TDT4137 (Cognitive Architectures) Assignment Sheet 4

Danilas Miscenko
Student ID: 536546

October 26, 2020

Acknowledgements

While working on this assignment, some topics were discussed in a group. The group members are Piri Babayev, Danilas Miscenko (me) and Aleksander Simmersholm. Because of the discussion that has taken place, some similarities between answers may arrise, but all of the assignment tasks were completed individually, as the point of the discussions is to get a better understanding of the subject and not copy each others work.

1 Connectionism

Task 1.1: What are some key characteristics of connectionist approaches?

Connectionist systems rely on parallel processing. The most common instantiations of connectionist systems uses neural networks, with the pattern of activation (and non-activation) of neurons being the way to store information. The system is usually non-symbolic with no explicit production rules. The systems should be dynamic and capture statistical regularities or associations observed in the input data or the consequences of interactions with the environment.

Task 1.2: optional Explain some ways connectionist approaches are inspired by human biology?

Studies of the brain have shown that the brain consists of vast networks of neurons. These neurons can fire of signals that reach other neurons that might prompt them to fire as well. These networks are called neural networks. Information is believed to be stored in the brain as patterns of neurons firing. Connectionist systems attempt to replicate this by using artificial neural networks to mimic the behaviour of the brain.

1.1 Artificial neural networks

Task 1.3: What is meant by the statement that artificial neural networks are universal function approximators?

This statement means that any function can be computed (or approximated) by a neural network.

Task 1.4: Explain, using also a formal description, why image classification can be addressed by a universal function approximator.

By definition of the universal function approximator, if we can define a problem as a function with n inputs and m outputs, then we can compute it with a neural network to some level of accuracy (or approximate the results). Image classification is a problem that isn't very hard to define as a function, because we could have $x_1, x_2, x_3, ...x_n$ inputs, which are the pixel values of the image, and $f(x_1, x_2, x_3, ...x_n)$ being the classification of the image.

Task 1.5: optional Explain some characteristics of a problem domain (e.g. the form of the input data) that would encourage the use of neural nets

If the data can be vectorised in some way, then it can be used as input data for a neural network.

Task 1.6: What is the most important difference between an sigmoid activation function and the ReLU activation function?

The sigmoid function maps all input values (positive and negative) to a value between -1 and 1, whereas the Rectified Linear Unit maps all negative values to 0, and all positive values to a linearly increasing value from 0 to (potentially) infinity.

Task 1.7: Explain the "association black box" analogy of artificial neural networks.

The term "black box" is used to explain that the process is not necessarily known. When talking about ANN's, the neural network basically takes in a vector of some kind, and by some weighted processes inside of the neural network maps the input to some sort of predetermined output, or in other words, associates the input to some output, making it an "association black box".

Task 1.8: optional What are the reasons why there has been a huge increase in the use of deep neural in the last years?

Some of the factors which contribute to the increase in interest of deep neural networks are:

- Massive amounts of data readily available
- Computational power of the hardware has increased (faster CPU's, more cores, more powerful GPU's)
- Many new optimization and learning methods are being developed

Task 1.9: optional Explain the biological inspiration for Convolutional Neural Networks

Researchers have found out that some areas in the visual cortex are meant for detecting specific features, such as different orientations etc. Convolutional Neural Networks mimic this by having neurons that are looking for specific patterns in an image by convolving a filter with their pattern over the image.

1.2 Perceptron

Task 1.10: Give an example of a problem that can be solved by a perceptron. You might sketch this.

A perceptron is able to preform linearly separable binary classification. If the classification problem is not linearly separable from the get go, the data can be pre-proccesed first to re-classify the problem as linearly separable (feature engineering). An example of a ploblem that can be tackled by a perceptron is classifying whether a picture depicts a house or a boat.

Task 1.11: Give an example of a problem that cannot be solved by a perceptron.

A single perceptron is unable to solve any non-linear function.

Task 1.12: Implement the perceptron model using python. Test your model with the **AND** and **OR** functions. Provide your code and explain your results

The perceptron class has four functions, activate, traint, test and print, as well as the variables weights, threshold, bias and alpha. Upon instantiation of the class, the variables threshold and bias are set to 0, the weights are set to [0, 0] and alpha is set to 0.01. The alpha value is the step value used in the training function. The rest of the values are supposed to be set to some random values, but that is just to increase efficiency of the algorithm, ergo, it doesn't matter what they are set to initially.

The activate function adds the bias together with the weighted inputs to the neuron and returns 0 if the sum is below the threshold, and 1 if the sum is equal to or above the threshold:

```python
def activate(self, inputs):
    Sum = self.bias + numpy.dot(self.weights, inputs)
    return 0 if self.threshold > Sum else 1
```

The train function checks whether the predicted value corresponds with the actual value in the data, and if not, corrects the weights and bias by using the Heavyside function:

```python
def train(self, data):
    for i in range(100):
        for row in data:
            result = self.activate(row[0:len(row)-1])
            self.bias += self.alpha * (row[len(row)-1] - result)

            for i in row[0:len(row)-1]:
                for w in range(0, len(self.weights)):
                    self.weights[w] += self.alpha * (row[len(row)-1] - result) * i
```

The test function just counts how many of the predicted values are correct according to the actual data, and returns that counter number:

```
def test(self, data):
    correct = 0
    for row in data:
        if self.activate(row[0:len(row)-1]) == row[len(row)-1]:
            correct += 1

    print(str(correct) + "/" + str(len(data)) + " correct")
```

The print function just prints the weights and the bias of the perceptron:

```
def print(self):
    print("Weights: " + str(self.weights))
    print("Bias: " + str(self.bias))
```

