|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **교과목 명** | | 종합설계1 | | | | | | | |
| **설계 제목** | | 스타크래프트 AI | | | | | | | |
| **설계 기간** | | 2019년도 2학기 | | | | | | | |
| **지도교수** | | 이재환 | | | | | | | |
| **팀원** | | **이름** | 김범준 | **학번** | 2015124035 | **☎** | 01041781793 | **E-mail** | tk2010kbj@naver.com |
| **이름** | 이태형 | **학번** | 2015124176 | **☎** | 01053759177 | **E-mail** | rkrp15@gmail.com |
| **이름** |  | **학번** |  | **☎** |  | **E-mail** |  |
| **목표설정** | **설계 목표** | 딥러닝 기술이 등장한 이례로 이 딥러닝 기술이 여러 분야에 광범위한 적용이 가능하다는 것이 확인되었고, 실제로 적용되어 좋은 성능을 보여주며 우리 생활에 밀접한 영향을 끼치게 되었다. 그 중에서도 초기에 딥러닝 기술의 등장을 세상에 알렸던 것은 2015년 알파고의 등장이었다. 알파고는 기계가 인간을 이길 수 없다고 평가 받고 있던 바둑에서 이세돌에게 승리하며 인공지능 기술의 새로운 전환점이 등장했음을 알렸다. 그 이후 바둑 뿐만 아니라 수많은 게임에서 딥러닝 기술을 적용한 인공지능이 만들어져, 인간을 쉽게 능가하는 모습을 보여주고 있다.  우리는 딥러닝 기술을 사용한 스타크래프트 인공지능을 실제로 설계해보려고 한다. 스타크래프트는 알파고가 등장하기전 경우의 수가 너무 많아서 인고지능이 인간을 능가할 수 없다고 평가받았던 바둑보다도 훨씬 더 많은 경우의 수가 존재하는 게임이다. (바둑: 10^170가지 경우의 수. 스타크래프트: 10^1685가지 경우의 수) 그럼에도 불구하고 이미 많은 스타크래프트 인공지능이 만들어져 인간과 대등하게 겨루고 있다. 우리는 기존의 스타크래프트 인공지능과 다른 게임 인공지능들이 어떤 방식으로, 어느 정도의 성능을 달성했는지 조사하여, 여러가지 기법들을 적용해가며 설계를 진행할 계획이다. 어떤 딥러닝 알고리즘을 적용했을 때 다음에 제시한 설계 목표들에 대해서 얼만큼의 성능을 보여주는지를 설계 과정에서 확인할 것이다.  현재까지의 구체적인 설계 목표는 다음과 같다.  1. 뛰어난 게임 플레이 실력  가장 1차적이고 근본적인 설계 목표는 인공지능이 게임을 잘하게 하는 것이다.  2. 학습의 효율성  얼마나 잘하는가 라는 주제를 넘어서, 얼마나 빠르게 학습시킬 수 있는가 라는 주제가 딥러닝 분야에서 대두되고 있다. 다른 인공지능들에 적용되는 여러 알고리즘들을 적용시켜보며 학습의 효율성면에서 좋은 성능을 보여주는 알고리즘이 무엇인지 확인한다. | | | | | | | |
| **설계 규격** | 스타크래프트 인공지능 설계에서 사용하는 2가지 오픈소스 프레임워크  1. BWAPI  스타크래프트 게임 환경 정보를 프로그래밍 환경에서 받아볼 수 있고, 게임 컨트롤을 프로그래밍적으로 처리할 수 있도록 만들어진 프레임워크이다. C++로 작성되었고, visaul studio에서 컴파일 된 .dll 파일을 chaoslaucher로 삽입하는 방식으로 작동한다.  2. tensorflow  구글에서 만들어 배포한 현재 가장 유명한 딥러닝 오픈소스 소프트웨어 라이브러리 중 하나이다. Python, java, C++, javascript 등 다양한 프로그래밍 언어를 지원한다. 그 중에서 python으로 작성된 API가 가장 다양한 기능을 지원한다.  # C++ 기반 BWAPI와 Python 기반 tensorflow 연동하기  BWAPI은 C++로 작성되었고, tensorflow는 C++ API를 지원하기는 하지만 대부분의 기능들이 python API로 작성되어 있다. 그래서 BWAPI C++ 함수를 python 래퍼로 감싸, BWAPI 환경과 python tensorflow 환경을 연동하여 스타크래프트 인공지능에 딥러닝 기법을 적용한다. | | | | | | | |
