Machine Learning Club Meeting #2

- SIDDRRSH@GMAIL.COM
- RONITHGANJIGUNTA@GMAIL.COM

Today's Meeting

- Review of Last Meeting's Concepts + Computational Skills
- Machine Learning Gradients
- Loss Models
- Videos (if we have time)

Last week's content:

- Basic ML Terminology
- The Mechanics of a Machine Learning Model
- Supervised Learning
- Regression v. Classification

Review of Previous Meeting

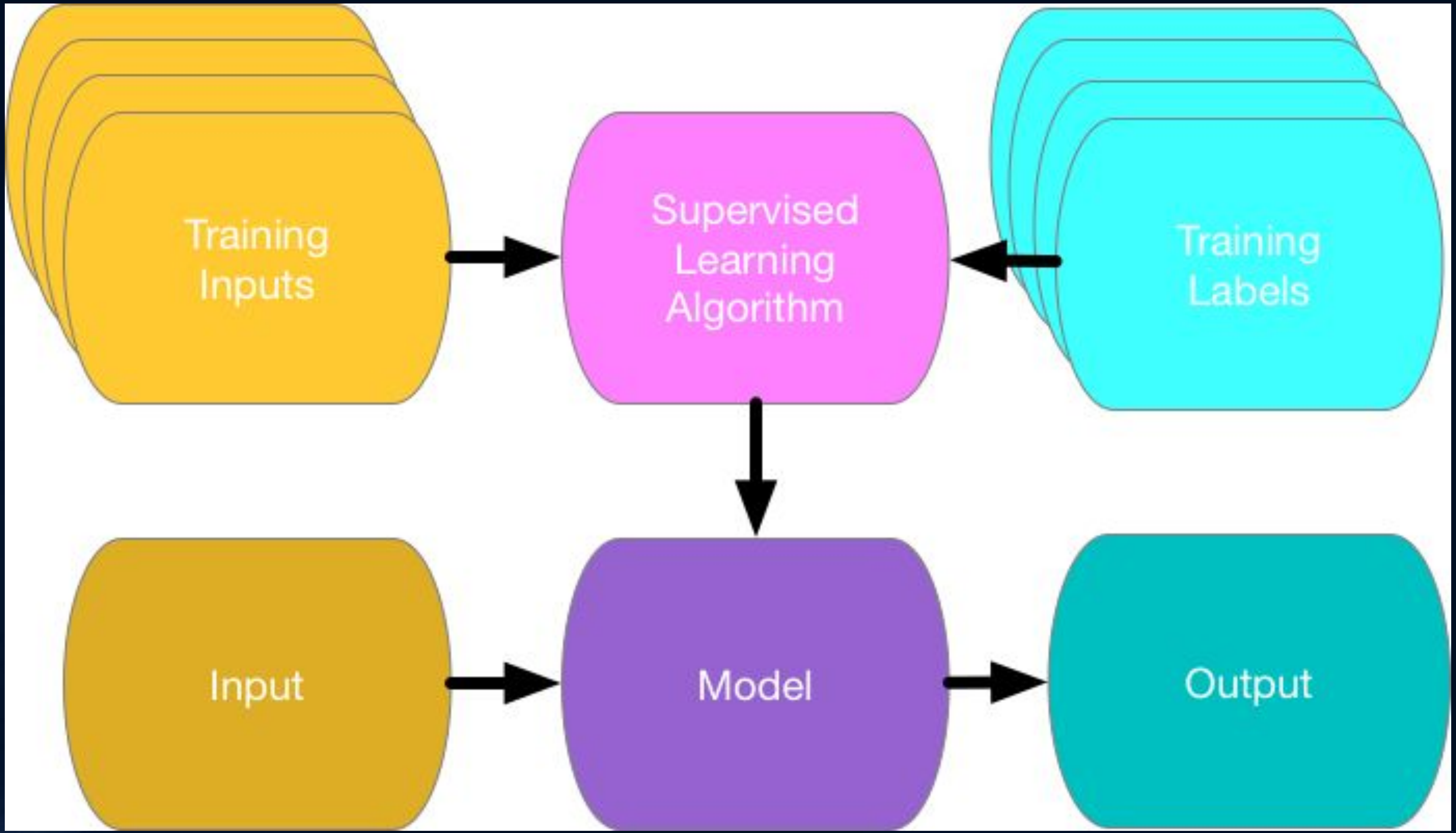
A FEW PROBLEMS AND EXAMPLES

ML Terminology

- Features - input variable (x)
- Labels - the thing we're predicting (y)
- Example - particular instance of data (unlabeled vs labeled ex.)
- Model - relationship between features and labels
- Training v. Inference
  - Show model learned examples to gradually learn
  - Apply model to unlabeled examples

Supervised Learning and ML Models

- The task of predicting targets given input data
- map each input (x) to a prediction (*f*(x))
- i.e. - predict cancer or not cancer based on CT image


Training
Inputs
Supervised
Learning
Algorithm
Training
Labels
Input
Model
Output

Regression v. Classification

Regression

- predict continuous values
- quantitative
- ex. number of puppies in a shelter

Classification

- predict discrete values
- qualitative
- ex. Is this animal an adult or child?

