

Profielwerkstuk
Machine Learning, Neural Networks and
the evolution of Neural Networks through
Augmenting Topologies

-
describing the underlying methods and the particular
application of these principles to making a computer teach
itself how to play a video game

by Maxim L. van den Berg, Sam R.W. van Kampen,
Wilco M. Stam and Robin A.J. Wacanno

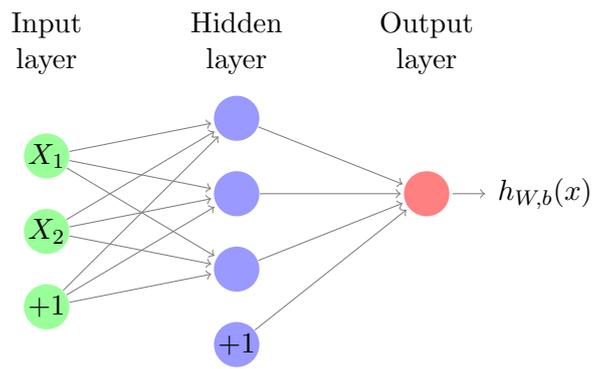
23-12-2016

Abstract

Lekker abstract

Contents

1	Introduction	5
2	[temp]	5
2.1	Background information	5
2.1.1	Solving mathematical problems with a computer	5
2.1.2	Neural networks - a simulation of the brain [temp]/ An introduction to neural networks	5
2.1.3	Examples of applications of machine learning and neural networks .	5
2.2	Machine learning	5
2.2.1	Mathematical theory and model	5
2.2.2	Implementation in the programming language Octave	5
2.3	Neural networks	6
2.3.1	Mathematical theory and model	6
2.3.2	Implementation in the programming language Octave	6
2.3.3	Alternative algorithms to gradient descent	6
2.4	Evolving neural networks and self-learning artificial intelligence	6
2.4.1	Self improving neural networks	6
2.4.2	The application of neural network based artificial intelligence to video games	6
2.4.3	Implementation in the programming language Octave	6
2.5	Experiment: The application of neural network based artificial intelligence to video games	6
2.5.1	Choosing an emulator and transferring the program to the appropriate language	6
2.5.2	The setup and time partition of the experiment	6
2.5.3	[after the experiment] Analysis of the data	6
3	Conclusion	6



Listing 1: Data definitions (nnrv.m)

```

1 W1 = stdnormal_rnd(2,3);
2 W2 = stdnormal_rnd(3,1);
3 b1 = stdnormal_rnd(1,3);
4 b2 = stdnormal_rnd(1,1);
5
6
7
8
9 exx1 = [1; 0; 1; 1; 0; 0; 0; 1; 1; 0; 1; 0; 0; 0];
10 exx2 = [0; 1; 0; 1; 0; 0; 1; 1; 1; 0; 0; 1; 1; 0];
11
12 exy = [0; 0; 0; 1; 0; 0; 0; 1; 1; 0; 0; 0; 0; 0];

```

Listing 2: Implementation of the XNOR gate in a neural network in the Octave language (nnt1.m)

```

1 % An implementation of the XNOR-gate using a neural network
2 % Uses data values defined in nnrv.m
3 %
4 % by Maxim van den Berg, Sam van Kampen, Robin Wacanno and Wilco Stam
5
6 % The learning rate
7 alpha = 1;
8
9 % The effect of weight decay is negligible in this network, and is therefore
10 % zero.
11 lambda = 0;
12
13 maxi = input("maxi: ");
14 m = rows(exx1);
15
16 function s = f(z)
17     s = 1./(1 + exp(-z));
18 end
19
20
21 %
22 disp("W2 (begin) =");
23 disp(W2);
24 disp("W1 (begin) =");
25 disp(W1);
26 disp("b2 (begin) =");
27 disp(b2);
28 disp("b1 (begin) =");
29 disp(b1);
30
31 % Compute starting values, so that they can be displayed
32 a1 = [exx1, exx2];
33 z2 = a1 * W1 + b1;
34 a2 = f(z2);
35 z3 = a2 * W2 + b2;
36 h = f(z3);
37 a3 = h;

