# Explanations

Spring 2020

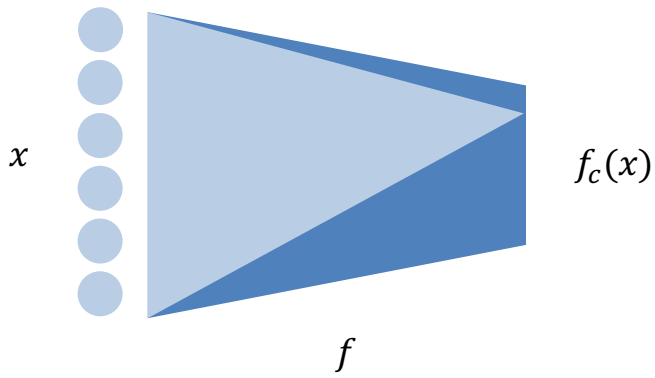
# Today

- (finish up) Influence directed explanations
- Explanations overview
- Linear vs non-linear models / coding practice

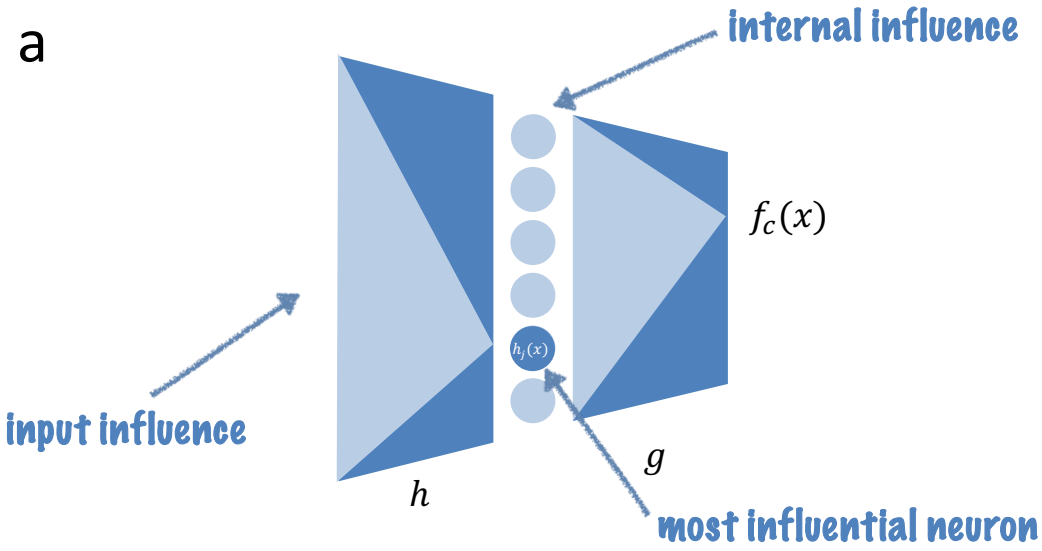
# Influence Directed Explanations

# Influence Directed Explanations

- Input Influence: Saliency, Integrated Gradients, many others




- Use input influence with a quantity of interest that selects a particular internal neuron



# Influence-Directed Explanations for CNNs

# Explanations Overview

# Explanations Overview

- Covered
  - Representer point selection for DNN
  - Understanding Black-box Predictions via Influence Functions
  - Axiomatic Attribution for Deep Networks
  - Influence-Directed Explanations for CNNs
- Categorize methods on
  - Explanation of …
  - Explanation form
  - Requirements
  - Evaluations
  - Strengths, weaknesses

# Explanations of …

- Prediction F(X) = Y
- Class Score F(X) = Y, explain $Y_c$
- Quantity of Interest q(F(X)) = I

# Form / Interpretation

- Shadow interpret able model.
  - Global shadow.
  - Local linear model.
- Input's (pixels) importance on score
  - Distributions of interest
- Input's (pixels) importance on QoI
- Training instances' importance on score
- Input's importance on "experts"
  - Distributions of interest

# Requirements

- Model requirements
  - (optimal/convex)
- Training dataset
- Test instances
- Computational power

