

Syllabus

1. Introduction to Artificial Intelligence (A.I.)

- 1.1. Origins of A.I.
- 1.2. A.I. and Other Disciplines
- 1.3. Agents (Reactive)
- 1.4. Environment Traits

2. Search Algorithms

- 2.1. Solution Space
- 2.2. Uninformed Search Algorithms
- 2.3. Informed Search Algorithms
- 2.4. Constraint Satisfaction

3. Uncertainty

- 3.1. Uncertainty
- 3.2. Bayes Networks

4. Machine Learning

- 4.1. Difference between supervised and unsupervised learning
- 4.2. Decision Trees (supervised Classifiers)
- 4.3. Similarity Metrics and Clustering (unsupervised learning)
- 4.4. Artificial Neural Networks (supervised Classifiers)

Class Schedule

See calendar in Schoology.

Activities

1. Class: Getting to know the Origins of A.I.

Goal: understand the origins and the principles of A.I. **(3 hours)**

Home reading

Russell & Norvig Introduction to Artificial Intelligence a Modern Approach, 3rd Edition

- Introduction page 1
- What is AI? page 2 - 4

- Foundations page 5 - 16
- State of the Art page 28 – 30

Watch the following video, which discusses the question: “can an AI ever be considered a Person”? <https://www.youtube.com/watch?v=39EdqUbj92U>

Include the following in your Research notes:

1. Write half page discussing how A.I. relates to other disciplines mentioned in the chapter.
2. Make a diagram of the most important theories.
3. What is artificial general intelligence (or Strong AI)? Do you think this is achievable? (Watch the video above for the necessary background).

2. Class: Analyse the relationship between A.I. and Other Disciplines

Goal: understand how A.I. techniques are used in or relate to other areas (**3 hours**)

For this activity, you have to research the following questions by yourself. You can use any (re)source you like.

Include the following in your Research notes:

1. Describe in your own words what is intelligence and how can you recognize it.
2. Search for competitions related to A.I. e.g., DARPA challenges or ROBOCUP. Explain briefly what they are and how they relate to A.I. (What techniques they use? Do they show intelligence? What is their real purpose?)
3. Search for a video of IBM Watson and analyse what it can do. Could you do this with regular programming – yes or no? Could you consider Watson intelligent?
4. Search for 2 everyday apps that you use that have A.I. components. Describe how they work. Are they intelligent?
5. Complement your original description 1), with new or more specific elements you find out after researching points 2, 3 and 4.
6. What is the relationship between Data Science and Artificial Intelligence?

I would recommend watching this documentary if you want more details about Watson:
<https://www.youtube.com/watch?v=4svcCJJ6ciw>

3. Lab: Using Robots for AI

Goal: understand how to program the Lego robots with the LEJOS software. Future labs will use this software (4 hours)

MAX 4 students per team. Minimum of 2. Ruben deserves the right to merge teams if there are not enough robots.

3.1 Preparation for the lab. **Do this *before* class: this counts for 10% of your grade.**

Goal of preparation:

1. Choose between the Parallax or LEJOS robot
2. You have to have the development environment installed.
3. You need to show a simple program that makes the robot move in a straight line.

Robot Types. CHOOSE ONE AND ONLY 1

Robot Type 1: Parallax

Important change for 2018: you are allowed to use the Parallax Robots instead of the LEJOS ones. Ask Alejandra Rosado (Directora de Carrera of Mechatronics) for one. This is experimental. I won't be able to help you if it doesn't work. However, you will receive my eternal gratitude if you can write instructions for setting it up and using it for this lab.

Robot Type 2: LEJOS

You can get the LEGO robots from the lab warehouse on level 2 in building 2.

