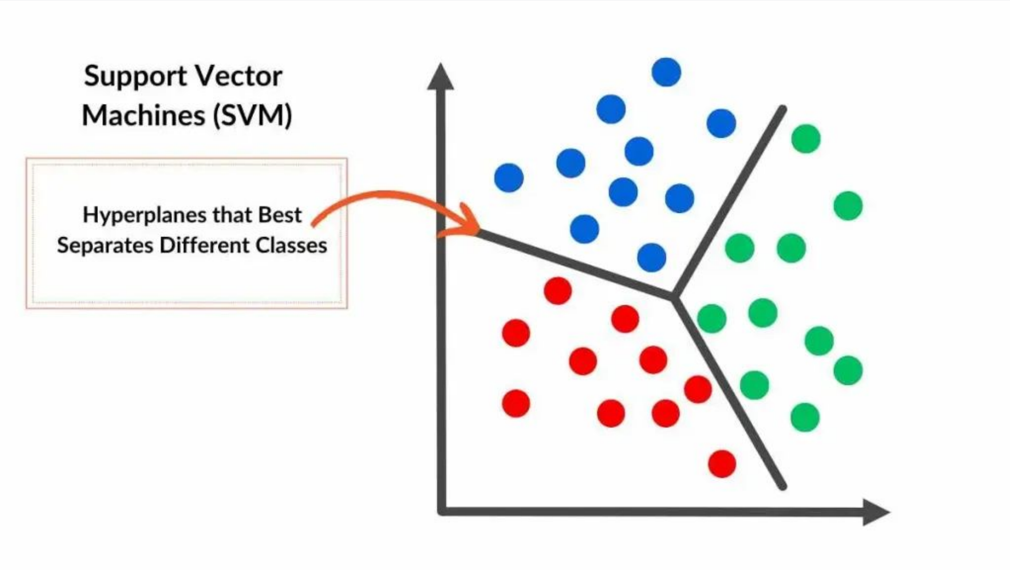
Multi-class SVM Loss

Multiclass SVM Loss



(two-class SVM - example)

Multiclass SVM Loss



(multi-class SVM - example)

Multiclass SVM Loss

Given an example (x_i, y_i)
(x_i is image, y_i is label)

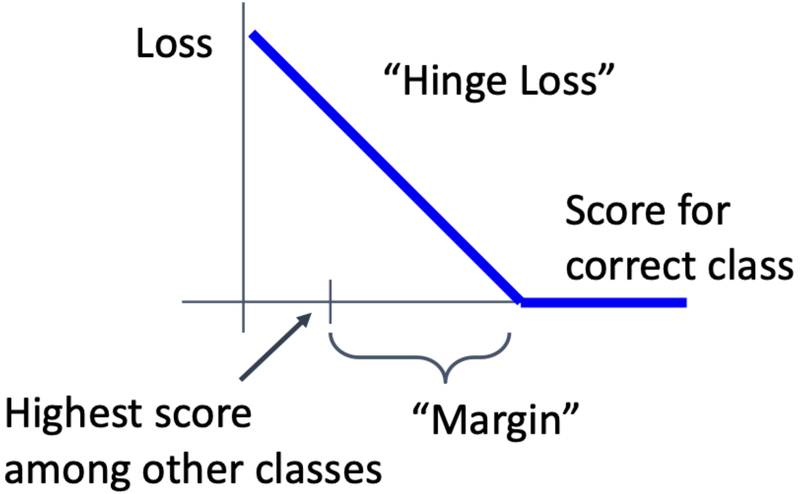
Let $s = f(x_i, W)$ be scores

Then the SVM loss has the form:

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

Multiclass SVM Loss

"The score of the correct class should be higher than all the other scores"



Given an example (x_i, y_i)
(x_i is image, y_i is label)

Let $s = f(x_i, W)$ be scores

Then the SVM loss has the form:

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

P1

Multiclass SVM Loss - Concrete Example


cracker	**3.2**	1.3	2.2
mug	5.1	**4.9**	2.5
sugar	-1.7	2.0	**-3.1**
Loss	2.9		

Given an example (x_i, y_i)
(x_i is image, y_i is label)

Let $s = f(x_i, W)$ be scores

Then the SVM loss has the form:

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

= 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

Multiclass SVM Loss - Concrete Example


cracker	**3.2**	1.3	2.2
mug	5.1	**4.9**	2.5
sugar	-1.7	2.0	**-3.1**
Loss	2.9	0	?

Given an example (x_i, y_i)
(x_i is image, y_i is label)

Let $s = f(x_i, W)$ be scores

Then the SVM loss has the form:

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

= 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

Aha Slides (In-class participation)

https://ahaslides.com/YAFWJ

Q3: What is the multi-class SVM loss for Slide 55, third image?

Q4: For multi-class SVM loss, what if the loss uses a mean instead of a sum? Would the prediction results be the same or different?

Multiclass SVM Loss - Concrete Example


cracker	**3.2**	1.3	2.2
mug	5.1	**4.9**	2.5
sugar	-1.7	2.0	**-3.1**
Loss	2.9	0	

Given an example (x_i, y_i)
(x_i is image, y_i is label)

Let $s = f(x_i, W)$ be scores

Then the SVM loss has the form:

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

Loss over the dataset is:

L = (2.9 + 0.0 + 12.9) / 3
= 5.27

Cross Entropy Loss
vs. Multi-class SVM Loss

Example

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

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

assume scores:

[10, -2, 3]

[10, 9, 9]

[10, -100, -100]

and $y_i = 0$

Q: What is cross-entropy loss?