| **합성/분석** | **관련 기술** | TensorFlow  구글에서 오픈소스로 공개한 기계학습 라이브러리이며 이것을 사용해서 아래의 방법들을 사용해 스타인공지능을 학습시킬 예정이다.  CNN(Convolutional Neural Network)  컨볼루션 신경망은 입력된 이미지에서 다시 한번 특징을 추출하기 위해 마스크를 도입하는 기법이다.  맥스 풀링 층을 추가하여 컨볼루션 층을 통해 이미지의 특징을 도출했을 때 그 결과가 크고 복잡한 것을 축소할 수 있다.  드롭아웃 기법을 사용해서 은닉층의 일부노드를 임의로 꺼서 학습데이터에 지나치게 치우쳐서 학습되는 과적합을 방지할 수 있다.  플래튼 함수는 2차원 배열상태인 이미지를 1차원으로 바꿀 수 있다.  RNN(Recurrent Neural Network)  순서대로 입력된 데이터에서 앞서 입력받은 데이터를 기억해둠  기억된 데이터가 얼마나 중요한가 판단하고 별도의 가중치를 두어 다음 데이터로 넘어감  입력 값, 출력 값의 설정에 따라서 여러 상황에 적용가능  - 다수 입력 단일 출력 (ex 문장을 읽고 뜻을 파악)  - 단일 입력 다수 출력 (ex 사진의 캡션 만들기)  - 다수 입력 다수 출력 (ex 문장번역)  DQN(Deep-Q-Network)  구글 딥마인드에서 개발한 알고리즘으로 state, action에 따른 reward를 받아 학습을 하며 learning rate(α)와 discount rate(γ)를 사용하여 알고리즘의 정확성을 높혔다.  Dummy Q Learning  이번에 얻은 reward와 그 다음 Q의 최대값으로 Q를 업데이트한다.    한번길을 찾은 후 새로운 길을 찾지 않으며 찾은 길이 최단거리가 아닐 수 있다는 문제점이 있음    E-Greedy  Dummy Q Learning의 문제점을 해결  e라는 값을 설정해두고 랜덤값이 e보다 작으면 랜덤한 action을 취한다.    이 방법은 쓸데없이 잘못된 길을 너무 많이 갈 수 있음  Decaying E-Greedy  E-Greedy의 문제를 e값이 후반부에 갈수록 작아지게 하여 해결함    Add Random Noise  위의 E-greedy방법은 원래의 reward값과 상관없이 랜덤으로 길을 선택하지만 reward가 더 큰 쪽을 더 자주 선택하도록 만든 방법이다.  각각의 값에 랜덤한 값을 더한 후 최대값을 따라감  원래 큰 값을 따라갈 확률이 높음  E-greedy에서 처럼 시간이 지날수록 작은 값을 더하도록 할 수도 있음      Learning Q with discounted reward  앞의 해결책들을 이용하여 아래그림과 같은 결과를 얻을 수 있음  새로운 길을 찾았지만 (3, 3)의 위치에서 두 길의 Qvalue가 모두 1이기 때문에 더 좋은 길을 선택할 수 없다.  이것을 해결하기 위해 discounted reward를 사용한다    로 하고 식을 아래와 같이 바꾼다.  를 사용해 reward를 일찍 받을 수록 더 크도록 한다.    Deterministic vs Stochastic  Deterministic에서는 오른쪽으로 갈 때 오른쪽으로 가고 왼쪽으로 갈 때 왼쪽으로 가는 항상 정해진대로 움직임  Stochastic에서는 오른쪽으로 움직여도 오른쪽으로 가는 것이 정해져 있지 않음  Stochastic에서 이전과 같은 방법을 사용하면 의도하지 않은 방법으로 움직인것도 의도한대로 움직인걸로 생각하고 학습하기 때문에 성공률이 매우 낮아진다.  Stochastic에서의 Q-Learning  Learning rate 를 두고 이전의 식을 아래의 식처럼 변경한다.  새로운 식은 업데이트전 자신의 Q에서 다음의 Q의 영향을 조금만 받은 상태로 업데이트한다.    BWAPI(Brood War Application Programming Interface)  BWAPI는 스타크래프트와 상호작용하는데 사용되는 오픈소스 C++ 프레임 워크이다. BWAPI를 사용하여 스타크래프트내 유닛들의 상태 등 게임내 정보를 가져오거나 유닛들에게 명령을 내릴 수 있다.  BWAPI를 설치하면 ExampleBot의 코드가 있으며 자원수집과 인구수 확보 2가지의 명령을 계속해서 수행하도록 구현이 되어있음  코드를 수정하여 유닛의 번호를 화면에 출력거나 특정유닛을 선택해 특정좌표로 이동시키는 동작들을 할 수 있음을 확인했음  버스, 모니터, 트럭, 앉아있는이(가) 표시된 사진  자동 생성된 설명 | | | | | | | |
