

CS 331

Computer Vision

10 / 15 / 2026

Instructor: Michael Eckmann

Today's Topics

- Questions / Comments?
- How is the programming assignment going?
- Linear Regression

Regression vs. Classification

- **Regression** is where we take in a vector of real values and **output a scalar real value** (that is a value from among a continuous set of reals)
- **Classification** on the other hand, takes in a vector of real values and **outputs one of a discrete set** of labels
- Linear Regression learns a linear function from a set of training examples (data points). The linear function produces a scalar real value.

Linear Regression

- First, let's recall what a derivative of a function means
- It's a function that gives us the slope at each point on the original function
- Example:
- $f(x) = x * x$
- $f'(x) = 2 * x$
- When is the original function $x * x$ minimum?
- Let me draw the graph of $x * x$

Linear Regression

- $f(x) = x^2$
- Derivative $f'(x) = 2x$
- When is the original function x^2 minimum?
- When $f'(x) = 0$ is when x^2 is minimum
- When a function is convex like x^2 then this always works.
- See: https://en.wikipedia.org/wiki/Convex_function

Linear Regression

- We talk about derivatives when dealing with functions of one variable
- For functions of more than one variable the analogous concept is a gradient
- The gradient of a function of multiple variables is a vector of partial derivatives.
- Suppose these are the multiple variables $\mathbf{x} = (x_1, x_2, x_3, \dots, x_n)$
- A partial derivative is taken with respect to one of the variables in that vector.
- Partial derivative of f with respect to x_1 , is found by treating all the variables other than x_1 as constant

Linear Regression

- There will be n partial derivatives which make up the gradient vector.
- Let me show a concrete example on the board of computing the Gradient of a function of 3 variables.

Linear Regression

- For the linear regression examples to come, we can think in terms of derivative in the examples that are functions of 1 variable.
- If we are dealing with functions of multiple variables then we need to deal with the gradient.

Linear Regression

As the name implies, linear regression solves a regression problem. In other words, the goal is to build a system that can take a vector $\mathbf{x} \in \mathbb{R}^n$ as input and predict the value of a scalar $y \in \mathbb{R}$ as its output. The output of linear regression is a linear function of the input. Let $\hat{y}$ be the value that our model predicts y should take on. We define the output to be

$$\hat{y} = \mathbf{w}^\top \mathbf{x}, \quad (5.3)$$

where $\mathbf{w} \in \mathbb{R}^n$ is a vector of **parameters**.

Parameters are values that control the behavior of the system. In this case, w_i is the coefficient that we multiply by feature x_i before summing up the contributions from all the features. We can think of $\mathbf{w}$ as a set of **weights** that determine how each feature affects the prediction. If a feature x_i receives a positive weight w_i , then increasing the value of that feature increases the value of our prediction $\hat{y}$. If a feature receives a negative weight, then increasing the value of that feature decreases the value of our prediction. If a feature's weight is large in magnitude, then it has a large effect on the prediction. If a feature's weight is zero, it has no effect on the prediction.

Linear Regression

$$\text{MSE}_{\text{test}} = \frac{1}{m} \sum_i (\hat{\mathbf{y}}^{(\text{test})} - \mathbf{y}^{(\text{test})})_i^2. \quad (5.4)$$

Linear Regression

- Conventions
- m is the name we give to the number of training examples
- Input variable(s) are named x
- Output variable is y
- $\hat{y}$ is the predicted value of y during training
- w indicates a weight
- $\hat{y}$ is a linear function of w 's and x 's
- vectors are $1 \times n$ matrices, aka column vectors

Linear Regression

- The most common loss function for linear regression is the Mean Squared Error.
- It is similar to the mean Euclidean distance between the predicted y 's and the actual y 's except no square root is taken.
- We are going to minimize the loss function.
- Since we are summing up positive numbers, minimizing MSE and minimizing Mean Euclidean Distance would give same result.