Review Game

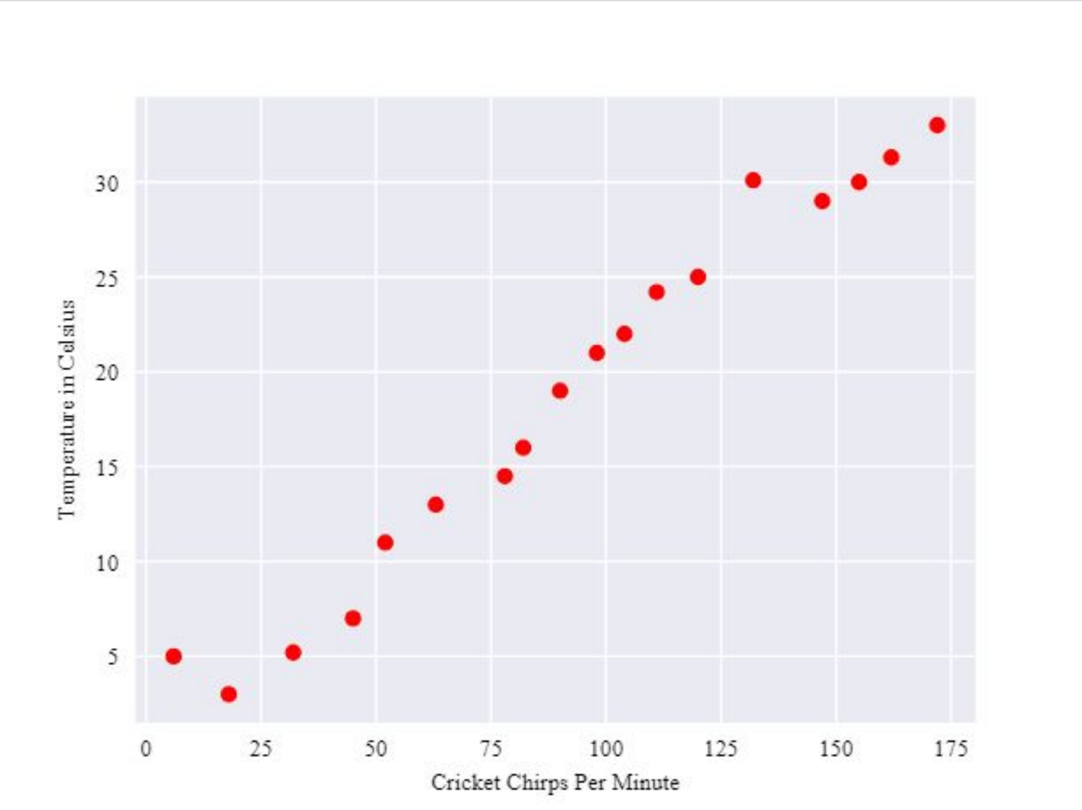
Quick Review of Computational Skills

Training and Loss with Gradients

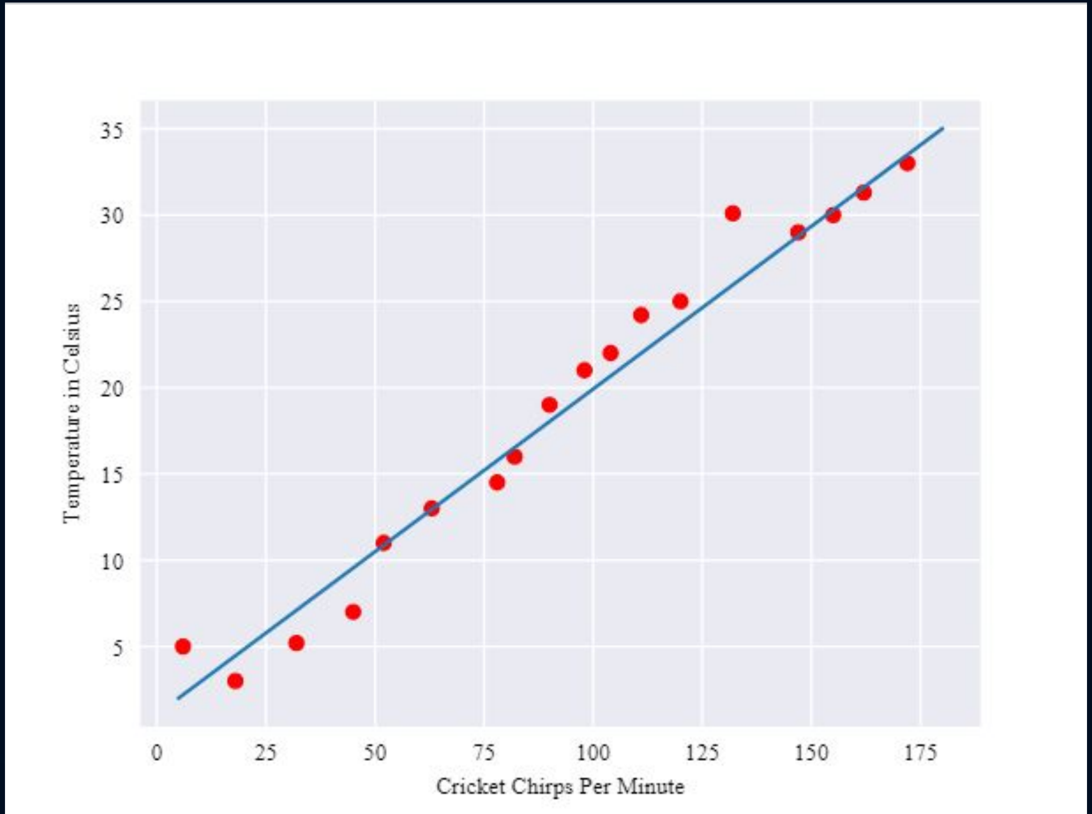
LOSS FUNCTIONS AND OPTIMIZATION ALGORITHMS

Linear Model

- classic slope-intercept equation - $y = mx + b$
- equation for linear model - $y' = b + w_1 x_1$
- y' - predicted label, "b" is bias (y-intercept)
- w1 - weight of input (*x*), same as slope (m), x1 is a feature (x-input)
- Linear regression utilizes a line of best fit to predict continuous values



