Sensors and Regression Overview

CMPT 882

Mar. 8

Outline

- Sensors Overview
  - More details in Siegwart, Nourbakhsh, Scarmuzza, “Introduction to Autonomous Mobile Robots,” MIT Press 2011
- Regression Overview
  - More details in the next lectures
- Neural Networks Overview
  - More details in
    - CMPT 726: Machine Learning
    - CMPT 822: Computer Vision

Classification of sensors

- Proprioceptive: measurements of internal values
  - Motor speed, heading
- Exteroceptive: measurements of the environment
  - Distance measurements, light intensity, sound

- Passive: measure of signals from the environment
  - Temperature sensors, cameras
- Active: send a signal to the environment and measure the response
  - Ultrasonic sensors, Laser rangefinders
  - May affect the environment

Sensor Performance

- **Dynamic range**: ratio between maximum and minimum input values that can be measured accurately
- **Resolution**: smallest difference in signal that can be detected
- **Linearity**
- **Bandwidth** or frequency: how often a measurement is made

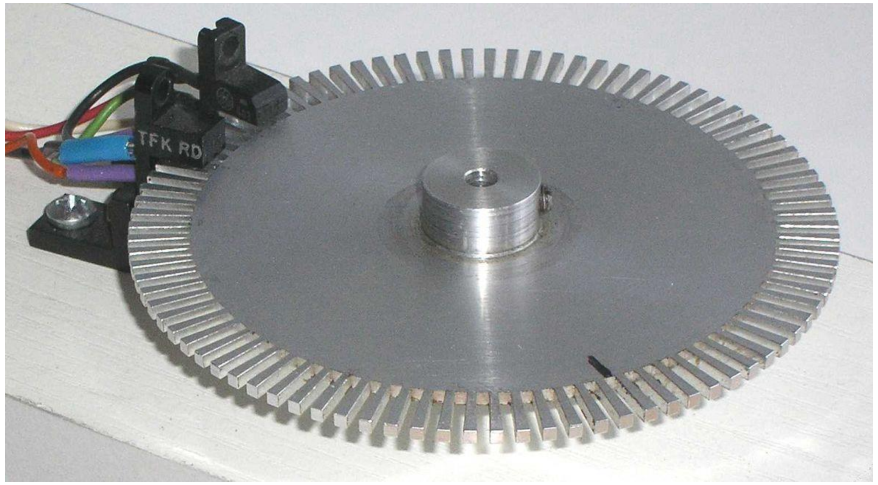
Sensor Performance

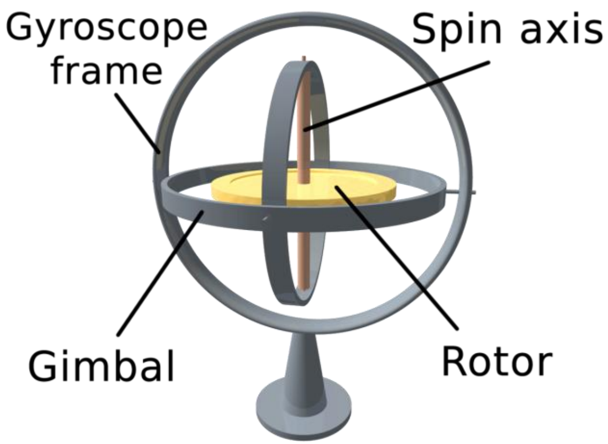
- **Sensitivity**: ratio of output change to input change
  - May vary with input signal, if sensor is nonlinear
  - Cross-sensitivity: sensitivity to unrelated factors in the environment
- **Error**: different between sensor measurement and true value
- **Accuracy**: absolute error relative to true value as a percentage
- **Precision**: consistency/reproducibility of measurements

- Sensor models: probabilistic description of sensor measurements
  - Will discuss more in localization and mapping lectures

Types of sensors

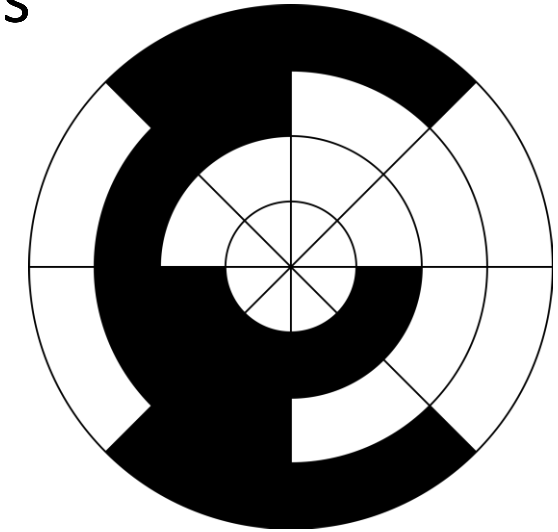
- Encoders
- Heading sensors
- Accelerometers and IMU
- Beacons
- Active ranging
- Cameras



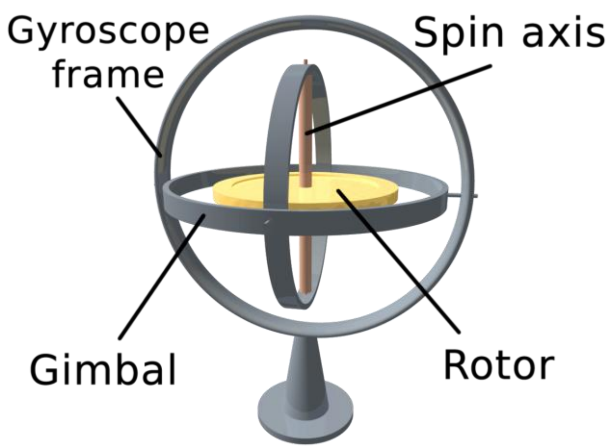


Encoders

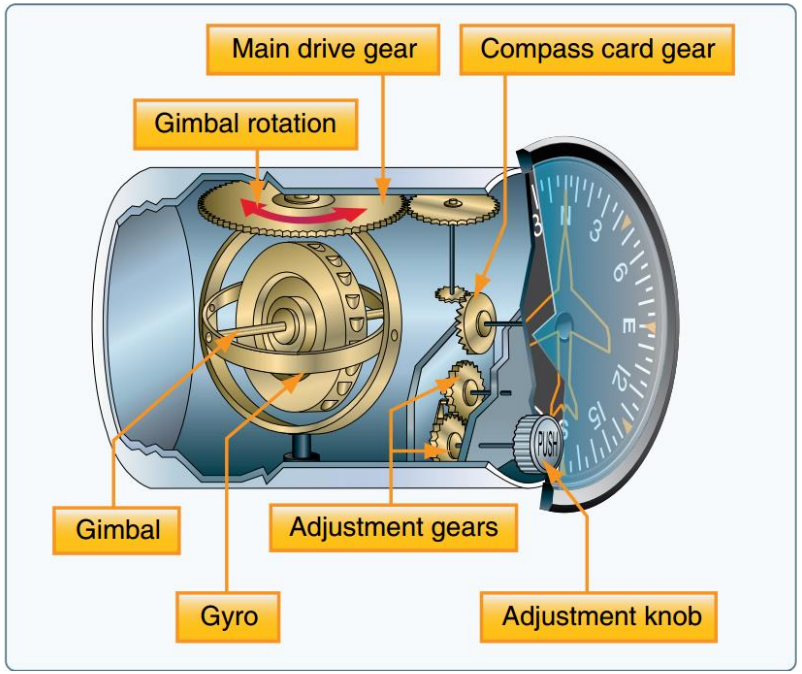
- Measures position by shining light through slits and counting number of interruptions
- Converts motion into a sequence of digital pulses
  - Proprioceptive
  - Can (kind of) be used for localization



Heading Sensors



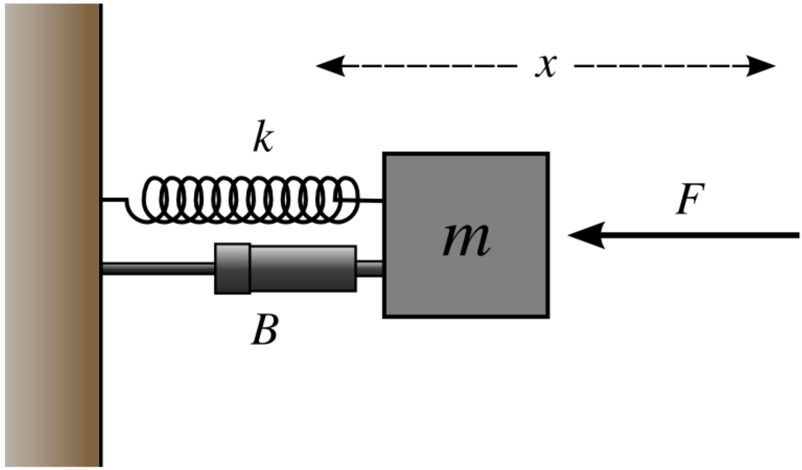
- Measures orientation or heading
  - Gyroscope: proprioceptive
    - Mechanical: up to three gimbals freely rotate without affecting axis of rotation of rotor
    - Optical: pair of lasers fired into circular optical fibre in opposite directions; rotations cause Doppler shift
  - Compass: exteroceptive



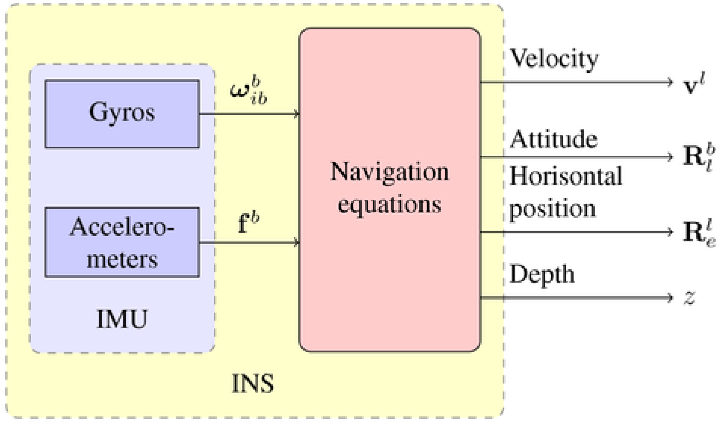
- Can be combined with velocity measurements to obtain position estimate

Accelerometer and Inertial Measurement Unit (IMU)

- Accelerometer: Measures all external forces acting on the sensor
  - Mechanical accelerometer: $F_{\text{applied}} = m\ddot{x} + c\dot{x} + kx$
    - $\Rightarrow a_{\text{applied}} = \frac{kx}{m}$ in steady state
    - Measure x, obtain a_{applied}
  - Modern accelerometers:
    - Micro Electro-Mechanical Systems (MEMS)
    - Capacitative: capacitance changes with force
    - Piezoelectric: voltage changes with force



- Inertial measurement unit (IMU)
  - Synonymous with Inertial Navigation System (INS)
  - Sensor package that measures position, orientation, and their rates
  - Combines gyroscopes and accelerometers



