Image Representation and Neural Networks

Andrew Kondrich

August 2, 2019

Class schedule

1. Linear Models for Regression and Classification
2. Optimization
3. Vectors
4. Matrices
5. Image Representation
6. Image Classification
7. Challenges of Computer Vision
8. Neural Networks

Curriculum Plan

1. Linear Models for Regression and Classification
2. Optimization
3. Vectors
4. Matrices
5. Image Representation
6. Image Classification
7. Challenges of Computer Vision
8. Neural Networks

Linear Models for Regression

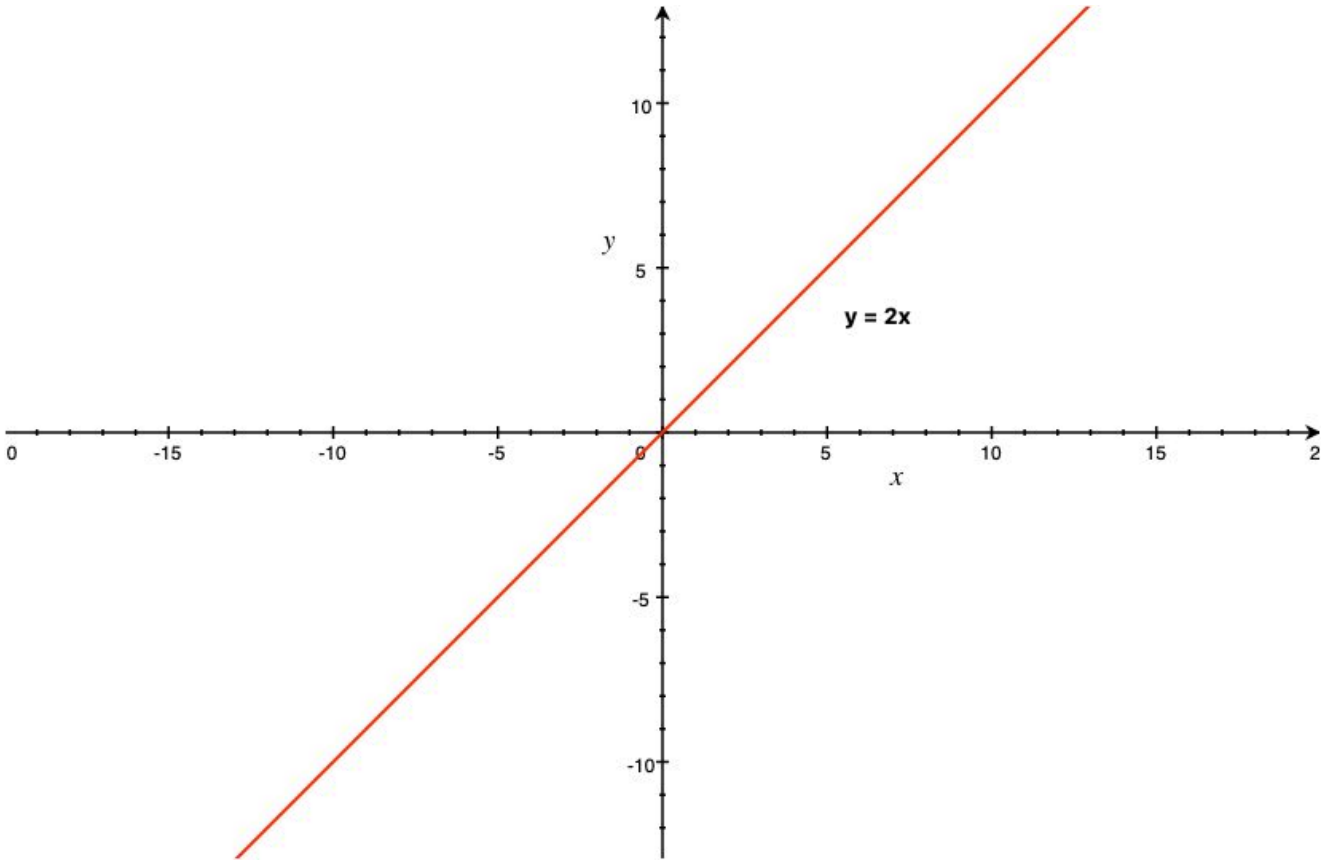


Figure: y is given as a function of x in the form $y = mx + b$

Linear Models for Regression

- Data is a collection of paired points such as

$$(0, 0), (1, 1), (2.5, 2), (5, 6), ...$$

- We assume data can be fit by a linear hypothesis function

$$y = w_1 x + b$$

- **Our job:** Given just this data, how to choose w_1 and b?

Linear Models for Regression

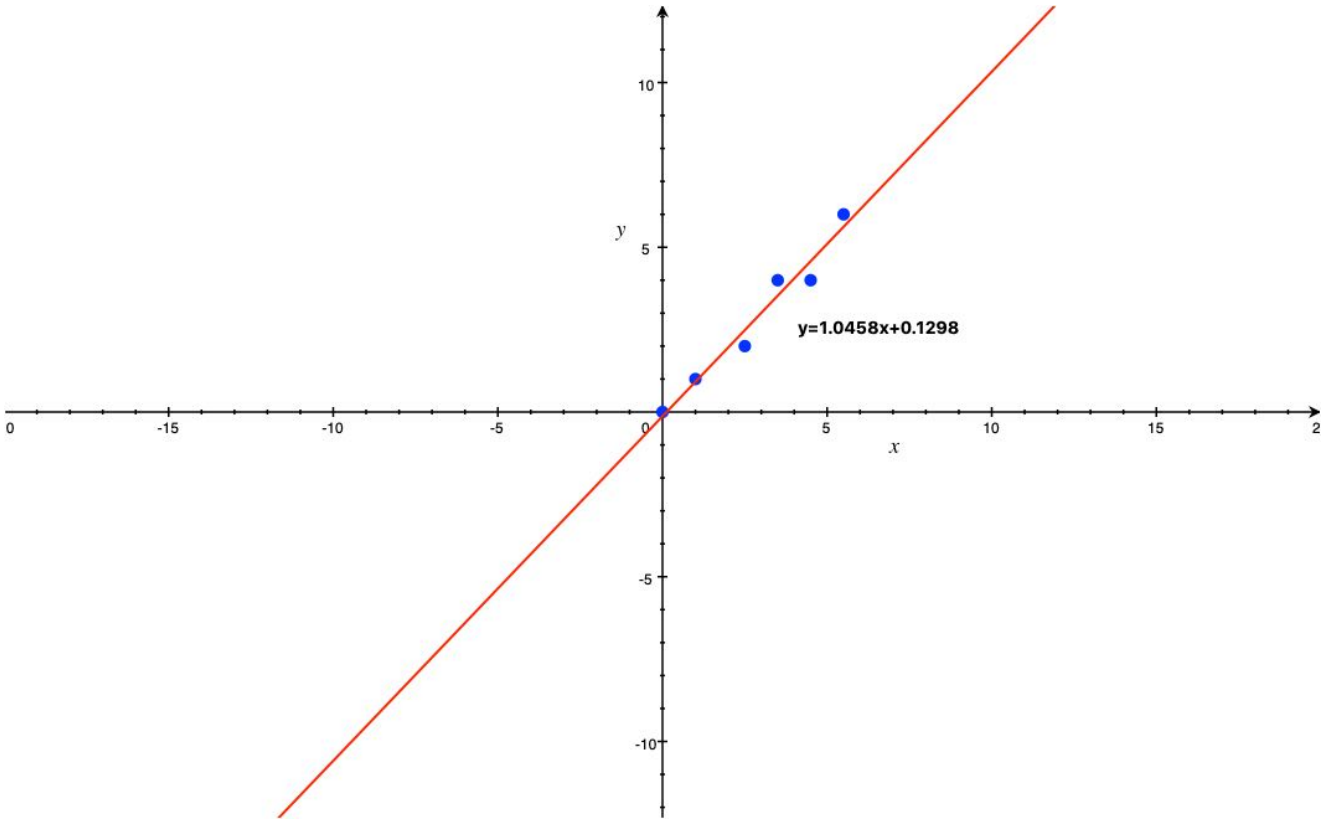


Figure: Line interpolated for 2D blue point dataset

Linear Models for Regression

- First, we generalize the problem statement a little bit...
- You can have more than just one feature x.
- Instead of just using ft^2 to predict price of house, you also include which city your house is in because apartment in San Francisco is small but expensive!
- Data of any number of x can be fit by a linear hypothesis function