The results are as expected, with both the AND and the OR gates. As both AND and OR are linearly separable, both perceptrons are able to get 4/4 correct:

```
AND gate
Weights: [0.01, 0.01]
Bias: -0.02
4/4 correct
```

```
OR gate
Weights: [0.01, 0.01]
Bias: -0.01
4/4 correct
```

Task 1.13: optional Plot the learnt decision boundary for both the **AND** and **OR** functions.

choose to obstain from this one due to time constraints

1.2.1 XOR

Task 1.14: optional What happens when run with the **XOR** function? Plot the decision boundary and explain what is happening.

choose to obstain from this one due to time constraints

1.2.2 IRIS

Task 1.15: optional The Iris data sets consists of 3 different types of irises' (Setosa, Versicolour, and Virginica) petal and sepal length. We have grouped the Virginica and the Versicolor in one class and the Setosa one class. The task is to use the perceptron to separate the Setosa from the other flowers using only the Sepal Width and the Petal Width with your perceptron model. The data is provided in the data.csv and the true classes in the target.csv. Plot the learnt decision boundary

choose to obstain from this one due to time constraints

1.3 General discussion

Task 1.16: Do ANN's use symbols? Explain how or how it is not possible.

Artificial neural networks do not use symbols, instead representing data by patterns of excited (or non-excited) neurons.

Task 1.17: optional In your opinion, is association making the same thing as real intelligence? If not, how do they relate to each other? Motivate your view.

While intelligent beings exhibit association making, it alone is not intelligent behaviour in my opinion. While an artificial neural network might look at a picture of a cat, and predict that the picture is indeed a cat and not a dog with 97 percent confidence, the neural network doesn't actually know what a cat is, and how it is different than a dog (other than the features it uses to compare the pictures). No deliberation was involved in this association, as the neural network matches the picture to some patterns that it has in it's memory to discern that the picture includes features that are typical of a cat.

It is the same as the Chinese room thought experiment, where a man sitting in an isolated room receives Chinese characters, and based on some rules he has in a book sends out different Chinese characters. The man is therefore creating associations of what characters need to be sent out, but he does not need to understand the characters to do so.

Task 1.18: Is there something that coarsely resembles production rules in ANNs?

Paul Smolensky is famous for inventing so called tensor product methods for simulating the process of variable binding, where symbolic information is stored at and retrieved from known "locations". If one were to stretch a bit, one could argue that these tensor product methods are ”kind of” production rules.

One might also argue that the weights at each neuron are also ”kind of” production rules, because they are what determine the output of the neural network and the relationships between which neurons fire when to which stimuli. They are also the mechanism by which the network learns, as the weights get tuned to better suit reality.

Task 1.19: Some connectionists argue that the brain's neural net indeed implements a symbolic processor. How do they argue for this claim?

Quoting from the entry on Connectionism in the Stanford Encyclopedia of Philosophy:

”True, the mind is a neural net; but it is also a symbolic processor at a higher and more abstract level of description.”

Task 1.20: optional In your own words, what do you believe are the main challenges that the connectionist perspective has to face in the coming decades?

ANN's are based on an extremely surface level abstraction of how we currently perceive the brain to work. The challenge in the coming decades in my opinion will be to adapt the model to new discoveries that will be made about how the human brain works, and if our current model is flawed or incorrect, create a new model that implements the discoveries made.

2 Belief-Desire-Intention (BDI) architecture

Task 1.21: Explain the different architecture components in the BDI architecture.

- **Belief:** Information that the agent percieves to be true about its environment.
- **Desire:** The goal state(s) of the agent, or states of the environment that the agent wishes to achieve.
- **Intention:** In general, not all desires can be achieved, so the intentions of the agent are the desire(s) the agent fixates on, or has commited to achieving.

Task 1.22: Comment on the view of 'practical reasoning' as referred to in the context of BDI?

Practical reasoning in the context of BDI means that the reasoning is directed towards some actions, or in other words, the process by which the agent figures out what to do next. The counterpart to this is theoretical reasoning where the agent contemplates on beliefs, or what it considers to be true.

3 Subsumption

Task 1.23: What are the key ideas behind the subsumption architecture?

- Situatedness/embodiment: Real intelligence is situated in the world, not in disembodied systems.
- Intelligence and emergence: Intelligent behaviour results from agent's interaction with its environment. Also, intelligence is "in the eye of the beholder".

Task 1.24: optional Many cognitive architectures make use of an explicit model of the world to be used for cognitive tasks, e.g. problem solving and learning. Does that also include Brooks' Subsumption Architecture? How does this architecture relate to the Physical Symbol Systems Hypothesis and the Heuristic Search Hypothesis?

choose to obstain from this one due to time constraints

Task 1.25: Describe the layered structure of the Subsumption Architecture. How are the layers related to each other, how are their dependencies? How are the layers linked to sensor data in and actions out?

The subsumption architecture couples sensory information to action selection in an intimate and bottom-up fashion, meaning that complete behaviours are decomposed into smaller sub-behaviours which are chosen in real time via the sensory information. The sub-behaviours are organized into a hierarchy of layers, each implementing some primitive competence. The lower layers have precedent over higher layers. The higher layers use the competence of the lower ones to achieve some behaviour which requires said competence. The layers work in parallel where the communication between them is done in the form of inhibition and suppression signals. Inhibition is used to block signals to actuators or layers, while suppression is for blocking or replacing inputs to the layers.

4 Hybrid architectures

Task 1.26: What is the benefit of combining reactive and deliberative sub-components to create a hybrid architecture?

The reactive sub-components would be capable of responding to changes in the environment without any complex reasoning and/or decision making, while the deliberative sub-components would be responsible for abstract planning and decision-making using symbolic representations.