



SUMMER TRAINING COURSE

CS50's Introduction to Artificial Intelligence with Python



PREPARED BY: NAIF GANADILY STUDENT ID: QA0272
Academic Supervisor: Dr. Ali El-Rashidi

Table of Contents

Introduction

- edX 2
- CS50's Introduction to Artificial Intelligence with Python 2

Lecture 0

- 0.1 Search in AI 3
- 0.2 Project Details..... 4

Lecture 1

- 1.1 Knowledge in AI 5
- 1.2 Project Details..... 6

Lecture 2

- 2.1 Uncertainty in AI 7
- 2.2 Project Details..... 8

Lecture 3

- 3.1 Optimization in AI 9
- 3.2 Project Details..... 10

Lecture 4

- 4.1 Learning in AI 11
- 4.1 Learning in AI (Cont.) 12
- 4.2 Project Details 13

Lecture 5

- 5.1 Neural Networking in AI 14
- 5.2 Project Details..... 15

Lecture 6

- 6.1 Language in AI..... 16
- 6.1 Language in AI (Cont.)..... 17
- 6.2 Project Details 18

Conclusion 19

Introduction

edX is the trusted platform for education and learning. Founded by Harvard and MIT, edX is home to more than 20 million learners, many top-ranked universities in the world and industry-leading companies. As a global nonprofit, edX is transforming traditional education, removing the barriers of cost, location, and access. Fulfilling the demand for people to learn on their own terms, edX is reimagining the possibilities of education, providing the highest-quality, stackable learning experiences including the groundbreaking MicroMasters® programs. Supporting learners at every stage, whether entering the job market, changing fields, seeking a promotion, or exploring new interests, edX delivers courses for curious minds on topics ranging from data and computer science to leadership and communications. edX is where you go to learn.

CS50's Introduction to Artificial Intelligence with Python

This course explores the concepts and algorithms at the foundation of modern artificial intelligence, diving into the ideas that give rise to technologies like game-playing engines, handwriting recognition, and machine translation. Through hands-on projects, students gain exposure to the theory behind graph search algorithms, classification, optimization, reinforcement learning, and other topics in artificial intelligence and machine learning as they incorporate them into their own Python programs. By course's end, students emerge with experience in libraries for machine learning as well as knowledge of artificial intelligence principles that enable them to design intelligent systems of their own.

0.1 Search in AI

Concepts

- **Agent:** entity that perceives its environment and acts upon that environment.
- **State:** a configuration of the agent and its environment.
- **Actions:** choices that can be made in a state.
- **Transition model:** a description of what state results from performing any applicable action in any state.
- **Path cost:** numerical cost associated with a given path.
- **Evaluation function:** function that estimates the expected utility of the game from a given state.

Algorithms

- **DFS** (depth first search): search algorithm that always expands the deepest node in the frontier.
- **BFS** (breadth first search): search algorithm that always expands the shallowest node in the frontier.
- **Greedy best-first search:** search algorithm that expands the node that is closest to the goal, as estimated by an heuristic function $h(n)$.
- **A* search:** search algorithm that expands node with lowest value of the "cost to reach node" plus the "estimated goal cost".
- **Minimax:** adversarial search algorithm.

Lecture 0

0.2 Project Details

Description: -

My Projects during this part of the course (Mega Download Folder) in the links below

Requirements: -

Python Editor

-I used PyCharm (Community Edition)

-pygame modulator

-nltk modulator

Visual Studio is also recommended

DON'T USE THONNY !!!

Project 0a: Degrees:-

<https://mega.nz/folder/oEAihS5D#CGEro3aLmll9pMVpucjcbA>

Project 0b: Tic-Tac-Toe:-

<https://mega.nz/folder/REQ2mAYL#XQf0R7vUPOlgsqKW1qY-A>

Lecture 1

1.1 Knowledge in AI

Concepts

- **Sentence:** an assertion about the world in a knowledge representation language.
- **Knowledge base:** a set of sentences known by a knowledge-based agent.
- **Entailment:** a entails b if in every model in which sentence a is true, sentence b is also true.
- **Inference:** the process of deriving new sentences from old ones.
- **Conjunctive normal form:** logical sentence that is a conjunction of clauses.
- **First order logic:** Propositional logic.
- **Second order logic:** Proposition logic with universal and existential quantification.

Algorithms

- **Model checking:** enumerate all possible models and see if a proposition is true in every one of them.
- **Conversion to CNF** and **Inference by resolution**

Lecture 1

1.2 Project Details

Description: -

My Projects during this part of the course (Mega Download Folder) in the links below

Requirements: -

Python Editor

-I used PyCharm (Community Edition)

-pygame modulator

-nltk modulator

Visual Studio is also recommended

DON'T USE THONNY !!!