$$y = w_0 + w_1 x_1 + ... + w_D x_D$$

Linear Models for Regression

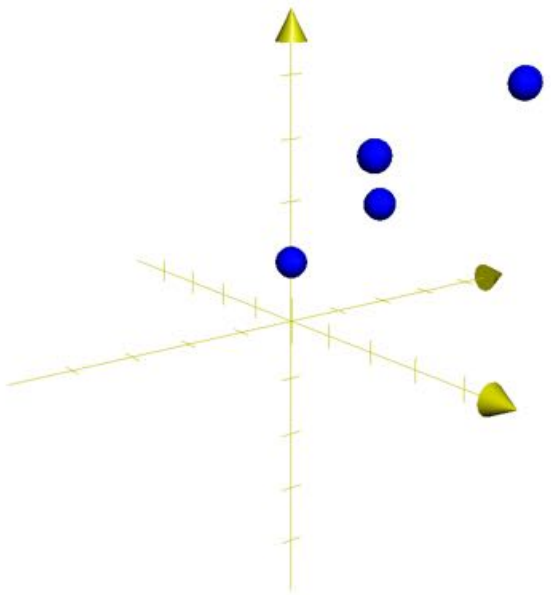


Figure: 3D blue point dataset (x_1, x_2, y)

Linear Models for Regression

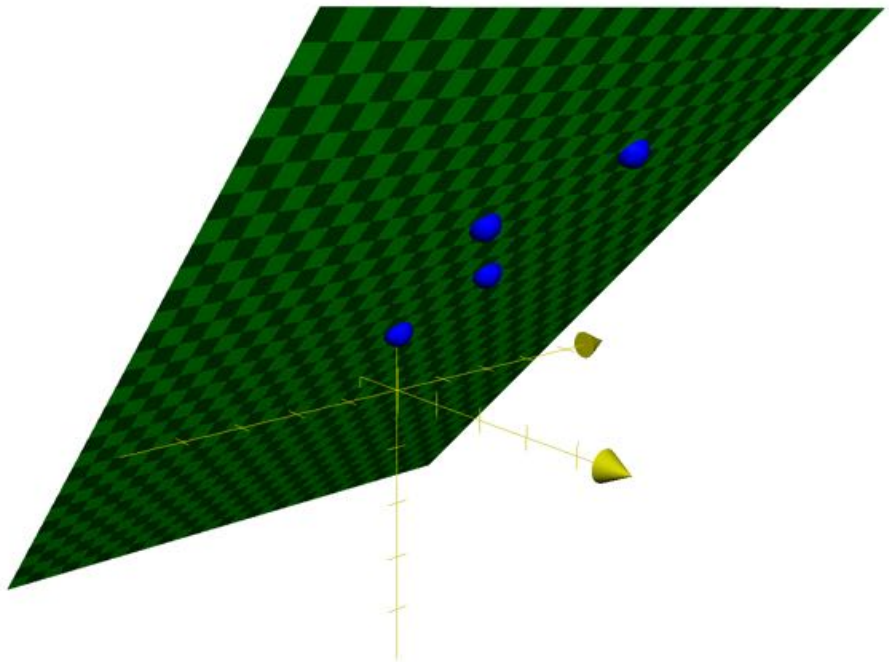


Figure: Regressed *plane*: $y = x_1 - x_2 + 1$

Linear Models for Classification

- In the regression case, we take input $(x_1, x_2, ..., x_D)$ and return the point y that fits our linear hypothesis function

$$y = w_0 + w_1 x_1 + ... + w_D x_D$$

where

$$y, x_1, x_2, ..., x_D \in \mathbb{R}$$

- The goal in classification is to take input $(x_1, x_2, ..., x_D)$ and assign it to one of K groups.
- Example of K $= 2$?

Linear Models for Classification

- In the regression case, we take input $(x_1, x_2, ..., x_D)$ and return the point y that fits our linear hypothesis function

$$y = w_0 + w_1 x_1 + ... + w_D x_D$$

where

$$y, x_1, x_2, ..., x_D \in \mathbb{R}$$

- The goal in classification is to take input $(x_1, x_2, ..., x_D)$ and assign it to one of K groups.
- Example of $K = 2$?
pass/fail, win/lose, healthy/sick, poor/not poor, hotdog/not hotdog,
- We will continue to focus on $K = 2$ case, known as **binary classification**.

Linear Models for Classification

- Regression returns the "real" numeric value we want, e.g. house price in dollars.
- Less intuitive: How to capture "group assignment" numerically?

Linear Models for Classification

- Regression returns the "real" numeric value we want, e.g. house price in dollars.
- Less intuitive: How to capture "group assignment" numerically?
- Classification models return the **probability** that input belongs to a group.

Linear Models for Classification

- Regression returns the "real" numeric value we want, e.g. house price in dollars.
- Less intuitive: How to capture "group assignment" numerically?
- Classification models return the **probability** that input belongs to a group.
- For binary classification: We need to find some function f defined as

$$p = f(x_1, x_2, ..., x_D)$$

  that takes $(x_1, ..., x_D)$, does some math, and returns a probability p of belonging to Group 1.
- Probability of being in Group 2 is given by $1 - p$.

Linear Models for Classification

- We use a modified version of our linear regression function:

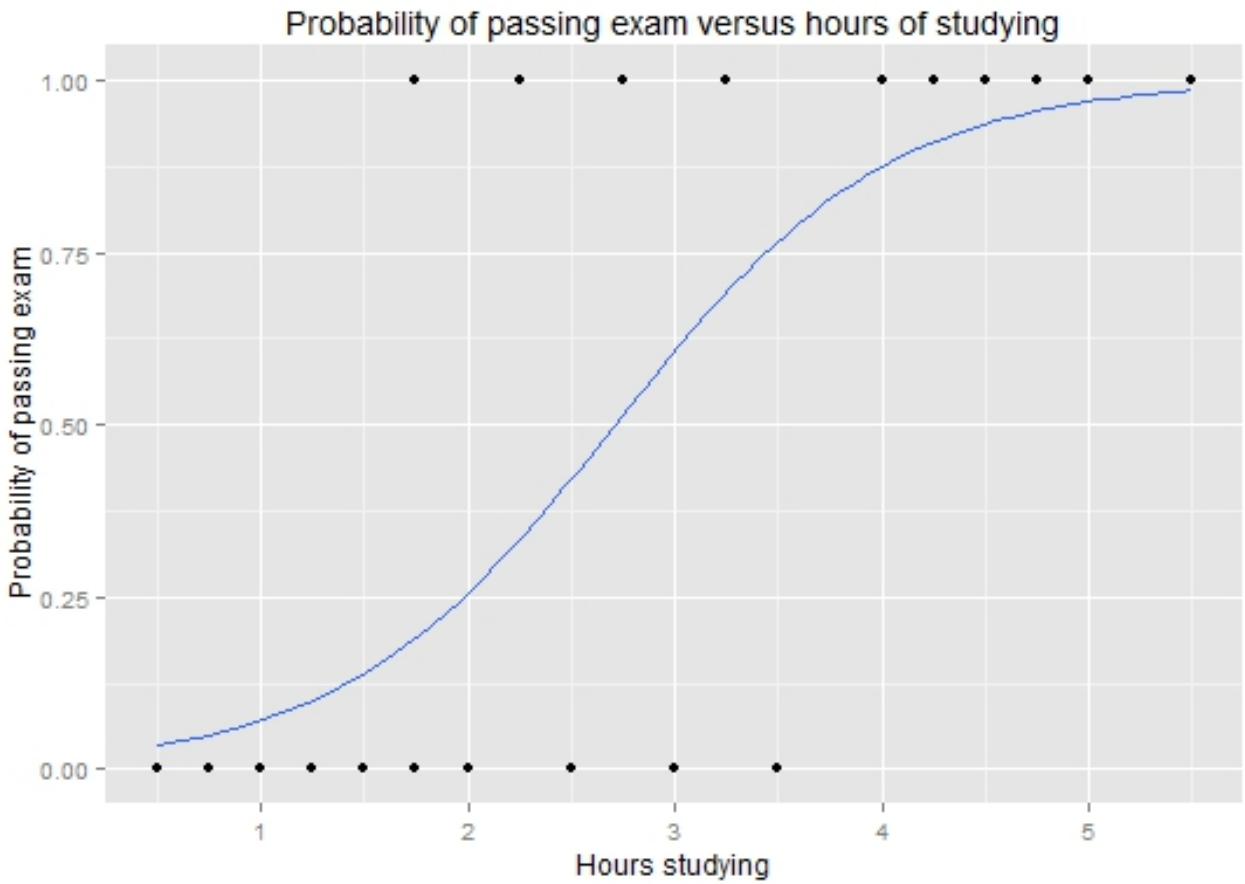
$$y = \sigma(w_0 + w_1 x_1 + ... + w_D x_D)$$

- Where σ is the **sigmoid** function, a type of logistic funciton given by

$$\frac{1}{1 + e^{-x}}$$

- In statistics, this model is known as **logistic regression**, although it should be emphasized that this is a model for classification rather than regression.

