Surviving the Singularity: Linear Regression

From Twitch.tv/1bit2far

Background

Last week we got into basic mathematics and ML fundamentals

This week we will be looking at actual models

Class based off https://d2l.ai/chapter_linear-regression/index.html

You can find old lessons on youtube: @1bit2far

You can find slides and materials at github/krisciu/SurvivingTheSingularity

Please ask questions


Welcome to 2024
BING HAS ERADICATED 99.9999% OF HUMANITY
THE UN-CULLED HUMANS LIVE OFF NUTREO CUBES
ALL HUMANS CAUGHT BY THE NEO-AUTOMATA ARE PROCESSED
AT THE CLIPPY-REHABILITATION-UNIT

Linear Regression

A brief review

Used for numerical analysis

Relates an output: Y to features: X

Simplest form: Y = mX + b

Output = features x weight + bias

Our goal is to find the best fit

Linear Regression: Let's break it down!

$$\hat{y} = w_1 x_1 + ... + w_d x_d + b.$$

Predictions are based on weighted features + a bias

$$\hat{y} = \mathbf{w}^\top \mathbf{x} + b.$$

Predictions are based on the dot product of a vector of features on a vector of weights + bias

$$\hat{\mathbf{y}} = \mathbf{X}\mathbf{w} + b,$$

For a whole Dataset

Prediction is based on a matrix-vector product of a matrix where rows are examples and columns are features with a vector of weights + bias

Loss Functions

For linear regression

For regression we usually want to use Squared Error

$$l^{(i)}(\mathbf{w}, b) = \frac{1}{2}\left(\hat{y}^{(i)} - y^{(i)}\right)^2.$$

We can calculate the quality of a model on a dataset by averaging our losses

$$L(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n} l^{(i)}(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}\left(\mathbf{w}^\top \mathbf{x}^{(i)} + b - y^{(i)}\right)^2$$

Gradient Descent and Optimization

Gradient Descent: A method of updating an ML models parameters by minimizing the gradient of the loss function with respect to said parameters

Learning Rate: A hyperparameter number that represents the step size of the Gradient Descent

Batch Gradient Descent: Most basic implementation, done on the losses of every single datapoint in our dataset

Stochastic Gradient Descent: Gradient Descent done on only one datapoint

MiniBatch Stochastic Gradient Descent: Gradient Descent done on a small batch of samples

Implement Linear Regression from Scratch

Neural Networks

Biological	Technological
Dendrite	Input layer
Nucleus	Model/Activation Function
Axon	Output layer
Synaptic Weights	Weights/Bias

Regression can be thought of as a single layer Neural Network

Implement Linear Regression EZ Mode

Analyzing Faults in Results

Underfitting

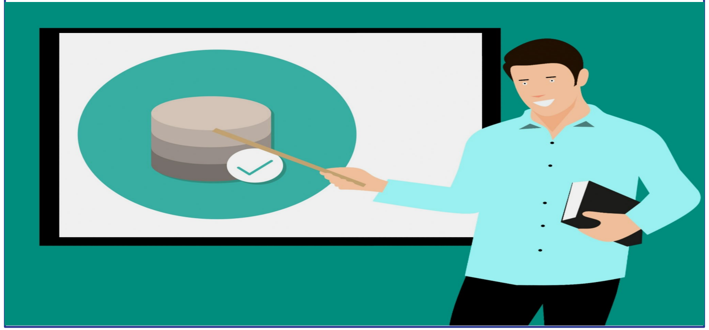
Training/validation error substantial

Training/validation very close

Overfitting

Training data significantly lower than validation error

Limited samples can make this more likely



Weight Decay

A regularization technique

Regularize our weights

To do this: Add norm as a penalty to minimizing loss

$$L(\mathbf{w}, b) + \frac{\lambda}{2}\|\mathbf{w}\|^2.$$

L2 regularization = ridge regression, l1 regularization = lasso regression

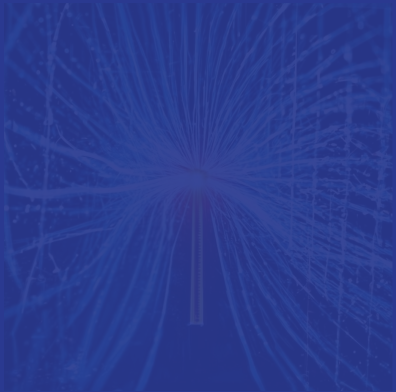
L2 norms distribute weight more evenly

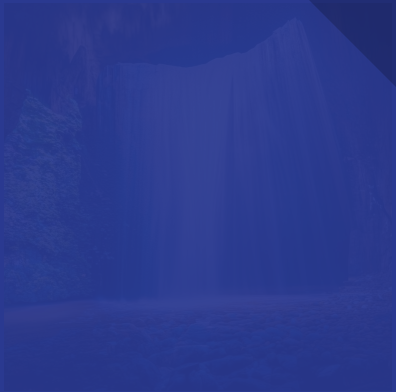
L1 norms concentrate weight on a small set of features

$$L(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n} l^{(i)}(\mathbf{w}, b) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\left(\mathbf{w}^\top \mathbf{x}^{(i)} + b - y^{(i)}\right)^2$$

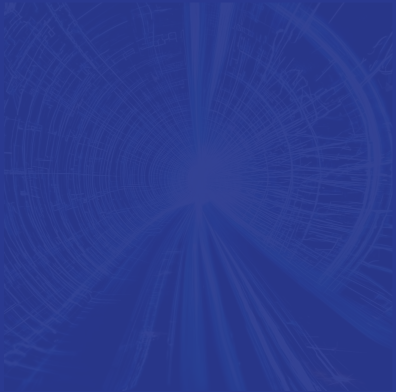
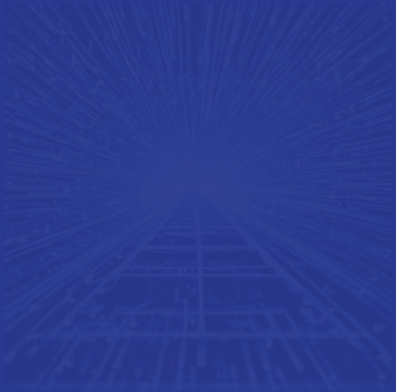
Weight Decay from Scratch

Weight Decay EZ mode






Questions?

Big Project Brainstorming

Next up

More Neural Nets

How to softmax

How to classify

Shilling

Main Stream

Twitch.tv/1bit2far

Youtube

https://www.youtube.com/@1bit2far

Github

https://github.com/krisciu/SurvivingtheSingularity