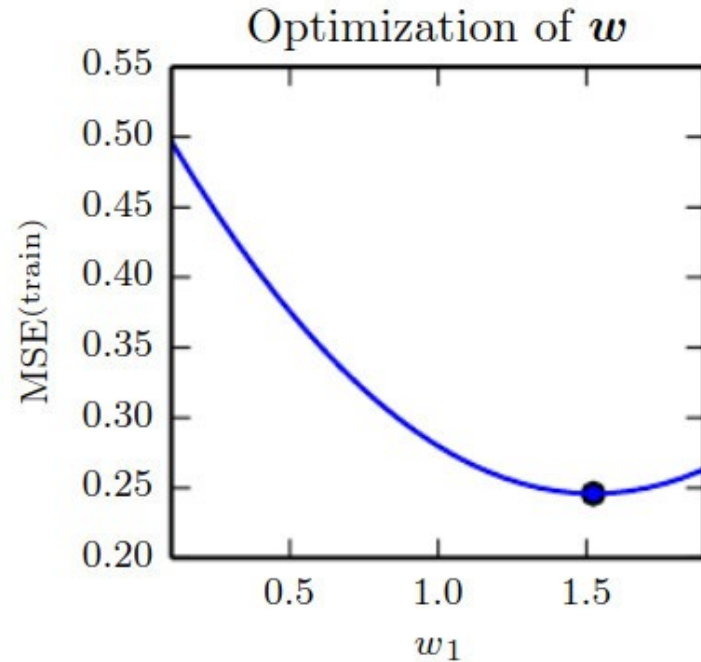
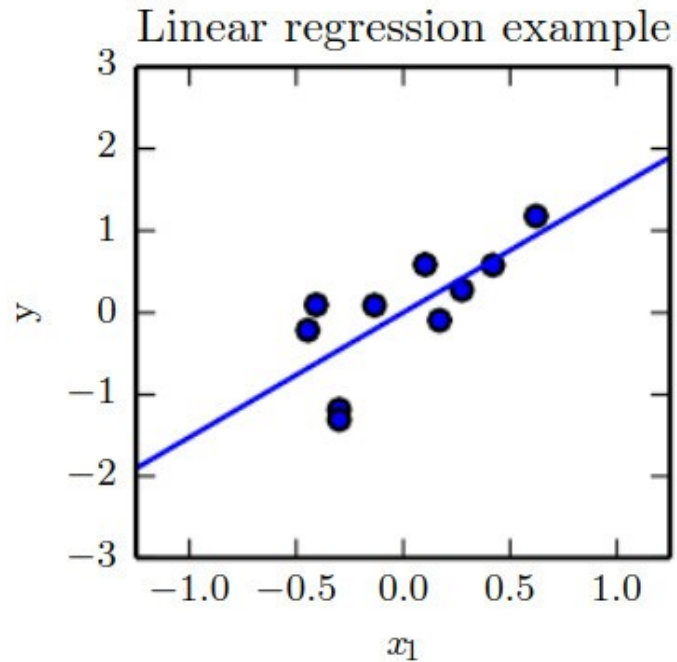
Linear Regression

- Examples on next slides uses a simpler form of a linear function. No y-intercept is used so it is 0.
- This will only learn a line that goes through the origin.
- Only 1 weight (for the slope) is to be learned.
- What's the general equation of a line?

Linear Regression

- $y = mx + b$
- $m = \text{slope}$, $b = \text{y-intercept}$
- But since the learned values are called weights we use w_1 for slope and for this simple example, no y-intercept (so line goes through origin).
- $y = w_1 x_1$