Linear Models for Classification



[0]Michaelg2015, commons.wikimedia.org/w/index.php?curid=42442194

Curriculum Plan

1. Linear Models for Regression and Classification
2. Optimization
3. Vectors
4. Matrices
5. Image Representation
6. Image Classification
7. Challenges of Computer Vision
8. Neural Networks

Optimization

- We know how to represent our logistic and linear regression functions:

$$y = (w_0 + w_1 x_1 + ... + w_D x_D)$$

$$y = \sigma(w_0 + w_1 x_1 + ... + w_D x_D)$$

- We need some algorithm for finding what the best choices of $(w_0, ..., w_D)$ are.
- **Gradient Descent** is a process by which we iteratively improve our choices for $(w_0, ..., w_D)$ by computing the **gradient** of the **loss** of our predictions.

Optimization

- **Gradients** equal the slope of the line that is *tangent* to your function.
- Imagine that one of these lines exists for each point along your function.

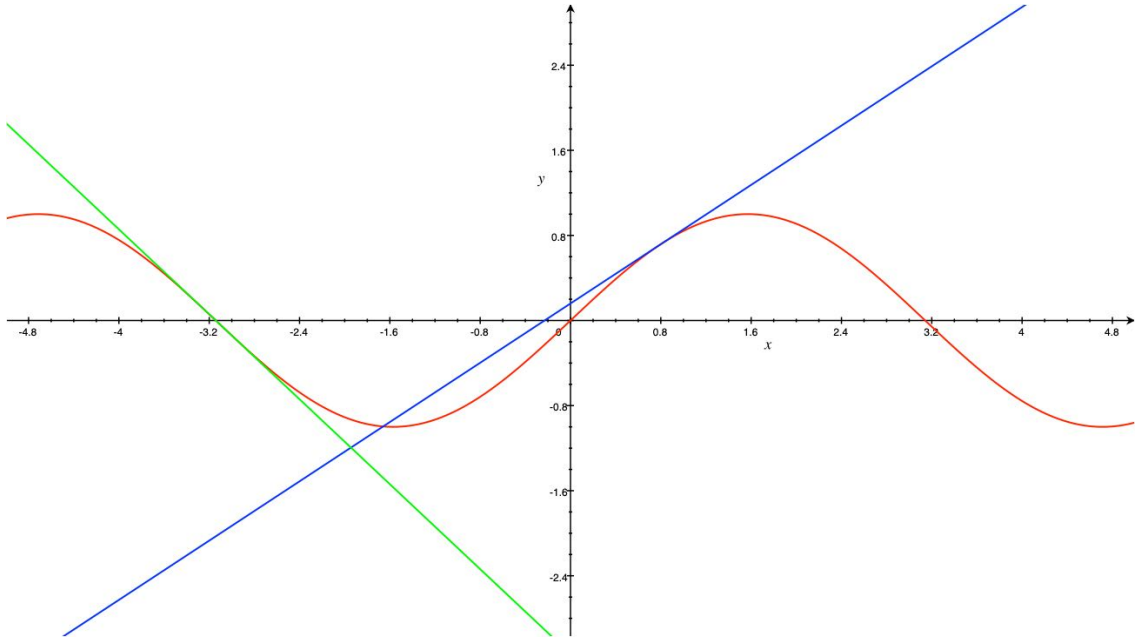


Figure: Lines tangent to $y = sin(x)$ at $x = 0.8$ and $x = -3.2$

- They indicate the direction in which the function is increasing.

Optimization

Physical Metaphor: The function is a hill. A **positive** gradient indicates a ball would roll **left**, a negative gradient indicates a **ball** would roll **right**.

Optimization

- A common loss function is the **Mean Squared Error** $\mathcal{L}(y, \hat{y}) = (y - \hat{y})^2$. Why do you think that is?

Optimization

- A common loss function is the **Mean Squared Error** $\mathcal{L}(y, \hat{y}) = (y - \hat{y})^2$. Why do you think that is?

