USCViterbi

CSCI 599 Applied Machine Learning for Games

Units: 4-1
Term—Day—Time: Spring – Tuesdays – 10am to 1pm, Lab 1pm to 2pm.

Location: SCI-106

Instructor: Michael Zyda
Office: EGG-209
Office Hours: Tuesdays 1pm

Contact Info: zyda@usc.edu

Teaching Assistant:
Office: EGG-205
Office Hours:
Contact Info:

IT Help: EGG Laboratory Manager
Hours of Service:
Contact Info:

Course Description

This course covers the fundamentals of machine learning applicable to the development and analysis of videogames. The objective of the course is to prepare the student for research and development of machine learning technologies for interacting with and analyzing games.

Learning Objectives

The topics covered include neural networks, convolutional neural networks, long-short term memory, recurrent neural networks, generative adversarial networks, reinforcement learning, Q-learning, deep Q-learning, Markov models, data analysis and AI character/bot creation and understanding using these techniques.

Prerequisite(s): CSCI-561 Foundations of AI or CSCI-567 Machine Learning or equivalent course/experience in artificial intelligence/machine learning or permission of the instructor.

Required Readings and Supplementary Materials

Andrew Glassner, Deep Learning, Vol. 1: From Basics to Practice [Kindle]
Andrew Glassner, Deep Learning, Vol. 2: From Basics to Practice [Kindle]
Embedded readings as directed below.

Description and Assessment of Assignments

Week 1 – Introduction to machine learning for games

Artificial Intelligence (AI) vs. Machine Learning vs. Deep Learning
https://skymind.ai/wiki/ai-vs-machine-learning-vs-deep-learning
Deep learning for games
https://arxiv.org/pdf/1708.07902.pdf
https://www.logikk.com/articles/machine-learning-in-game-development/
https://www.kickstarter.com/projects/513736598/build-bots-to-play-games-machine-learning-ai-with
Project 1: Select a machine learning project your group will work on all semester, including the game to be utilized in that project. Draft the initial Engineering Design Document (EDD) for that project. The EDD will be maintained and updated as the semester progresses.

Week 2 – Neural Networks

A Beginner's Guide to Neural Networks and Deep Learning
https://skymind.ai/wiki/neural-network
Neural Network Tuning
https://skymind.ai/wiki/neural-network-tuning
Case analysis: Flappy Bird
Examine the Flappy Bird AI agent as an example - it uses Neural Networks and Genetic Algorithms to find the best way to get rid of the barriers.
http://www.askforgametask.com/html5/tutorials/flappy/
Source code: https://github.com/ssusnic/Machine-Learning-Flappy-Bird
Project 2: How will you utilize neural networks in your proposed EDD? What will the use of neural networks provide with respect to your chosen project?

Week 3 – Neural Network Tools

Tensorflow - An end-to-end open source machine learning platform
https://www.tensorflow.org/
Keras – an open-source neural network library written in Python that is now integrated with and built on-top of Tensorflow.
https://keras.io

Project 3: Download Tensorflow and Keras and work through the tutorial material so that your team understands how to deploy them on your project.

Week 4 – Convolutional Neural Networks (CNNs)

A Beginner's Guide to Convolutional Neural Networks (CNNs)

https://skymind.ai/wiki/convolutional-network

Understanding convolutional neural networks

https://arxiv.org/pdf/1605.09081.pdf

Project 4: Perform the assigned readings on CNNs and revise the EDD based on potential requirements of that project.

Week 5 – Convolutional Neural Networks (CNNs) for Object Detection

R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms

Understanding object detection algorithms

https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e

Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

https://arxiv.org/pdf/1506.01497.pdf

Yolov3 – real-time object detection with neural networks

https://pjreddie.com/media/files/papers/YOLOv3.pdf

https://pjreddie.com/darknet/yolo/

Darknet: Open Source Neural Networks in C

https://pjreddie.com/darknet/

Case analysis: League of Legends

Using the application of CNN, such as YoloV3, for League of Legends' champion detection and recognition.

https://medium.com/@farzatv/deepleague-leveraging-computer-vision-and-deep-learning-on-the-league-of-legends-mini-map-giving-d275fd17c4e0

Project 5: Download the latest version of Yolo (V3), run the tutorials to understand what can be achieved with respect to object detection.