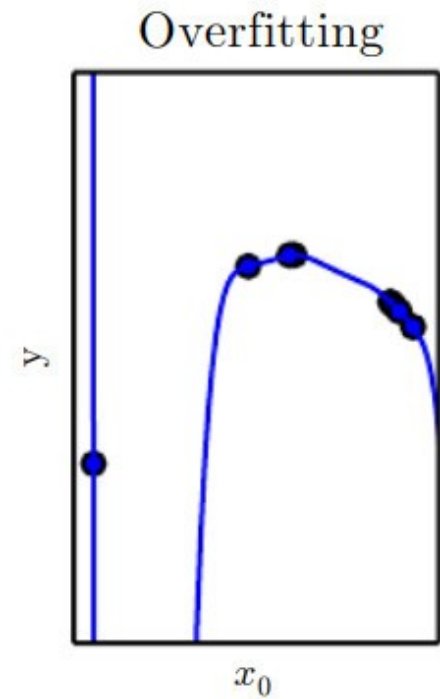
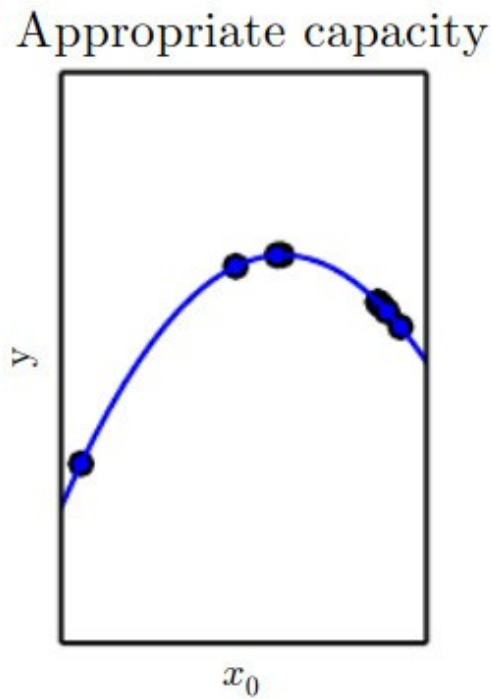
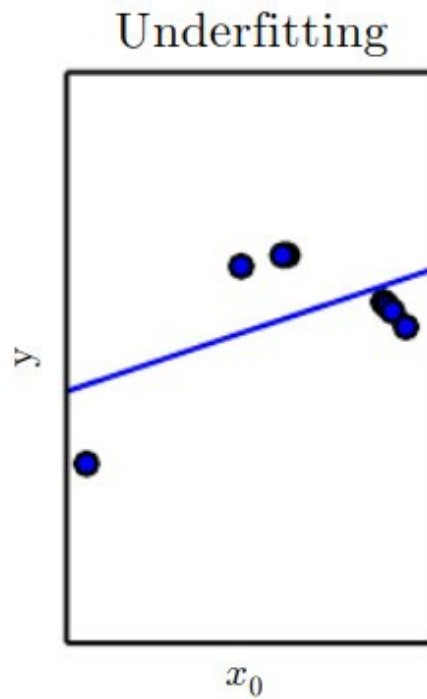
Linear Regression



Capacity, Overfitting, Underfitting

- Capacity relates to the kinds of functions to which we are trying to fit the data.
- Overfitting is when the functions are more complex (e.g. higher order polynomials) than those that actually represent the true population of data
- Underfitting is when the functions are less complex than those that actually represent the true population of data
- When we train, our training data is a *sample* of the data that exists. It is not the entire population of data (e.g. it is not every image to ever exist)

Capacity, Overfitting, Underfitting



Recap

- Experience
- Regression vs. Classification
- Loss function
- Capacity, Overfitting and Underfitting

Next topics

Material in Chapter 4 Numerical Computation in Goodfellow, et.al.'s book titled *Deep Learning*

- <https://www.deeplearningbook.org/>
- Specifically section 4.3

<https://www.deeplearningbook.org/contents/numerical.html>

- from page 80 – middle of page 84

Topics

- Gradient
- Gradient Descent in general
- Gradient Descent for Linear Regression

Derivative, Partial Derivative, Gradient

- Derivatives are for functions of one variable
- Functions of n variables ($n > 1$)
 - Take the partial derivative with respect to each variable x_i (which is just the derivative of the function with all other x 's considered constant except x_i)
 - The n partial derivatives taken together as a vector is the gradient