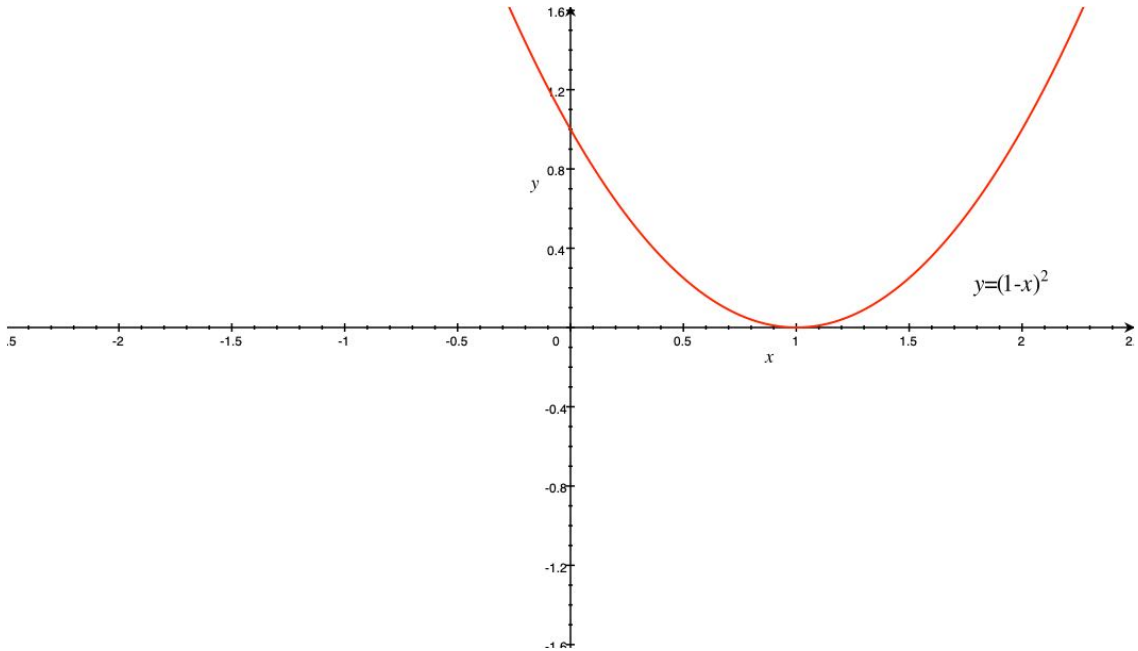


Figure: Graph of $y = (1 - x)^2$

Optimization

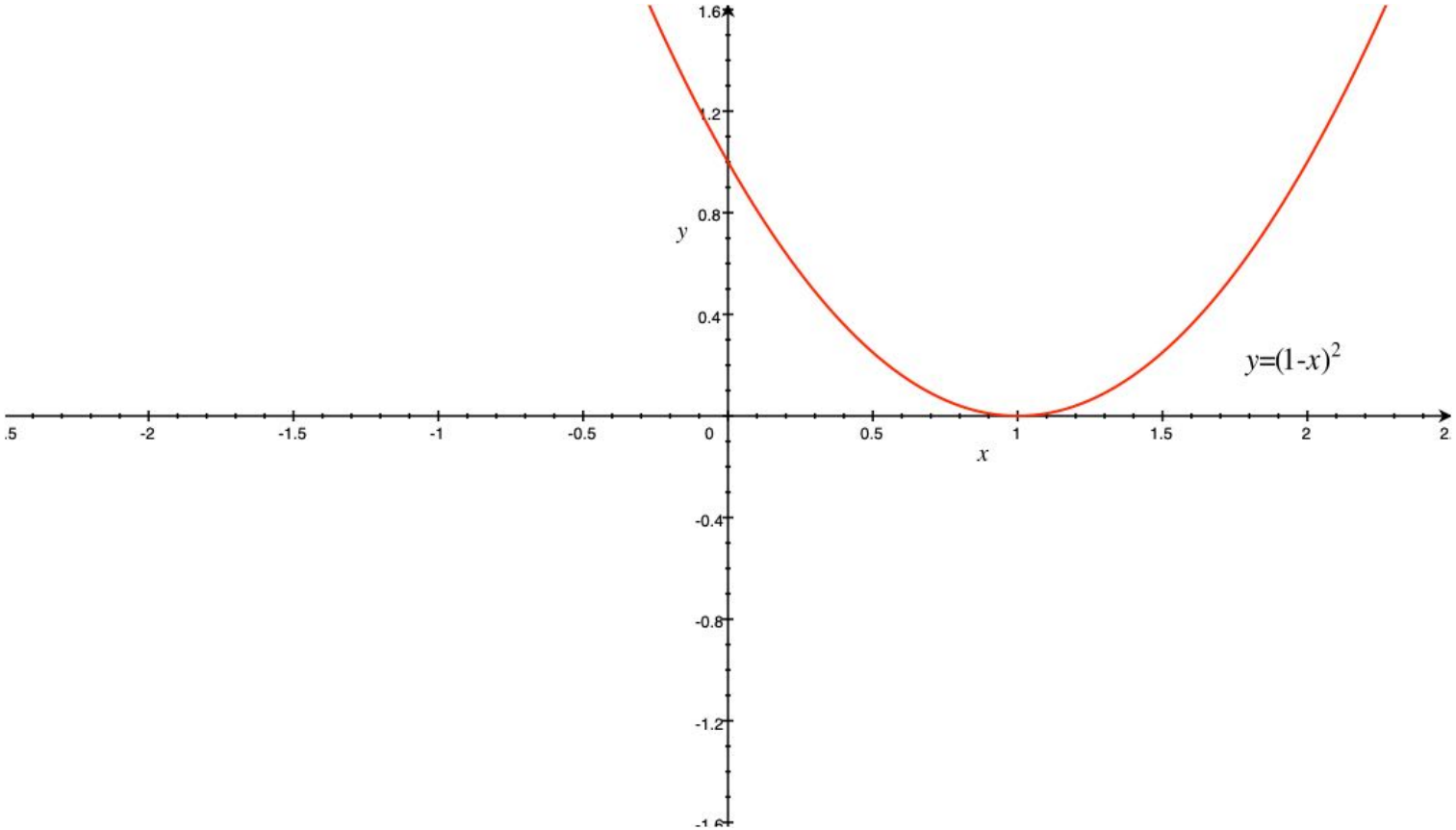


Figure: Graph of $y = (1 - x)^2$

or $\mathcal{L}(y = 1, \hat{y}) = (1 - \hat{y})^2$

Optimization

- **Mean Squared Error** (MSE) defined by $\mathcal{L}(y, \hat{y}) = (y - \hat{y})^2$.
- What is $\hat{y}$? Its the output of linear or logistic regression!

$$\hat{y} = (w_0 + w_1 x_1 + ... + w_D x_D)$$

$$\hat{y} = \sigma(w_0 + w_1 x_1 + ... + w_D x_D)$$

- y is the corresponding label to datapoint $(x_1, x_2, ..., x_D)$!

Optimization

- Collect a dataset of $(y, x_1, x_2, ..., x_D)$ sequences.
- Choose some initial weights randomly, $(w_0, w_1, ..., w_D)$

while *loss is not absolute minimum* **do**
  find $\hat{y} = \sigma(w_0 + w_1 x_1 + ... + w_D x_D)$ for each $(y, x_1, x_2, ..., x_D)$;
  compute MSE $\mathcal{L}(y, \hat{y}) = (y - \hat{y})^2$;
  recompute weights using **gradient** of your loss (shows direction to nudge your weights);
end

Algorithm 1: Gradient Descent Algorithm for Binary Classification

Curriculum Plan

1. Linear Models for Regression and Classification
2. Optimization
3. Vectors
4. Matrices
5. Image Representation
6. Image Classification
7. Challenges of Computer Vision
8. Neural Networks

Vectors

- Recap: Vectors are a sequence of numbers

$$[1.0, 2.0]; [6.0, 5.3, 2.1, -6.7]$$

- Vector Spaces

$$\mathbb{R}^2 = \{(x, y) : x, y \in \mathbb{R}\}$$
$$\mathbb{R}^4 = \{(a, b, c, d) : a, b, c, d \in \mathbb{R}\}$$

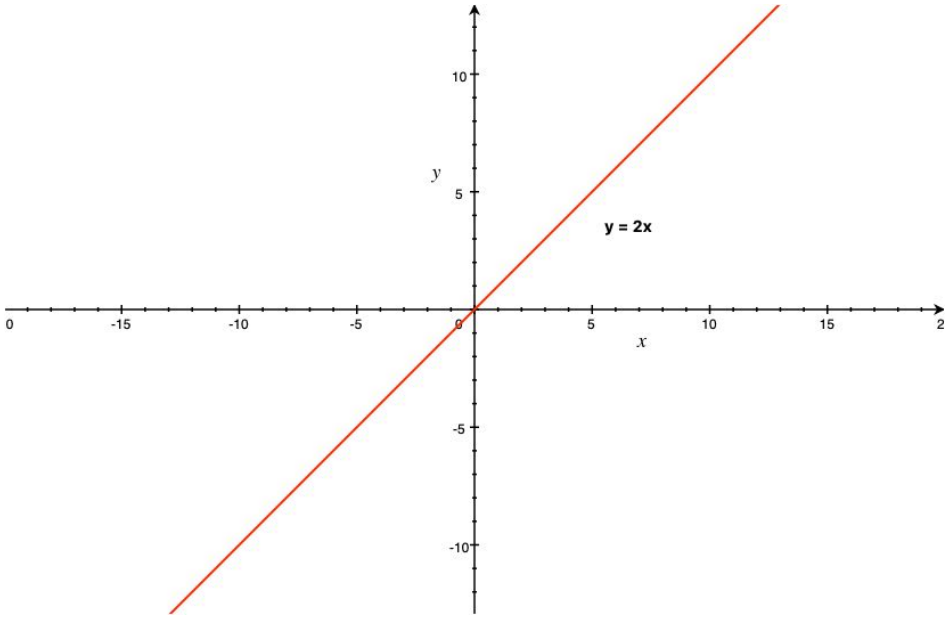


Figure: Vectors in the 2D plane (x, y) are in $\mathbb{R}^2$

Linear Models for Regression

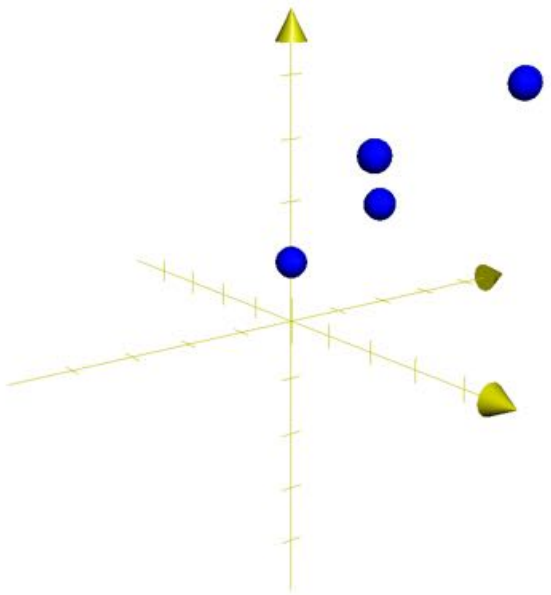


Figure: 3D blue point dataset $(x_1, x_2, y) \in \mathbb{R}^3$

Vectors

- In logistic regression: $\hat{y} = \sigma(w_0 + w_1 x_1 + ... + w_D x_D)$

$$\mathbf{w} = (w_0, ..., w_D) \in \mathbb{R}^D, \mathbf{x} = (x_0, ..., x_D) \in \mathbb{R}^D$$

$$\hat{y} = \sigma(\mathbf{w} \cdot \mathbf{x})$$

Curriculum Plan

1. Linear Models for Regression and Classification
2. Optimization
3. Vectors
4. Matrices
5. Image Representation
6. Image Classification
7. Challenges of Computer Vision
8. Neural Networks