*Note: the installation of the leJOS development environment can be a little tricky. Follow the instructions below **to the letter, and don't skip any steps**. Make sure you follow the instructions for the MINDSTORMS NXT platform.*

TIPS & TRICKS

1. If your Mindstorm robot makes "clicking" sounds, you have to install the LeJoS firmware again.
Run `sudo ./nxjflash` from the /bin directory of your LeJoS installation
2. For more tips and tricks for Windows, see
https://docs.google.com/document/d/1mHTmvV73DWrAEC1mHfZ7GbLJvgWlau7EAe7o0hPs6_U/edit?usp=sharing

Installation of drivers and communication with LEJOS

1. Follow the instructions for your platform:
<http://www.lejos.org/nxt/nxj/tutorial/Preliminaries/GettingStarted.htm>
2. The Fantom Driver can be downloaded here:
<https://lc-www-live-s.legocdn.com/r/www/r/mindstorms/-/media/franchises/mindstorms%202014/downloads/firmware%20and%20software/nxt%20software/nxt%20fantom%20drivers%20v120.zip?l.r2=-964392510> (if this link is dead, see <https://www.lego.com/en-us/mindstorms/downloads>)

Testing your installation

1. Connect the robot via USB and turn it on
2. Open the command line and execute “nxjbrowse”. Ensure the robot shows up in the list
3. Follow these instructions to the letter to run a HelloWorld program:
<http://www.lejos.org/nxt/nxj/tutorial/Preliminaries/FirstProgram.htm>

Configure IDE:

1. Download Eclipse from <http://www.eclipse.org>. Use "Eclipse for Java Developers"
2. Install LEJOS plugin and configure it:
<http://www.lejos.org/nxt/nxj/tutorial/Preliminaries/UsingEclipse.htm>
3. Follow each step until the Importing Sample Projects section.

Note: if you're getting compile errors in Eclipse, make sure you've created a LEJOS project, and not a plain Java project. New -> Project -> Lejos -> Lejos NXJ project.

For a full tutorial see: <http://www.lejos.org/nxt/nxj/tutorial/index.htm>

3.2 Lab

Each team must program a behaviour for the robot. The behaviour must include a movement, the drawing of a pattern, and the use of a sensor to change the behaviour.

Program:

1. Each team must program a different behaviour to other teams choose a specific pattern for movement, for instance:
 - a. Square, circle, hexagon, spiral, infinite loop, or any pattern from your own design. (The objective is to learn to create complex movements which you will use in future labs.)
2. Use a sensor to modify the behaviour of your robot, e.g. stop, turn, change pattern, etc...
When the sensor is triggered, the first behaviour must be interrupted immediately, and new behaviour must start.

3.3 Report

Explain not what you did, but the ideas behind it, i.e. the reasoning behind your decisions. Use the following questions to guide your deliberations:

1. What problems did you come across during the lab?
2. What modification would be required to turn your behaviour into a solution for any real problem? i.e. Where could it be used and how?

3.4 Evaluation

See rubric on Schoology for the Lab hand-in

4. Class: Understanding Agents and Environments

Goal: understand the concept of Agent and Environment from the perspective of A.I. **(3 hours)**

Home reading

Russell & Norvig Introduction to Artificial Intelligence a Modern Approach, 3rd Edition

- 2.1 Agents and Environments page 34-36
- 2.2 Good Behaviour page 37
- 2.2.1 Rationality page 38
- 2.2.2 Omniscience page 39
- 2.3 The Nature of Environments page 40
- 2.3.1 Specifying Task Environment page 40
- 2.3.2 Properties of Task Environments page 41 - 46

Include the following in your Research notes:

Answer the following questions:

1. How can we differentiate an agent from any other piece of software? What are the special qualities that make it an agent?

2. Provide 3 examples of **task environments**. Describe them accordingly to the properties of environments.
3. Are there any parts that are not fundamental for agents? i.e. parts you could take away and have the program remain an agent.
4. Google the concept of a computational reactive agent, then look in formal sources and write half a page (200 – 250 words) describing what a reactive agent is, and what other types of agents exist. Include your references.

