CP468 Assignments 5, 6 & 7 Report

Daniel Crha | 190891000

Mera Fares | 190322170

Ruben Halanen | 180573480

Samuel Tessema | 170535150

Abstract

This report presents an overview of three different types of artificial intelligence systems: rule-based systems, expert systems, and artificial neural networks (ANNs). Rule-based systems use a set of pre-defined rules to make decisions or provide answers to specific questions. Expert systems are a type of rule-based system that incorporate domain-specific knowledge and can reason with uncertainty. ANNs are a class of machine learning algorithms that are modeled after the human brain and are able to learn from data to make predictions or classifications. We demonstrate these systems in assignments 5, 6, and 7.

Assignment 5

Code

```
digit(0).
digit(1).
digit(2).
digit(3).
digit(4).
digit(5).
digit(6).
digit(7).
digit(8).
digit(9).

valid_digits(X, Y, Z) :-
    digit(X),
    digit(Y),
    digit(Z),
    X \== Y,
    Y \== Z,
    Z \== X.

clue1(X, Y, Z) :-
    valid_digits(X, Y, Z),
    ((X = 7, Y \== 8, Z \== 3);
     (X \== 7, Y = 8, Z \== 3);
     (X \== 7, Y \== 8, Z = 3)).

clue2(X, Y, Z) :-
    valid_digits(X, Y, Z),
   ((X \== 1, Y = 7, Z \== 6);
    ( X \== 1, Y\== 6, Z = 7);
    ( X = 1, Y \== 6, Z \== 7);
    ( X \== 6, Y\== 7, Z = 1);
    ( X = 6, Y \== 7, Z \== 1);
    ( X \== 1, Y = 6, Z \== 7)).


clue3(X, Y, Z) :-
    valid_digits(X, Y, Z),
    ((X = 0, Y = 3, Z \== 7);
    ( X \== 7, Y = 3, Z = 0);
    ( X = 0, Y \== 7, Z = 3);

    (X = 0, Y = 7, Z \== 3);
    (X \== 3, Y = 7, Z = 0);
    (X = 7, Y \== 3, Z = 0);

    (X = 7, Y\==0, Z = 3);
    (X \== 0, Y = 7, Z = 3);
    (X = 7, Y = 3, Z \== 0)).
```

```
clue4(X, Y, Z) :-
    valid_digits(X, Y, Z),
    X \== 4,
    X \== 2,
    X \== 8,
    Y \== 4,
    Y \== 2,
    Y \== 8,
    Z \== 4,
    Z \== 2,
    Z \== 8.


clue5(X, Y, Z) :-
    valid_digits(X, Y, Z),
    ((X \== 8, Y = 4, Z \== 0);
    ( X \== 8, Y\== 0, Z = 4);
    ( X = 8, Y \== 4, Z \== 0);
    ( X \== 4, Y\== 0, Z = 8);
    ( X = 0, Y \== 8, Z \== 4);
    ( X \== 4, Y = 0, Z \== 8)).

crack_code(X, Y, Z) :-
    clue1(X, Y, Z),
    clue2(X, Y, Z),
    clue3(X, Y, Z),
    clue4(X, Y, Z),
    clue5(X, Y, Z).
```

Code Explanation

The code starts by defining all of the possible digits could fill each square. Then, define each clue given, and check the if digits are valid. We then use the "crack_code" predicate to come up with a code using all of the clues given. With all of the clues taken into account, the only possible answer is "063". If we get rid of clue 1, the possible codes become "031", "063", "075", "079". If we use only the first 3 rules, possible results are "063", "173", "273", "473", "573","673", "731", "760", "873", "973"

Results

Result using all of the clues:

```
?- crack_code(X,Y,Z).
X = 0,
Y = 6,
Z = 3 ;
```

Result not using rule 1:

```
?- crack_code(X,Y,Z).
X = 0,
Y = 3,
Z = 1 ;
X = 0,
Y = 6,
Z = 3 ;
X = 0,
Y = 7,
Z = 1 ;
X = 0,
Y = 7,
Z = 5 ;
X = 0,
Y = 7,
Z = 9 ;
false.
```