Matrices

- An m-by-n matrix is a rectangular array with m rows and n columns that looks like this:

$$\begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & & \vdots \\ a_{m,1} & \cdots & a_{m,n} \end{bmatrix}.$$

- Use matrices to represent collections of data (each row is a vector)
- Matrices can represent images

Figure: An example matrix filled with numbers

Curriculum Plan

1. Linear Models for Regression and Classification
2. Optimization
3. Vectors
4. Matrices
5. **Image Representation**
6. Image Classification
7. Challenges of Computer Vision
8. Neural Networks

Image Representation

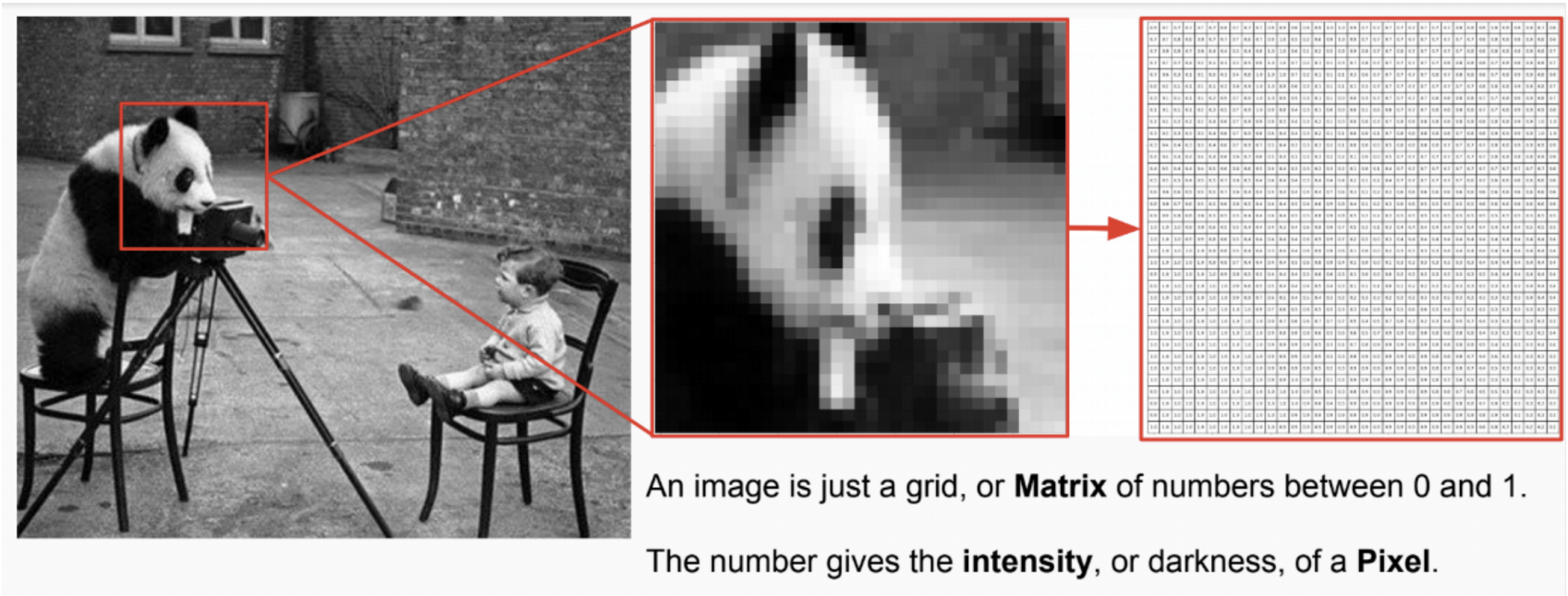


Figure: Black & White image of panda is a matrix

Representing Color

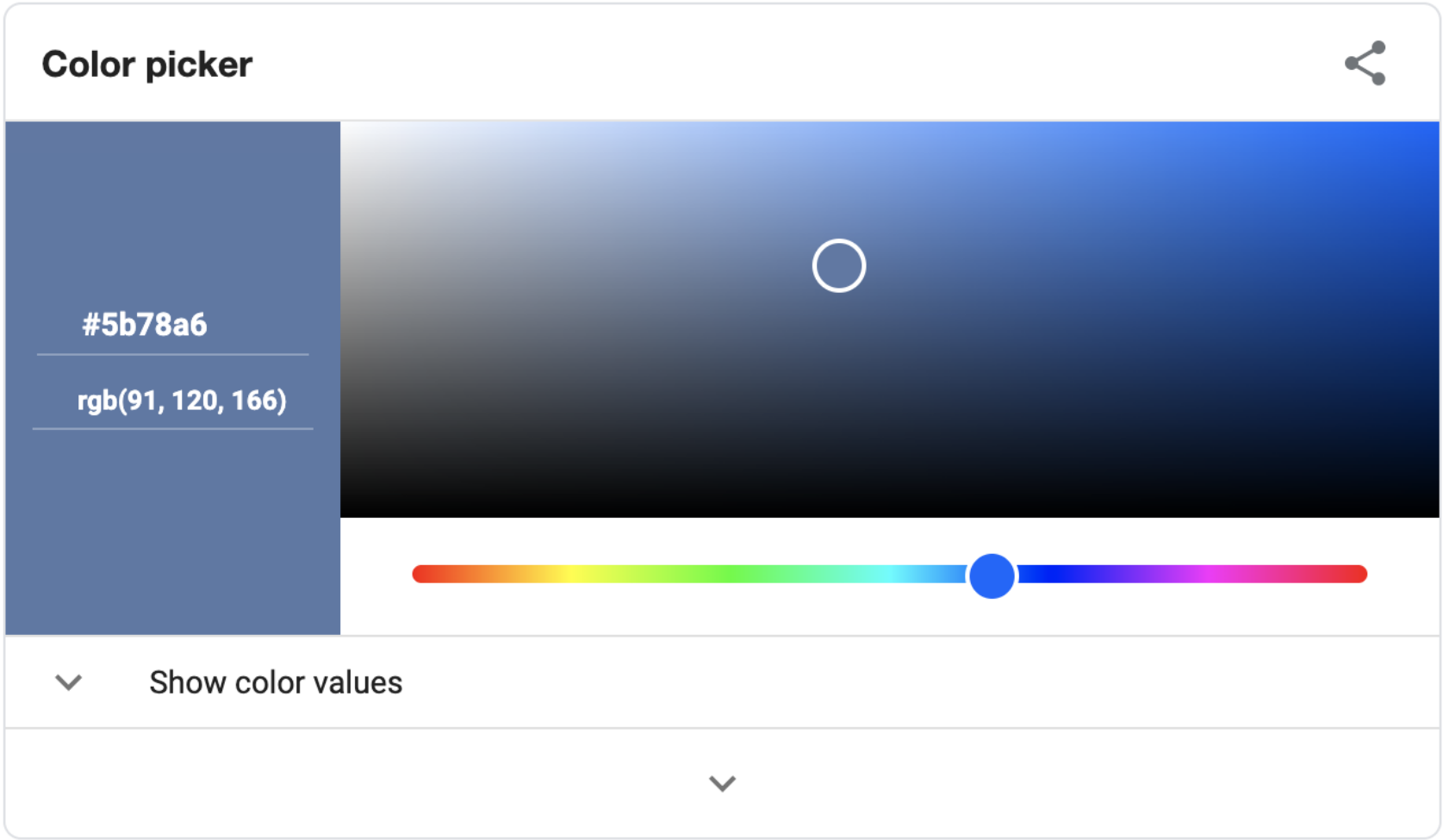


Figure: Color of single pixel is represented by combination of 3 RGB values

Representing Color

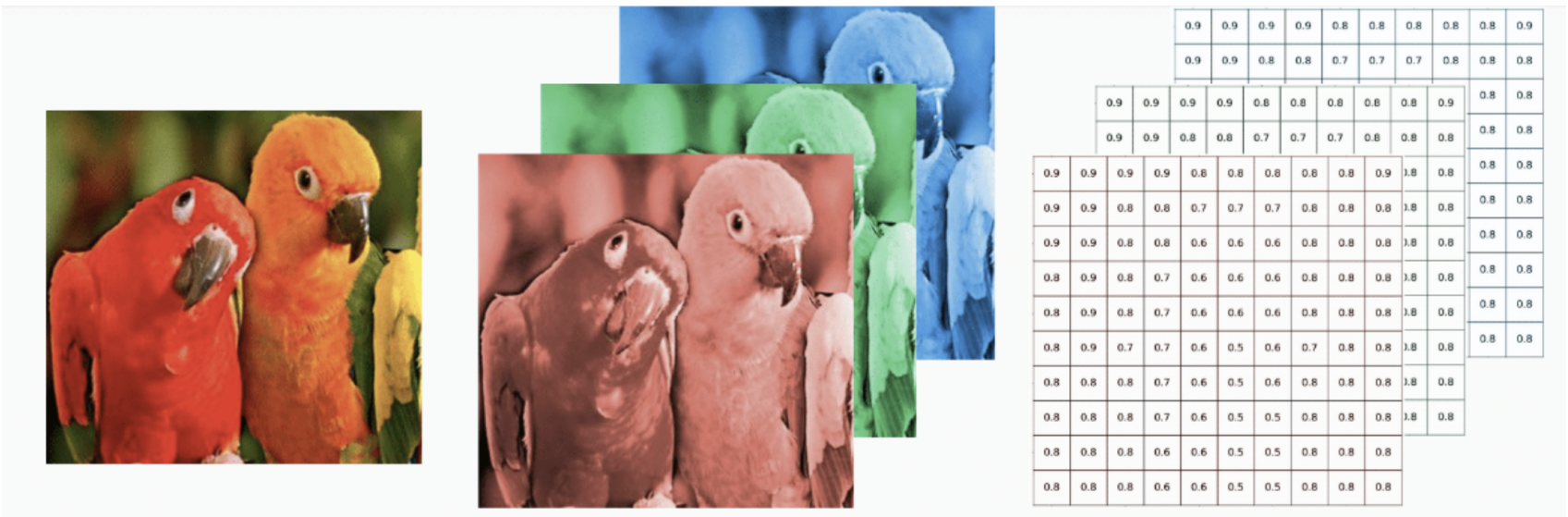


Figure: Stacking RGB layers forms final picture

Representing Color

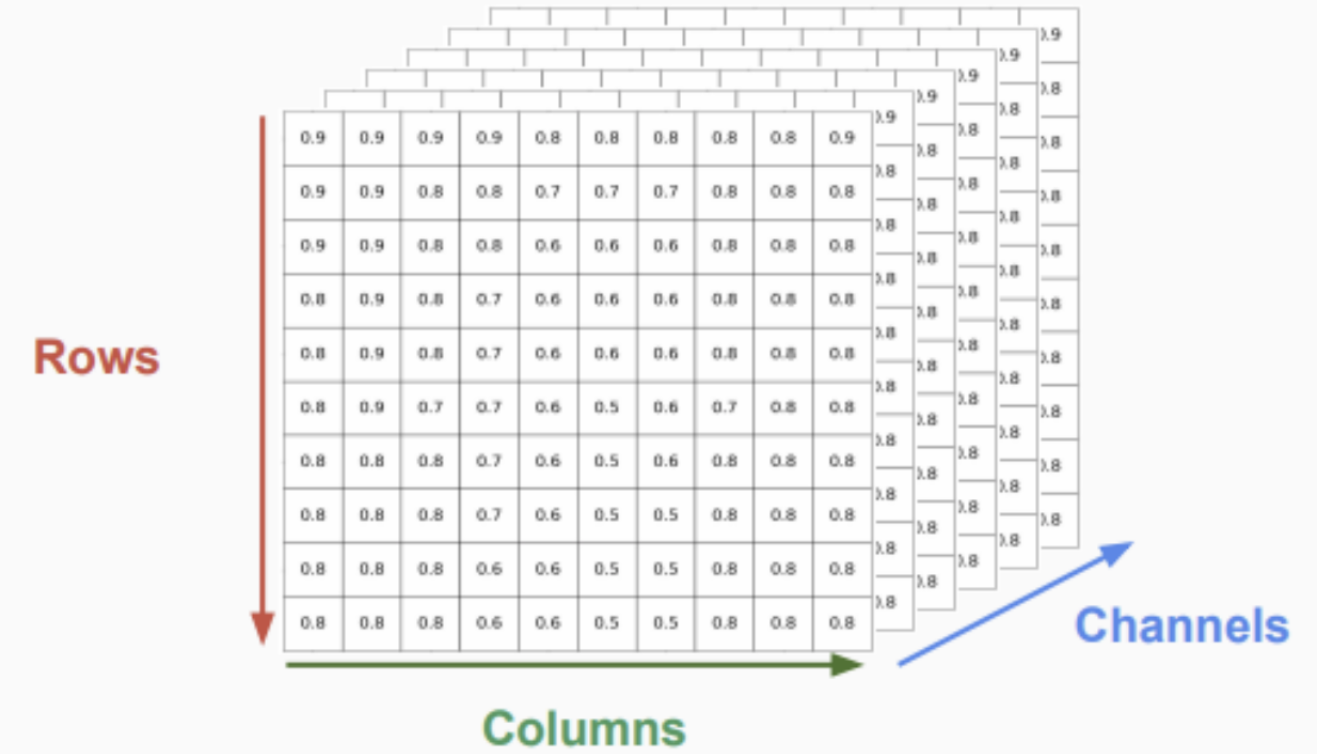


Figure: Numerically, images are 3-dimensional matrices with rows, columns, and 3 **channels**. We also call these **volumes**.