| **제작** | **설계 계획** | 뛰어난 게임실력과 높은 학습효율성을 목표로 설계를 함  스타크래프트에서는 고려해야할 것이 아주 많기 때문에 스타크래프트의 여러가지 상황들을 나누고 기계학습으로 높은 성능을 보여줄 수 있는 몇 가지 상황들에 기계학습을 적용할 것이다. 스타크래프트에서 게임에 큰 영향을 미치는 것들로 전투와 빌드 오더 등이 있다. 이 두가지 상황에 기계학습을 적용할 예정이며 전투에서는 전략판단(전투, 후퇴)과 유닛 컨트롤에서 적용할 예정이고 빌드 오더에서는 정찰로 얻은 정보와 자원 상황에 따른 빌드 오더의 결정에 적용할 예정이다.  스타크래프트의 여러가지 상황들을 나누어 처리하기 위해 아래와 같은 8가지 클래스를 만들 예정이다.  StrategyManager  전략판단, 유닛 컨트롤 : 크게 3가지 동작을 함 유닛들을 몇 개의 그룹으로 나누어 관리, 전투에서 이득을 볼지 손해를 볼지를 계산하여 전투를 할지 도망갈지 결정, 전투상황에서의 컨트롤  전투상황의 컨트롤, 유닛 나누기, 전투 손익계산 순으로 딥러닝 기술을 적용해볼 예정  WorkerManager  일꾼관리 : 일꾼의 숫자를 조정하고 위험한 경우 안전한 곳으로 이동시킨다.  ScoutManager  정찰관리 : 정찰유닛을 지정하고 컨트롤 한다.  BuildManager  빌드 오더 큐 관리 : 사용할 전체적인 빌드를 정하고 지어야 할 건물의 순서와 현재까지 지어진 건물을 관리한다.  전체적인 빌드는 몇가지를 고정적으로 사용하고 StrategyManager에서의 딥러닝 적용이 잘되면 이쪽도 시도해볼 예정  InformationManager  데이터관리 : 베이스, 유닛 상태 등의 데이터를 자료구조에 저장한다.  ConstructionManager  건설관리 : 빌드 오더 큐 중 건물 건설에 관련해서 일꾼 선정, 위치선정, 건설 실시, 중단된 건물 빌드 재개를 담당한다.  MapGridManager  유닛 위치 관리 : 유닛 위치 업데이트 담당  TensorflowManager  텐서플로 관리 : 텐서플로를 사용하기 위해 파이썬과의 연결을 담당한다.  처음에는 정해진 명령만을 수행하도록 만들고 이후 tensorflow와 bwapi를 함께 사용하기 위한 연결을 하고 전략판단, 유닛 컨트롤, 빌드 오더순으로 하나씩 기계학습을 적용하여 볼 예정이다. | | | | | | | |
| **시험/평가** | **검증 계획** | 컴퓨터와의 게임에서 승패 여부  이득이 될 만한 전투를 먼저 거는지  무리한 전투를 하지 않는지  전투에 참여한 유닛의 차이를 고려했을 때 전투시 잃은 유닛과 잡은 유닛에 따른 손익  정찰로 얻어낸 정보나 현재의 상황에 따른 빌드 오더의 효율적인 선택 | | | | | | | |
| **일정** | | 설계 내용 및 검증 방법에 언급된 전체 진행 과정의 일 정을 자세히 기술한다.  # 3학년 2학기  총 16주차  1~9주차: 딥러닝 이론 공부(신경망, DNN, RNN, Q-learning)  10~11주차: 스타크래프트 인공지능 관련 조사 수행  12~13주차: 구현 방법 조사 및 기본 클래스 설계 구상  14~16주차: 스타크래프트 인공지능 설계 환경 구축 | | | | | | | |
| **역할 분담** | | 코드설계과정에서는 각자 코드의 일부를 설계하고 구현한 후 합치고 필요한 자료와 라이브러리의 조사는 함께 찾아보고 찾은 자료를 공유하는 방식으로 진행함  김범준  StrategyManager클래스의 설계와 구현  WorkerManager클래스의 설계와 구현  ScoutManager클래스의 설계와 구현  MapGridManager클래스의 설계와 구현  이태형  BuildManager클래스의 설계와 구현  InformationManager클래스의 설계와 구현  ConstructionManager클래스의 설계와 구현  TensorflowManager클래스의 설계와 구현  공동  RNN, DQN등의 알고리즘 공부와 tensorflow를 사용해 직접 예제를 수행해봄  기계학습을 하기위한 라이브러리 tensorflow의 사용법 조사  게임내의 정보를 가져오고 명령을 내리기 위한 bwapi의 사용법 조사  프로젝트설계에 참고할 수 있는 스타크래프트 ai들을 찾아보고 사용해봄  C++코드인 bwapi와 python에서 사용할 수 있는 tensorflow를 함께 사용하기 위한 방법 조사  각자 구현한 코드를 합치고 디버깅과 수정이 필요한 부분 수정 | | | | | | | |