Summary

Arthur J. Redfern
axr180074@utdallas.edu
Dec 03, 2018

Big Picture

Thought Chain

- Why study this topic?

- Almost any problem you can think of can be cast as 1 of 2 types of problems
  - Classification / regression
  - Generation

- Classification / regression and generation can be put into a general framework
  - Pre processing - optional domain specific processing
  - Classification / regression or generation - input output mapping
  - Post processing - optional domain specific processing

- xNNs can be used for the input output mapping
  - xNNs are universal approximators so if a mapping exists they can approximate it (under some technical conditions)
  - So potentially they can be used to solve almost any problem you can think of
  - And in practice they're a key component of the top performing methods for many important tasks

Thought Chain

- What is the problem with this?

- We don't know the input output mapping that we need
  - It's not given what the optimal network structure is to approximate the input output mapping (design) and how to find the parameters that control the network structure mapping (training)
  - It's not given how to practically and efficiently realize network structures (implementation)
  - It's not given what different applications require in terms of pre and post processing and along with different design and training strategies (vision, speech, language, ...)

- So the obvious purpose of the classes this semester was
  - To review the basic math needed for pre processing, xNNs and post processing
  - To learn network design strategies to accomplish different goals
  - To learn end to end network training from data
  - To learn efficient network implementations
  - To learn specific requirements of common applications

Thought Chain

- However, an equally important purpose of this course is to learn how to learn
  - The last paper isn't written, there are many more each day
  - You have a good base to work from
  - You have the potential to understand new ideas from others and create new ideas yourselves
  - Your ability to do this improves with practice

Topics We Covered

Observation: We Covered A Lot Of Material

That's by design: if you want to do meaningful work in this area, it helps to know a lot about a lot

- The first 3 topics in the linear algebra lecture were sets, fields and vectors
- The last 3 flagship topics in the application lectures were object based image segmentation, speech to text transduction and language translation
- And somewhere in the middle we talked a little bit about semiconductor device physics

Math

- Linear algebra
  - Sets, fields, vectors, matrices, tensors, functions, vector spaces, normed vector spaces, inner product spaces, matrix vector multiplication, matrix matrix multiplication, CNN style 2D convolution, DFTs and PCA
- Calculus
  - Derivatives, sub derivatives, partial derivatives, gradients, Jacobians, chain rule, critical points, gradient descent, automatic differentiation with reverse more accumulation and universal approximation
- Probability
  - Probability spaces, events, random variables, expected value, normalization, law of large numbers, central limit theorem, random processes, stationarity, time averages, ergodicity, entropy, mutual information, Kullback Leibler divergence, data processing inequality, compression, Huffman and arithmetic coding
- Algorithms
  - Comparison sorts, sequential merge sort, parallel merge sort, pooling, median and rank order filtering

Networks

- Design
  - Goals, size considerations of the network, feature maps and filter coefficients, problem complexity, graph specification, layer types, tail body head decomposition, tail designs, head designs, body designs including chain, parallel, dense and residual structures, optimized architecture search and visualization

- Training
  - Supervised learning, differences with function optimization, convergence, overfitting, regularization and generalization, the curse of dimensionality, training validation testing data splits, natural data, labeling, cleaning, synthetic data, hand engineered generation, learned generation, data augmentation, random initialization, transfer learning, curriculum learning, batch normalization, group normalization, stochastic width and depth, classification and regression losses, loss surface shapes, unequal class weighting, aux network heads, multiple network heads, check pointing, reversible architectures, batch size, weight update methods including SGD, momentum, AdaGrad, RMSProb and Adam, learning a solver, weight decay, gradient noise, gradient clipping, synchronous stochastic gradient descent and hyper parameter selection