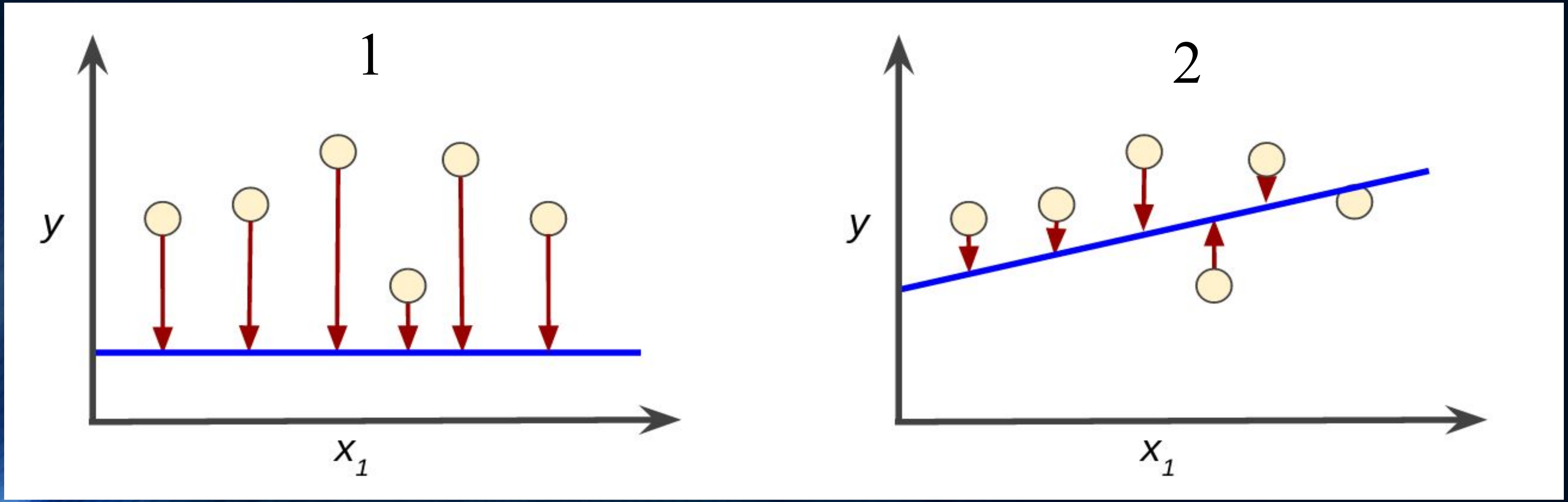
Temperature in Celsius
Cricket Chirps Per Minute



Temperature in Celsius
Cricket Chirps Per Minute

Loss at the surface level

- how “bad” the model’s prediction was on a single example
- Which model below has higher loss?



