SYMBILITY
INTERSECT

Machine Learning, Regression, and You

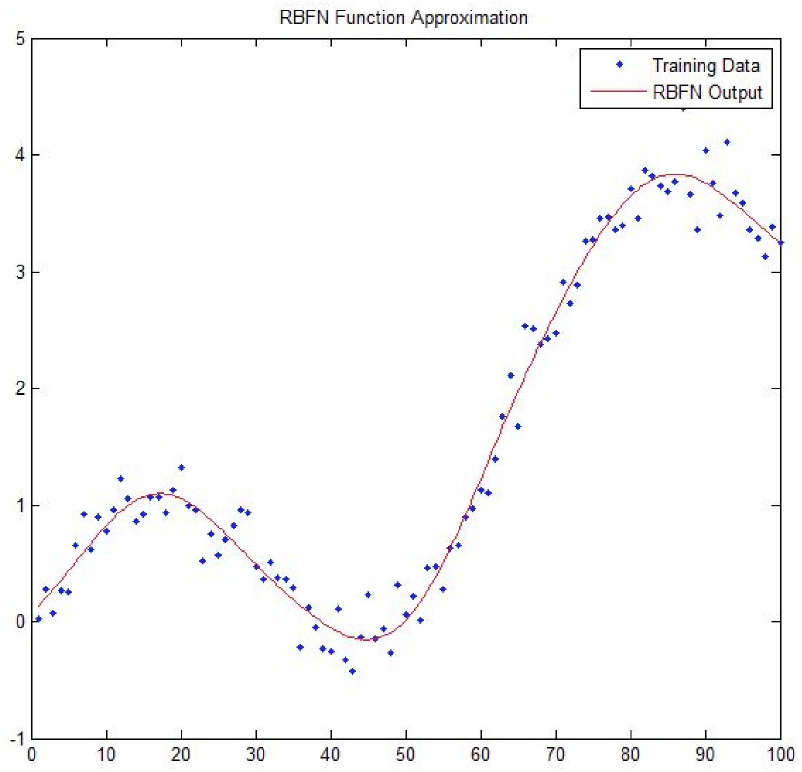
Prepared By Lindsay-Rick

What Is Machine Learning?

Function Approximation - making an educated guess

- Clustering
- Classification
- Nearest Neighbour
- Feature Analysis
- Linear Regression
- Reinforcement Learning
- ..and more

Function Approximation



The function (red) matches the pattern of the data (blue)

The function is “close enough” to the data set

Disclaimer

Machine Learning is Almost <u>Never</u> the Answer!!!

…but can be invaluable for solving certain problems

When is ML Useful?

- Big Data (millions of data points!)
- Complex Problem
- Non-Deterministic
- Many-Dimensional (rule of thumb: n>=9)
- Too difficult to solve by ‘hand’

Non-Deterministic: the same input into the same system multiple times may have different outputs(results) each time

Handling Non-Determinism

- Can’t guarantee the output of a system
- Must try to predict the output of the system

Almost all ML theory is creative use of math!

  - statistics/probabilities
  - vectors
  - optimization

Core ML Problem Types

- Classification
  - Classify data into categories
  - Used in facial recognition, recommendation engines
- Regression (NOT the method)
  - Predict continuous data and trends
  - Used in weather forecasting, algorithmic trading
- Clustering
  - Find natural groupings and patterns in data
  - Used in pattern mining, medical imaging, object recognition

Core ML Training Types

- Supervised Learning
  - The input and output are known
  - y = wx+b, solve for w and b
- Unsupervised Learning
  - The input is known, the output is not
  - y = wx+b, solve for y approximating w and b
- Reinforcement Learning
  - Deterministic state based learning

