Comp 6321 - Machine Learning - Project report

Federico O'Reilly Regueiro

December 15^{th}, 2016

1 Introduction and motivation

The current proliferation of readily available data collections that has been spawned by the inetrnet has set the stage for machine learning as a pervasive means to develop different kinds of intelligent systems.

It is, however, the heterogeneity and inconsistency of such data that limits what can be done by regular supervised learning. Some of the biggest efforts put forth while tackling a supervised learning problem continue to be collection and curation of data sets.

Additionally, the notion of labeling all or most input instances becomes unwieldy under certain circumstances.

The lack of clear or constant labels to inputs gives rise to two forms of learning that are not supervised. Unsupervised learning, where algorithms aim at finding underlying structure in the data and reinforcement learning; which is in some ways between supervised learning and unsupervised learning, where 'labels' are sparse and time-delayed [**?**]. This same sparsity and delay give rise to what is referred to as 'the credit assignment problem', where given a certain outcome from a series of actions, it is difficult to ascertain which, if any, of the actions leading to the outcome bears the largest responsibility for the outcome.

There are several approaches to reinforcement learning, we focus on a particular type which tries to assign the credit of a given outcome somewhat evenly amonge the events leading to it. It does so by supposing that at any point in time the reward (or penalty) is equal to the reward gained at that point plus the sum of possible discounted future rewards. The rationale for discounting rewards over time corresponds to the notion of rewards being more desirable now than later on[1]. This idea, central to temporal difference learning will be presented more fully in section **??**.

Applications of reinforcement learning are varied and curretly under development in fields such as vehicle control, robotics, prediction of streaming data such as that of financial applications and gaming to name a few.

As Moriarty and Miikkulainen put it, 'games are an important domain for studying problem-solving strategies.' [**?**] Traditionally games, due to their well-

[1] As Andrew Ng puts it in his online lecture on the topic, we might be dead tomorrow. [**?**]

defined rules, state-transitions and goal, have made a good sandbox for the development of any form of intelligent agents.

Much has been achieved in this field, recently Google's Alpha-go bested top-ranking go-player Lee Sedol. Go had traditionally been seen as the unattainable goal in cmoputer game-playing given the vastness of it's state-space. Alpha-go achieved this outstanding result with the combination of different strategies, via 3 Neural Networks [**?**]. Although there are several formulations, traditionally there have been two main approaches to temporal difference: to temporal difference learning

2 Description of approach

-Many forms of TD learning (cite paper)

-What we do, see the ideal sequence of time-discounted predictions as a smooth progression (*Think of a more mathematically friendly term) —describe the td error equation (requires reading Tesauro a bit more)

-We use a convolutional net and a fc net (sort of justification) —Justification and description of convolutional nets are intermingled—justify the lack of pooling (look for citation)

Other approaches and why we don' t use them —Q-learning, currently mentioned often. Difference between Value expansion and q or policy expansion and implications for the problem —Older - ENN - interesting but requires starting generations very much in advance

3 Setup

-using tensorflow, w / python 3.5 api and numpy —how tensorflow works, still getting the hang of it

-Brief description of the board model as an input matrix -Description of the network (Latex NN package?)

–Avoue qu' il y a knowledge representation - we avoided the symmetry proposed by Ugosky(cite) given what he describes as a problem but expanded the idea

–Important to mention: output is squashed token ratio

-Routine for training / testing - given time

Justification of training against a random player, brief mention of next steps —Exploration of the space (exploration/exploitation) –question: is play-

ing black generally a disadvantage? nets tend to start with more losses than wins –question: is playing whites (times -1) much different

4 Results

-Expected- evaluations of the board should move towards 0 at the onset and relatively smoothly progress towards the end result

Problems constraining the obtained results - memory leak, resulting lack of time due to long training

No comparison of different lambda, gamma, lr dropout values – many hyperparams to tune!

convergence? How does it stack up. How many epochs, plots of wins

plots of token ratios per epoch

5 Future work

Compare initialization vs randomPlayer with different parameters
Initialize several nets against randomPlayer
pit them against each other for training
they should all develop unique strategies so then they help each other out

Maybe combining training with random player and then Gen Algo approach?