Implementation

- Networks
  - Theoretical complexity, precision, hardware size vs model size, training vs testing, data formats, quantization, network sizing and network simplification
- Hardware
  - Moore's law, Dennard scaling, dark silicon and dark memory, power, roofline models, SoC architectures, domain specific architectures, control, memory, data movement, compression, Amdahl's law, computational basis, matrix multiplication primitive, inner, outer, Strassen style, input power of 2, sparse and analog matrix multiplication, Winograd style convolution, sort primitive, tree configurations, torus configurations and hardware design examples
- Software
  - High level graph specification, high level graph transformations, static vs dynamic graphs, sessions, graph compilers, low level graphs, runtime initialization, runtime execution, software design examples, predicting performance and benchmarking

Applications

- Vision
  - Image capture and processing, hardware design examples, classification, pixel segmentation, up sampling, encoder decoder with skip connections, Atrous convolution, spatial pyramid pooling, 1 and 2 stage approaches to multiple object detection, feature pyramids, anchor boxes, spatial pyramid pooling, RoI pooling, region proposal networks, iterative methods, non maximal suppression, confidence threshold, intersection over union, precision, recall, precision recall curve, 2 and 3 stage approaches to object based segmentation, RoI align, depth estimation, stereo fundamentals, motion estimation and motion fundamentals

- Speech
  - Speech and audio signal chain, sampling, pre emphasis, windowing and spectrograms, MFCC, RNNs, GRUs, LSTMs, bi directional, pyramidal, sources of variability, speaker verification, speaker recognition, wake up, limited vocabulary speech recognition, confusion matrix, speech to text, sequence to sequence models, CTC, beam search, language model, auto segmentation, RNN transducer, attention, transition possibilities, alignments, text to speech, intermediate representation conversion and audio signal conversion

- Language
  - Word embeddings, the distributional hypothesis, SVD based, continuous bag of words, skip grams, visualization, word similarity and analogies, task specific optimization, language modeling, N grams, neural language models, perplexity, character based, translation, sequence to sequence, greedy and beam search decoding, structured prediction, attention, architecture exploration, self attention, transformer and BLEU

Random Generally Useful Stuff

- An incomplete laundry list
  - Paper reading and summarization
  - Paper writing
  - Presentation styles and formats
  - arXiv paper dates
  - Programming interview jokes
  - Guessing numbers between 10 and 50
  - Computing the square root of 2 in elementary school
  - Hobbies
  - ...

Topics We Didn’t Cover

Observation: There's A Lot Of Material We Didn't Cover

- Not sure if this is good or bad
  - Perhaps it's just the reality of a large field viewed through the lens of a 1 semester class
- Realize that these uncovered topics are everywhere
  - Within the topics areas that we covered
  - Topics that we didn't cover
- Realize that there are new developments every day in this field
- Your only hope: learn how to learn
  - Hopefully this class contributes positively to your ability to do that
  - In general, the more stuff you know the easier it is to learn something new (as it's likely similar to something you already know)

Ex: Networks

- While it feels like we covered a lot of network topics
  - Layers, design, training, implementation
- There's a lot more that we didn't cover
  - For example, networks with an external memory developed in the context of language applications, interesting to think about in the context of other applications
- On the upside, the network information presented during this semester should provide a good basis for understanding new network designs

Ex: Vision, Speech And Language

- While it feels like we covered a lot of vision topics
  - Classification, pixel segmentation, multiple object detection, object based segmentation, depth estimation and motion estimation
- There's a lot more that we didn't cover
  - Variations within the topics that we covered
  - Other topics that we didn't cover like point clouds, facial recognition, handwriting recognition, optimizing object boundaries, ...
- The same observation is true for speech and language
- On the upside, the vision, speech and language information presented during this semester should provide a good basis for understanding different topics in these fields

Ex: Games And Art

- A lot of games related work is leading to interesting intersections of reinforcement learning with xNNs
  - Board games: backgammon, go, chess, ...
  - Video games: Atari variants, StarCraft, Dota, ...
  - The creation of new simulated environments for training and testing algorithms
- A lot of art related work is leading to interesting network structures for generative systems
  - Style transfer
  - GANs
  - Music, images and video
- My goal is to add these 2 topics to the course next semester
  - They'll be posted to GitHub if you would like to see them
  - Alternatively, if you're interested in 1 or both of these, you should teach yourself (all the information is out there, it just takes time to go through and organize it for understanding)