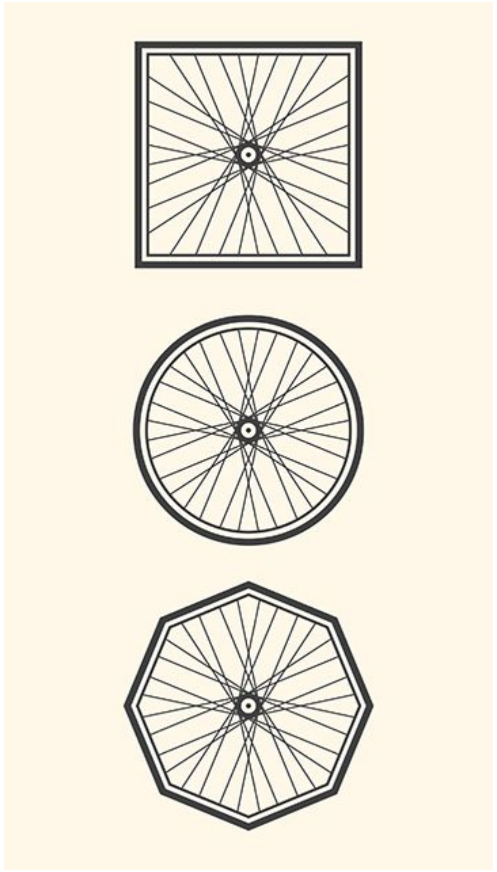
These training styles will determine how your ultimate system learns!

Solving a Problem with ML

1) Define meaningful data
    - What attributes matter?
    - What is a “good” datapoint versus a “bad” one? (define metrics)
2) Define the problem
    - Classification problem? Clustering? Weighted Probability? Image Processing?
3) Determine the method of attack
    - In this case, we have a classification problem, and to solve it we are going to use softmax regression (the method, not the type of problem)
4) Generate Training and Test Data
    - Loose rule of thumb: 70% train, 30% test
5) Train and Test the algorithm

Common ML Mistakes

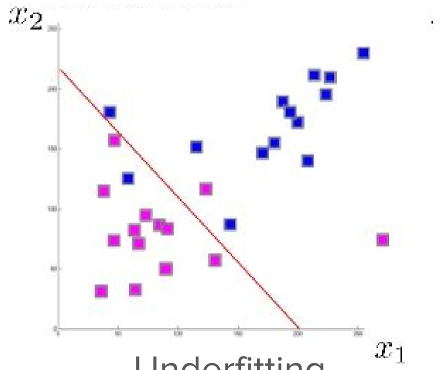
- Over/underfitting
- Solving problems unsuited for ML
- Incorrect classification of the problem
- Rebuilding the wheel
- Ignoring outliers
- Meaningless training data
- No/little test data



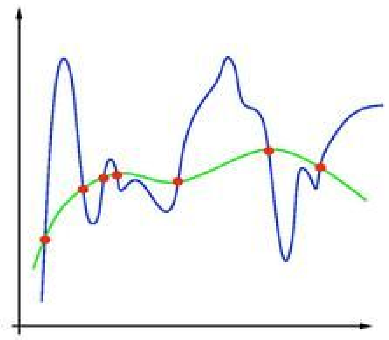
Overfitting and Underfitting

Underfitting (high bias): approximated function is too simple, favours only a few data points

Overfitting (high variance): approximated function is too complicated, does not represent data pattern



Underfitting



Overfitting

The most common and fatal failures in machine learning algorithms are based in over- (and sometimes under) fitting functions!

Regression (and you)

Regression is:

- statistics
- has a dependent variable (y) and one or more independent variables (x)
- usually takes an average value to estimate behaviour of data points
- trying to find a pattern that reflects the trend of the data

Linear Regression (Method)

- used in solving a regression ML problem
- not practically used, but easy to understand
- 2D space
- straight line function (y=wx+b)
- w: weights
- b: bias
- unbound - y doesn’t have to be between 0 and 1
- minimize the error

Error and Gradient Descent

- minimize the “cost” or “error” of an approximation to get our best guess
- Gradient Descent: find the local minimum (the least error, reasonably)
- Slightly change our weights or bias until the algorithm minimizes error

Logistic Regression (Method)

- used in solving a classification ML problem
- aka “S-curve model”
- based in probability, decide between two classes y = {1, 0}
- we’re looking for w and b like linear regression, but using a probability P(y=1/x)=wx+b instead
- When P(y=1/x) is big, our input (x) is in class 1. Small, and our input is in class 0

Softmax Regression

- a generalization of logistic regression that allows more than 2 classes (multinomial logistic regression)
- vectors! bigger than 2D
- literally “soft maximum”- local maximum
- keeps all classes, including redundancies, in an array instead of defining directly
- much more complexity, much higher time cost
- “overparameterized”

On to Tensorflow!