TDT4137 (Cognitive Architectures) Assignment 4

1 Connectionism

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

Connectionist approaches tries to explain mental phenomena using artificial neural networks. That is, it uses neurons that have input signals, which are weighted, and connected to some output signals. If we look at connectionism from a 'non-technical' view, it is to gather information from a lot of sources to understand a subject. For instance like I did when answering this question. I read the lecture slides, watched a Youtube video, read the provided article and found information on Wikipedia. This is also what happens in the ANN connectionist approaches uses. We can think about the input signals as information from different sources (e.g. my information from slides, articles, Youtube), their weight as how trustworthy the information is (Youtube and Wikipedia are less trustworthy than slides and articles), and the output signals as the answer I now have given.

1.1 Artificial neural networks

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

Basically, it means that an artificial neural network can generate a class of functions that gives a valid output value, based on input value and training data.

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

Image classification can be addressed by a universal function approximator because of something called convolution. Input is an image, seen as RGB values, output is a classification of the image. The input values are fed to filter units, that detects certain features of the image, such as a dog's nose or elephant ear. After detecting all these features, the image are reduces to a *feature map* (which consists of all unique features found in the input values.). The feature map are then put into so called hidden layers, for further detection of features in the image. Following this, the output from the hidden layers are passed to a classification layer, where a label to the image is assigned. I.e "dog".

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

The most important difference between a sigmoid activation function and a ReLU activation function is that a sigmoid activation function will map all all negative values close to 0 and all positive closer to 1 ($\frac{1}{1+e^{-x}}$), giving it a steep curve from x = 0 to x = 1, where the ReLU function maps all negative values to 0 and all positive to themselves ($f(x) = max(0, x)$), giving it a linear curve from x = 0 to x = 1.

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

The association black box analogy is that in a deep neural network, all computation happens in "a black box". We can not percieve the computation, only the input and output values.

1.2 Perceptron

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

Classification of elephants and mice, based on their weigth can be solved by a perceptron. The elephants will be heavy, while the mice will be light. This way, they are linearly seperable in i graph, and a perceptron can solve the problem.

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

XOR-operations cannot be solved by a perceptron, due to the fact that they are not linearly separable.

> **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

Code:

```
import numpy as np
import random

class Percepton:
    def __init__(self, learning_rate=0.01):
        self.learning_rate = learning_rate
        self.threshold = 1000
        self.weights = np.zeros(3)
        self.bias = random.uniform(-0.5, 0.5)

    def predict(self, inputs, train=True):
        yp = np.dot(inputs, self.weights)
        return 1 if yp > 0 else 0

    def train(self, train_data):
        for i in range(self.threshold):
            x, expected = random.choice(train_data)
            prediction = self.predict(x)
            self.weights += self.learning_rate * (expected - prediction) *
x
```

Results:

```
------------ AND ------------             ------------ OR ------------
[0, 0] --> 0                              [0, 0] --> 0
[0, 1] --> 0                              [0, 1] --> 1
[1, 0] --> 0                              [1, 0] --> 1
[1, 1] --> 1                              [1, 1] --> 1


Weights: [ 0.02  0.01 -0.02]              Weights: [0.01 0.01 0.  ]
Bias: -0.02                               Bias: 0.0
```

We see that for predicting the AND function, each input has to be 1, for the prediction to be 1. Having the weights $[0.02, 0.01]$ and bias -0.02, we need both inputs to be one, otherwise, the prediction will return 0 because the dot product would result in either $0.02 + 0 - 0.02 = 0$ or $0 + 0.01 - 0.02 = -0.01$. For the OR-function, it is the other way around. We need *at least* one input to be 1, or else the predict function will return 0: $0 + 0 - 0 = 0$.

1.3 General discussion

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

Atrificial neural networks uses distributed representation. This means that the state of the network is represented by "activating" different nodes (or neurons) and not by symbols. This is also how the brain works. Different neurons are activated to represent a state. This is also closely related to connectionism. The system finds it's information/knowledge throughout different neurones/nodes.

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

The input-specific weights of the different nodes in an ANN coarsely resembles production rules.

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

Most connectionists claim that information is stored non-symbolically in the nodes' weigths, while others claim that the brain's net implements a symbolic processor. They think that yes, the brain is still a neural net, but also a symbolic processor at a higher and more abstract level. Following this, the connectionist research is to descover how the machinery for symbolic processing can be forged from a neural network.

2 Belief-Desire-Intention (BDI) architecture

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

The different attribute components in the BDI architecture is *Beliefs*, *Desires* and *Intentions*.

- Belief
  - Consists of information the agent has about the world. This information may or may not be true, it is what the agent *belives*.
- Desires
  - States of affairs that the agent would wish to be brought about
- Intentions
  - Desires that the agent has committed to achieving

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

Practical reasnoning is the process of finding out what to do. This comprises two activities; deliberaion and means-end reasoning. Deliberations means *what* state of affairs wr want to achieve and means-end reasoning means deciding *how* we want to achieve this state.

In a BDI, practical reasoning is done continously. The architechture runs in a loop that can be roughly summarized with the following loop:

```
while true:
    Observe the world
    Update internl workl model
    _Deliberate about what intention to achieve next_
    _Use means-end reasoning to get a plan for the intention_
    Execute plan
end while
```

This means that a BDI will not remember what it has previously done, only what is best to do and what it should do right now.

3 Subsumption

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

The subsumption architecture mainly incorporates four priciples:

1. It is essentially a hierarchical structure og task-accomplishing behaviours. This means that the architecture has rules that are competing for control over the agent's behaviour.
2. The architechture has behaviours, which basically are rules as to what to do. Theese rules can fire simultaneously.
3. It is a layered structure, where lower layers correspond to primitive behaviours, such as "avoid obstacles", have precedence over higher, more abstract levels, such as "drive to goal".
4. The architecture is extremely simple in computational terms, but very effective.

> **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 has a layered structure where lower, more primitive layers, has precedence over hoigher, more abstract, layers. Each layer is responsible for a singe behaviour goal, such as "avoid obstacle". Each layer is made up of connected, simple modules/processors that works as finite state machines. The layers communicate through the suppression of signals from other layers, which are communicated to an actuator. The actuator will co the output and percept new inputs.

4 Hybrid architectures

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

The benefit is that you "get the best of both worlds". You combine both perspectives in one arcitechture. The sub-components gathered from the reactive architecture will be able to respind to ral world changes without the need for ant complex reasoning and decision-making, and the deliberative sub-system will provide responsibility for abstract planning and decision-making using symbolic representations.