Curriculum Plan

1. Linear Models for Regression and Classification
2. Optimization
3. Vectors
4. Matrices
5. Image Representation
6. Image Classification
7. Challenges of Computer Vision
8. Neural Networks

Image Classification

- We can represent an image as a vector in a very high dimensional space, $\mathbb{R}^D$

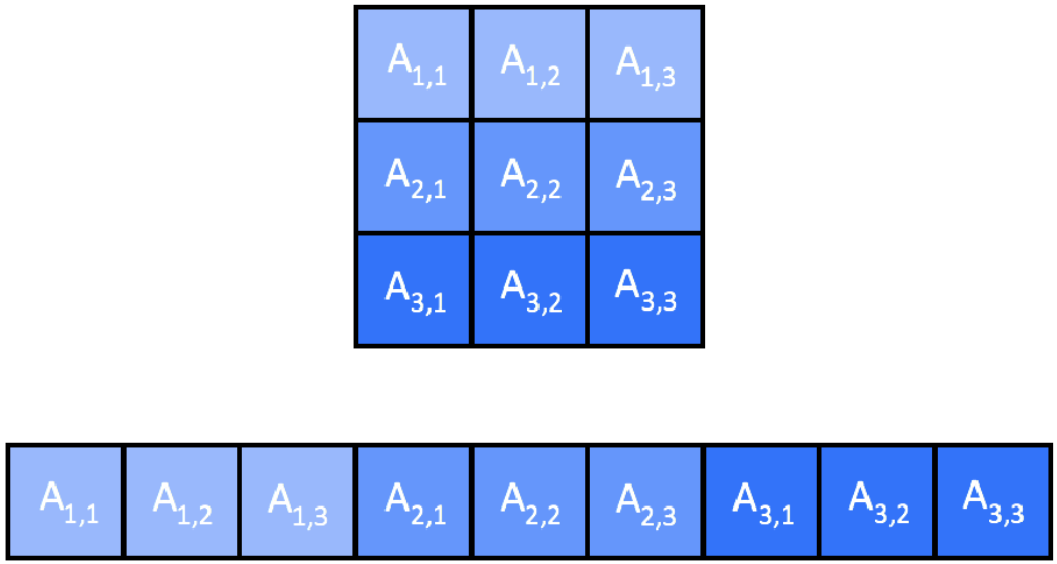


Figure: Flatten a matrix A into a vector[1]

[1] quantstart.com

Image Classification

- After flattening image $\mathbf{x} \in \mathbb{R}^D$, we want to classify as cat/ not cat using logistic regression

$$y = \sigma(w_0 + w_1 x_1 + ... + w_D x_D)$$

where each $x_i \in \mathbf{x}$ is a value for one pixel in one RGB channel.

- What could be a problem with this approach?

Image Classification

- Interpretation of linear classifiers as template matching.
- Weights W corresponds to a template for the cat class.
- Whether or not an image is a cat is decided by comparing the template with the image using dot product to find if the score exceeds the 0.5 decision boundary.