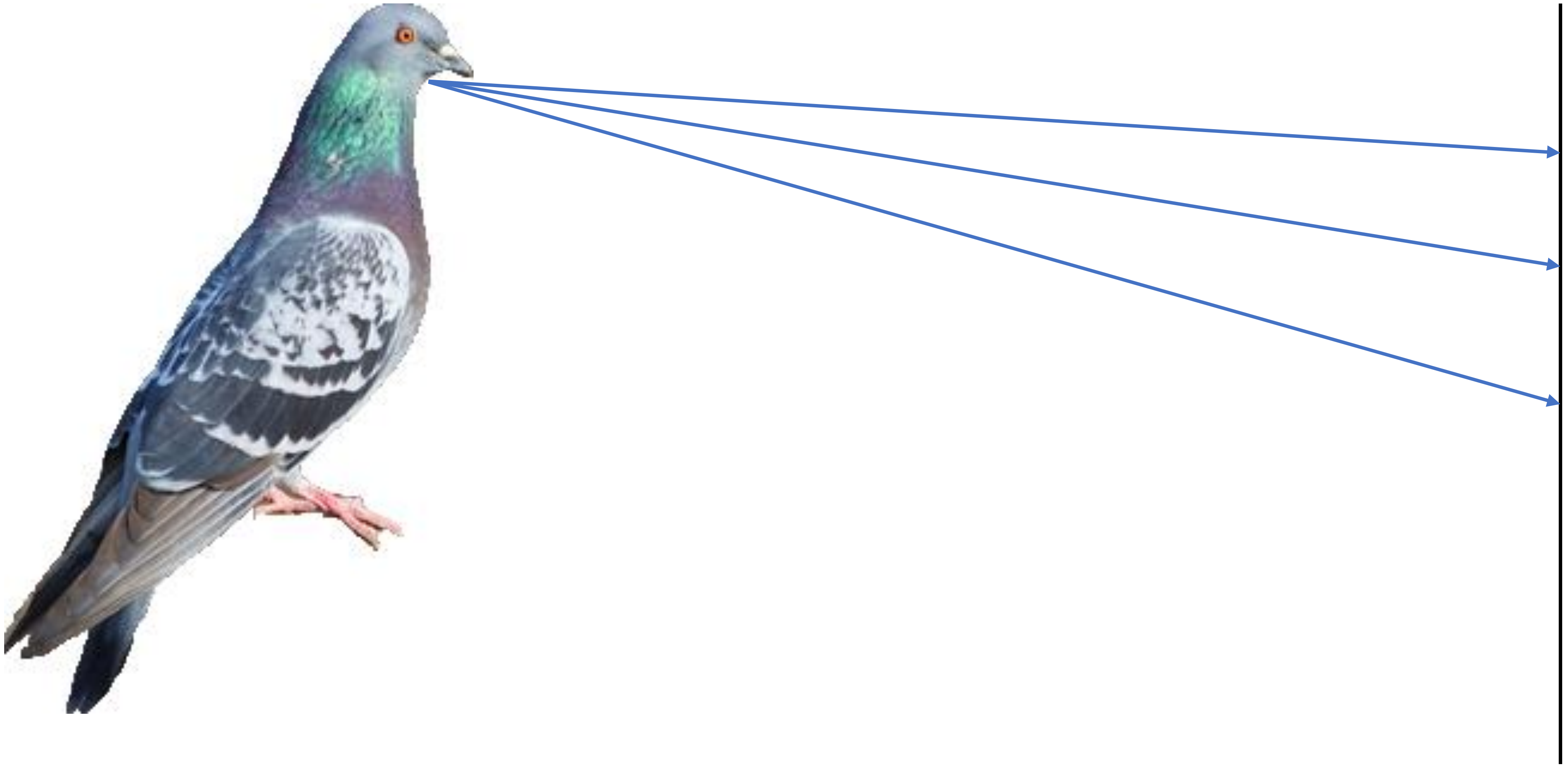
Beacons

- A device or structure with precisely known position
- Stars, lighthouses, landmarks
- GPS, motion capture systems
- Required for accurate measurement of position
  - Used in combination with IMU

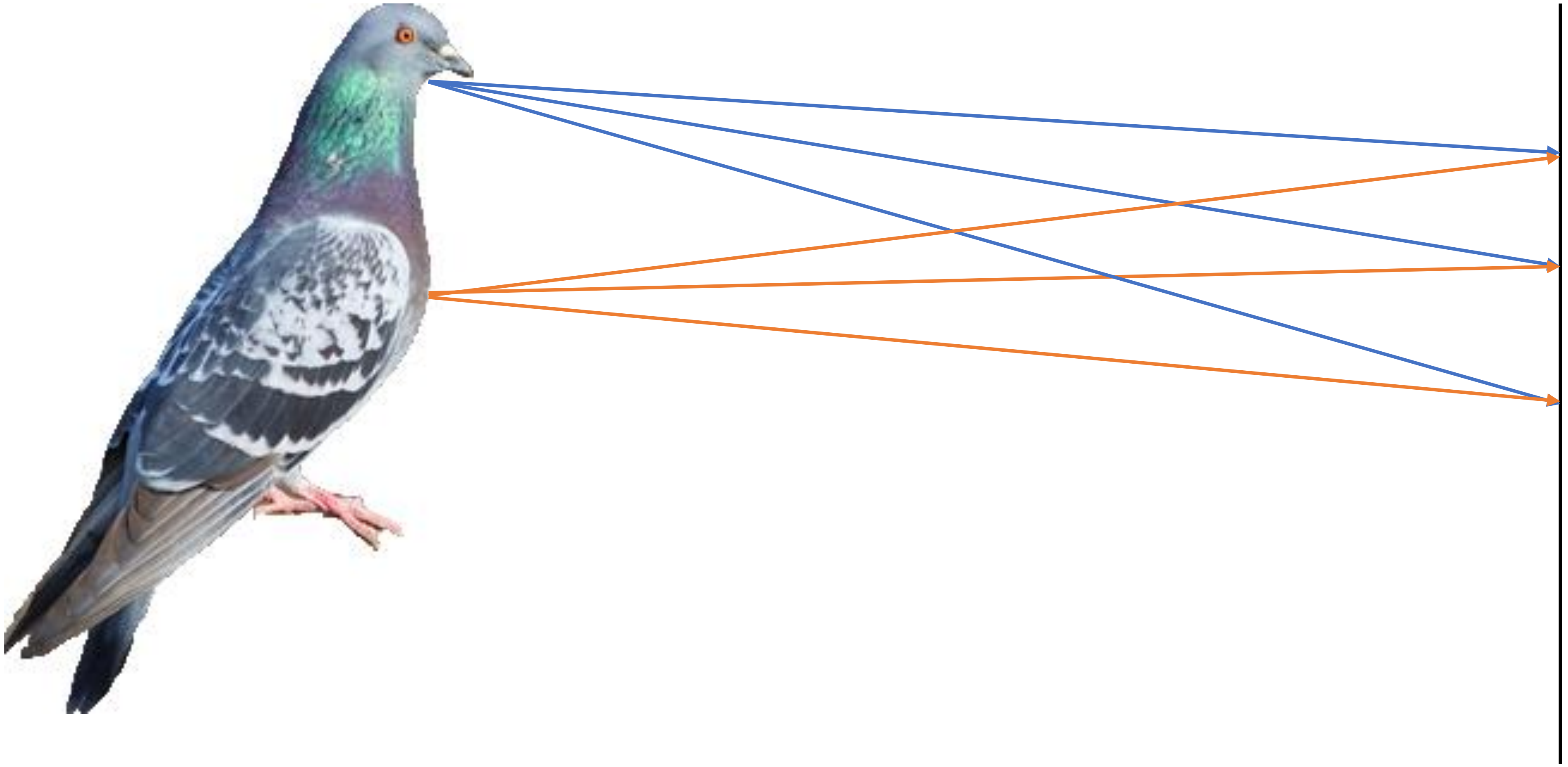
Active Ranging

- Measures distances to nearby objects
- Time-of-flight active ranging sensors
    - Travel distance: $d = ct$, where c is the speed of wave propagation and t is time of flight
    - Sonar: uses sound waves, $c = 343\ \mathrm{m/s}$
    - Lidar/radar: uses light waves, $c = 300\ \mathrm{m/\mu s}$
        - In general, longer wavelength → longer range, but cannot detect small features
- Geometric active ranging sensors

