课题来源及意义：

Deep Reinforcement Learning (DRL)由于其既有Deep Learning的表示能力，同时也具有Reinforcement Learning的 trial-and-error 的学习能力，被视为通往AGI的一种方法和手段。但由于Deep Learning在训练模型需要巨大的计算量，同时Reinforcement Learning在前期的学习过程中无法产生高质量的训练数据，使得DRN在实际运用时，面临着Policy震荡，不能学习到最优策略，训练时间长等问题。

Multi-agent System中包含多个agent的交互与控制，被广泛运用于智能机器人，交通控制，分布式预测、监控及诊断等具有高价值的领域。但是由于具有多个agent，不同agent可能具有不同或相同的目标，为获得最优策略，agent必须根据其他agent的行为来调整策略。为了将DRN有效运用Multi-agent System中就必须解决DRN收敛速度的问题。

所以设计一种高效算法加速DRN的收敛速度具有现实意义。

国内外发展状况：

2013年，Volodymyr Mnih等人在NIPS发表《Playing Atari with Deep Reinforcement Learning》，首次提出Deep Q-learning的概念，同时将DRN运用在Atari的游戏上，第一次在end-to-end的游戏控制上达到了人类的水平，并在2015年NATURE 杂志上发表《Human-level control through deep reinforcement learning》进一步改进网络结构，提高性能。为了进一步减小对于Q值估计的偏差，在2015年，Hado van Hasselt 等人提出了Double Q-learning 学习方法。同时为了解决在序列决策中reward稀疏所带来的Experience reply效率低下的问题，Tom Schaul等人在2016年提出使用Prioritized Experience Reply来改变Memory抽样策略。Ziyu Wang等人在2016年提出一种新的网络结构，将Q值分解为state-value和advantages for each action，提高了DRN在Atari上最后效果。为了改进DRN的学习速度，Volodymyr Mnih等人在2016年提出一种异步学习算法：asynchronous advantage actor-critic（A3C）。

此外，结合DRN和Monte Carlo tree search的AlphaGo在2015年4:1击败顶尖职业棋手李世乭，代表着DRN在实际运用中具有重要的价值。

研究目标：

改进已有DRN网络结构，设计新的学习目标，加速DRN的收敛速度，同时探索在Multi-agent System下基于DRL的协作算法设计。

研究内容：

探索如果在已经有一部分Demonstration的情况下，如何设计网络结构与学习目标，加速DRN的收敛速度与提高性能；探索如果不同任务间具有相似性，如何利用已有网络，加速DRN的收敛速度与提高性能。

研究方法与手段：

首先实现《Playing Atari with Deep Reinforcement Learning》中描述的DQN算法，然后在此基础上实现《Deep reinforcement learning with double Q-learning》中改进的DQN算法，同时使用OpenAI提供的gym环境进行学习，在Atari的游戏上验证DQN算法的可行性，对比A3C等其他DRN算法，并调节不同参数，观察分析不同参数对于收敛速率的影响。其次，通过训练出的网络与人来产生示例，尝试利用示例来调整参数，加速DRN网络的收敛速度，或者通过示例来影响DQN算法在前期的动作选择，比如启发选择较优的动作，避免产生低质量的样例。最后，利用已经训练出的网络，并找到相似的任务，尝试使用已有网络加速相似网络的收敛速度。

进度安排：

阅读相关文献，如《Playing Atari with Deep Reinforcement Learning》，《Deep reinforcement learning with double Q-learning》等，并实现最原始的DQN与Double DQN，了解参数意义，同时与A3C等算法进行比较，加深对DRN的认识与学习，同时对比基于表格Reinforcement Learning，进一步深化了解DQN预计一个月左右。

尝试不同方法比如对Memory的添加作限制与过滤，保留有意义的学习样例；采用启发式的学习目标，在前期就可以避免生成低质量的样例；根据不同表现来选择模型更新速度，如在前期的时候可以快速更新网络权重，使得网络调整速度加速，在中后期减小权重更新频率，降低更新权重带来的时间开销等来加速收敛速度，预计一个月左右。

阅读迁移学习的相关文献，尝试找到不同任务相似性的衡量方法，然后使用相似性的度量来利用已有任务训练完的模型来加速未知相似任务的学习速度，同时阅读CNN迁移的相关文献，参试利用已有权重迁移到未知相似任务上，预计一个月左右。

