Brian Craft

CSC 480 Presentation Notes

* Generalizing AI
  + MarI/O grows a neural network to play Mario.
  + MarI/O is built off an algorithm called NEAT.
  + LearnFun is a pair of algorithm that learns to play from a recording of you playing a game. Then creates heuristics for which bytes are successful and which aren’t. So this is more general, not specific to one level or one game. Can play any NES game.
* Genetic Algorithm for Elevator Dispatch
  + How do you map elevators to the optimal people for them to pickup.
  + As it stands, we try to minimize pickup time.
  + But it costs money to move elevators up and down, due to electricity, but could have heuristics for this as opposed to just time.
  + Use approximation algorithms, since the problem is NP hard.
  + Long story short…look at genetic algorithms.
* Intelligent Tutoring Systems
  + Personalized systems for each student.
  + Detect learning emotion and provide feedback based on emotions.
  + These are learning agents. Environment is the learner and the sensor can be keyboard inputs and speech recognition.
  + Can you Bayesian Models, Hidden Markov Models or Fuzzy Decision Trees.
  + Can improve learning emotion and performance and increase motivation for learning.
* Deep Learning in NLP
  + Good performance using linear models.
  + Need large corpus to train on.
  + Can find semantically similar words.
  + Use deep neural network architecture.
* Jokebot using Wit.AI
  + Part of facebook used to help add NLP to apps.
  + Build on facebook deep text, which uses deep learning neural networks to learn nlp.
  + Wit.Ai can build a chat bot system that will call a function on certain messages.
  + Can add in sentiments as well.
  + You can make some cool Chuck Norris jokes.
* Starcraft
  + Real time strategy game
  + Command and Conquer style
  + Game state isn’t complete
  + Partially observable, deterministic,
  + Durative Actions…actions that take some time.
  + Can bridge the real world a bit.
  + Imperfect info, durative actions and real time information.
  + Deep Mind is more general intelligence.
* NN in Computational Neuroscience
  + Study of the brain as an information processing system.
  + NN and brain are quite similar how they are structured.
  + Brain can juggle information and separate out concepts and items better then a neural network. Not necessarily just from how or what neuron it arrives from.
  + Neural Networks can model neuron connections and make a giant artificial image of a brain and how it’s connected.
* Piano Mover
  + Minimum constraint removal problem, if no obstacle free path, what is the minimum you can move to have a constraint free path.
  + These can be NP hard problems.
  + The greedy algorithm isn’t always optimal.
* Is your path obstructed by polygons?
  + How can you remove shapes to get from one point to another.
  + Chain vertices to goal to make it.
  + MCR in 2D is NP hard so a good heuristic is needed.
  + Best first solved the problem.
  + Could be used for the Mars Rover.
  + Could apply different costs to different polygons.
* Google Translate Neural Machine
  + Use a recurrent neural network that didn’t work until better hardware was available.
  + Used to be word-based models. Problem is sequence of words have different meanings and words aren’t a one to one match.
  + Could also cause odd orders of the words.
  + Neural machine translation does entire sentence translation. Uses LSTR NN with 8 encoder and decoder layers.
  + Model can still make large errors.
* Deep Learning for Autonomous Vehicles
  + Computer vision processing.
  + Layers analyze each pixel of the image.
  + Analyzing the image tells the car what action to take.
  + Some companies are trying to create autonomous racecars.
* Metacognition
  + What humans have to do when they don’t know something or create a strategy to learn what they need too?
  + Applicable because agents have to solve problems and figure out the best way to learn what they need to too to get to an inefficient solution.
  + Long term autonomous decision making.
  + Want to get towards an agent that can reason about the world.
  + Dual-Cycle Arch (MIDCA)
* Solve Rubik’s Cube
  + State is 6 2D matrix that are 6x6.
  + Successors are turning the rows and columns.
  + Need a heuristic because the branding factor and search space is so large.
  + Could use the manhattan distance of the tiles. But since it’s 3D quite challenging.