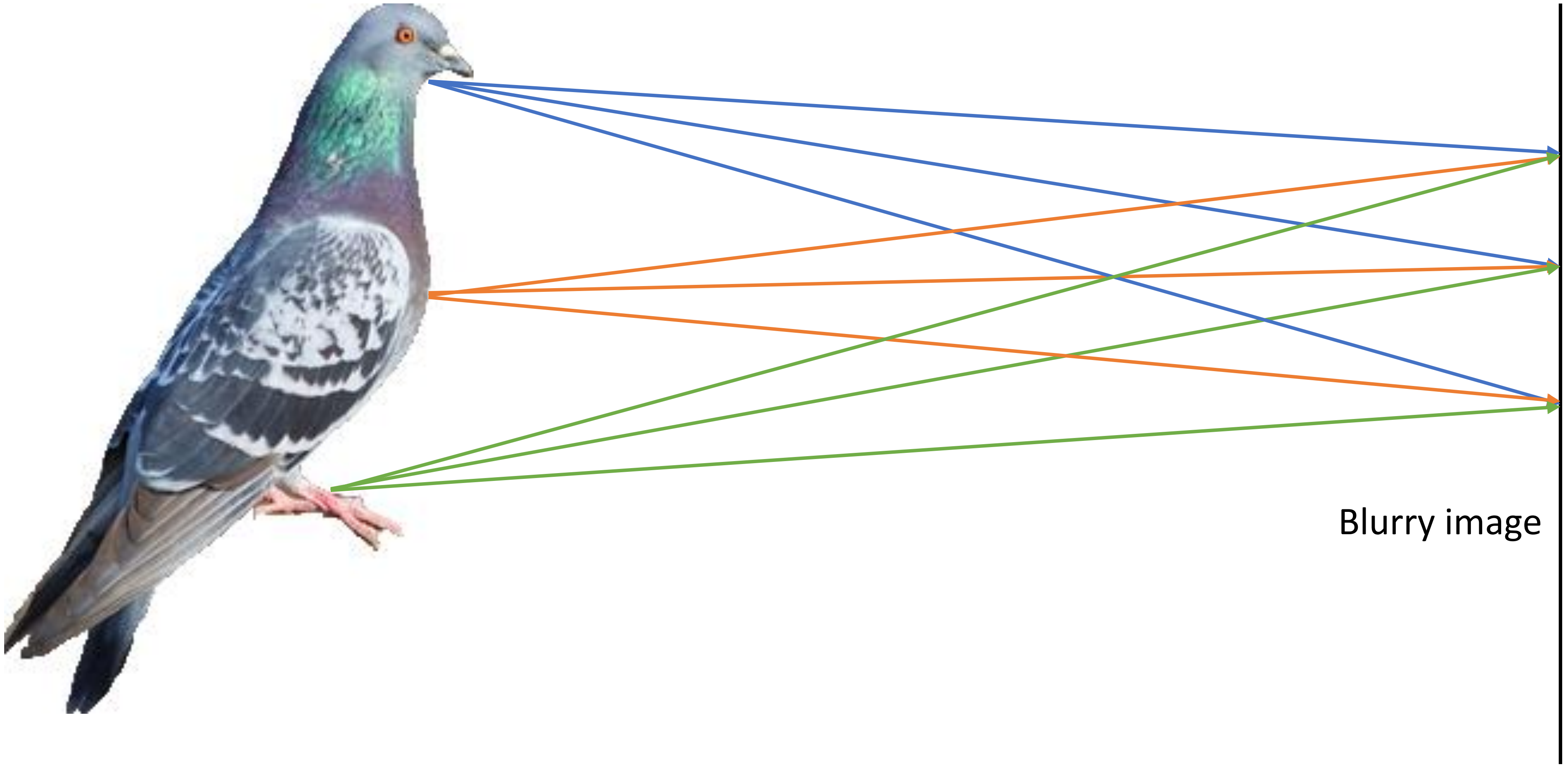
Cameras



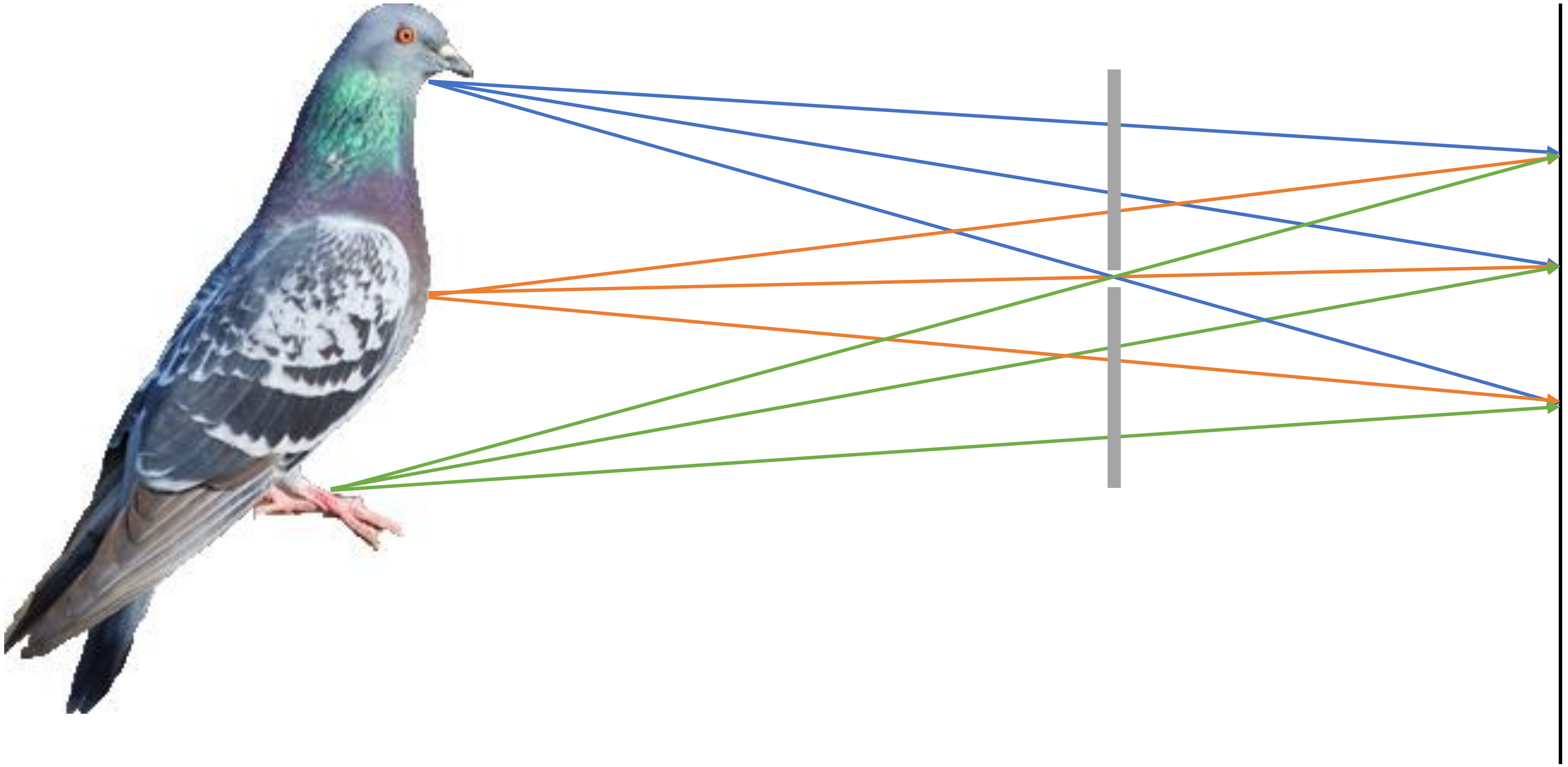
Cameras



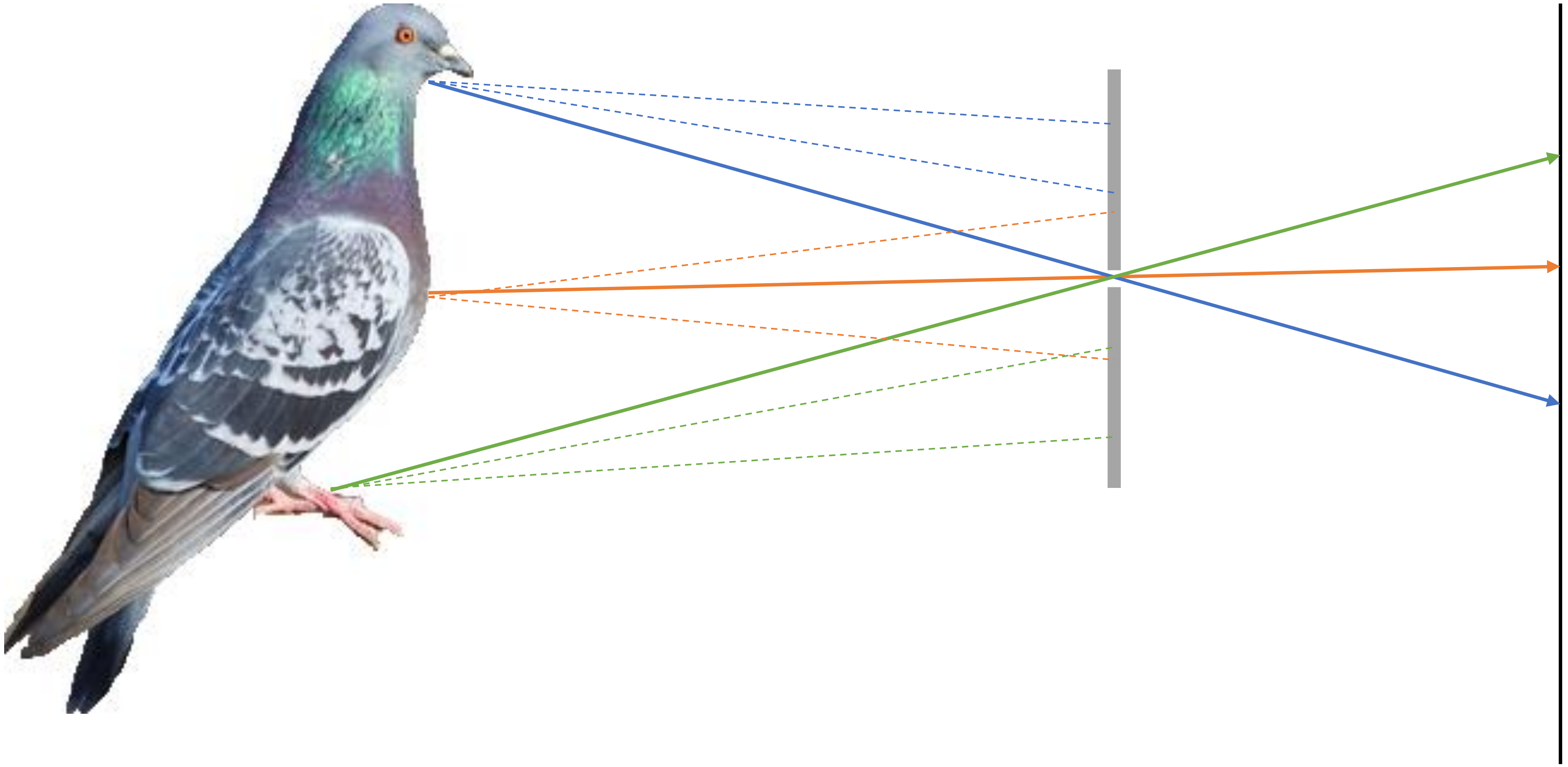
Cameras



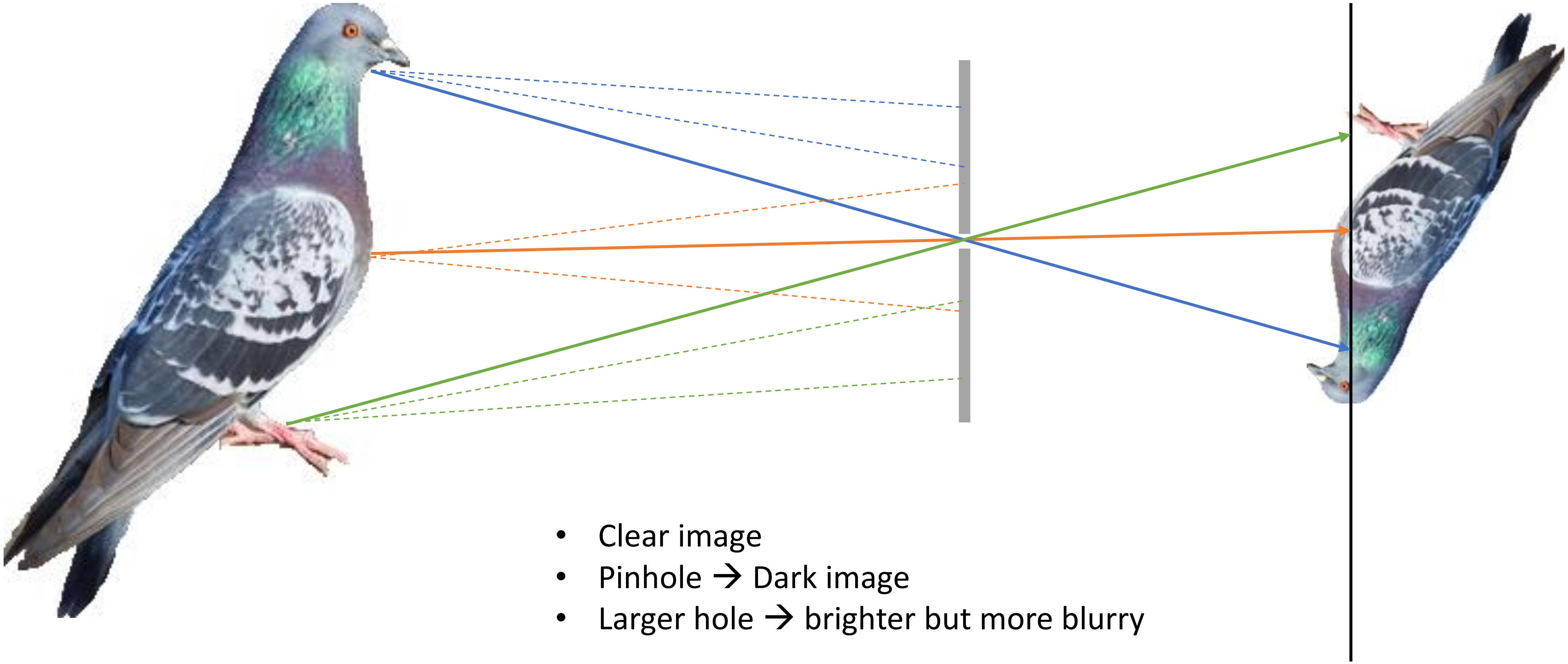
Pinhole Camera



Pinhole Camera



Pinhole Camera

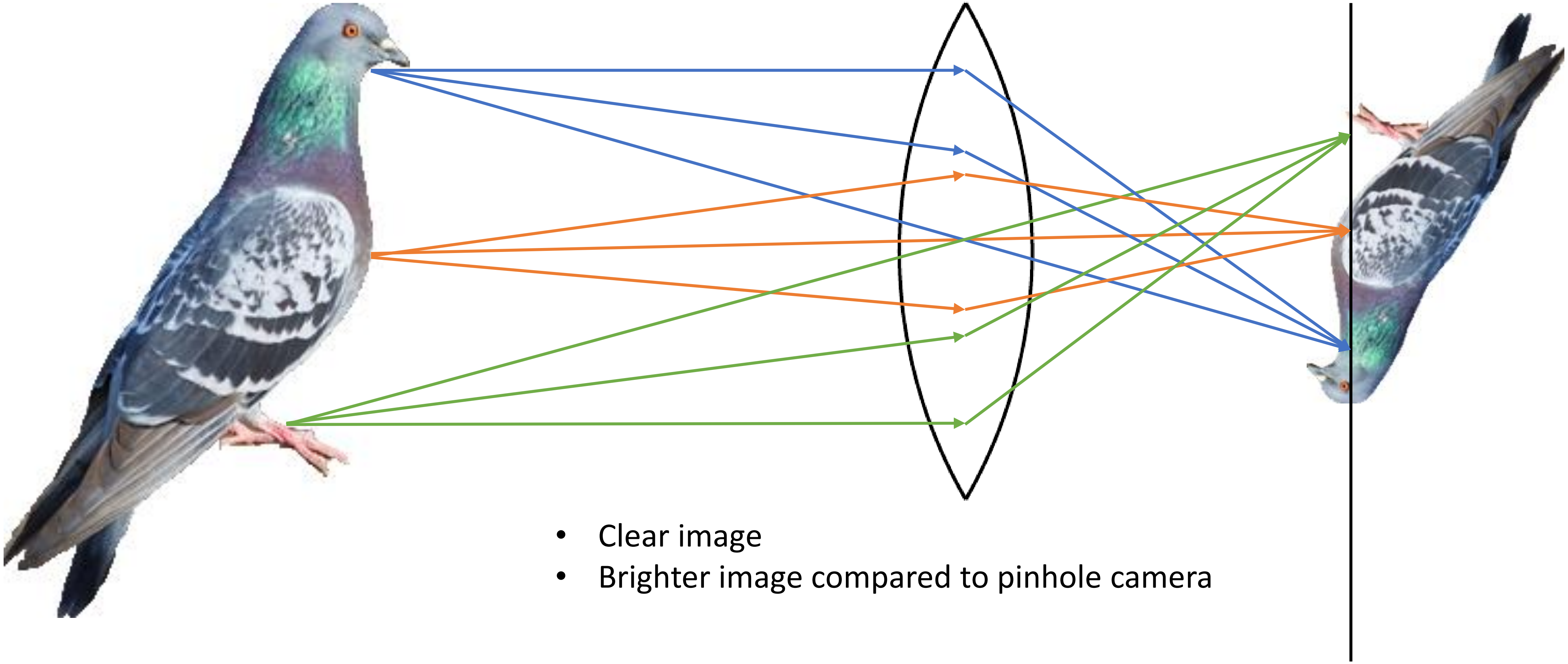


- Clear image
- Pinhole → Dark image
- Larger hole → brighter but more blurry

Solar Eclipse

- Gaps between leaves act as pinholes
- The shape of the sun is projected on the screen (ground)

Lenses



- Clear image
- Brighter image compared to pinhole camera

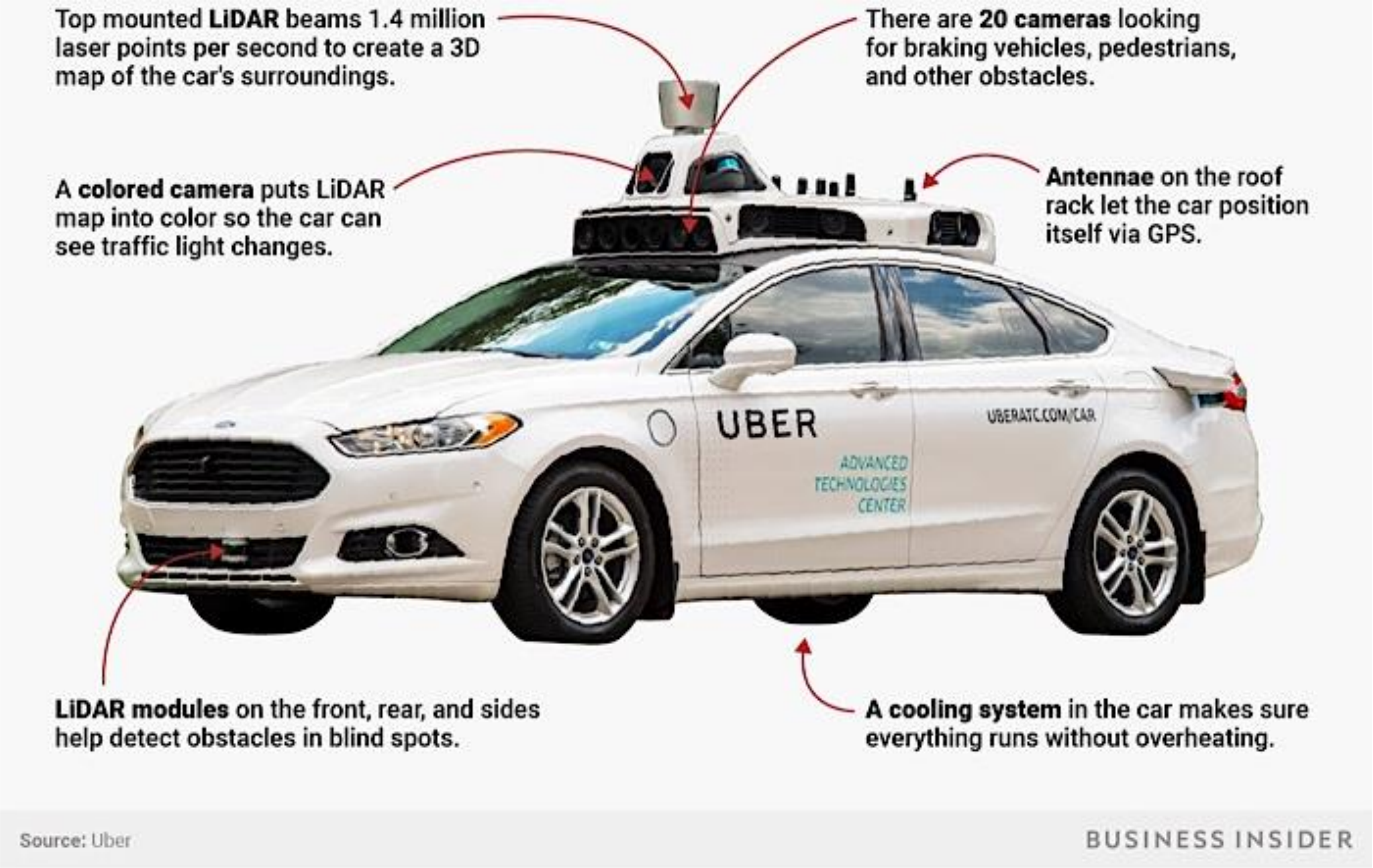
3D Scene Reconstruction From 2D Images

- Depth from focus
- Stereo vision: two images taken at different locations at the same time
- Structure from motion: two images of the same object taken at different times

Image Processing and Understanding

- Pixel data need to be converted into useful features
- Common operations
  - Image filtering, enhancement, compression
  - Geometric feature extraction
    - corner, edge, plane, etc.