Training and Loss

Training

- Determine good values
- Use labeled examples

Minimize Loss

- empirical risk minimization
- find weights and biases

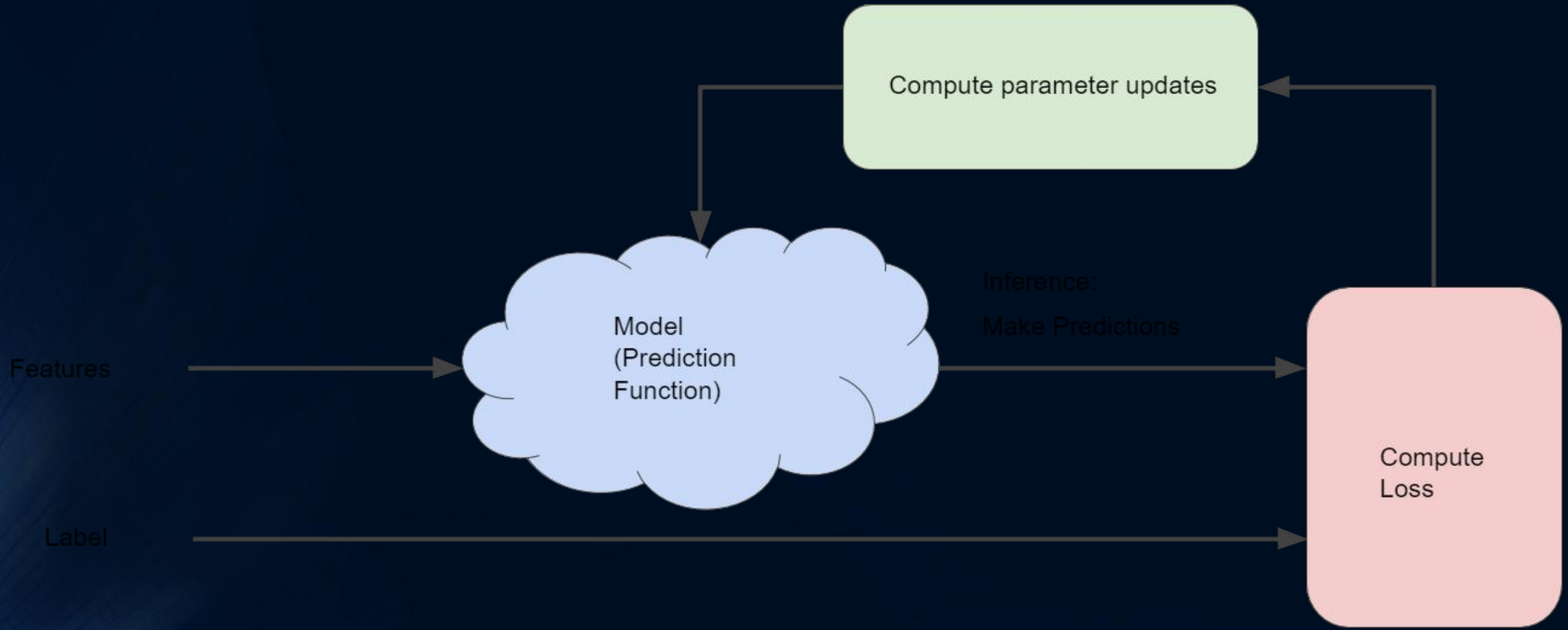
Loss Function

- Use MSE as math function
- examine squared loss

Iterative Approach to Reducing Loss

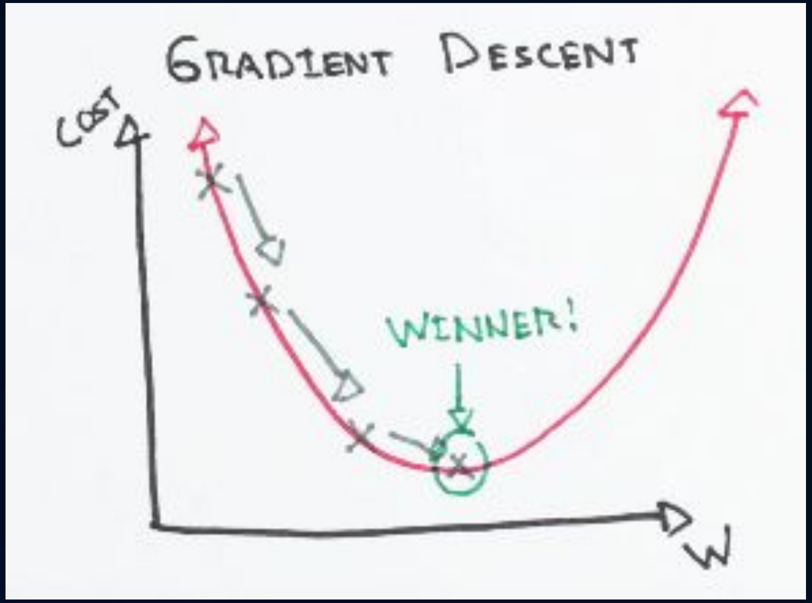
- $y' = b + w_1 x_1$
- A Machine Learning model is trained by …
  - starting with an initial guess for the weights and bias (perhaps 0,0)
  - iteratively adjusting those guesses until learning the weights and bias with the lowest possible loss.
- machine learning model will examine system and generate new values for b, w1