5. Lab: Programming a Reactive Agent

Goal: understand the implications and difficulties of agent design and agent programming (4 hours)

Lab

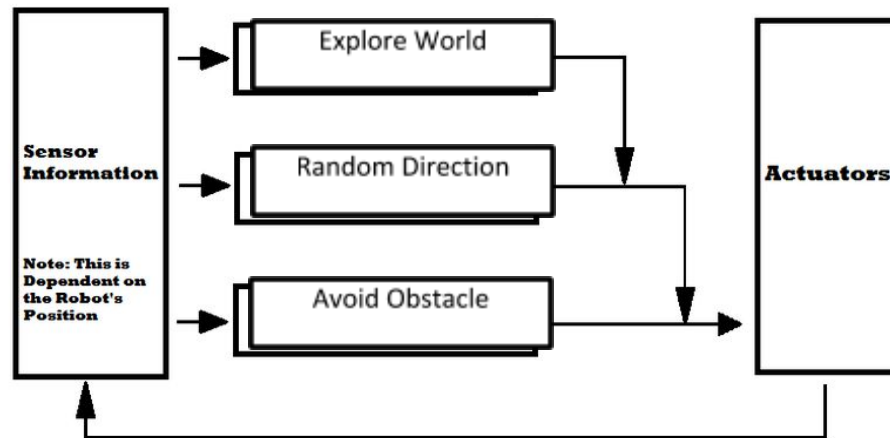
Each team will create an architecture for a simple agent and will program its behaviours in a robot.

Preparation for the lab. Do this *before* class: this counts for 10% of your grade.

1. Read slides 24-33 and 37-45 in the presentation on reactive agents. In Schoology, see Labs -> Lab 2 -> “Session 2 - Agent Architecture.ppt”
2. Run the Mars Explorer demo: In Schoology, see Supporting Materials -> Agents and Environments -> MarsExplorer.jar
3. Decide which type of behavior you want to implement. See instructions below. You have to come up with at least 3 levels (sub-behaviors) that make up your agent’s subsumption architecture. I will ask you in class. (I might also ask you a number of questions about the subsumption architecture in general. You should understand how it works.)

Basic Architecture

- 1) Implement a reactive agent that follows these rules:



2) Modify the architecture: you can include other behaviours or change all of them.

Report

1. Describe the behaviours you changed and the reasons why you chose it in 100-200 words.
2. Include your agent final architecture. Use the concept of "Subsumption Architecture" (see next item) to explain your architecture. Explain behaviors, the hierarchy and the conditions under which the behaviors activate.
3. Based on what you saw in this lab, what are the advantages and disadvantages of reactive agents? Can they achieve complex tasks? (Explain your answer in 100 - 200 words).

Hint: look at the slide deck accompanying this class for some theoretical background:

<https://app.schoolology.com/course/1147623146/materials/gp/1147623294> (From slide 30)

Additional Information: <https://www.udacity.com/course/cs373>

Evaluation

Evaluation will be based on:

1. Implementation of the basic agent
2. The agent architecture
3. The report

Idea for a challenge (this is not part of the lab)

Create an agent that will perform a **useful task** and embed it in physical hardware (Lego, Quadcopter, AIBO) you can modify the Lego robot in any way you decide but you must put it back the way it was at the end and check it work properly. Some examples are: exit a complex maze, interact with other agents, move objects, play a game, search for something, rescue operations, etc... Each application must be different.

Create a software agent that performs a **useful task** (web crawling, file organization, assistant, companion, information searcher, intelligent agent in a game), you must prove and justify the agent's autonomy.