- Deep learning computer vision techniques

Example: Self-Driving Car



Outline

- Sensors Overview
- Regression Overview
- Neural Networks Overview

Regression

- Supervised learning / classification / regression
  - Give data $(x_1, y_1), \ldots, (x_n, y_n)$, choose a function f such that $y \approx f(x)$
  - x_i are the inputs/independent variables
  - y_i are the outputs/dependent variables

- Due to noise of measurements, choose f such that $f(x) \approx y$
  - Choose f from a class of functions with parameter f_θ
  - $\mathrm{minimize}_\theta \underbrace{\sum_i |f_\theta(x_i) - y_i|^2}_{\text{“Loss function”}}$

Another choice: $\sum_i |f_\theta(x_i) - y_i|^1$

Linear Regression

- Scalar example: line fitting
  - $y = f_{m,b}(x) = mx + b$
  - Data: $\{(x_i, y_i)\}_{i=1}^{N}$

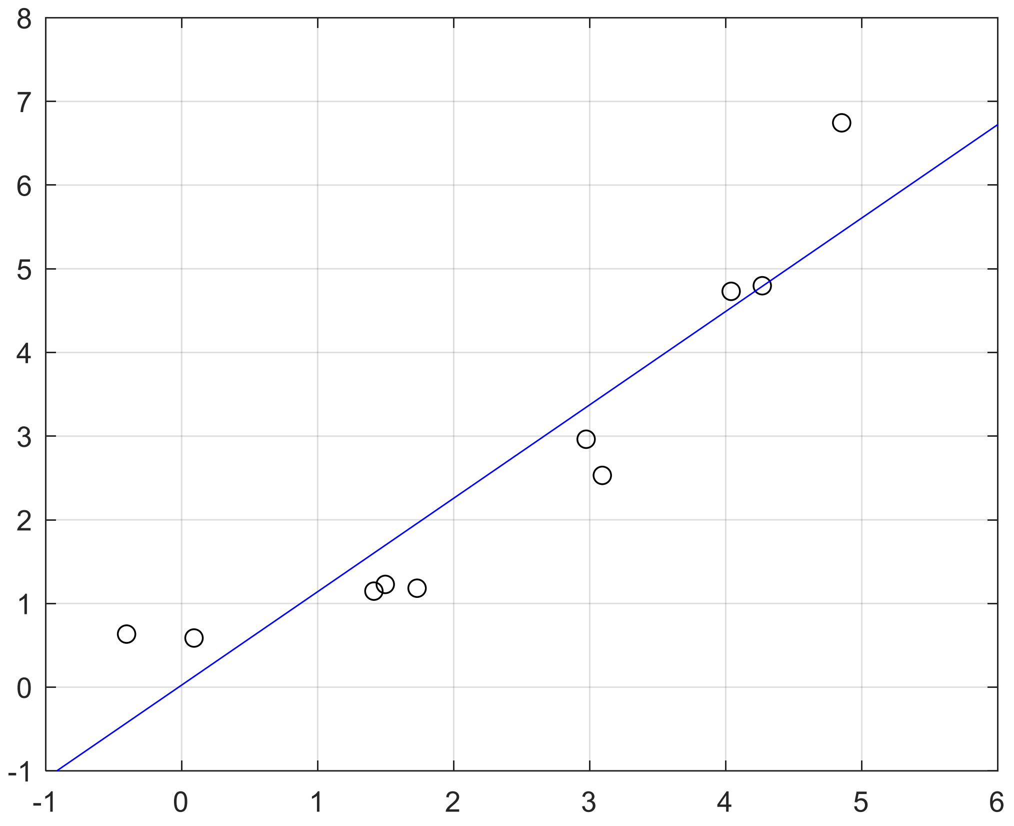
- $\text{minimize}_{m,b} \ \sum_i |mx_i + b - y_i|^2$
  - Let $\hat{X} = (x_1, x_2, \ldots, x_N), \ Y = (y_1, y_2, \ldots, y_N)$
  - Let $\theta = (m, b), X = [\hat{X} \quad 1_{N\times 1}]$
  - Differentiate w.r.t. θ and set to zero