Gradient Descent

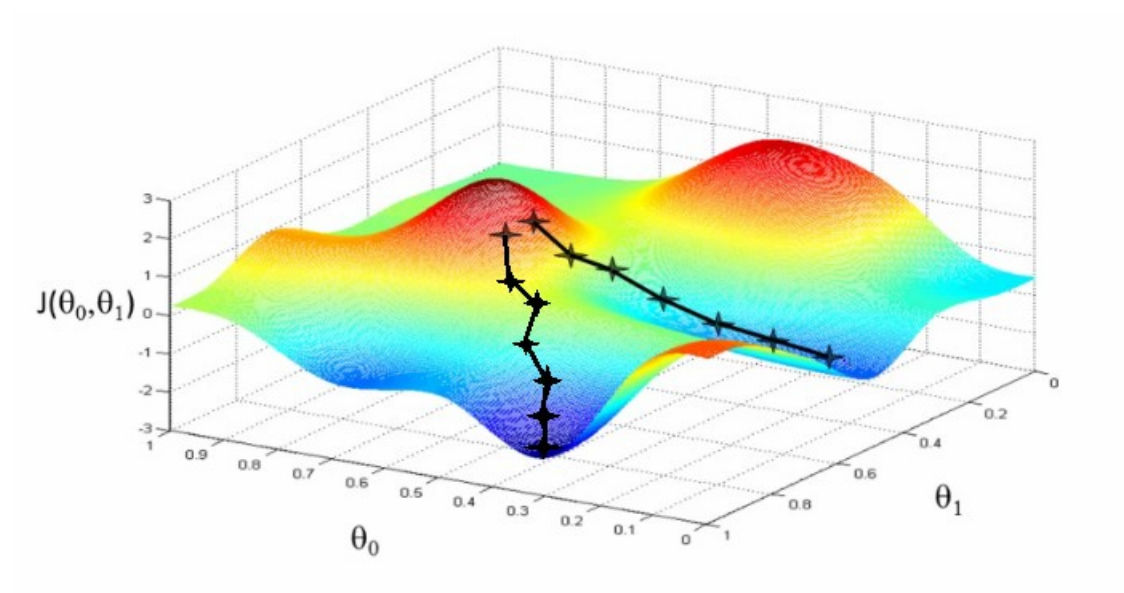
- Gradient Descent
- Imagine the next pictures as surfaces (functions of two variables)
- We'll start at some random place on the surface and want to travel down to the lowest point (the minimum value of the surface)
- When we are at a particular place on the surface we want to figure out which direction to travel, only from very nearby surroundings.

Gradient Descent



Michael Eckmann - Skidmore
College - CS 331 - Fall 2026

Gradient Descent



Gradient Descent

- Gradient Descent for linear regression
 - The surface isn't as complicated as those --- it is bowl shaped (convex) and has no minima other than the global minimum
 - Recall, the function we are trying to minimize is the loss function
 - The loss function was the Mean Squared Error
 - We'll take the gradient of the loss function together shortly

Linear Regression

- Reminders:
- m is the name we give to the number of training examples
- Input variable(s) are named x
- Output variable is y
- $\hat{y}$ is the predicted value of y during training
- w indicates a weight
- $\hat{y}$ is a linear function of w 's and x 's
- The loss function is the Mean Squared Error

Gradient Descent

- Gradient Descent for linear regression
 - The linear function we're trying to learn has two learnable parameters (aka weights) w_0 is the y-intercept and w_1 is the slope
 - The cost function for this is the MSE over all the m training examples
 - Formula: (on the board)
 - Gradient of MSE is made up of two partial derivatives (one with respect to w_0 and the other with respect to w_1)

Gradient Descent

- On the board MSE loss function
- Partial Derivative of MSE w/r/t w_0 is
- Partial Derivative of MSE w/r/t w_1 is
- The gradient of MSE is a vector of those two partials.