6. Class: Describing Environments and Search Spaces

Goal: understand the nature of problems, how they can be classified and what kind of approaches one should take to solve them (**2 hours**)

Home Reading

Russell & Norvig Introduction to Artificial Intelligence a Modern Approach, 3rd Edition

- 3. Intro page 64-66
- 3.1.1 Well-defined Problems and Solutions page 66-68
- 3.2 Example Problems page 69
- 3.2.1 Toy Problems page 70-73
- 3.2.2 Real World Problems page 73 -75
- 3.3 Searching for Solutions page 75 -78

Include the following in your Research notes:

1. Represent the problem of finding the exit of a maze as search space. Describe all 5 components.
2. Represent a search space (for search algorithms) from your favourite video game/novel/comic/sport, etc... Remember that the search space represents possible states, but it is different from a state machine or an automata.

In both cases, clearly define your search space using **all five required properties**.

7. Class: Getting to know Search Algorithms

Goal: Learn different types of search algorithms which will allow you to search in search spaces (**3 hours**)

Home Reading

Russell & Norvig Introduction to Artificial Intelligence a Modern Approach, 3rd Edition

- 3.3 Searching for Solutions page 75-78
- 3.3.1 Infrastructure for Search Algorithms page 78-80
- 3.3.2 Measuring Problem Solving Performance page 80-81
- 3.4 Uninformed Search Strategies page 81
- 3.4.1 Breadth First Search page 81-83
- 3.4.2 Uniform-Cost Search page 83 -85
- 3.4.3 Depth First Search page 85-87
- 3.4.4 Depth Limited Search page 87-88
- 3.4.5 Iterative Deepening page 88 -90

Some animations of the algorithms discussed in the book:

1. http://will.thimbleby.net/algorithms/doku.php?id=breadth-first_search
2. http://will.thimbleby.net/algorithms/doku.php?id=depth-first_search
3. http://will.thimbleby.net/algorithms/doku.php?id=iterative_deepening_depth-first_search

Include the following in your Research notes:

Go through the search space you represented in your last homework assignment using at least 3 types of uninformed searches (correct or finish previous homework if necessary). Write down the complete path made by each search. Compare the searches you choose by using the properties for search strategies.

8. Class: Using Informed Search Algorithms and Heuristics

Goal: Learn how informed strategies are used (3 hours)

Home Reading

Russell & Norvig Introduction to Artificial Intelligence a Modern Approach, 3rd Edition

- 3.5 Informed Search Strategies page 92
- 3.5.1 Greedy Best-first Search page 92-93
- 3.5.2 A*Search (until "Optimality of A*") page 93-95
- 3.6 Heuristic Functions page 102 -103
- 3.6.1 The Effect of Heuristic Accuracy page 103 -104

Include the following in your Research notes:

1. Assign weight to the arcs in the graph you have modelled in previous homework assignments. Use a Greedy search and a A* search to find the goal in the graph. Since A* is an informed search algorithm, you also have to come up with your own heuristic. Compare the paths between them with those done in previous homework assignments.
2. Explain what a heuristic is
3. Search for 2 examples of heuristics for **search problems**. (The term heuristic is used for other things as well)

Suggestions for Challenge:

Create a **useful** program of your own choosing that uses an informed search algorithm. Examples include a gps path finder, model a problem where the solutions are in a search space and take decisions there using searches.

Nice examples <http://movingai.com/astar-var.html>

Madisson U. example <http://pages.cs.wisc.edu/~dyer/cs540/notes/search2.html>

9. Lab: Implementing (Un)informed Search Algorithms

Goal: Understand the use of various search algorithms and the effort involved in implementing them. (7 hours)

Lab

In this lab, we will make use of your programming skills to create practical implementations of several (un)informed search algorithms.

Size of teams: each team must be made up of at most 2 students.

Preparation (*Note: Do this before class. This counts for 10% of your grade)

1. Have an account on Alphagrader, and submit a correct solution to the example Calculator assignment: <https://www.alphagrader.com/courses/17/assignments/163>
2. Read the instructions for the report and program here: <https://www.alphagrader.com/courses/17/assignments/162>
3. Write a parser for the input, and submit this to AlphaGrader (this will fail, but that's OK)

Talk to me first if you plan to use a programming language that is not in this list: Python, Java, C, C++, Ruby.