Trial and Error

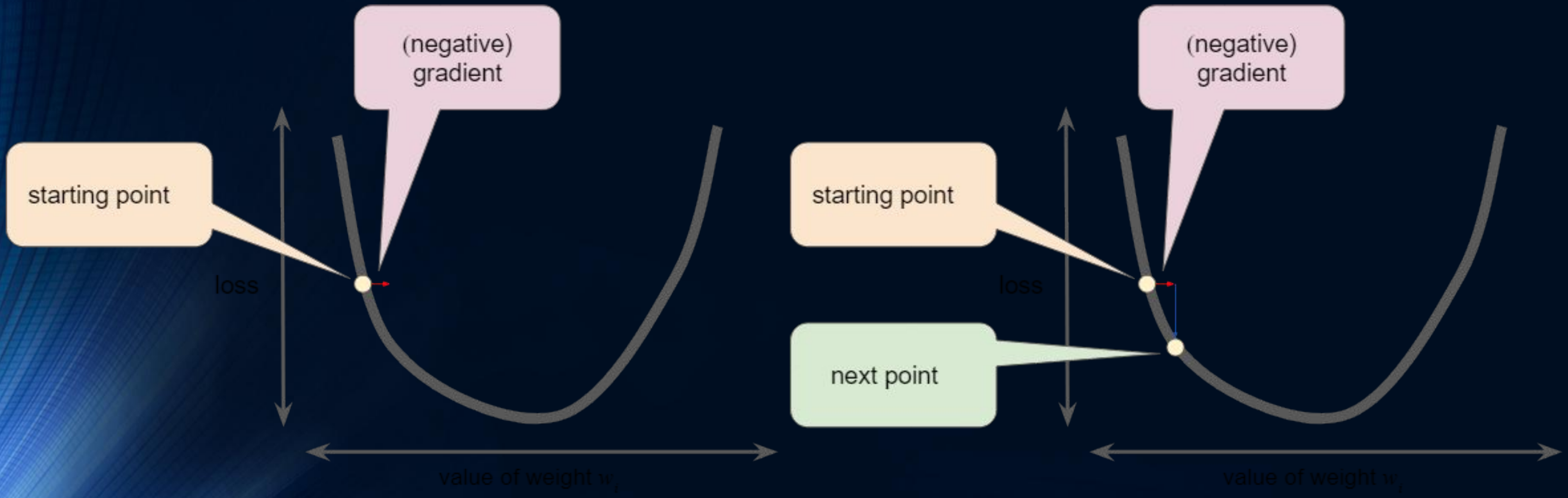


Simple Gradient Descent

- we want to find the point where the loss is at a minimum and converges
- minimum on the loss v. weight graph is optimal
- gradient descent is how we get that point
  - general steps
    - pick a starting point
    - gradient of loss is equal to derivative (slope)!



Gradient Descent simplified



These are both loss v. weight graphs

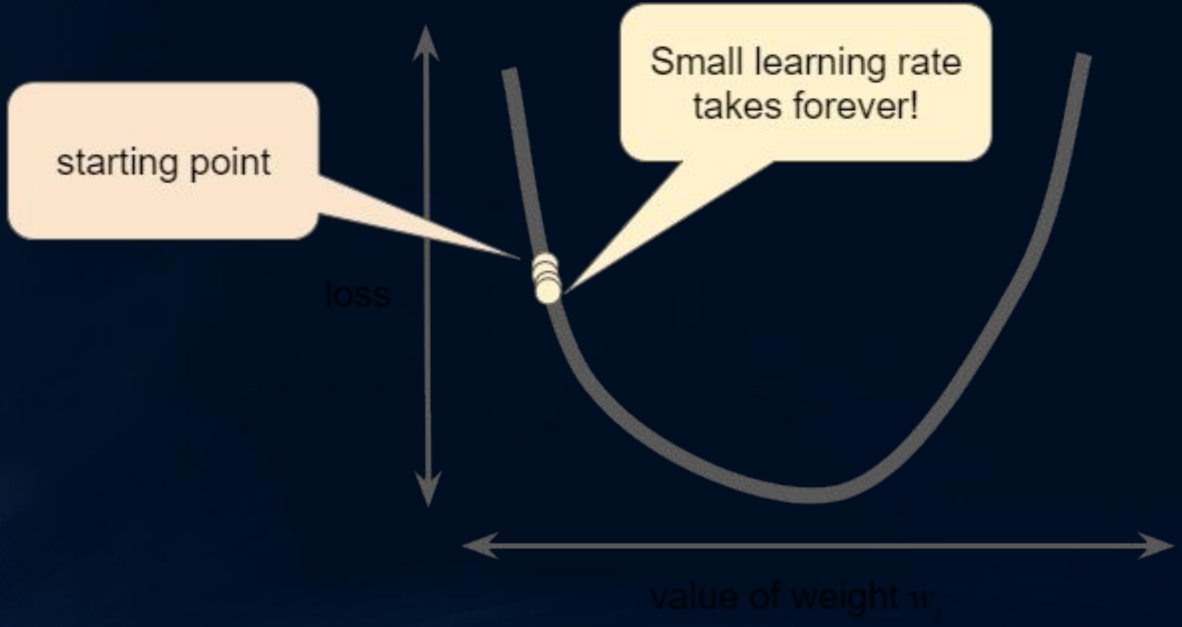
- the gradient vector has both direction and magnitude
- at the simplest level, the learning rate is what we multiply the gradient descent by to get the next point
- hyperparameters are knobs that we tweak in ML algorithms, (ex. picking best learning rate)