Project 1a: Knights:-

<https://mega.nz/folder/gRBA2a4R#sgvuyJPI6UYxPPPIcPuuJw>

Project 1b: Minesweeper:-

<https://mega.nz/folder/1VZ2CIKC#Pvw7WDcPQ4DN3GSdUxLh3w>

Lecture 2

2.1 Uncertainty in AI

Concepts

- **Unconditional probability:** degree of belief in a proposition in the absence of any other evidence.
- **Conditional probability:** degree of belief in a proposition given some evidence that has already been revealed.
- **Random variable:** a variable in probability theory with a domain of possible values it can take on.
- **Independence:** the knowledge that one event occurs does not affect the probability of the other event.
- **Bayes' Rule:** $P(a) P(b|a) = P(b) P(a|b)$
- **Bayesian network:** data structure that represents the dependencies among random variables.
- **Markov assumption:** the assumption that the current state depends on only a finite fixed number of previous states.
- **Markov chain:** a sequence of random variables where the distribution of each variable follows the Markov assumption.
- **Hidden Markov Model:** a Markov model for a system with hidden states that generate some observed event.

Algorithms

- **Inference by enumeration**
- **Sampling**
- **Likelihood weighting**

Lecture 2

2.2 Project Details

Description: -

My Projects during this part of the course (Mega Download Folder) in the links below

Requirements: -

Python Editor

-I used PyCharm (Community Edition)

-pygame modulator

-nltk modulator

Visual Studio is also recommended

DON'T USE THONNY !!!

Project 2a: PageRank:-

https://mega.nz/folder/hNJA1QiK#cVnm5lDwUJp_dykGMidZgQ

Project 2b: Heredity:-

<https://mega.nz/folder/AJQwwlSZ#uMTDA5pzj57cMksoUJT8sA>

Lecture 3

3.1 Optimization in AI

Concepts

- **Optimization:** choosing the best option from a set of options.

Algorithms

- **Local Search Hill climbing**
 - **steepest-ascent:** choose the highest-valued neighbor.
 - **stochastic:** choose randomly from higher-valued neighbors.
 - **first-choice:** choose the first higher-valued neighbor.
 - **random-restart:** conduct hill climbing multiple times.
 - **local beam search:** chooses the k highest-valued neighbors.
- **Simulated annealing:** early on, more likely to accept worse-valued neighbors than the current state.
- **Linear programming**
 - **Simplex**
 - **Interior-Point**
- **Constraint satisfaction problems**
 - **Arc consistency:** to make X arc-consistent with respect to Y , removing elements from X 's domain until every choice for X has a possible choice for Y
 - **Backtracking search**

Lecture 3

3.2 Project Details

Description: -

My Projects during this part of the course (Mega Download Folder) in the links below

Requirements: -

Python Editor

-I used PyCharm (Community Edition)

-pygame modulator

-nltk modulator

Visual Studio is also recommended

DON'T USE THONNY !!!

Project 3: Crossword:-

https://mega.nz/folder/kQZ2EKLb#2v7S6h3Llm_qK7BQc9UE8g

Lecture 4

4.1 Learning in AI

Concepts

- **Supervised learning:** given a data set of input-output pairs, learn a function to map inputs to outputs.
 - **Classification:** supervised learning task of learning a function mapping an input point to a discrete category.
 - **Regression:** supervised learning task of learning a function mapping an input point to a continuous value.
 - **Loss function:** function that express how poorly our hypothesis performs (L1, L2).
 - **Overfitting:** when a model fits too closely to a particular data set and therefore may fail to generalize to future data.
 - **Regularization:** penalizing hypotheses that are more complex to favor simpler, more general hypotheses.
 - **Holdout cross-validation:** splitting data into a training set and a test set, such that learning happens on the training set and is evaluated on the test set.
 - **k-fold cross-validation:** splitting data into k sets, and experimenting k times, using each set as a test set once, and using remaining data as training set.
- **Reinforcement learning:** given a set of rewards or punishments, learn what actions to take in the future.
- **Unsupervised learning:** given input data without any additional feedback, learn patterns.
- **Clustering:** organizing a set of objects into groups in such a way that similar objects tend to be in the same group.

4.1 Learning in AI (Cont.)

Algorithms

- **k-nearest-neighbor classification:** given an input, chooses the most common class out of the k nearest data points to that input.
- **Support Vector Machines (SVM)**
- **Markov decision process:** model for decision-making, representing states, actions and their rewards.
- **Q-learning:** method for learning a function $Q(s, a)$, estimate of the value of performing action a in state s .
- **Greedy decision-making**
- **epsilon-greedy**
- **k-means clustering:** clustering data based on repeatedly assigning points to clusters and updating those clusters' centers.

Lecture 4

4.2 Project Details

Description: -

My Projects during this part of the course (Mega Download Folder) in the links below

Requirements: -

Python Editor

-I used PyCharm (Community Edition)

-pygame modulator

-nltk modulator

Visual Studio is also recommended

DON'T USE THONNY !!!