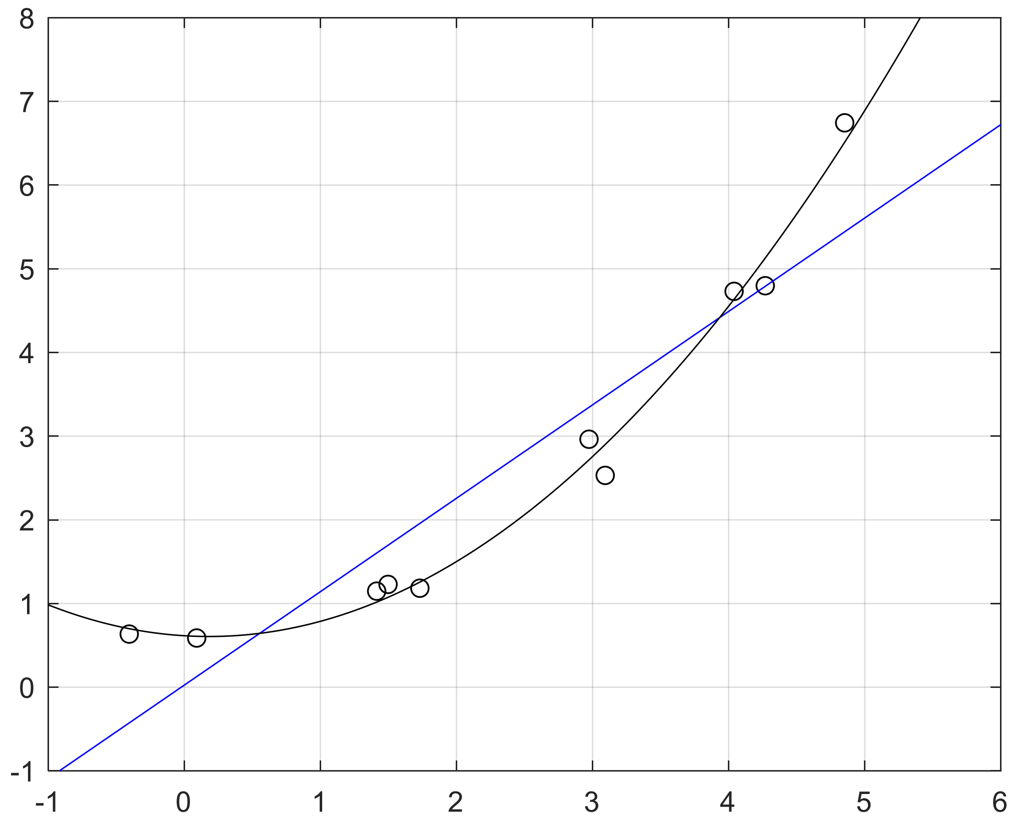
$$\sum_i |mx_i + b - y_i|^2 = \left\|m\hat{X} + b - Y\right\|^2$$

$$\left\|m\hat{X} + b - Y\right\|^2 = \|X\theta - Y\|^2$$

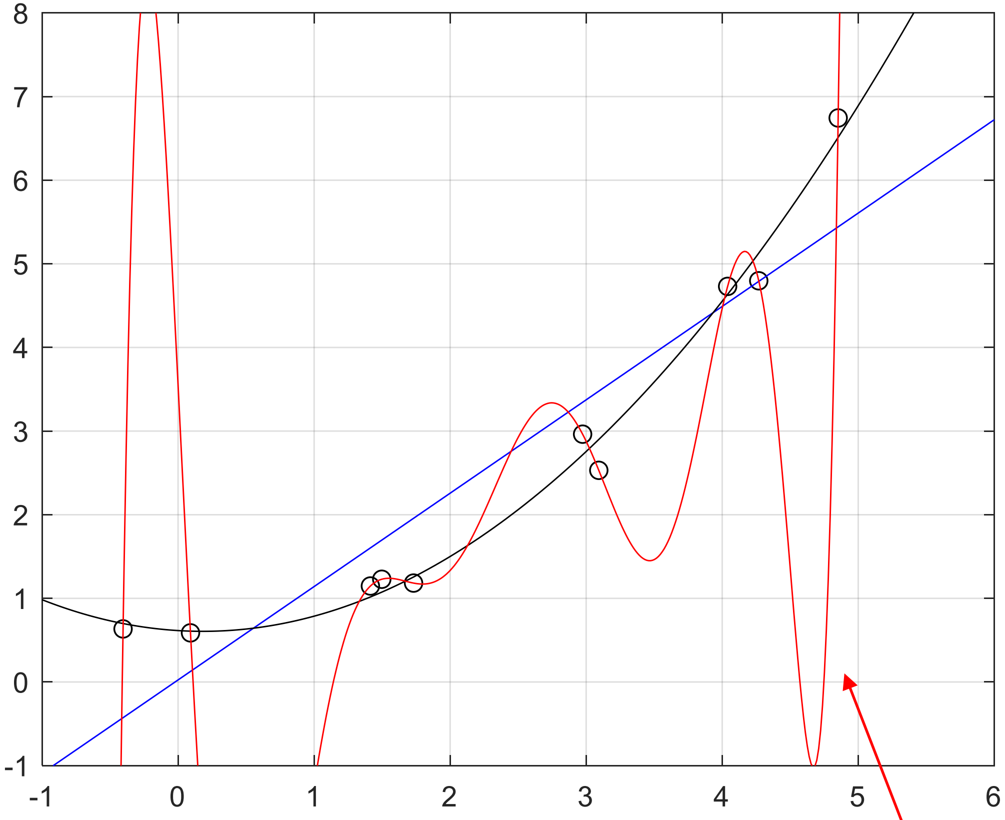
$$2X^\top(X\theta - Y) = 0$$
$$\Rightarrow \theta^* = (X^\top X)^{-1} X^\top Y$$

Regression





Regularization

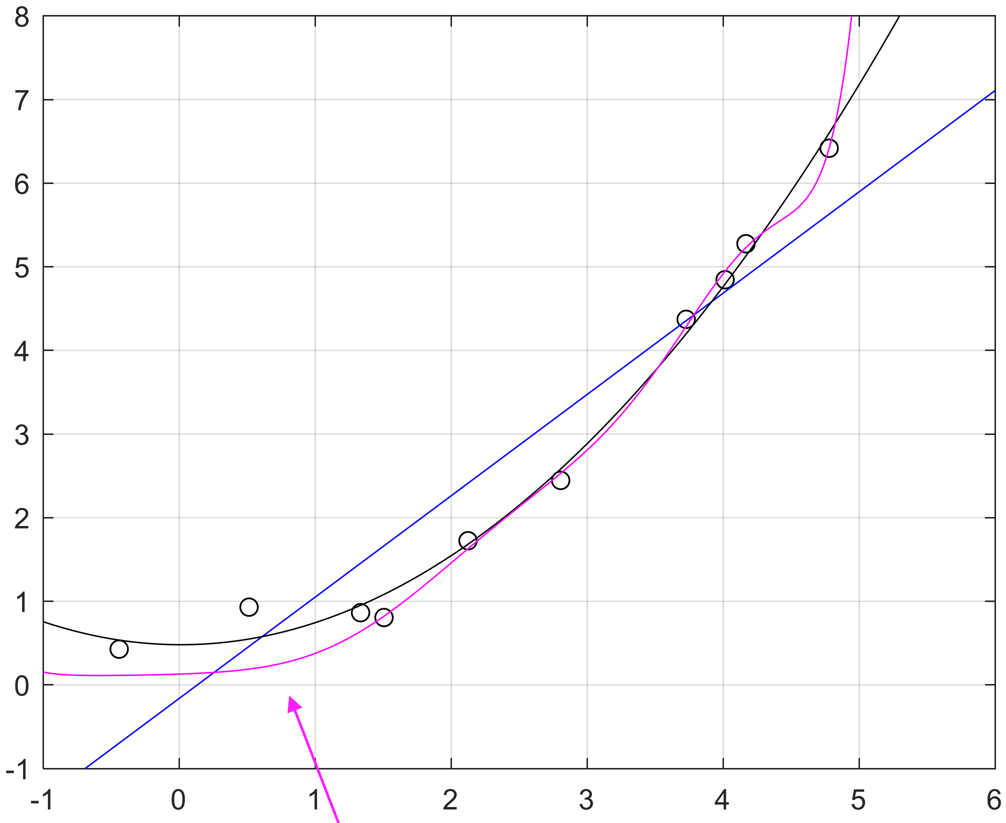


- Penalize size of parameters
  - $\text{minimize}_\theta \|X\theta - Y\|^2 + \lambda\|\theta\|^2$

$$2X^\top(X\theta - Y) + 2\lambda\theta = 0$$
$$X^\top X\theta - X^\top Y + \lambda\theta = 0$$
$$(X^\top X + \lambda I)\theta = X^\top Y$$
$$\theta = (X^\top X + \lambda I)^{-1}X^\top Y$$

10th order polynomial given by this θ

Regularization

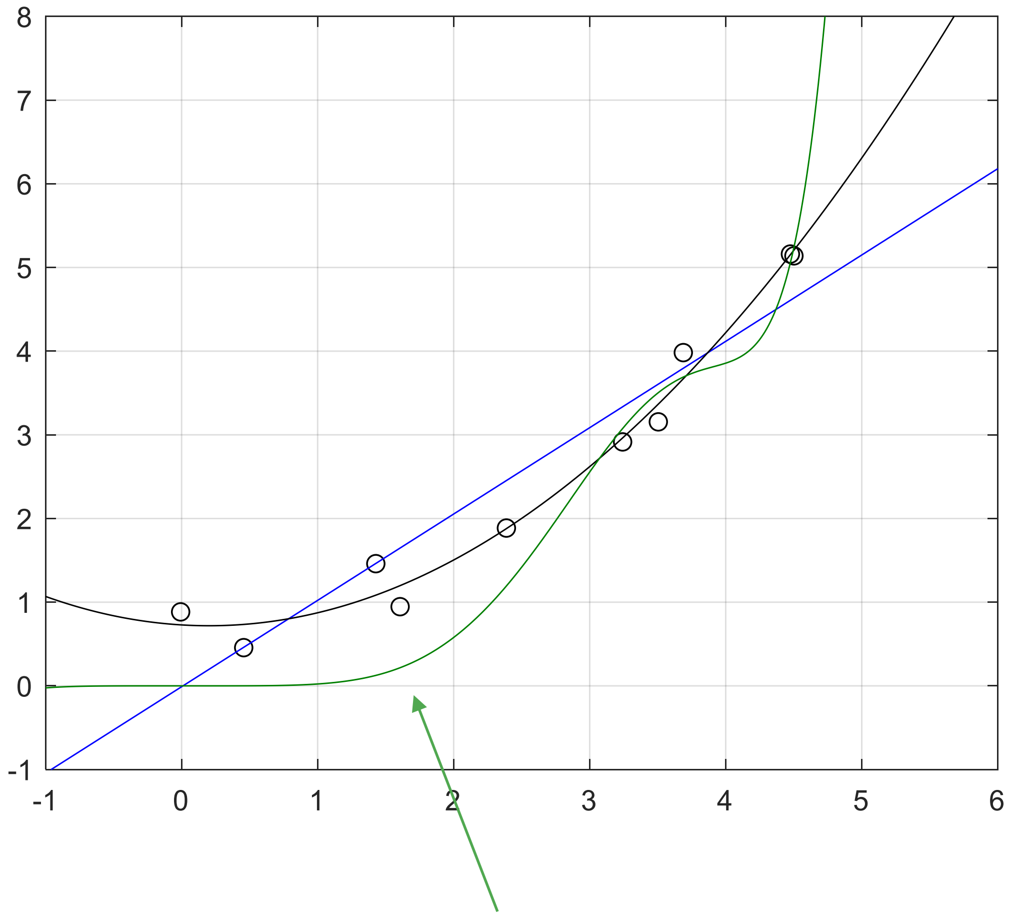


10th order polynomial given by this θ

- Penalize size of parameters
  - $\text{minimize}_\theta \|X\theta - Y\|^2 + \lambda\|\theta\|^2$

$$2X^\top(X\theta - Y) + 2\lambda\theta = 0$$
$$X^\top X\theta - X^\top Y + \lambda\theta = 0$$
$$(X^\top X + \lambda I)\theta = X^\top Y$$
$$\theta = (X^\top X + \lambda I)^{-1}X^\top Y$$