Week 6 – Long short-term memory (LSTM) & Recurrent Neural Networks (RNNs)

LSTM – is an artificial recurrent neural network (RNN) architecture.

Unlike standard feedforward neural networks, LSTM has feedback connections that make it a "general purpose computer" (that is, it can compute anything that a Turing machine can). It can not only process single data points (such as images), but also entire sequences of data (such as speech or video).

https://en.wikipedia.org/wiki/Long_short-term_memory

A Beginner's Guide to LSTMs and Recurrent Neural Networks

https://skymind.ai/wiki/lstm

Case Analysis: FIFA

Applying an RNN to the FIFA game.

https://towardsdatascience.com/building-a-deep-neural-network-to-play-fifa-18-dce54d45e675

Project 6: Read the FIFA article and download the code from Github. Run that code and understand what it is capable of.

Week 7 – Generative Adversarial Networks

A Beginner's Guide to Generative Adversarial Networks (GANs)

https://skymind.ai/wiki/generative-adversarial-network-gan

Case Analysis: Generating High Quality Anime Characters

https://nips2017creativity.github.io/doc/High_Quality_Anime.pdf

Source code: https://github.com/forcecore/Keras-GAN-Animeface-Character

Project 7: Do the reading on GANs. Download the source code from the case analysis and try it out.

Week 8 – Reinforcement Learning, Q-learning, Deep Q-learning

What is Reinforcement Learning?
Why is it needed in games?
What are its advantages in games?
Why can't we use supervised learning in games?

https://arxiv.org/pdf/1707.01310.pdf

https://papers.nips.cc/paper/1953-reinforcement-learning-with-long-short-term-memory.pdf

A Beginner's Guide to Deep Reinforcement Learning

https://skymind.ai/wiki/deep-reinforcement-learning

Project 8: Read the material on reinforcement learning and determine how to use reinforcement learning in the project described in your EDD.

Week 9 – Reinforcement Learning, Q-learning, Deep Q-learning Part 2

Hidden Markov Models

https://en.wikipedia.org/wiki/Hidden_Markov_model

A Beginner's Guide to Markov Chain Monte Carlo, Machine Learning & Markov Blankets

https://skymind.ai/wiki/markov-chain-monte-carlo

Markov Decision Processes

POMDP, policy iteration, value iteration/Bellman equation, learning from examples, policy generation algorithms.

Reinforcement Learning — Markov Decision Processes

https://medium.com/machine-learning-bites/machine-learning-reinforcement-learning-markov-decision-processes-431762c7515b

Project 9: Read the material on reinforcement learning and Markov models.

Week 10 – Reinforcement Learning, Q-learning, Deep Q-learning Part 3

Reinforcement learning applied to AI character behavior authoring.
Case Analysis: Torc
Take the racing car game Torc as an example, which applies Reinforcement Learning and Q-Learning.

https://yanpanlau.github.io/2016/10/11/Torcs-Keras.html

Case Analysis: Dino Run
Students will have the chance to develop their own agent to play the game Dino Run.

https://blog.paperspace.com/dino-run/

Source Code: https://github.com/Paperspace/DinoRunTutorial

Playing Atari with Deep Reinforcement Learning

https://www.cs.toronto.edu/~vmnih/docs/dqn.pdf

Project 10: Read the case analysis materials above and determine how to use reinforcement learning in the authoring of your AI characters.

Week 11 – Game Data Analysis

Datasets and Machine Learning

https://skymind.ai/wiki/datasets-ml

Data for Deep Learning

https://skymind.ai/wiki/data-for-deep-learning

Data Analysis

https://www.sciencedirect.com/science/article/pii/S1877050917314989

Reinforcement learning in a game of tic-tac-toe

https://github.com/Neeratyoy/TicTacToe

Snake game with neural network

https://github.com/TheAILearner/Snake-Game-with-Deep-learning

How to teach an AI to play games using deep reinforcement learning

https://towardsdatascience.com/how-to-teach-an-ai-to-play-games-deep-reinforcement-learning-28f9b920440a

Project 11: Read the materials cited above and determine their relevance to your team's project.