Project 4a: Shopping:-

<https://mega.nz/folder/tYAURKBA#ZxBNvv1dY1FHcAcbbJSXMA>

Project 4b: Nim:-

<https://mega.nz/folder/5BAEDQYL#h3dE1XTEjabYjEwBtv3JgA>

Lecture 5

5.1 Neural Networks in AI

Concepts

- **Artificial neural network:** mathematical model for learning inspired by biological neural networks.
- **Multilayer neural network:** artificial neural network with an input layer, an output layer, and at least one hidden layer.
- **Deep neural network:** neural network with multiple hidden layer.
- **Dropout:** temporarily removing units - selected at random - from a neural network to prevent over-reliance on certain units.
- **Image convolution:** applying a filter that adds each pixel value of an image to its neighbors, weighted according to a kernel matrix.
- **Pooling:** reducing the size of an input by sampling from regions in the input.
- **Convolutional neural network:** neural networks that use convolution, usually for analyzing images.
- **Recurrent neural network:** neural network that generates output that feeds back into its own inputs.

Algorithms

- **Gradient descent:** algorithm for minimizing loss when training neural network.
- **Backpropagation:** algorithm for training neural networks with hidden layers.

Lecture 5

5.2 Project Details

Description: -

My Projects during this part of the course (Mega Download Folder) in the links below

Requirements: -

Python Editor

-I used PyCharm (Community Edition)

-pygame modulator

-nltk modulator

Visual Studio is also recommended

DON'T USE THONNY !!!

Project 5: Traffic:-

https://mega.nz/folder/0FQkTCil#zxThl5uUy_6Fc-yIAMoxdA

6.1 Language in AI

Concepts

- **Natural language processing**
- **n-gram**: a continuous sequence of n items inside of a text.
- **Tokenization**: the task of splitting a sequence of characters into pieces (tokens).
- **Text Categorization**
 - **Bag-of-words model**: represent text as an unordered collection of words.
- **Information retrieval**: the task of finding relevant documents in response to a user query.
 - **Topic modeling**: models for discovering the topics for a set of documents.
 - **Term frequency**: number of times a term appears in a document.
 - **Function words**: words that have little meaning on their own, but are used to grammatically connect other words.
 - **Content words**: words that carry meaning independently.
 - **Inverse document frequency**: measure of how common or rare a word is across documents.
- **Information extraction**: the task of extracting knowledge from documents.
- **WordNet**: a lexical database of semantic relations between words.
- **Word representation**: looking for a way to represent the meaning of a word for further processing.
 - **one-hot**: representation of meaning as a vector with a single 1, and with other values as 0.
 - **distribution**: representation of meaning distributed across multiple values.

6.1 Language in AI (Cont.)

Algorithms

- **Markov model applied to language:** generating the next word based on the previous words and a probability.
- **Naive Bayes:** based on the Bayes' Rule to calculate probability of a text being in a certain category, given it contains specific words. Assuming every word is independent of each other.
 - **Additive smoothing:** adding a value α to each value in our distribution to smooth the data.
 - **Laplace smoothing:** adding 1 to each value in our distribution (pretending we've seen each value one more time than we actually have).
- **tf-idf:** ranking of what words are important in a document by multiplying term frequency (TF) by inverse document frequency (IDF).
- **Automated template generation:** giving AI some terms and let it look into a corpus for patterns where those terms show up together. Then it can use those templates to extract new knowledge from the corpus.
- **word2vec:** model for generating word vectors.
- **skip-gram architecture:** neural network architecture for predicting context words given a target word.

Lecture 6

6.2 Project Details

Description: -

My Projects during this part of the course (Mega Download Folder) in the links below

Requirements: -

Python Editor

-I used PyCharm (Community Edition)

-pygame modulator

-nltk modulator

Visual Studio is also recommended

DON'T USE THONNY !!!

Project 6a: Parser:-

https://mega.nz/folder/1NBglKyZ#WvbFI_gAL1p67T_gqES0VQ

Project 6b: Questions:-

https://mega.nz/folder/INQ0IIRd#6_bBa6rIT6JqxdAZIXOfbw

Conclusion

Harvard CS50's Introduction to Artificial Intelligence online course has opened my eyes to new possibilities of what's to come. I have only learned a fraction what AI can do with what limitations we have given it in this introductory course. The course has covered the basics of each application AI is used for in our everyday lives. AI is rapidly growing, with 'Big Data' on the rise, thus have made the perfect environment for new AI applications and services to grow. AI is for now controlled by the user and Big Data, but with Neural Networking it might not need a "User" anymore.

Special Thanks to edX who have given me the opportunity to learn from one of the best university's in the world, also thanks to CS50 for the outstanding up to date knowledge that has provided during the course. An honor it is to be supervised under Dr. Ali Elrashidi and thanks to everyone in the University of Business and Technology for the love and support they have given me over the years.

AI doesn't have to be evil to destroy humanity – if **AI** has a goal and humanity just happens to come in the way, it will destroy humanity as a matter of course without even thinking about it, no hard feelings.” **Elon Musk, Technology Entrepreneur, and Investor.**