Regularization



10th order polynomial given by this θ

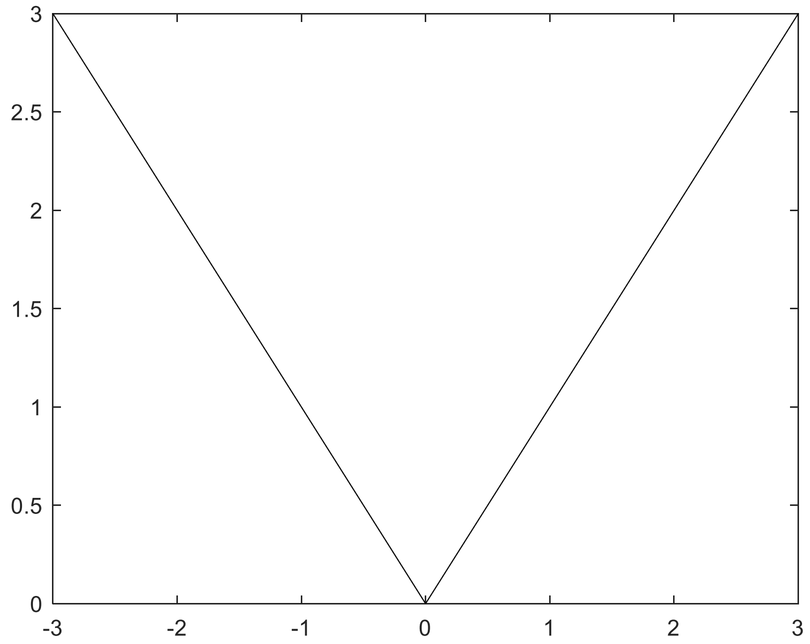
- Penalize size of parameters
  - $\text{minimize}_\theta \|X\theta - Y\|^2 + \lambda\|\theta\|_1$
    - Optimize using gradient descent
    - 1-norm encourages sparsity

```
Theta_opt =
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0232
    0.0000
    0.0000
   -0.0010
    0.0002
   -0.0000
```

Regularization

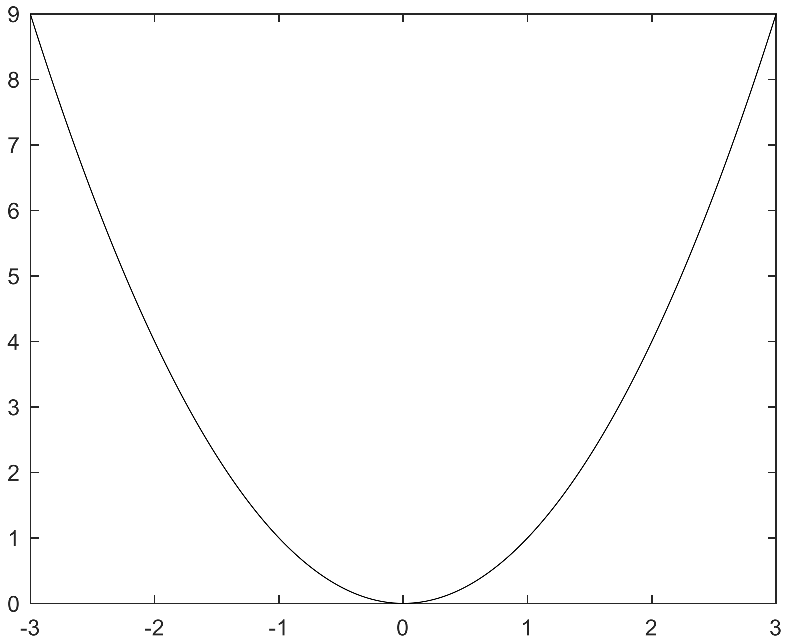
L1: $\|\boldsymbol{\theta}\|_1 = \sum_i |\boldsymbol{\theta}_i|$

- Does not prioritize reduction of any component of θ
- Encourages sparsity



L2: $\|\boldsymbol{\theta}\|_2 = \sum_i \boldsymbol{\theta}_i^2$

- Prioritizes reduction of large components of θ



Regression

- In general, minimize "**loss function**":
  - $\text{minimize}_\theta\, l(\theta; X, Y)$, where $l\colon \mathbb{R}^n \to \mathbb{R}^m$
  - Scalar case: $X = (x_1, x_2, \dots, x_N) = [x_1, x_2, \dots, x_N]^\top$
  - General case: $X = \begin{bmatrix} x_1^\top \\ x_2^\top \\ \vdots \\ x_N^\top \end{bmatrix}$, where $x_i \in \mathbb{R}^n$, $Y = \begin{bmatrix} y_1^\top \\ y_2^\top \\ \vdots \\ y_N^\top \end{bmatrix}$, where $y_i \in \mathbb{R}^m$

Stochastic Gradient Descent

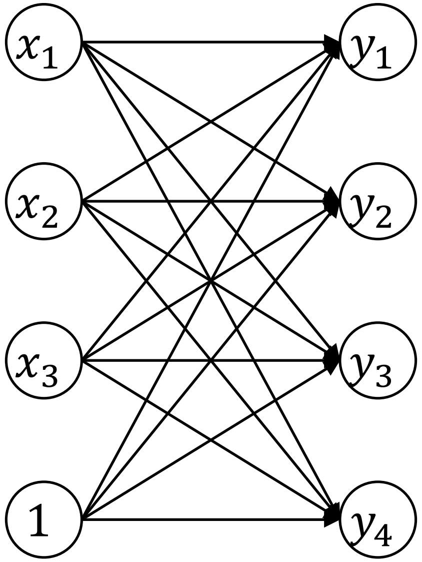
- Gradient descent: $\theta^{k+1} = \theta^k - \alpha^k \nabla l(\theta)$
  - If f has many parameters, then the gradient $\nabla l(\theta)$is difficult to compute
- Idea: Only compute a few components of the gradient
- Which components?
  - Cyclical choice: eg. Components 1 and 2, then 3 and 4, etc.
  - Random choice: **stochastic gradient descent**

Outline

- Sensors Overview
- Regression Overview
- Neural Networks Overview

Neural Networks

- A specific form of $f_\theta(x)$



$y_1 = f(x^\top w_1 + b_1)$

$y_2 = f(x^\top w_2 + b_2)$

$y_3 = f(x^\top w_3 + b_3)$

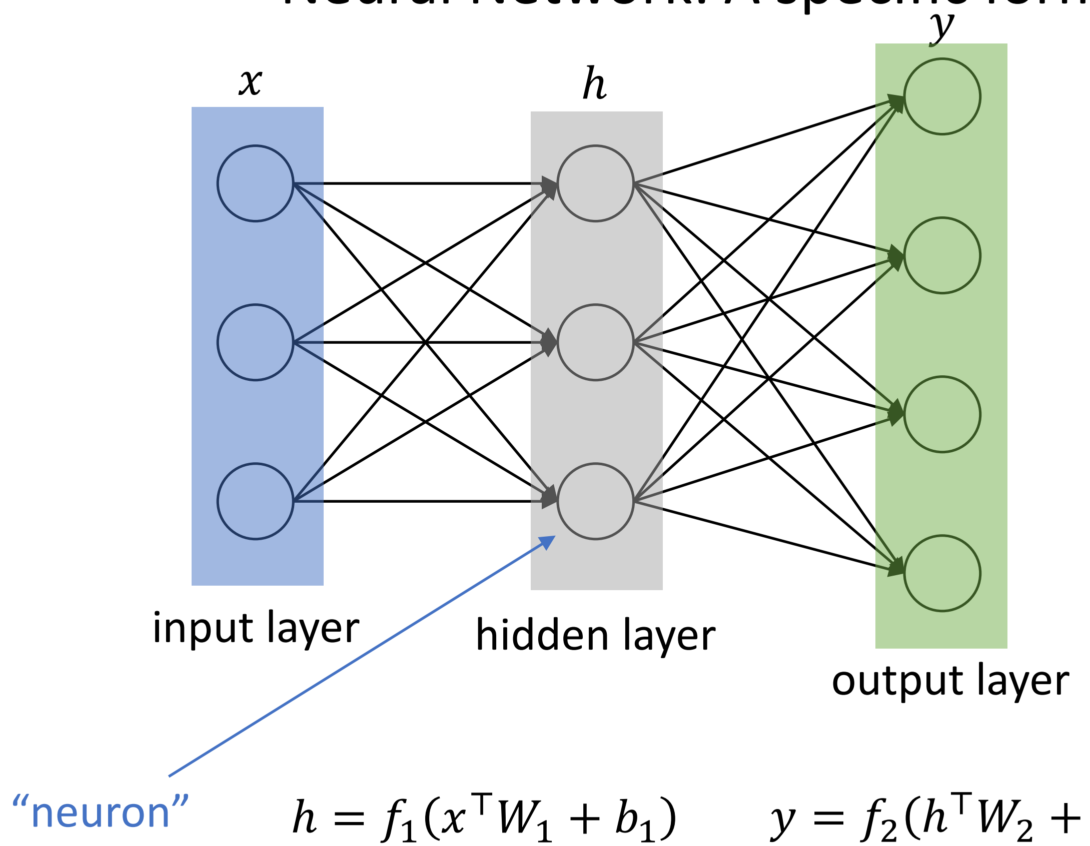
$y_4 = f(x^\top w_4 + b_4)$

$$y = f(x^\top W + b)$$

- Parameters θ are W and b
- "**Weights**"

Neural Networks

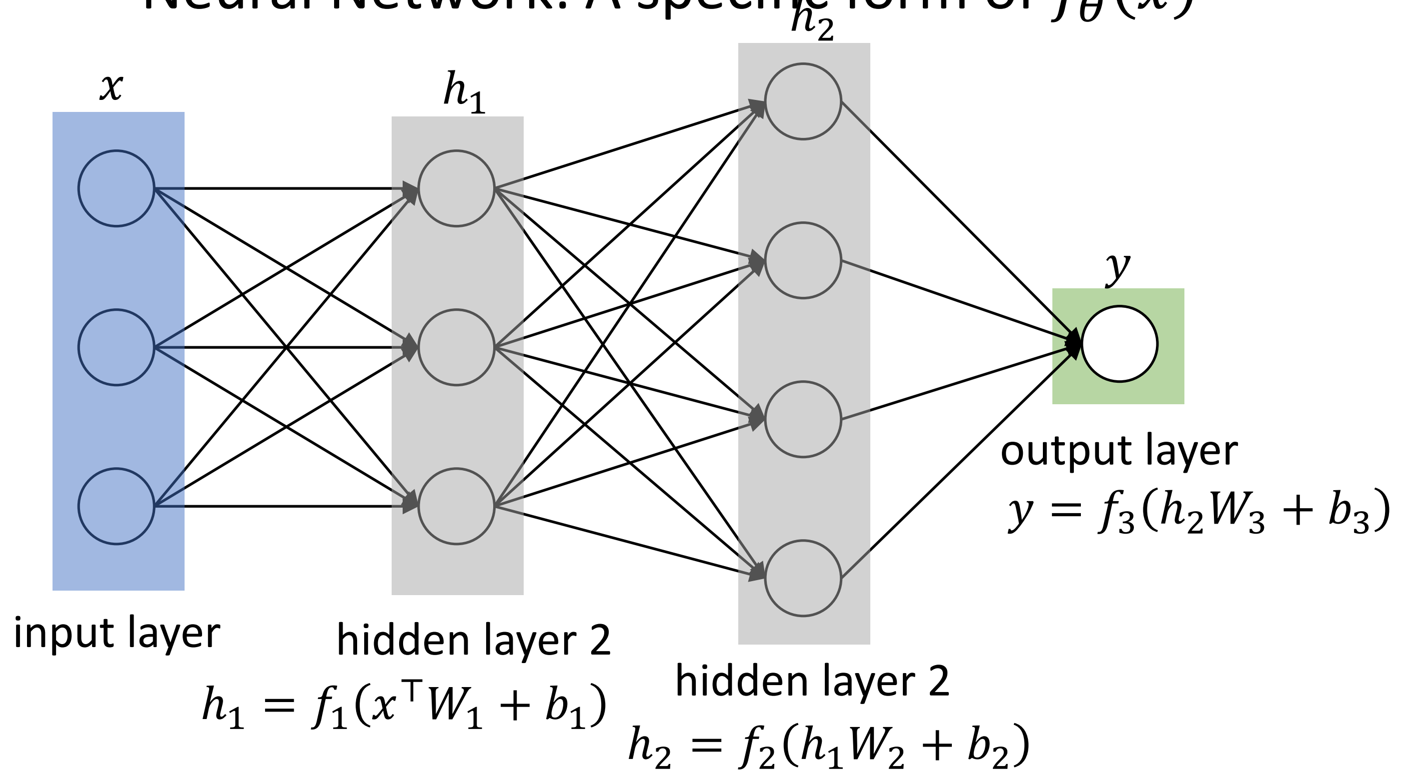
- Regression: Choose θ such that $y \approx f_\theta(x)$
  - Neural Network: A specific form of $f_\theta(x)$