# Evaluations (was explanation good?)

- Subjective (human, typically the author) evaluation.
- Usefulness
- Objective
  - Compression
  - Ablation

# Strengths / Weaknesses

- Requirements
  - Computational power
  - Scalability vs dataset
    - Test instance
- Objective evaluation
- Implementation invariance
  - Interpretation
- Hyperparameters
  - Baselines
- Approximations for requirements

# Linear vs. Non-linear models

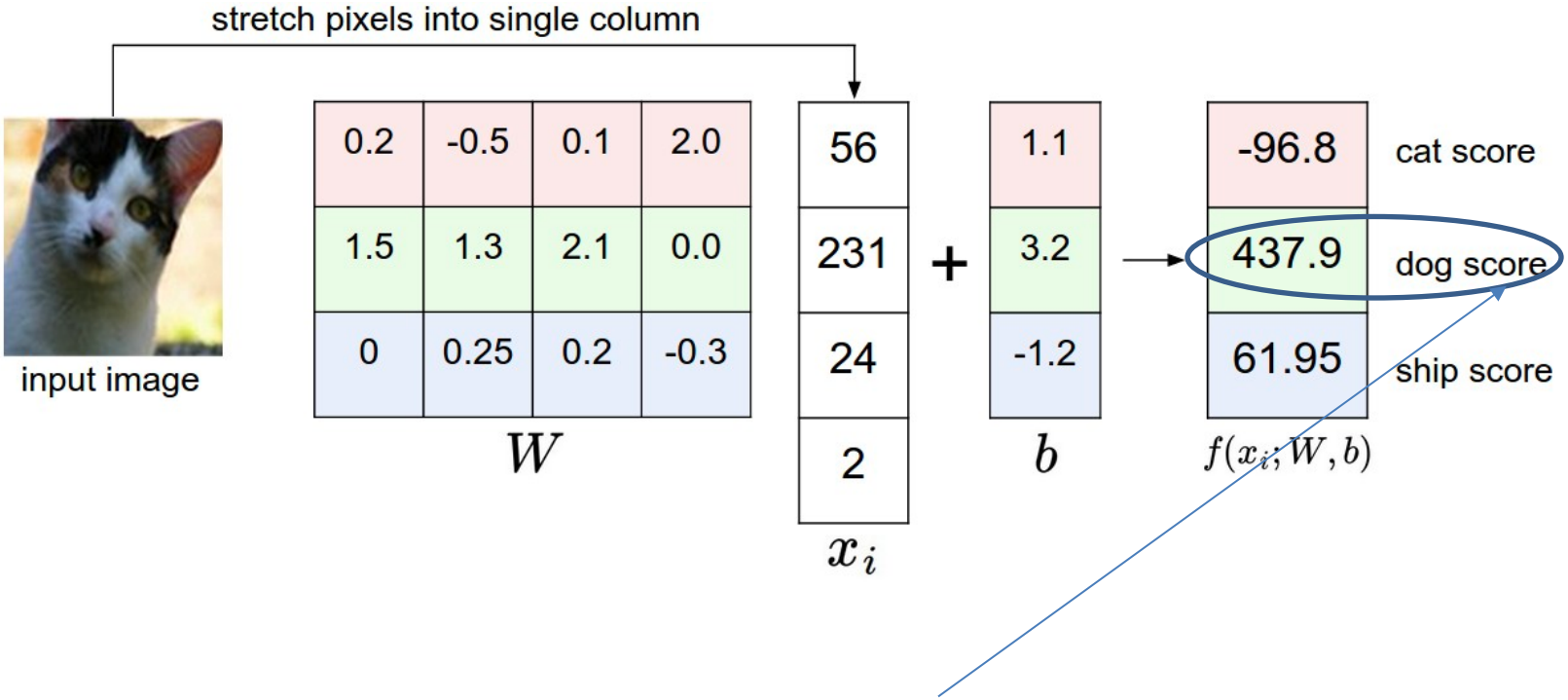