Week 12 – AI Bot Creation

OpenAI Five (for bot creation)

https://en.wikipedia.org/wiki/OpenAI_Five

https://openai.com/five/

OpenAI Five is the name of a machine learning project and system that performs as a team of video game bots playing against human players in the competitive five-on-five video game *Dota 2*. The system was developed by OpenAI, an American artificial intelligence (AI) research and development company founded with the mission to develop safe AI in a way that benefits humanity. OpenAI Five's first public appearance occurred in 2017, where it was demonstrated in a live one-on-one game against a professional player of the game known as Dendi, who lost to it. The following year, the system had advanced to the point of performing as a full team of five, and began playing against and showing the capability to defeat professional teams.

Project 12: Download Open AI Five and install it on your machine. Work through the tutorials to determine its relevance to your team's project.

Week 13 – Advanced Topics - Highlights from recent papers on machine learning & games

Week 14 – Advanced Topics - Highlights from recent papers on machine learning & games

Week 15 – Final Presentations

Final project demos/presentations, writeups & video demo are due.

Grading Breakdown

Students in this class will utilize the covered ML fundamentals and openly available ML tools, to create bots that play against commercial and/or student-built games or utilize those tools to analyze those same games. Each student team will spend the first weeks of the semester creating an engineering design document (EDD) that describes the proposed project - that EDD will be maintained and updated as the semester progresses. The defined project runs continuously all semester, the code testing the student's knowledge of the in-class lectures and reading material. Final demos will be presented on the last day of class, including live demo, trailer of what has been produced and the EDD enhanced with a results and limitations section as for a publishable technical paper.

The project is 90% of the class grade, with in-class attendance and engagement being the remaining 10%. There will be a midterm presentation of intermediate project results (40% of the grade) as well as a final presentation on the last day of class (40% of the grade). The grading of each project will be on whether the project fulfills the requirements for that project. The grading will be either "fulfills the goals" or "does not fulfill the goals". Students in this class will build and maintain a web site describing their group's work, and a personal web page describing their work in particular. Each project will be accompanied by a live in-class demo. That web site must be maintained weekly to advise the professor on group and individual status. Daily class and lab attendance is required for full participation and for full credit for this course.

Statement on Academic Conduct and Support Systems

Academic Conduct

Plagiarism – presenting someone else's ideas as your own, either verbatim or recast in your own words – is a serious academic offense with serious consequences. Please familiarize yourself with the discussion of plagiarism in *SCampus* in Section 11, *Behavior Violating University Standards* https://scampus.usc.edu/1100-behavior-violating-university-standards-and-appropriate-sanctions. Other forms of academic dishonesty are equally unacceptable. See additional information in *SCampus* and university policies on scientific misconduct, http://policy.usc.edu/scientific-misconduct.

Discrimination, sexual assault, and harassment are not tolerated by the university. You are encouraged to report any incidents to the *Office of Equity and Diversity* http://equity.usc.edu or to the *Department of Public Safety* http://adminopsnet.usc.edu/department/department-public-safety. This is important for the safety of the whole USC community. Another member of the university community – such as a friend, classmate, advisor, or faculty member – can help initiate the report, or can initiate the report on behalf of another person. *The Center for Women and Men* http://www.usc.edu/student-affairs/cwm/ provides 24/7 confidential support, and the sexual assault resource center webpage http://sarc.usc.edu describes reporting options and other resources.

Support Systems

A number of USC's schools provide support for students who need help with scholarly writing. Check with your advisor or program staff to find out more. Students whose primary language is not English should check with the *American Language Institute* http://dornsife.usc.edu/ali, which sponsors courses and workshops specifically for international graduate students. *The Office of Disability Services and Programs* http://sait.usc.edu/academicsupport/centerprograms/dsp/home_index.html provides certification for students with disabilities and helps arrange the relevant accommodations. If an officially declared emergency makes travel to campus infeasible, *USC Emergency Information* *http://emergency.usc.edu* will provide safety and other updates, including ways in which instruction will be continued by means of blackboard, teleconferencing, and other technology.