```

```

38 disp("h (begin) =");
39 disp(h);
40
41
42 for i = 1:maxi
43
44 % Compute the current output of the neural net
45 a1 = [exx1, exx2];
46
47 z2 = a1 * W1 + b1;
48 a2 = f(z2);
49
50 z3 = a2 * W2 + b2;
51 h = f(z3);
52 a3 = h;
53
54 % Start computing deltas for backprop
55
56 %delta[laagnummer]
57 delta3 = -(exy - h) .* (a3 .* (1 - a3)); % f'(z) = f(z)(1 - f(z)).
58 delta2 = (delta3 * W2') .* (a2 .* (1 - a2));
59
60 gradW2 = a2' * delta3;
61 gradW1 = a1' * delta2;
62 gradb2 = sum(delta3);
63 gradb1 = sum(delta2);
64
65
66 W2 = W2 - alpha * (gradW2 * 1/m) + lambda * W2);
67 W1 = W1 - alpha * (gradW1 * 1/m) + lambda * W1);
68
69 b2 = b2 - alpha * (gradb2 * 1/m);
70 b1 = b1 - alpha * (gradb1 * 1/m);
71
72 end
73
74
75 disp("h =");
76 disp(h);
77
78 disp("round(h) =");
79 disp(round(h));
80
81 disp("J / cost =");
82 disp(delta3);
83
84 disp("W2 =");
85 disp(W2);
86 disp("W1 =");
87 disp(W1);
88
89 disp("b2 =");
90 disp(b2);
91 disp("b1 =");
92 disp(b1);

```

(temporary titles)

1 Introduction

Define what the reader should now already. For example: "It is expected that basic matrix functions and basic programming syntax are understood."

2 [temp]

2.1 Background information

2.1.1 Solving mathematical problems with a computer

- History
- Basic idea (fitting a formula to a data set and such)

"[Machine Learning is] The field of study that gives computers the ability to learn without being explicitly programmed."

Arthur Lee Samuel (1959), a pioneer in the field of computer gaming, artificial intelligence, and machine learning.

"Well-posed Learning Problem: A computer program is said to learn from experience E with respect to some task T and some performance measure P , if its performance on T , as measured by P , improves with experience E ."

Tom M. Mitchell, a famous computer scientist and university professor.

2.1.2 Neural networks - a simulation of the brain [temp]/ An introduction to neural networks

- Biology lesson about the brain and how neural networks work alike, especially evolving neural networks

2.1.3 Examples of applications of machine learning and neural networks

- Examples of neural network implementation and what they have achieved

2.2 Machine learning

2.2.1 Mathematical theory and model

- Gradient descent
- Logistical regression
- formulas and alpurrr
- Explanation vectorized implementation

2.2.2 Implementation in the programming language Octave

- First without vectorization.
- Then with vectorized implementation - Determining the amount of iterations and the value for alpha.

2.3 Neural networks

2.3.1 Mathematical theory and model

- Unsupervised versus supervised learning
- Autoencoders
- formulas and alpurrr
- Explanation vectorized implementation

2.3.2 Implementation in the programming language Octave

- First without vectorization.
- Then with vectorized implementation - Determining the amount of iterations and the value for alpha and labda.

2.3.3 Alternative algorithms to gradient descent

2.4 Evolving neural networks and self-learning artificial intelligence

2.4.1 Self improving neural networks

neural networks are braaaains, etc. etc. etc.

2.4.2 The application of neural network based artificial intelligence to video games

- theory
- input handling
- Which games is is applicable to?

2.4.3 Implementation in the programming language Octave

2.5 Experiment: The application of neural network based artificial intelligence to video games

2.5.1 Choosing an emulator and transferring the program to the appropriate language

2.5.2 The setup and time partition of the experiment

2.5.3 [after the experiment] Analysis of the data

- Did it work?
- How long did it take?
- Application on other levels and games
- How can the program be improved

3 Conclusion