Evaluation

Evaluation will be based on:

1. Problem Representation
2. Search algorithms. Your algorithms will be automatically tested using a large set of examples. Your grade depends on the number of examples on which your program gives correct output. Make sure you test them carefully! Some tests will be provided.
3. Report: analysis, presentation, conciseness.

Suggestions for Challenge:

Create a **useful** program of your own choosing that uses a search algorithm. Model a problem where the solutions are in a search space and take decisions there using searches.

10. Class: Familiarizing with General Concepts of Uncertainty

Note: this refresher class contains a lot of material. Preparing for this class will likely take you a lot of time. Take enough time for this.

Goal: Review necessary of probability principles to understand probability in A.I. (3-10 hours)

Study at Home

1. **Reading:** Read chapter 1 of “Korb & Nicholson (2010) Bayesian artificial intelligence, Boca Raton”, Highlighted parts are mandatory, the rest is optional. **If you do not understand the highlighted parts, then you should read the whole chapter and write down your doubts for class.**
See Schoology for the materials: **Supporting Materials -> Korb and Nicholson - Bayesian Artificial Intelligence**
2. A really good reminder on how the total probability theorem works in a real life example: <https://www.youtube.com/watch?v=d4WZHz1arG8>
3. **Watch video course: (before starting the video, read the assignments below)**
 - a. Create an account in UDACITY to access videos and record your progress, <https://www.udacity.com/>
 - b. Thurn, Sebastian (2010), Introduction to Artificial Intelligence: probability in A.I., Udacity course, Udacity.com class ID cs271

<https://classroom.udacity.com/courses/cs271/lessons/48624746/concepts/486095600923#>

- c. **From video 1 “Introduction” to video 18 “Quiz Exercise: Bayes Network” (18 videos in total)**

Include the following in your Research notes:

Answer the questions presented during the video. You are not required to answer every question correctly, however, pay attention to the explanations of the presented solutions. In your notebook, explain the following concepts:

- Conditional probability(formula)
- Total probability (formula)
- Chain Rule (formula)
- Prior
- Posterior
- Conditional Dependence
- Conditional Independence
- Distribution of Probability / Probability distribution
- Bayes Theorem
- Bayesian Network

11. Class: Modelling Bayesian Networks and Probabilistic Inference

Goal: understand how Bayes networks behave, how to use them and the basic principle behind these networks **(3 hours)**

Study at Home

* **Note:** before starting the videos, read the assignments below

1. **Reading:** Read chapter 2 of “Korb & Nicholson (2010) Bayesian artificial intelligence, Boca Raton”, Highlighted parts are mandatory, the rest is optional. **If you do not understand the highlighted parts, then you should read the whole chapter and write down your doubts for class.**
2. **Video course**
 - a. Thurn, Sebastian (2010), Introduction to Artificial Intelligence: probability in A.I., Udacity course, Udacity.com class ID cs271

<https://classroom.udacity.com/courses/cs271/lessons/48624746/concepts/479785610923>

- b. **Video 18 “Bayes Network” to final video “congratulations”**
3. **Video course (before starting the video, read the assignments below)**
 - a. Norvig, Peter (2010), Introduction to Artificial Intelligence: probabilistic inference, Udacity course, Udacity.com class ID cs271
<https://classroom.udacity.com/courses/cs271/lessons/48743138/concepts/483698960923>
 - b. Watch **video 1 and video 2 only**
4. Study a fully worked out example for applying the *Enumeration Algorithm* to calculate the result of querying a Bayes Network:

https://docs.google.com/document/d/15avLWz7875PnahtVDN_VUaAo9V-floK1ssWGR67bBJQ/edit

Include the following in your Research notes:

Answer the questions of the videos. In your notebook, explain in your own words the following concepts and how/where to use them:

- Bayesian Network
- Types of nodes
- Network Complexity
- Probabilistic inference
- Conditional tables
- Types of reasoning

12. Lab: Implementing Bayesian Networks

Goal: understand how to use and make a simple Bayesian network (7 hours)

***Note:** Be sure you understand the concepts viewed in previous sessions. If you are still uncertain, consult: Korb & Nicholson (2010) Bayesian artificial intelligence, Boca Raton, Fla.