Result using only the first 3 rules:

```
?- crack_code(X,Y,Z).
X = 0,
Y = 6,
Z = 3 ;
X = 1,
Y = 7,
Z = 3 ;
X = 2,
Y = 7,
Z = 3 ;
X = 4,
Y = 7,
Z = 3 ;
X = 5,
Y = 7,
Z = 3 ;
X = 6,
Y = 7,
Z = 3 ;
X = 7,
Y = 3,
Z = 1 ;
X = 7,
Y = 6,
Z = 0 ;
X = 8,
Y = 7,
Z = 3 ;
X = 9,
Y = 7,
Z = 3 ;
false.
```

Assignment 6

In assignment 6, we created a simple expert system symptom checker, which uses a series of questions about symptoms to identify potential health issues. The user is prompted to answer "yes" or "no" to questions about symptoms they are experiencing. Based on the combination of symptoms reported, the program uses a set of rules to suggest a possible diagnosis and provide advice for seeking medical attention or self-care. If the reported symptoms do not fit any of the pre-determined rules, a general recommendation to monitor symptoms and seek medical attention if necessary is given. This code demonstrates how an expert system can be used to assist in medical diagnosis by automating the decision-making process using artificial intelligence techniques.

Code

cp468_a6_expert_system.py > ...

```python
print("Welcome to our expert system, symptom checker!")

fever = input("Do you have a fever? (y/n) ")
cough = input("Do you have a cough? (y/n) ")
shortness_of_breath = input("Do you have shortness of breath? (y/n) ")
body_aches = input("Do you have body aches? (y/n) ")
headache = input("Do you have a headache? (y/n) ")
sore_throat = input("Do you have a sore throat? (y/n) ")
loss_of_taste_or_smell = input("Have you lost your sense of taste or smell? (y/n) ")
fatigue = input("Do you feel unusually tired or fatigued? (y/n) ")
nausea_or_vomiting = input("Do you have nausea or vomiting? (y/n) ")
diarrhea = input("Do you have diarrhea? (y/n) ")

if fever == "y" and cough == "y" and shortness_of_breath == "y":
    print("You may have COVID-19. Please seek medical attention.")
elif fever == "y" and body_aches == "y" and headache == "y" and fatigue == "y":
    print("You may have the flu. Rest and drink fluids.")
elif cough == "y" and shortness_of_breath == "y" and fever == "n" and sore_throat == "n" and loss_of_taste_or_smell == "n":
    print("You may have pneumonia. Please seek medical attention.")
elif sore_throat == "y" and loss_of_taste_or_smell == "y" and fever == "n" and cough == "n" and shortness_of_breath == "n":
    print("You may have COVID-19. Please seek medical attention.")
elif nausea_or_vomiting == "y" and diarrhea == "y" and fever == "n" and cough == "n" and shortness_of_breath == "n":
    print("You may have gastroenteritis. Rest and drink fluids.")
else:
    print("It's unclear what condition you may have. Please monitor your symptoms and seek medical attention if they worsen.")

```

Code Explanation

The code starts by printing a welcoming message, then prompts the user to answer a series of yes or no questions about their symptoms using the `input()` function. The answers are stored as strings in variables named after the corresponding symptom.

The program then uses a series of `if` statements to check for specific combinations of symptoms and provide a possible diagnosis and recommendation. For example, if the user reports having a fever, cough, and shortness of breath, the program will print a message suggesting that the user may have COVID-19 and advising them to seek medical attention.

If the reported symptoms do not match any of the pre-determined rules, the program prints a general message suggesting that the user monitor their symptoms and seek medical attention if necessary.

The code demonstrates a simple implementation of an expert system using a rule-based approach. However, in a real-world scenario, a more sophisticated system may be required, such as one that uses machine learning algorithms to improve the accuracy of the diagnosis.