Reducing Loss

Learning Rate

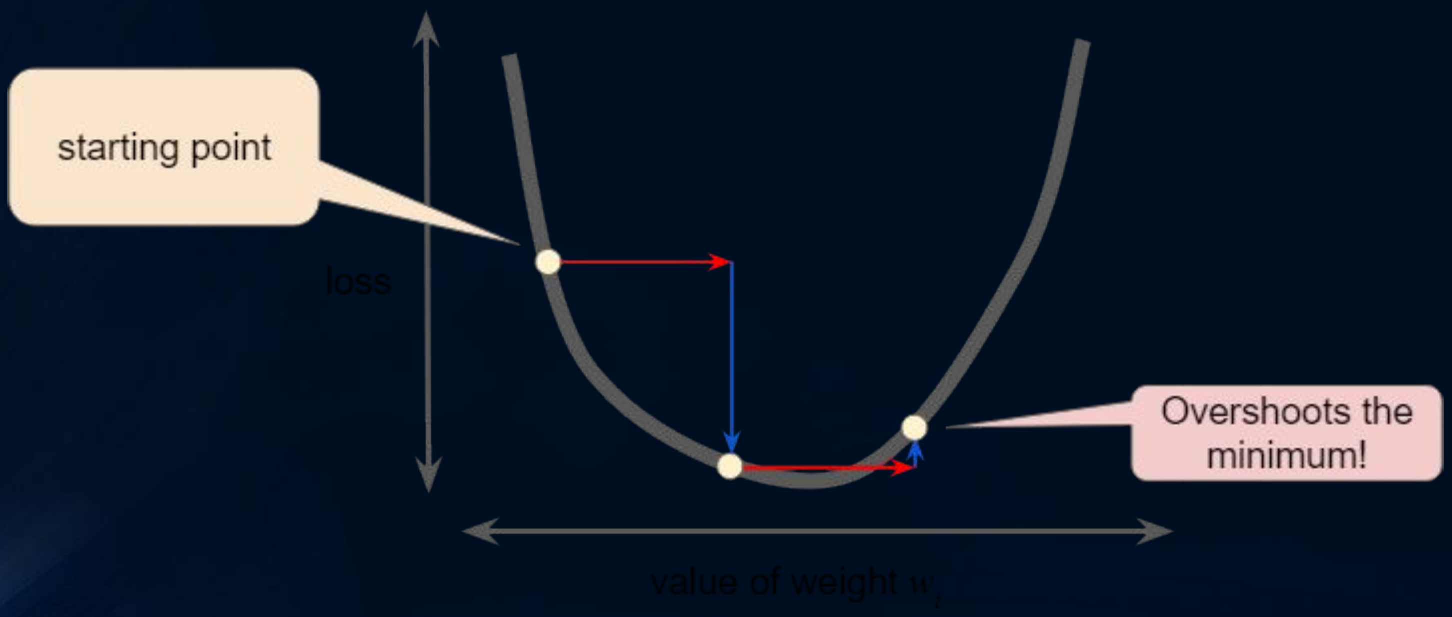
Goldilocks for Learning Rates (1)

