Deep Learning – Cho-Jui Hsieh

Application & Limitation

* Learn from data: ImageNet Challenge (recognition images – classification)
  + Prediction – error < human’s performance
  + Unable to check images when obstruct (challenges)
* Training
  + Input: vector of pixel values
  + Output: Boolean (1 or 0) -> true or false
  + Function – maps any image (vector) to 0/1
    - Fitting training data to output the correct solution
  + Use of linear function: f(x) = (W^T)\*x -> find the best W
    - Multi-dimensions
  + Each image -> point (classify input into circle and triangle)
    - Simpler data to analyze
    - W1 -> separating ~ determine the wrongfully predicted
      * Measurement of errors
    - W2 -> another vector – rotating -> less errors
    - W3 -> another – even less errors
  + Linear support vector machine (SVM)
  + Data: D = (x1, y1)… (x, y)
    - Training error: sum of loss (f(x),y)
  + Machine learning -> optimization
    - Min loss
    - Importance of data -> wrong hyperplane – false positive
    - More data -> higher confidence
* Optimization – minimizer (efficient on large datasets)
* Software: LIBLINEAR – all linear classification
* Regression: output – all real number
  + Ex. stock price prediction, genomic features
  + Loss is squared -> loss(f(x),y) = (y-f(x))^2
* Neural network – nonlinear function (pro: more complex mapping, con: data)
  + Simple model -> underfit ~ unable to improve
  + Too complex – overfit (fitting all training data, failed to perform in future)
  + 15 rules – error
  + Layering of linear models
  + Weights to min loss
* Challenges
  + Model Design – design network architecture (NLP, mining)
  + Training efficiency ~ large batch size (GPU)
  + Model size + inference speed ~ limited storage + in milliseconds
  + Sensitive to perturbation
    - Ex.

PPT Discussion 10/30/19

Deep Learning – Apps & Limitations (Cho-Jui Hsieh)

* Engine recognition – Data: image -> output: name of the dog
  + ImageNet Challenge – limited data w/ large variety of classes
  + Prediction error – outperform human perf

Roadmap: what’s machine learning? -> deep neural networks -> challenge/research

* Image classification – input (pixels) -> decision rule -> output (binary)
  + How to learn the decision rule?
  + Humans
    - Observation -> learning via observation -> memory
  + Machine learning
    - Training data (x, y) -> machine learning
      * X: vector of pixels
      * Y; 0 or 1 (one bit)
      * Decision rule: function – mapping the image vector to a result
    - Decision function – find the best function to map input to output
      * Ex: Linear function
        + Separating features (data point) via line division

Counting variable to measure loss (error range)

Minimize loss -> optimization

* + - * + Use of linear support vector machine (SVM)
        + Classification: training data -> pick the function w/ smallest loss
      * Error: loss function – loss(f(x), y) -> 0,1 – binary/ bool
      * Summation of all loss -> min function to pick the best
    - More data -> better performance – more representative – est.
      * Optimization: minimizer of a function
        + Efficient on large dataset
        + Ex: LIBLINEAR

Linear classification

* + - Regression: instead of 0/1 output, Y, f(x,) is a real number
      * Ex: stock price, genomic feature
      * Squared the loss
  + Neural network: nonlinear function – represent more complex mapping
    - Cons: required much more data
    - Model too simple: underfit (fail to improve when adding more data)
    - Model too complex: overfit (inaccurate future data – too specific)
      * Initialization of rules
* History
  + 1998: mainstream ML research
  + -2012: linear/kernel SVM
  + -now: big data + efficient network
* Nonlinear
  + Layering -> weight in each layer = determinant of matrix
    - Each layer – nonlinear function
  + Learning – min loss – solved the same optimization equation
  + Complex output space – object detection
  + Machine translation
* Challenges
  + Model design, efficiency, robustness, explainability
  + Design network architecture for each task (NLP, computer vision, mining)
  + Automatic
  + Training efficiency – transed by sampel data point -> check current pred -> update -> speed up using multiple GPUs/TPUs
  + Can be solved much shorter – power hungry
  + Model size + inference speed – limited storage + inference in msec
  + Sensitive to small perturbation
* Current: domain shift – self-driving car: env – medical data (race, sensor)
  + Different data scenarios
  + Model interpretability – why image is categorized the way they are
* Conclusion
  + Machine learning – function to map x to y
  + Training samples -> input (x) -> output (y) – function (decision rule)