$h = f_1(x^\top W_1 + b_1)$ $\quad$ $y = f_2(h^\top W_2 + b_2)$

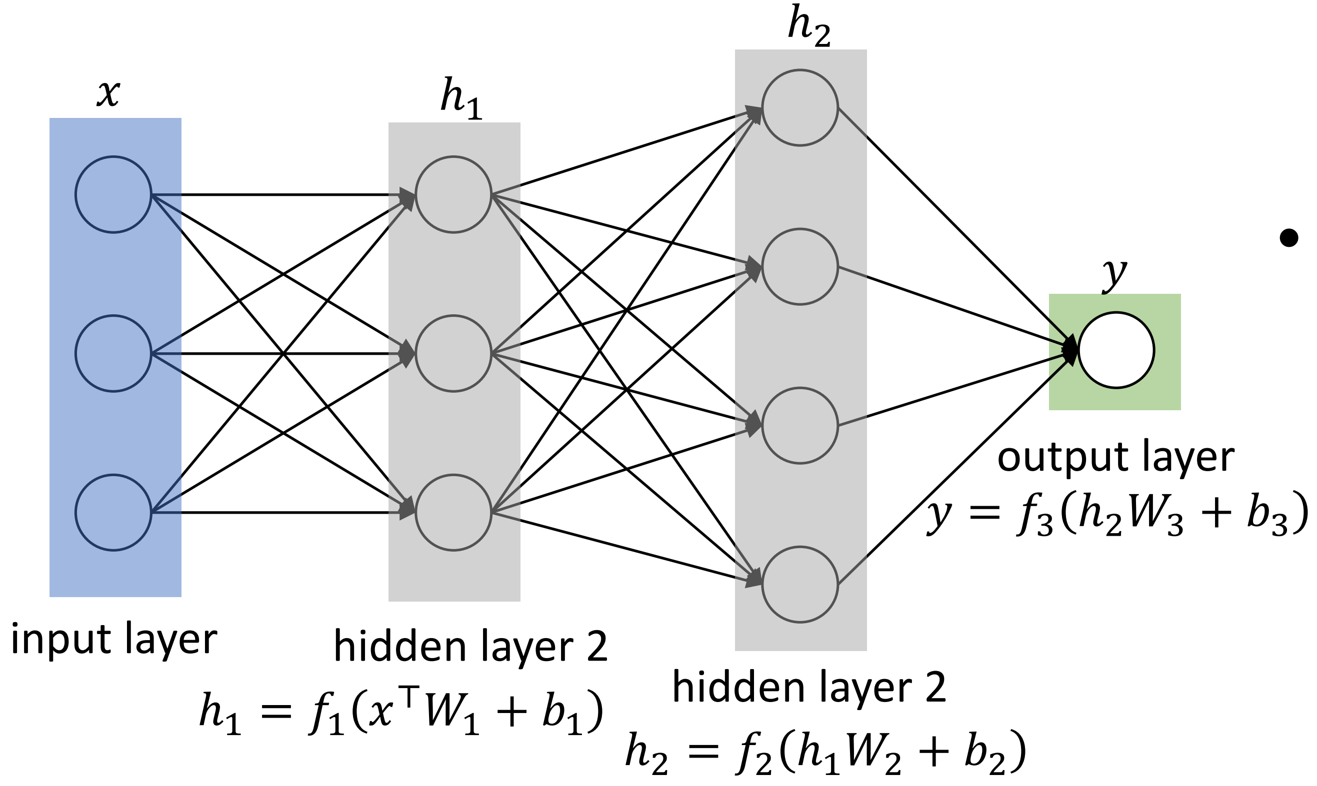
Neural Networks

- Regression: Choose θ such that $y \approx f_\theta(x)$
  - Neural Network: A specific form of $f_\theta(x)$



Neural Networks

- Regression: Choose θ such that $y \approx f_\theta(x)$
  - Neural Network: A specific form of $f_\theta(x)$



- Parameters θ are the weights W_i and b_i

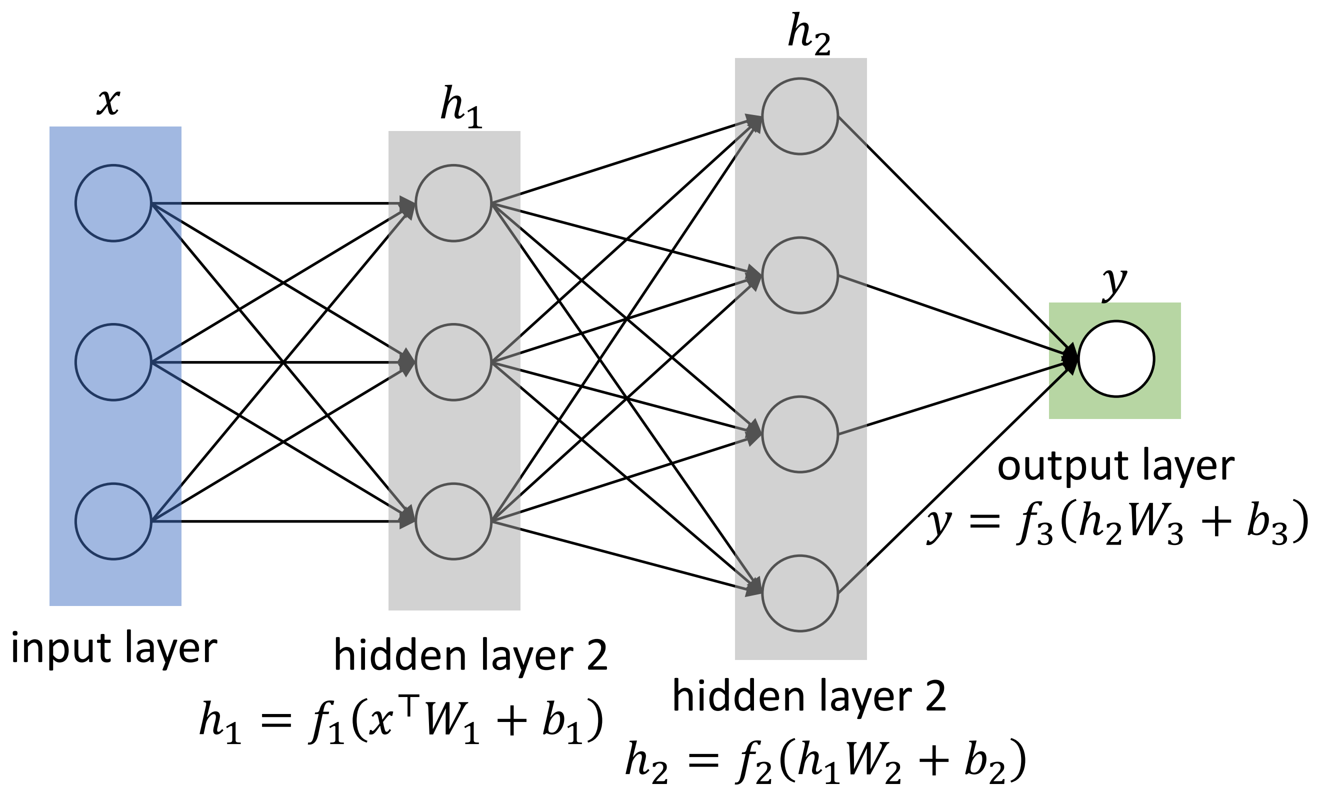
- f_1, f_2, f_3 are nonlinear
  - Otherwise f would just be a single linear function:

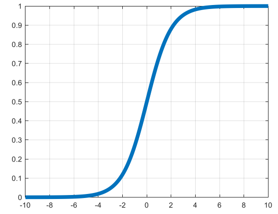
$$\begin{aligned} y &= \big((x^\top W_1 + b_1)W_2 + b_2\big)W_3 + b_3 \\ &= x^\top W_1 W_2 W_3 + b_1 W_2 W_3 + b_2 W_3 + b_3 \end{aligned}$$

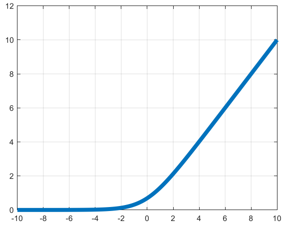
  - “Activation functions”

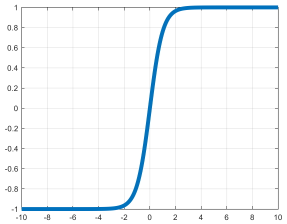
Neural Networks

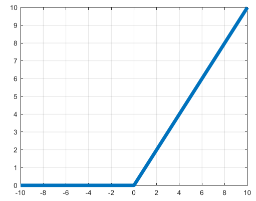
- Regression: Choose θ such that $y \approx f_\theta(x)$
  - Neural Network: A specific form of $f_\theta(x)$



- Common choices of activation functions
  - Sigmoid:
    $$\frac{1}{1+e^{-x}}$$


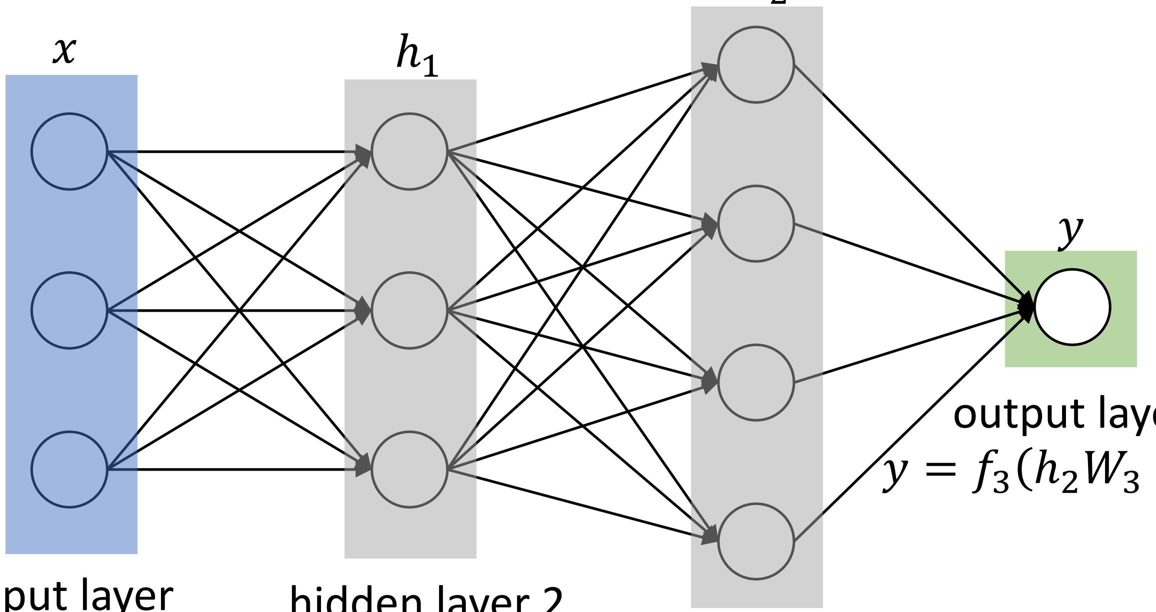
  - Softplus:
    $$\log(1+e^x)$$


  - Hyperbolic tangent:
    $$\tanh x$$


  - Rectified linear unit (ReLU):
    $$\max(0, x)$$


Training Neural Networks

- Regression: Choose θ such that $y \approx f_\theta(x)$
  - Neural Network: A specific form of $f_\theta(x)$