Size of teams: each team must be made up of at most 2 students.

Preparation before class: parse input and build representation of the network. Do this before class. This counts for 10% of your grade

See <https://www.alphagrader.com/courses/17/assignments/164>

Submitting your project

Submit your project through Alphagrader:

<https://www.alphagrader.com/courses/17/assignments/164>

Evaluation

Based on example network complexity, programs functionality and correctness, report

Suggestions for Challenge:

Create a predictor that uses a Bayesian network, the predictor should include an interface, be easy to use and show in some part the prediction values. Examples: prediction for a league of legends match, predicting a date outcome, detection of cancer, or an agent that decides to do something based on the bayes network, a game playing agent, etc...

13. Class: Getting to Know Machine Learning

Goal: understand the basic ideas behind machine learning (2 hours)

Study at Home

Watch Video from "1 introduction" until and including "6 Occams Razor" of Thurn, Sebastian (2010), Introduction to Artificial Intelligence: Machine Learning, Udacity course, Udacity.com class ID cs271

<https://classroom.udacity.com/courses/cs271/lessons/48703335/concepts/486836550923#>

Include the following in your Research notes:

Answer the questions presented during the video. You are not required to answer every question correctly, however, pay attention to the explanations of the presented solutions. In your notebook, explain the following concepts:

- Occam's Razor
- Unsupervised learning
- Supervised learning
- Training
- Error
- Research the concept of "overfitting":
 - What is it?
 - What is it caused by?

- o and why is it bad?

***Note for challenges:** data sets training and mining, can be found at the machine learning repository: <http://archive.ics.uci.edu/ml/datasets.html>

14. Class: Modelling Decision trees ID3, C45, J48

Goal: learn how to use basic gain algorithms, understand the concept of entropy and how to use it for classification purposes (2-3 hours)

Study at home

Here are a number of references for understanding decision trees in general, and ID3 in particular:

1. https://www.youtube.com/watch?v=eKD5gxPpEY0&list=PLBv09BD7ez_4temBw7vLA19p3tdQH6FY0 I especially recommend the first 4
2. Udacity course on ID3:
<https://classroom.udacity.com/courses/ud262/lessons/313488098/concepts/3140257610923#>
 - a. Watch videos 6 to 14 (“Example 1: dating” to “Decision Trees Expressiveness XOR”)
 - b. Watch video 18: “ID3”
3. In this link you will find an example of how ID3 algorithm works:
<http://www.cise.ufl.edu/~ddd/cap6635/Fall-97/Short-papers/2.htm>
4. Another decision tree tutorial:
<https://web.archive.org/web/20100212233007/http://www.decisiontrees.net/node/16>

Research

Answer the following questions in your notes:

1. What is entropy is and how do you calculate it?
2. What is information gain?
3. Why is information entropy relevant to computer science in general?
4. What are decision trees such as ID3, C4.5 and J48? What are they used for?
5. Explain how ID3 works in your own words.
6. Give a small example of a decision tree using an example in your daily life

15. Lab: Using Data Mining Tools WEKA

Goal: learn how to use data mining tools to rapidly test and make experiments with several machine learning algorithms (6 hours)

Size of teams: each team must be made up of at most 2 students.

See: <https://www.alphagrader.com/>. Go to Assignment “Decision Trees”

Suggestions for Challenge:

Create a useful program where you make your own implementation of an ID3 algorithm to solve a classification problem. The program must be able to classify provided examples. Some ideas include decision trees that decide if: a game is good match for you or not, if a person is viable for a loan, if some is more likely to vote for a political party, if someone is likely to fail a course, etc...