The Future

Interpolation

These are pretty easy to guess

- Gradually more and more theoretical questions are answered
  - People will quit complaining about a lack of theory
  - Note: there's already a lot of theory and the complaining seems unreasonable even now
- Everything gets better in proportion to computing
  - Maybe slightly faster than process technology but not radically faster
  - Will get a boost for going to lower precisions and building more efficient architectures
- Basic methods are applied to more and more fields
  - Just like there was computational <science> that became <science>, there will be machine learning for <science> that eventually becomes commonplace enough that it's again called <science>
  - Likely some developments in some of these fields propagate back to the core and become used in many fields
  - But in general convergence towards a few core techniques with some specialization (e.g., pre and post processing) around them

Extrapolation

These are comments and guesses with a much higher variance and by choice are somewhat constrained

- Theory
  - It would radically change what we're capable of if algorithms are developed with processing complexity proportional to the information rate vs the data rate
- Computation
  - Over the course of a longer time scale what happens if compute capability increased by 1000? 1000000? 1000000000?
  - What happens if a different computing paradigm (maybe quantum) becoming commonplace and let's us design algorithms that solve complementary problems and providing another fundamental step up in AI
- Applications
  - We figure out how to handle more and more complex problems by putting together more and more simple problems
  - To an outsider this starts to feel like real intelligence vs specific / isolated problem solving
- Humanity
  - We become much more closely integrated with AI; everyone's hand is forced in this direction or they're left out / disadvantaged
  - We observe a person over the course of their life and get good at simulating them
  - We question our own intelligence

Your Place In It

- A few suggestions
  - Add a new hobby every few years
  - Figure out what's at the intersection of what you love to do and what you're good at
  - Do that
  - Don't be afraid to change

Next Time

Spiral Presentation Of Material

- I like the spiral presentation of material with increasing levels of complexity
  - A 1st time at a basic level during the math lectures
  - A 2nd time at a more advanced level during the design, training and implementation lectures
  - A 3rd time focusing on specific cases during the application lectures
  - I'm going to stick with this basic format but make a fair amount of small refinements to the slides

- But I think some of the in class the material was too much to take in all at once
  - My strategy this semester for material presentation was
    - 1st view during the lecture at the lecture pace
    - 2nd view after the lecture at a relaxed offline pace
  - It may be better if I post lectures ahead of time so a person gets 3 looks and it makes the in class portion more effective
    - 1st view before the lecture at a relaxed offline pace
    - 2nd view during the lecture at the lecture pace
    - 3rd view after the lecture at a relaxed offline pace
  - I'll probably give this a try next semester

Implementation Tool And App Project

- I'm going to redo this part of the grading
- Fall 2018 strategy
  - Pencil and paper tests plus homework for the 1st half to acquire basic skills
  - Implementation tool and application project for the 2nd half to apply skills
- My observation
  - It was a reasonable strategy
  - But the 2nd part didn't work out like I thought
- Spring 2019 plan for grades
  - 3 pencil and paper tests (25% each) after the math, networks and applications portions of the semester
  - Continual homework assignments (25% total) throughout the semester

App Implementations As HW Assignments

- I still believe application implementation skills are important
- So my plan is
  - Get everyone using a high level software library on a regular basis sooner
  - Use homework assignments throughout the semester for application implementations
- I think this will serve as a better jumping off point for people who are interested in doing more on their own; I'll figure out an alternative way of encouraging this

One More Thing

Thank You!

- You're the 1st class I've taught
  - You're all special to me
- I appreciate you sticking with me though this
  - The last time I was in class on a regular basis was ~ 20 years ago as a student
  - So I know at times the lectures and assignments were a little rough
- I wish you all the best for happiness in your lives and careers
- Keep in touch!

The Final Slide

Slides	
Introduction	39
Linear algebra	84
Calculus	56
Probability	74
Algorithms	27
Design	117
Training	139
Implementation	183
Vision	110
Speech	124
Language	79
Games	2
Art	2
Summary	28
Total	**1064**
Days	106
Slides / day	10.0