What is SVM loss?

Example

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

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

assume scores:

[10, -2, 3]

[10, 9, 9]

[10, -100, -100]

and $y_i = 0$

Q: What is cross-entropy loss?

What is SVM loss?

A: Cross-entropy loss > 0

SVM loss = 0

Example

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

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

assume scores:

[10, -2, 3]

[10, 9, 9]

[10, -100, -100]

and $y_i = 0$

Q: What happens to each loss if I slightly change the scores of the last datapoint?

Example

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

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

assume scores:
[10, -2, 3]
[10, 9, 9]
[10, -100, -100]
and $y_i = 0$

Q: What happens to each loss if I slightly change the scores of the last datapoint?

A: Cross-Entropy Loss will change;
SVM loss will stay the same for 1st and 3rd cases;
SVM loss will change for the 2nd case

Example

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

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

assume scores:

[10, -2, 3]

[10, 9, 9]

[10, -100, -100]

and $y_i = 0$

Q: What happens to each loss if I double the score of the correct class from 10 to 20?

Example

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

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

assume scores:
[10, -2, 3]
[10, 9, 9]
[10, -100, -100]
and $y_i = 0$

Q: What happens to each loss if I double the score of the correct class from 10 to 20?

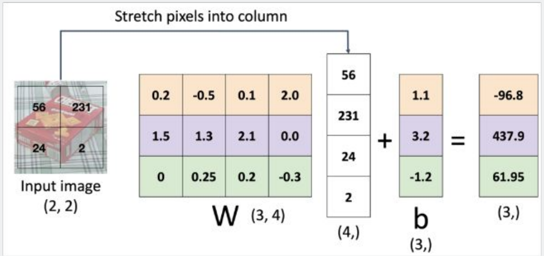
A: Cross-Entropy Loss will ????; (Canvas quiz)
SVM loss still 0

Summary

Linear Classifier - Three Viewpoints

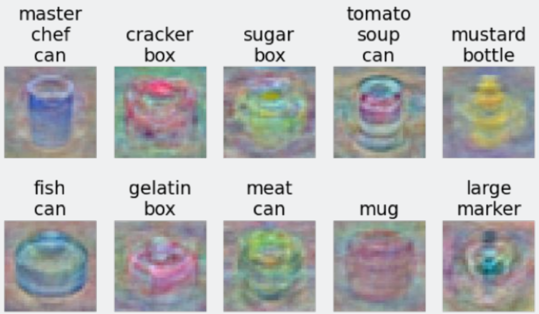
① Algebraic Viewpoint

$f(x,W) = Wx$



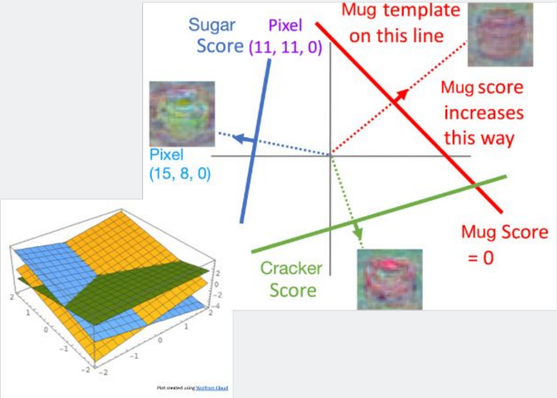
② Visual Viewpoint

One template
per class



③ Geometric Viewpoint

Hyperplanes
cutting up space



Loss Functions

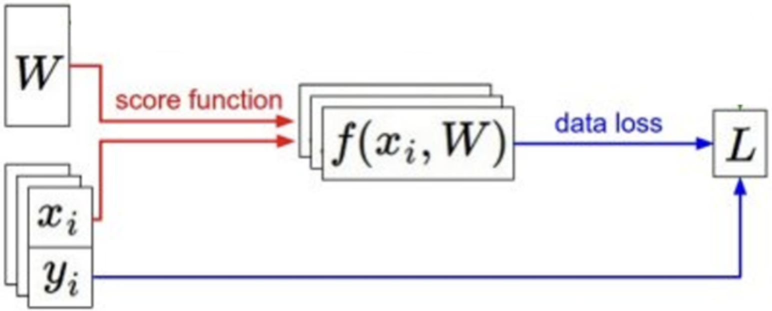
- We have some dataset of (x, y)
- We have a **score function:**
- We have a **loss function:**

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

Linear classifier

Softmax: $L_i = -\log\left(\frac{\exp(s_{y_i})}{\sum_j \exp(s_j)}\right)$

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



Next up: How to find best W and b? Optimization

Due dates

Canvas Assignment: 20260112 KNN Quiz

Scored - individual (as part of in-class activity points)

Due Jan. 14, 2026

Canvas Assignment: 20260114 Linear Classifier Quiz

Scored - individual (as part of in-class activity points)

Due Jan. 18, 2026

P0

5 submissions per day - Start today!!!

Due Jan. 18, 2026

ROBOTICS