# Linear score function

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

- $x_i$ input image
- $W$ weights
- $b$ bias

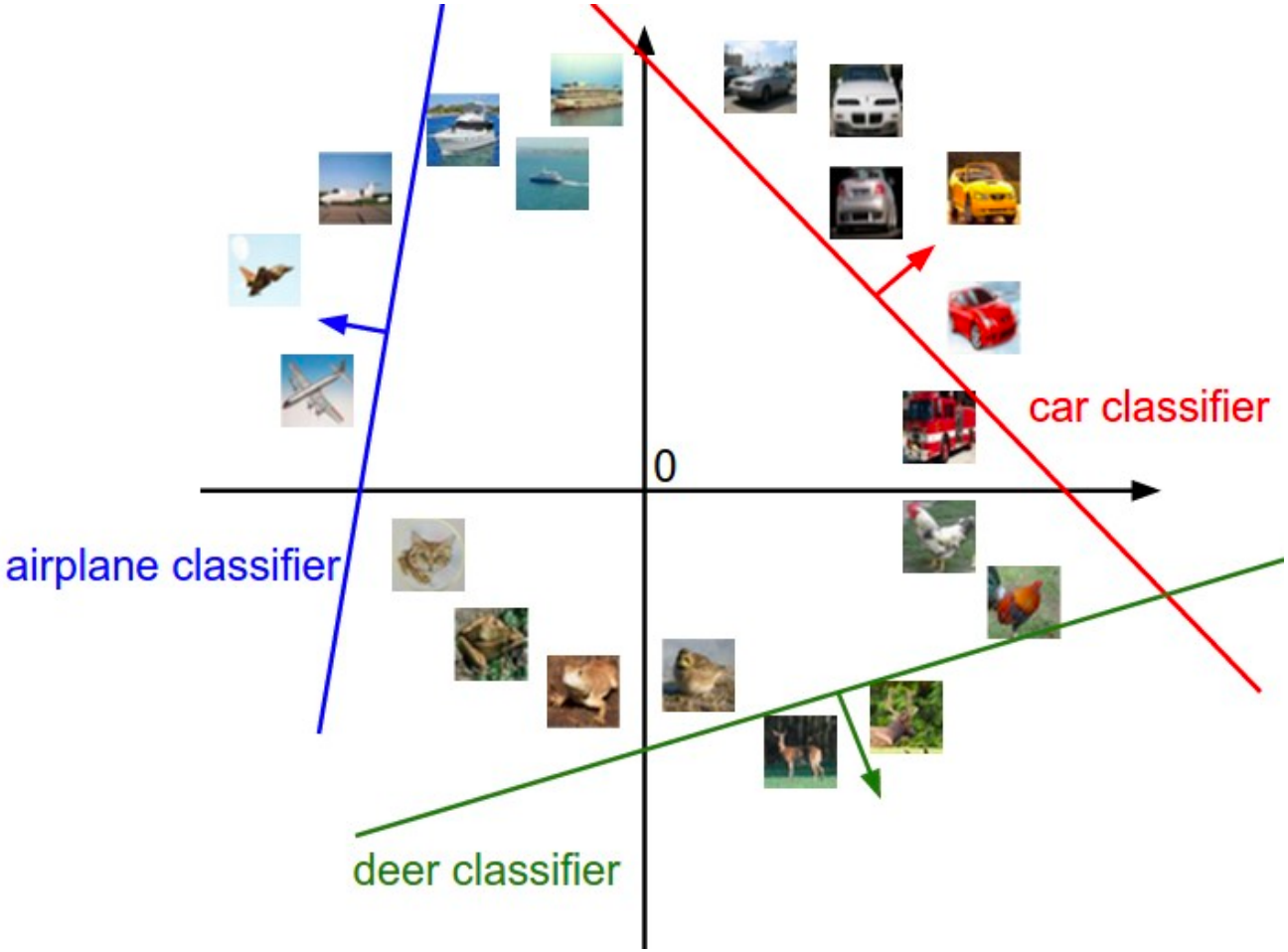
Learning goal:
Learn weights and bias that minimize loss

# Using score function




Predict class with highest score

# Linear classifiers as hyperplanes



# Acknowledgment


- Based on material from
  - Stanford CS231n http://cs231n.github.io/
  - Spring 2019 Course