Insert here your thesis' task.

FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Bachelor's thesis

Gesture detector with Leap Motion sensor

Anh Tran Viet

Department of Theoretical Computer Science
Supervisor: Tomáš Nováček

November 17, 2020

Acknowledgements

THANKS (remove entirely in case you do not with to thank anyone)

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as a school work under the provisions of Article 60 (1) of the Act.

In Prague on November 17, 2020 . . . .

Citation of this thesis

Tran, Anh Viet. *Gesture detector with Leap Motion sensor.* Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2020.

Abstrakt

V několika větách shrňte obsah a přínos této práce v českém jazyce.

Klíčová slova Replace with comma-separated list of keywords in Czech.

Abstract

Summarize the contents and contribution of your work in a few sentences in English language.

Keywords Replace with comma-separated list of keywords in English.

Contents

List of Figures

Introduction

LOREM IPSUM.

Chapter 1

Neural Networks

An artificial neural network (ANN) is a mathematical model mimicking biological neural networks, namely their ability to learn and correct errors from previous experience.[1][2]
The ANN subject was first introduced by Warren McCulloch and Walter Pitts in "A logical calculus of the ideas immanent in nervous activity" published in 1943.[6 M] But it was not until recent years when ANN has gained popularity with still increasing advancements in technology and availability of training data. ANN had become one of the default solutions for complex tasks which were previously thought be unsolvable by computers.[3]
This chapter will briefly explore different types of neural units and their activation functions, along with some exemplary network architectures.

1.1 Artificial Neuron

As previously mentioned, artificial neurons are units mimicking behaviors of biological neurons. Meaning it can receive as well as pass information between themselves.

1.1.1 Perceptron

Perceptron is the simplest class of artificial neurons developed by Frank Rosenblatt in 1958.[4] Perceptron takes several binary inputs, vector x = (x1, x2,...,xn), and outputs a single binary number.
To express the importance of respected input edges, perceptron uses real numbers called weights, assigned to each edge, vector w = (w1,w2,...,wn).

A step function calculates the perceptron's output. The function output is either 0 or 1 determined by whether its weighted sum w (SUM) is less or greater than its threshold value, a real number, usually represented as an incoming edge with a negative weight -1.

FIGURE
MATH

1.1.2 Sigmoid Neuron

Sigmoid neuron, similarly to perceptron, has inputs x and weights. The key difference comes in once we inspect the output value and its calculation. Instead of perceptron's binary output 0 or 1, a sigmoid neuron outputs a real number between 0 and 1 using a sigmoid function.[5][6]

MATH
PLOTS

As shown in Figure xx and Figure xx, the sigmoid function is a smoothed-out version of the step function.

1.1.3 Activation Function

An artificial neuron's activation function defines that neuron's output value for given inputs, commonly being f: R->R [7]. A significant trait of many activation functions is their differentiability, allowing them to be used for Backpropagation, ANN algorithm for training weights. Having derivative not saturating or exploding, heads towards 0 or inf, is necessary for activation functions.

For such reasons, the usage of step function or any linear function is unsuitable for ANN.

1.1.3.1 Sigmoid Function

The sigmoid function is commonly used in ANN as an alternative to the step function. A popular choice of the sigmoid function is a logistic sigmoid. Its output value is in the range of 0 and 1.

MATH

One of the reasons being the simplicity of derivative calculation:

MATH

One of its disadvantages being the vanishing gradient. A problem where for a given very high or very low input values, there would be almost no change in its prediction. Possibly resulting in training complications or performance issues.[**?**]

1.1.3.2 Hyperbolic Tangent

Hyperbolic tangent is similar to logistic sigmoid function with a key difference in its output, ranging between -1 and 1.

MATH

It shares sigmoid's simple calculation of its derivative.

MATH

By being only moved and scaled version of the sigmoid function, hyperbolic tangent does share sigmoid's advantages and its disadvantages.[7]

1.1.3.3 Rectified Linear Unit

The output of the rectified linear unit (ReLU) is defined as:

[MATH]

ReLU popularity is mainly for its computational efficiency.[8]

ReLu's disadvantages appear when inputs approach zero or are negative. Causing the so-called dying ReLu problem, where the network is unable to learn. There are many variations of ReLu to this date, e.g., Leaky ReLU, Parametric ReLU, ELU, ...