16. Similarity Metrics and Clustering

Goal: understand the criteria used to determine similarity and processes through which they are grouped (clustered) (3 hours)

Video course

1. Charles Isbell, Michale Littman (2015), Machine Learning: Unsupervised Learning, Udacity course, Udacity.com class ID cs271
<https://classroom.udacity.com/courses/ud741/lessons/644878538/concepts/6381886310923#>
2. Watch video 1 “unsupervised learning” to video 6 “Issues with SLC”

Include the following in your Research notes:

1. What is clustering used for?
2. To what previous content in the course can you relate clustering and why?
3. Search in a formal source (book, research article, or a certified online course) the concept of similarity in Artificial intelligence.

17. Lab: Programming Similarity Metrics and Clustering

Goal: learn how to use cluster algorithms (4 hours)

Lab

1. Create your will domain of instances in csv file, you will measure similarity among them.
2. Use Weka to analyse the data of your dataset and at least 2 other datasets.
3. Use the K-means algorithm and make several runs with different amount of clusters (K values). Choose the runs where the clusters show more interesting patterns.
4. Program either K-means or a SLC algorithm and run it for one of the same data set you created.

Report

What data sets did you choose, and why? Describe the dataset. Which are the interesting patterns you found in the clusters? Include the clusters reported by Weka. Comparing your implementation with that of weka, are the clusters the same? Do they have same members or centroids? Explain the similarities or the differences (300 - 400 words).

Evaluation

Based on: Data Analysis weka

Note: Helpful tutorial for Weka and K-means: <http://sentimentmining.net/weka/> check video 8 “clustering pt. 1”

Suggestions for Challenge:

Create a useful program where you make your own implementation of a similarity metric or a clustering algorithm find similar instances. Some ideas include, a case based reasoning system, a data analysis on a complex set of real data to determine a pattern hidden from the human eye, a classification program using clusters.

18. Modelling Neural Networks

Goal: understand what is an Artificial Neural Network and its uses (2 hours)

Study at home

1. **Video-** brief preview of NN. Suarez, José, (N.A.) Intro to Neural Networks, available at <http://www.youtube.com/watch?v=DG5-UyRBOD4> last visited 11 July 2014
2. **Reading-** Russell & Norvig Introduction to Artificial Intelligence a Modern Approach, 3rd Edition
 - 18.7 Artificial Neural Networks page 727 - 728
 - 18.7.1 Neural Network Structures page 728 - 729
 - 18.7.2 Single-Layer Feedforward page 729 - 731

- 18.7.3 Multi-Layer Feedforward page 731 - 733
 - 18.7.4 Learning in Multi-Layer Networks page 733 – 736
 - 18.7.5 Learning in Neural Networks Struct. page 736 - 737
3. **Video-** Interesting ted talk of how it is used to detect cancer. Wegner, Brittany (N.A.)
How to make a Neural Network in your bedroom, TED x CERN Talks
<http://www.youtube.com/watch?v=n-YbJi4EPxc> last visited 11 July 2014

Include the following in your Research notes:

In your notebook, answer the following questions:

1. What are Artificial Neural Networks (ANN)?
2. Explain the process by which they learn.
3. How would you classify the type of learning of the default ANN?
4. What is the difference between an ANN and a BN?
5. Name a couple of concrete examples for which ANN are used?

19. Lab: Implementing Neural Networks

Goal: understand how to create an ANN and the pros and cons of training it (**6 hours**)

See: <https://www.alphagrader.com/>. Go to Assignment “Perceptron / Neural Networks”

Suggestions for Challenge:

Create a useful program where you make your own implementation of a Perceptron algorithm or (an Artificial Neural Network) to solve a classification problem. Some ideas include to train a network to recognize a character, an object in a picture, a sound, a song, a word, digit recognition, symbol recognition, object recognition, sound recognition, etc...