Image Classification

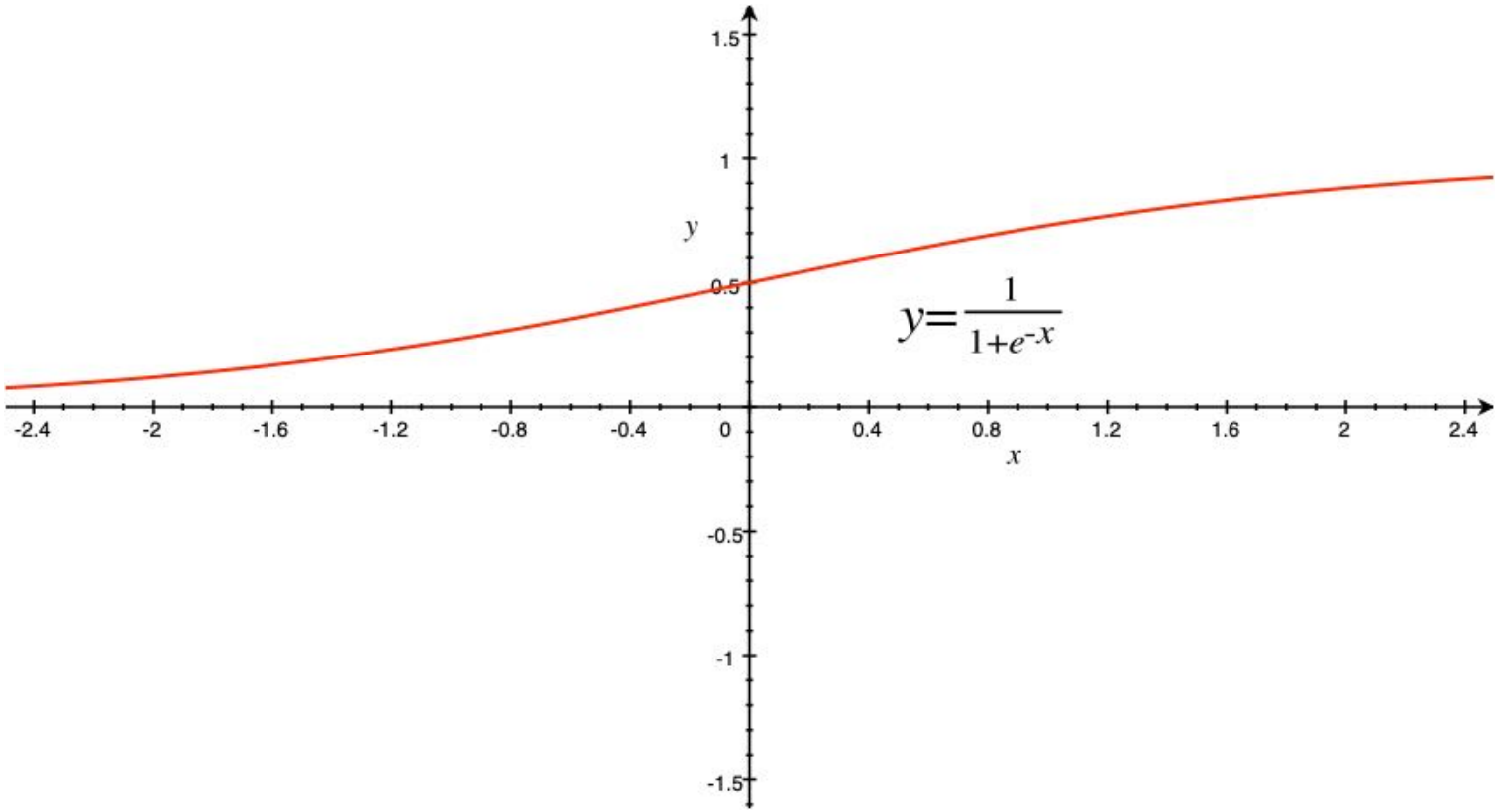


Figure: The sigmoid function σ will predict cat for higher value of $\mathbf{x} \cdot \mathbf{w}$, and not cat for lower.

Image Classification

- Interpretation of linear classifiers as template matching.
- Weights W corresponds to a template for the cat class.
- Whether or not an image is a cat is decided by comparing the template with the image using dot product to find if the score exceeds the 0.5 decision boundary.
- **The best template should look approximately like a cat!**

Image Classification

Figure: Using gradient descent, the weights **w** learn a visual resemblance of a class

- Now, what do you think you this is a problem with these templates?

[1] cs231n.github.io/linear-classify

Image Classification

Figure: Using gradient descent, the weights **w** learn a visual resemblance of a class

- Now, what do you think you this is a problem with these templates?
- It is very hard to capture all possible variations of an image using just a single template.

[1] cs231n.github.io/linear-classify

Curriculum Plan

1. Linear Models for Regression and Classification
2. Optimization
3. Vectors
4. Matrices
5. Image Representation
6. Image Classification
7. Challenges of Computer Vision
8. Neural Networks

Challenges of Computer Vision

- Because images are very high-dimensional data, it is difficult for models to capture all possible variety and edge cases included in your data.
- Example: When using logistic regression, what happens when you shift an entire image left by one pixel?

Challenges of Computer Vision

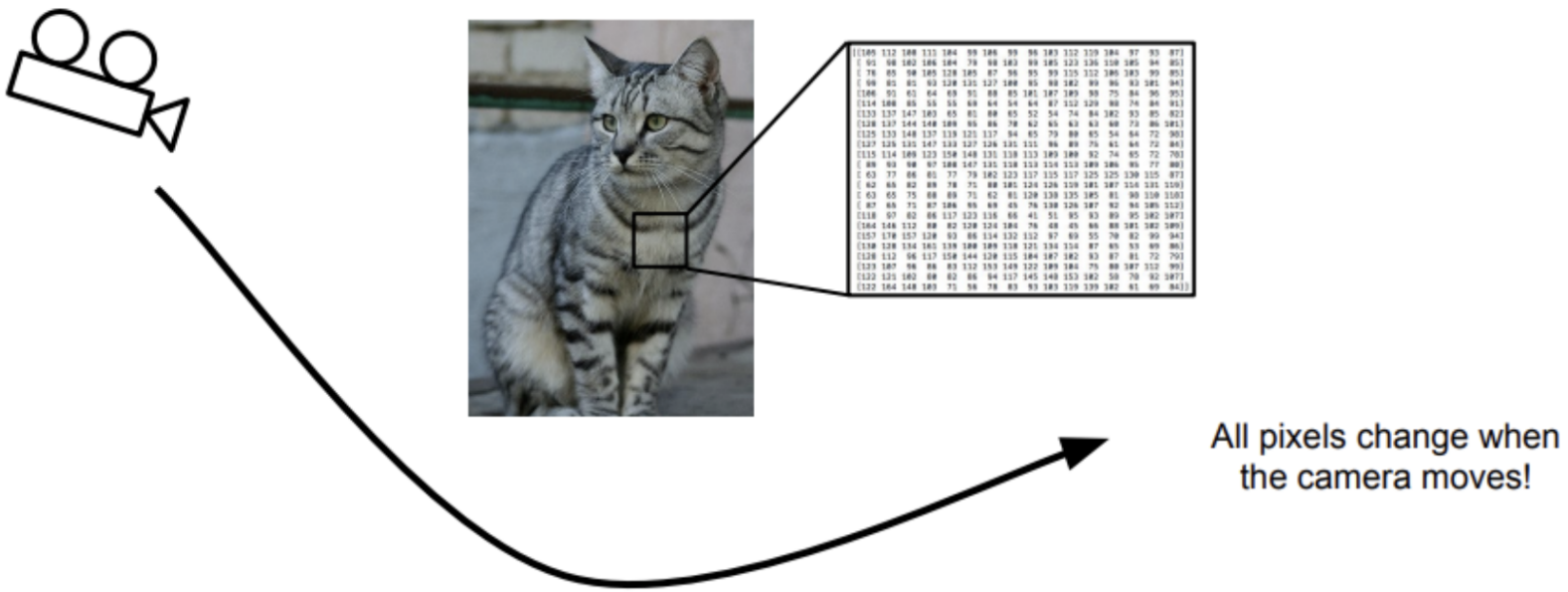


Figure: Viewpoint Variation

[1] http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture02.pdf

Challenges of Computer Vision

Figure: Background Clutter

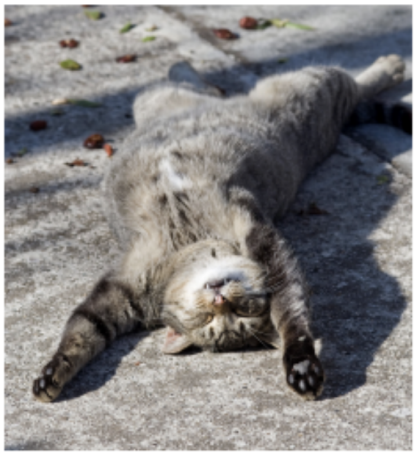
[1] http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture02.pdf