实验方案可行性分析：

目前网上相关的Deep Learning框架有[Caffe](https://github.com/BVLC/caffe)、[CNTK](https://cntk.codeplex.com/)、[TensorFlow](https://github.com/tensorflow/tensorflow)、[Theano](https://github.com/Theano/Theano)和[Torch](https://github.com/torch/torch7)，Keras等，通过使用比较，预计使用Google开源的TensorFlow与Keras进行Deep Learning模型的训练，应该能较好地完成深度学习模型的训练。Reinforcement Learning部分，预计使用keras-rl框架编写与修改。

关于加速传统Reinforcement Learning的方法一直是学术界研究热门，并结合提出一些列加速方法，比如对于基于Boltzmann的策略选择，基于模拟退火的策略选择，采用Demonstration对学习目标的reward-shaping，结合迁移学习对于相似环境的扩展等等。

在Deep Learning中，加速学习速度的方法也是层出不穷，比如随机梯度下降；mini-batch梯度下降；加入Momentum因子一方面加速下降速度，一方面避免陷入局部最优；使用Nesterov，对梯度更新对出校正；Adagrad，对学习率进行约束，Adadelta，添加固定大小的值，近似计算对应的平均值；Adaptive Moment Estimation，Adamax，Nadam等等

所以可以结合已有的加速策略，并进行改进，使其能够适用于DRN的学习框架，可行性较高。

此外，将已有任务的模型运用到未知相似任务中，一方面可以借鉴迁移学习中的概念，另一方面可以借鉴Deep Learning中类似图片风格转移的方法。同样也具有较高的可行性。

具备的实验条件：

指导教师一名，实验室电脑一台(配置：i7-600 cpu, 16G内存，AMD R7 430)，深度学习框架： TensorFlow，Keras，keras-rl，游戏环境：Open AI gym

主要参考文件：

Article (van2015deep) Van Hasselt, H.; Guez, A. & Silver, D. Deep reinforcement learning with double Q-learning CoRR, abs/1509.06461, 2015

Article (wang2015dueling) Wang, Z.; de Freitas, N. & Lanctot, M. Dueling network architectures for deep reinforcement learning arXiv preprint arXiv:1511.06581, 2015

Article (mnih2015human) Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G. & others Human-level control through deep reinforcement learning Nature, Nature Publishing Group, 2015, 518, 529-533

Article (silver2016mastering) Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M. & others Mastering the game of Go with deep neural networks and tree search Nature, Nature Publishing Group, 2016, 529, 484-489

Article (mnih2013playing) Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D. & Riedmiller, M. Playing atari with deep reinforcement learning arXiv preprint arXiv:1312.5602, 2013

Article (schaul2015prioritized) Schaul, T.; Quan, J.; Antonoglou, I. & Silver, D. Prioritized experience replay arXiv preprint arXiv:1511.05952, 2015

Book (sutton1998reinforcement) Sutton, R. S. & Barto, A. G. Reinforcement learning: An introduction MIT press Cambridge, 1998, 1

Misc (sutton2011reinforcement) Sutton, R. S. & Barto, A. G. Reinforcement learning: An introduction Cambridge Univ Press, 2011

Article (lecun2015deep) LeCun, Y.; Bengio, Y. & Hinton, G. Deep learning Nature, Nature Publishing Group, 2015, 521, 436-444

Article (bengio2009learning) Bengio, Y. Learning deep architectures for AI Foundations and trendstextregistered in Machine Learning, Now Publishers Inc., 2009, 2, 1-127

Article (pan2010survey) Pan, S. J. & Yang, Q. A survey on transfer learning IEEE Transactions on knowledge and data engineering, IEEE, 2010, 22, 1345-1359

Article (abadi2015tensorflow) Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M. & others TensorFlow: Large-scale machine learning on heterogeneous systems, 2015 Software available from tensorflow. org, 2015, 1

InProceedings (ngiam2011optimization) Ngiam, J.; Coates, A.; Lahiri, A.; Prochnow, B.; Le, Q. V. & Ng, A. Y. On optimization methods for deep learning Proceedings of the 28th International Conference on Machine Learning (ICML-11), 2011, 265-272