- Given current θ, X, Y, compute $l(\theta; X, Y)$
  - Compares $f_\theta(X)$ with ground truth Y
  - Evaluation of f: "**Forward propagation**"

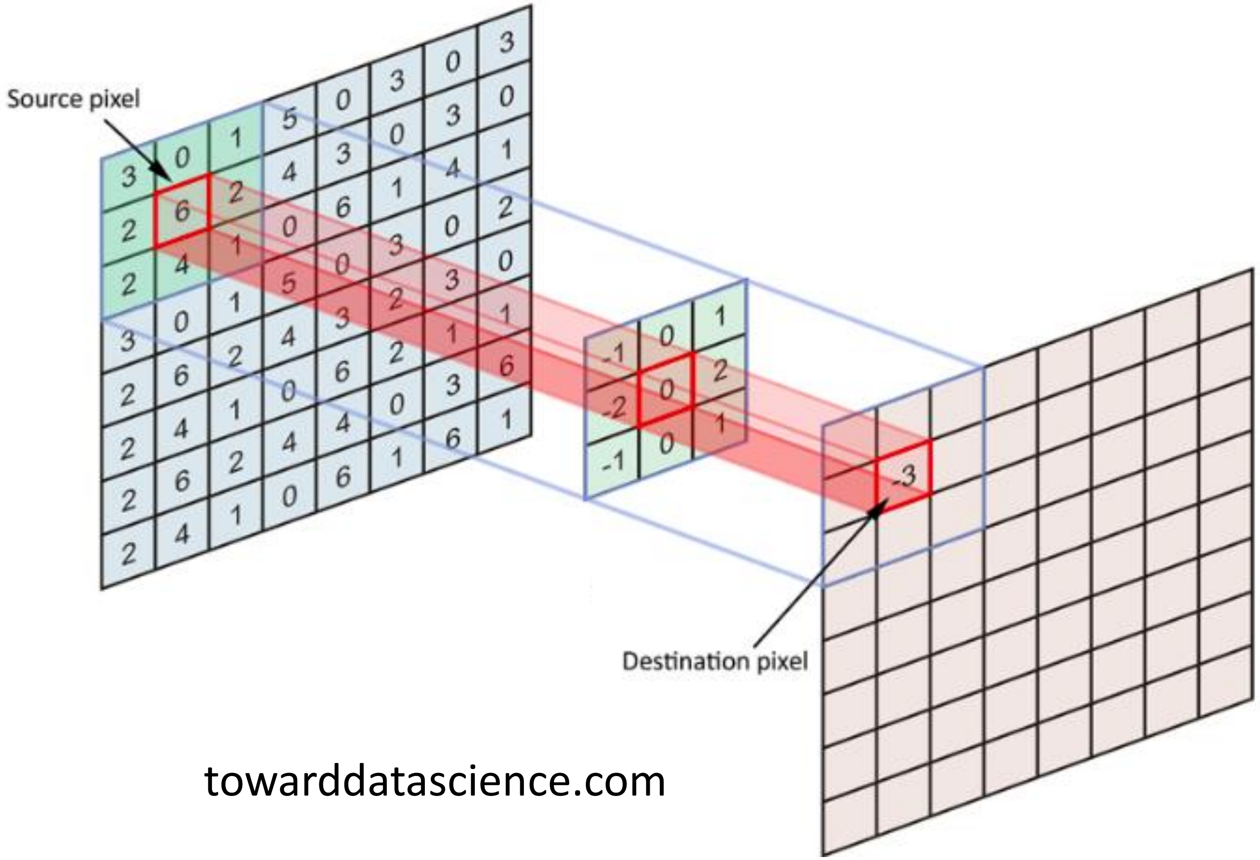
- Minimize $l(\theta; X, Y)$
  - Stochastic gradient descent

  - Computing the gradient
    - Chain rule: $\frac{dy}{dx} = \frac{dy}{h_2} \times \frac{dh_2}{dh_1} \times \frac{dh_1}{dx}$

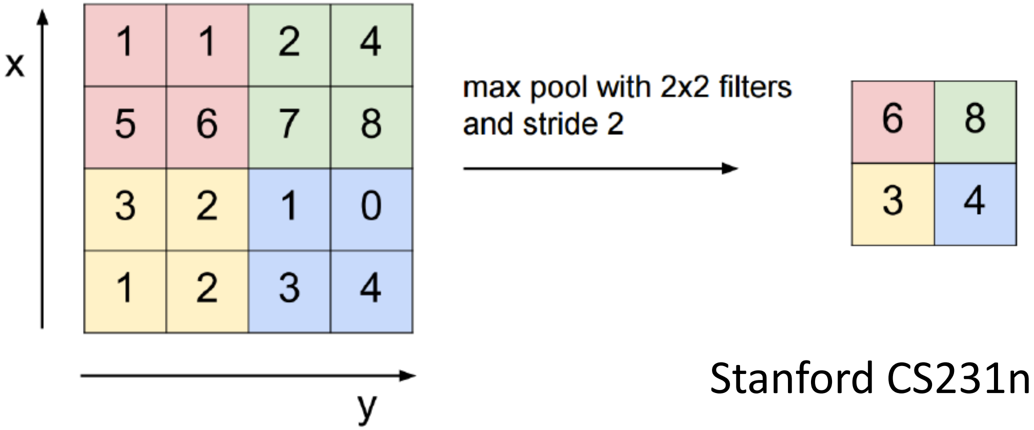
      Matrices
    - Evaluation of $\frac{dy}{dx}$: "**Back propagation**"

Common Operations

- Fully connected (dot product)
- Convolution
  - Translationally invariant
  - Controls overfitting
- Pooling (fixed function)
  - Down-sampling
  - Controls overfitting
- Nonlinearity layer (fixed function)
  - Activation functions, e.g. ReLU

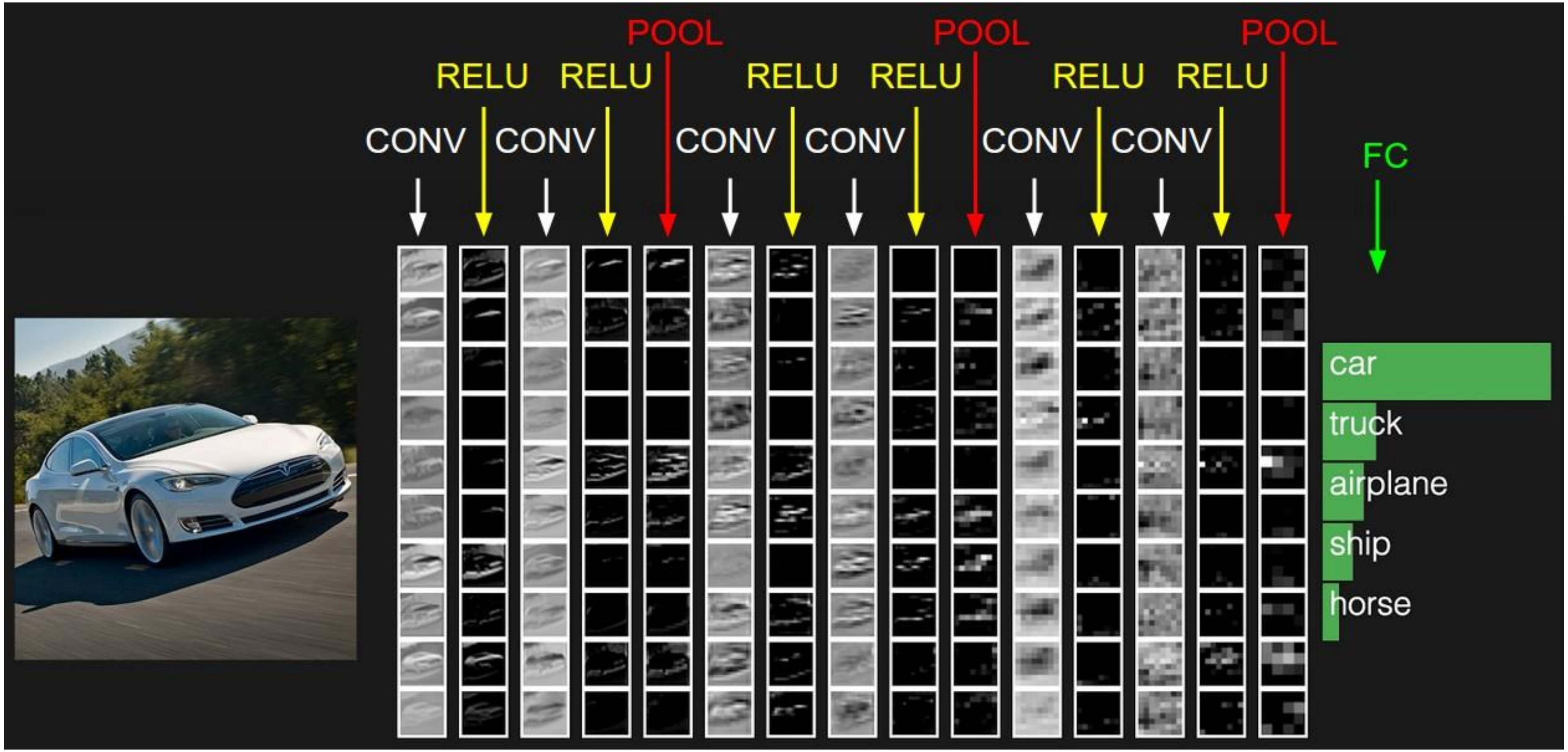


towarddatascience.com



Stanford CS231n

Example: Small VGG Net From Stanford CS231n



Neural Network Architectures

- Convolutional neural network (CNN)
  - Has translational invariance properties from convolution
  - Common used for computer vision
- Recurrent neural network RNN
  - Has feedback loops to capture temporal or sequential information
  - Useful for handwriting recognition, speech recognition, reinforcement learning
  - Long short-term memory (LSTM): special type of RNN with advantages in numerical properties
- Others
  - General feedforward networks, variational autoencoders (VAEs), conditional VAEs,

Training Neural Networks

- Training process (optimization algorithm)
  - Standard L1 and L2 regularization
  - Dropout: randomly set neurons to zero in each training iteration
  - Transform input data (e.g. rotating, stretching, adding noise)
  - Learning rate (step size) and other hyperparameter tuning

- Software packages: Efficient gradient computation
  - Caffe, Torch, Theano, Tensor Flow