Results

```
Welcome to our expert system, symptom checker!
Do you have a fever? (y/n) y
Do you have a cough? (y/n) y
Do you have shortness of breath? (y/n) y
Do you have body aches? (y/n) n
Do you have a headache? (y/n) n
Do you have a sore throat? (y/n) n
Have you lost your sense of taste or smell? (y/n) n
Do you feel unusually tired or fatigued? (y/n) n
Do you have nausea or vomiting? (y/n) n
Do you have diarrhea? (y/n) n
You may have COVID-19. Please seek medical attention.
```

```
Welcome to our expert system, symptom checker!
Do you have a fever? (y/n) y
Do you have a cough? (y/n) n
Do you have shortness of breath? (y/n) n
Do you have body aches? (y/n) y
Do you have a headache? (y/n) y
Do you have a sore throat? (y/n) n
Have you lost your sense of taste or smell? (y/n) n
Do you feel unusually tired or fatigued? (y/n) y
Do you have nausea or vomiting? (y/n) n
Do you have diarrhea? (y/n) n
You may have the flu. Rest and drink fluids.
```

```
Welcome to our expert system, symptom checker!
Do you have a fever? (y/n) y
Do you have a cough? (y/n) n
Do you have shortness of breath? (y/n) y
Do you have body aches? (y/n) n
Do you have a headache? (y/n) y
Do you have a sore throat? (y/n) n
Have you lost your sense of taste or smell? (y/n) y
Do you feel unusually tired or fatigued? (y/n) n
Do you have nausea or vomiting? (y/n) y
Do you have diarrhea? (y/n) n
It's unclear what condition you may have. Please monitor your symptoms and seek medical attention if they worsen.
```

Assignment 7

In order to implement the 2-layer ANN Perceptron with 3 input neurons and 1 output neuron, we use the following code in Python using the numpy library:

Code

```python
import numpy as np

iterations = 0
c = 0.2


weights = np.array([0.75, 0.5, -0.6])

# Define training data
data = np.array([[1.0, 1.0, 1.0, 1.0],
                 [9.4, 6.4, 1.0, -1.0],
                 [2.5, 2.1, 1.0, 1.0],
                 [8.0, 7.7, 1.0, -1.0],
                 [0.5, 2.2, 1.0, 1.0],
                 [7.9, 8.4, 1.0, -1.0],
                 [7.0, 7.0, 1.0, -1.0],
                 [2.8, 0.8, 1.0, 1.0],
                 [1.2, 3.0, 1.0, 1.0],
                 [7.8, 6.1, 1.0, -1.0]])
count = data.shape[0]

while count > 0:
    count = 0
    for i in range(data.shape[0]):
        x = data[i,:3] # input values
        desired = data[i,3]  # desired output
        output = np.sign(np.dot(weights, x)) # multiply the weights, the input values, and the sign
        if output != desired: # make sure output is different than desired before incrementing
            count += 1
            weights += c*(desired - output)*x # update weights using given formula
    iterations += 1

print("Weights after one full iteration:", weights)
print("Number of iterations through the data:", iterations)
```

Code explanation

In this implementation, we have a loop that goes through the whole data set and updates each data point using the given formula. One iteration is the program training through all of the data points. Thus, one iteration of training would update the weights 10 times. After one iteration, the weight will be closer to the optimal weight that minimize the error. First, we use np.dot to multiply the array of weights with the inputs and the correct sign. Then, if the output != the desired output, we increment the count and update the weights accordingly. After one iteration the updated weights are [-0.69, -0.26, 3.8]. The count variable tracks the error count. In this case, after 6 iterations through the data set, the count reaches 0 and the while loop ends. The difference between the last updated weights and the current updated weights was close enough to 0 to stop looping. Meaning there is now zero error after 6 iterations.

Results

```
Weights after one full iteration: [-0.69 -0.26  3.8 ]
Number of iterations through the data for zero error: 6
>
```