* Amazon Alexa
  + Can make an Alexa using a raspberry pi.
  + Can connect to the Amazon Alexa system.
  + Need a mic, speaker and raspberry pi
  + Took about 60 minutes. Need authentication keys.
  + Alexa has a server build in node JS which authenticates device, Java client is the bulk of it which is a large code base that manages devices, input and output and keeps track of user requests.
  + There is more skills then Amazon implemented. Actually could expand this.
* Maze Solving Program
  + Uses the pixels to analyze the maze state space.
  + The maze is thought of as a graph, or an adjaceny list.
  + Hash map to vertices that can be reached from each vertex
  + Use search to solve the solution
* AI for Gomku
  + Use score as the heuristic
  + Large branching factor
  + Use multiple heuristics and stack them in order of importance.
  + He made a really cool AI game play against itself.
* Sudoku
  + 2 methods to call back and forth until the solution is solved.
  + Basically solves the cell and checks for conflicts.
  + Moves down the first column, top to bottom then goes to the column to the right.
  + If there is a conflict, it resets the cell to blank and just keeps going. Uses search.
* Emotiv
  + Brainwaves to control.
  + Something that could observe brain waves and translate it into code.
  + Translates brainwaves into actions.
  + Made a cube move towards the screen and make the cube disappear by imaging it happening.
* Project Malmo
  + AI project out of Microsoft that is built on minecraft
  + Agent moves using x,y and z coordinates to model a 3d world.
  + Python implementation is quite possible. Not to tough to implement.
  + Learns the correct path as you encode a solution and an environment.
  + Map physical world to virtual world to test and create systems to solve problems.
* NLP and Deep Text
  + Prolog was built specifically for language processing.
  + CoreNLP is a Stanford library for basic NLP.
  + Sentences don’t abide by linguist models making it tough to model.
  + Ambiguous words are challenging. Words with same spelling but different meaning. This causes a lot of confusion.
  + Deep learning helps to solve for this.
  + Facebook uses Deep Text and implemented that paper.
* Hierarchical Temporal Memory
  + Want to learn how the brain runs and replicate it.
  + Brain is different from traditional AI.
  + Prediction is fundamental for the brain.
  + Only about 2% of neurons are active at once.
  + Brain learns in sequence and patterns.
* Reinforcement Learning
  + A game that has to eat food. Agent can go up and down or left and right.
  + Dynamic environment since random foot points are generated.
  + Multi agent trained jointly.
  + Monte Carlo Policy Gradients to make model.
* Sabermetrics
  + A way to analyze baseball statistically
  + As did nearly as good as the Yankees with less than a third of the budget.
  + How do you compete with big market teams in an unfair system.
  + New way to select players that are of value and constructing a team.
* Boston Dynamics
  + Owned by Google now.
  + Made a robotic dog that is a quadruped.
  + Robots with legs can cross certain terrains better then wheels.
  + Legs have extra degrees of freedom that add more complexity.
  + Lots of actuators
  + AI makes motor commands to get from the start position to the goal positions, but the world isn’t ideal and static since there is elements.
* Cyber Immune System
  + Data breaches are a huge cost to companies
  + Cyber security systems aren’t up to snuff to stop some of the attacks today.
  + Today, building models that mimic the human immune system to help stop cyber attacks.
  + System can create models that predict the probability of a threat.
* Killer Instinct
  + People hate unfair AIs.
  + Hard AIs are safe but boring.
  + They’re not fun…just lots of energy or equipment etc.
  + Too good an AI would be able to basically beat you at every move and become unbeatable.
  + Trains a personal AI based on your fighting style.
* Starcraft
  + Original AI was cheating which was making people mad.
  + Explore the map so the state isn’t always known. But the AI is given the full state in the beginning.
  + Used a decision tree to make a more basic AI. If the player expanded, attack otherwise don’t do much.
  + API for C++ and Java to make your own AI so it doesn’t cheat.
  + Would perform terrain analysis.