Challenges of Computer Vision

Figure: Illumination

[1] http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture02.pdf

Challenges of Computer Vision



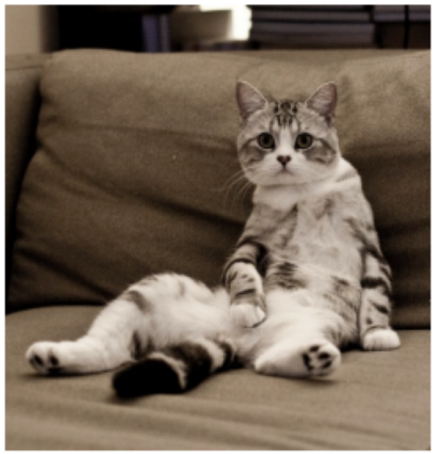


Figure: Deformation

[1] http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture02.pdf

Challenges of Computer Vision

Figure: Occlusion

[1] http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture02.pdf

Curriculum Plan

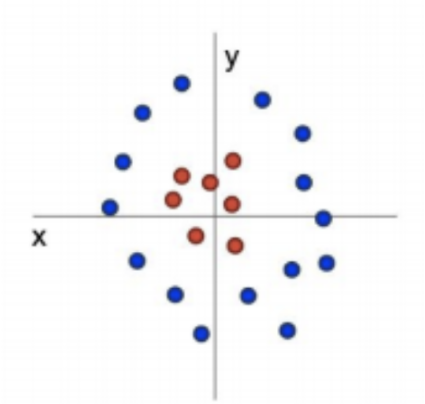
1. Linear Models for Regression and Classification
2. Optimization
3. Vectors
4. Matrices
5. Image Representation
6. Image Classification
7. Challenges of Computer Vision
8. Neural Networks

Neural Networks

Visual Viewpoint

Linear classifiers learn one template per class

Geometric Viewpoint



Linear classifiers can only draw linear decision boundaries

Figure: Motivation: Linear models are fairly weak

[1] http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture02.pdf

Neural Networks

- We can model more complex functions by "stacking" more weights.
- Instead of

$$\hat{y} = \sigma(\mathbf{w} \cdot \mathbf{x})$$

We can now also do this:

$$\hat{y} = \sigma(\mathbf{w}_1 \cdot \sigma(\mathbf{w}_2 \cdot \mathbf{x}))$$

$$\mathbf{w}_1 \in \mathbf{R}^M, \mathbf{w}_2 \in \mathbf{R}^{M \times D}, \mathbf{x} \in \mathbf{R}^D$$

Each row of the matrix $\mathbf{w}_2$ is called a "neuron"

Neural Networks

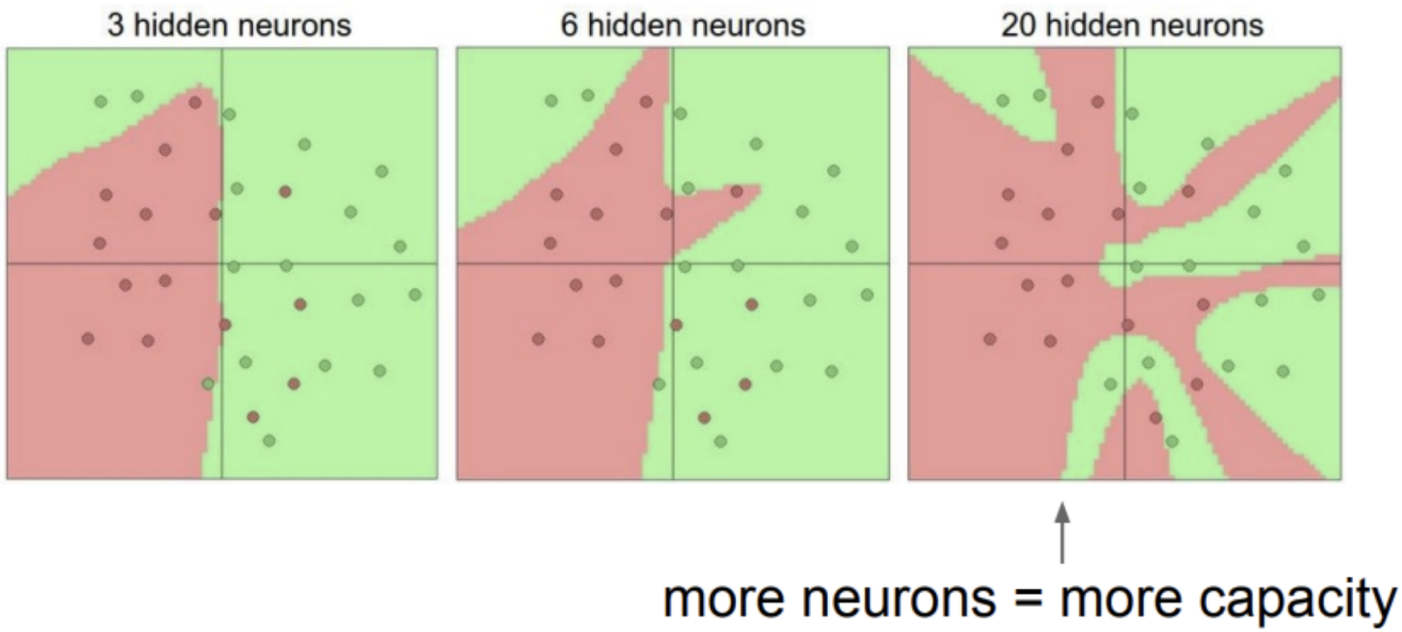


Figure: The more neurons you have, the more **expressive** your model is.

[1] http://cs231n.github.io/neural-networks-1/

Neural Networks

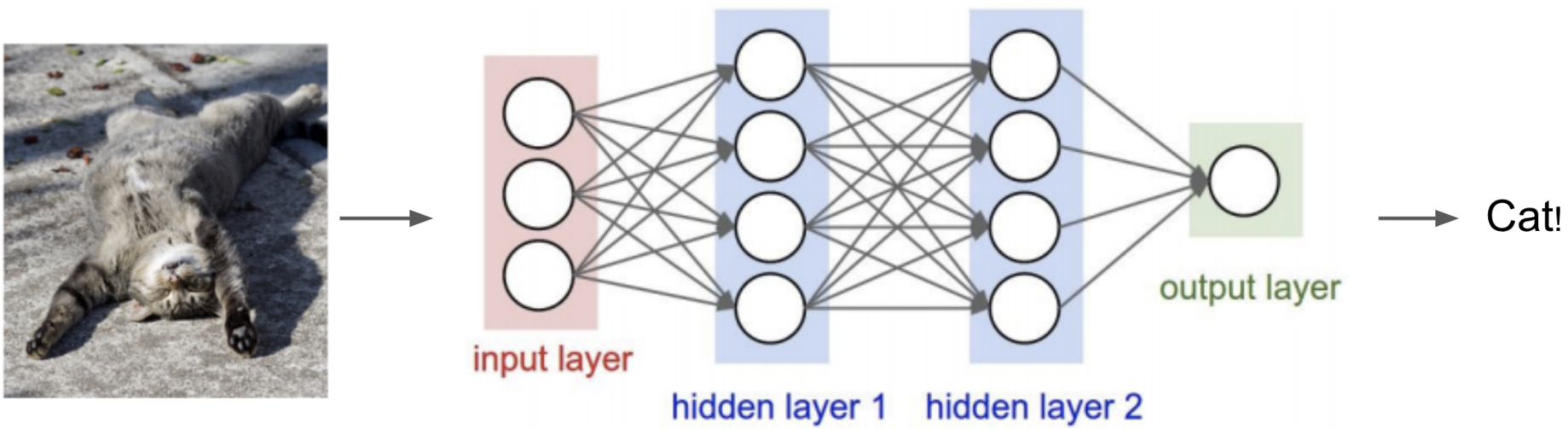


Figure: By stacking more and more layers and adding neurons, we can get models capable of representing all cats!

Image Representation and Neural Networks

Andrew Kondrich

August 2, 2019