Too small of a learning rate



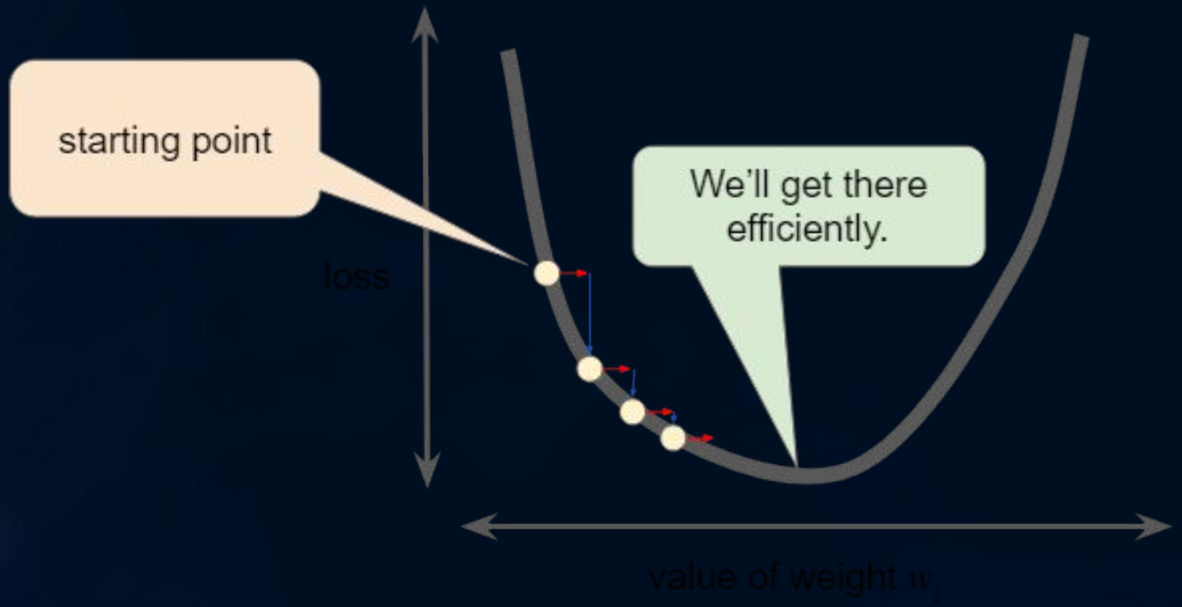
Goldilocks for Learning Rates (2)

Too large of a learning rate



Goldilocks for Learning Rates (3)

"Just right" learning rate



Optimizing the Learning Rate

Loss vs. Weight

Find optimal learning rate or a learning rate large enough that the gradient descent converges efficiently, but not so large that it never converges

Takeaways from today:

- linear regression uses a line of best fit to predict data
- goal is to reduce the loss of a machine learning model
- we reduce loss by employing a gradient descent (derivative/slope) to reach the minimum loss
- the learning rate is what we multiply by the gradient vector
  - it is commonly adjusted to find the optimal gradient for minimizing loss

Next week: TensorFlow Tutorials

INTERACTING WITH MACHINE LEARNING LIBRARIES

What do you need to know?

- Linear Algebra: Matrices and Vectors (We will cover this in the future)
- Probability and Statistics (Linear Regression)
- Programming: Python
- Logic and computational skills

Software to download

- Python Anaconda Version 4
- Sublime Text or a text editor
- Github
- Tensorflow Libraries (numpy and scipy)

Resources

- Stanford ML
  https://www.coursera.org/learn/machine-learning/home/welcome
- Google ML crash course
  https://developers.google.com/machine-learning/crash-course/ml-intro
- Princeton Deep Learning
  https://www.cs.princeton.edu/courses/archive/spring16/cos495/slides/Intro_to_course.pdf

Resources (cont'd)

- Glu-on tutorial
  https://gluon.mxnet.io/chapter01_crashcourse/introduction.html#Basics-of-machine-learning
- Udacity ML Nanodegree
  https://www.udacity.com/course/machine-learning-engineer-nanodegree--nd009t
- MIT ML course online
  https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-867-machine-learning-fall-2006/

Videos (if time permits)

- AI in a nutshell https://www.youtube.com/watch?v=mJeNghZXtMo
- Google Machine Learning https://www.youtube.com/watch?v=nKW8Ndu7Mjw
- AI applications https://www.youtube.com/watch?v=GapiDqifthM
- Simplified neural nets https://www.youtube.com/watch?v=rEDzUT3ymw4