1.1.3.4 Softmax

Softmax separates itself from all the previously mentioned functions by its ability to handle multiple input values in the form of a vector x = (x1,x2,...,xn) and output for each xi defined as:

MATH

For output being normalized probability distribution, ensuring SUM MATH.[9] It is being used as the last activation function of ANN to normalize the network's output into n probability groups.

1.2 Types of Neural Networks

To this day, there are many types and variations of ANN, each with its structure and use cases. Here we will briefly introduce the most common ones, such as feed-forward networks, convolutional neural networks, or recurrent neural networks.

1.2.1 Feed-forward Networks

Feed-forward network (FNN) has its data or input travel in one direction, oriented from the input layer to the output layer, without cycles.[10] FNN may or may not contain several hidden layers of various widths. By having no back-loops, FNN generally minimizes error in its prediction by using the backpropagation algorithm to update its weight values. [11]

GRAPH

The input layer takes input data, vector x, producing y at the output layer. The process of training weights consists of minimizing the loss function L(y,y), y being the target output of input x.[9]

1.2.1.1 Backpropagation

Backpropagation, short of backward propagation of errors, is a widely used algorithm in training FFN using gradient descent to update the weights. [12]

MATH

Its generalization is used for other ANNs. Providing a way to compute the gradient of the cost function, real number value expressing prediction incorrectness.[13]

1.2.2 Convolutional Neural Networks

Convolutional Neural network (CNN) is a variation of FNN primarily used for image classification, object recognition, and other two-dimensional inputs.[13] CNN is usually consisting of the input layer followed by multiple hidden layers, typically several convolutional layers with standard pooling layers, and ending with the output layer.

1.2.2.1 Convolutional Layer

The convolutional layers' objective is to extract key features from the input image by passing a matrix known as a kernel over the input image abstracted into a matrix.[14]

IMAGE

The convolution result can be of two types. One being the convolved feature is reduced in dimensionality compared to the input, valid padding, and the other in which the dimensionality is either increased or remains the same, same padding. latter. [https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53]

1.2.2.2 Pooling Layer

Similar to the previously mentioned convolutional layer, the pooling layer reduces the convolved feature's spatial size to decrease the computational power required for data processing. Furthermore, being useful by extracting dominant features, which are rotational and positional invariant, thus maintaining the process of effectively training the model.[15]

There are two types of pooling: max pooling and average pooling. Max Pooling returns the maximum value from the portion of the image covered by the kernel. It performs as a noise suppressant, discarding the noisy activations altogether and performing de-noising and dimensionality reduction. Where average pooling returns the average of all the values from the same covered portion, performing dimensionality reduction as a noise suppressing mechanism. Hence, it is possible to note that max-pooling performs better.[15]

IMAGE

1.2.3 Recurrent Neural Networks

Recurrent Neural Network (RNN) distinguished by its memory, RNN takes input sequence with no predetermined size. Its past predictions influence currently generated output. Thus for the same input, RNN could produce different results depending on previous inputs in the sequence.[16]

SCHEMATIC

Figure 1.7b shows the network for each time step, i.e., at time t, the input Xt goes into the network to produce output Yt, the next time step of the input is Xt+1 with additional input from the previous time step from the hidden state Ht. This way, the neural network looks at the current input and has the context from the previous inputs. With this structure, recurrent units hold the past values, referred to as memory. Making it possible to work with a context in data. [17]

The recurrent unit is calculated as follows:

MATH

f() being the activation function, W is the weight matrix, X is the input, and b is the vector of bias parameters. Unit at time step t=0 is initialized to (0,0,...0). The output Y is then calculated as:

MATH

g() also being an activation function, usually being softmax to ensure the output is in the desired class range. W is the weight matrix and b being a vector of biases determined during the learning process.

Training RNNs uses a modified version of the backpropagation algorithm called backpropagation through time (BPTT), working by unrolling the RNN [13], calculating the losses across time steps, then updating the weights with the backpropagation algorithm. More on RNN in [9] by Liton et al.

Bibliography

[1] Chen, Y.-Y.; Lin, Y.-H.; et al. Design and Implementation of Cloud Analytics-Assisted Smart Power Meters Considering Advanced Artificial Intelligence as Edge Analytics in Demand-Side Management for Smart Homes. *Sensors*, 05 2019, doi:10.3390/s19092047.

[2] Bengio, Y.; Goodfellow, I.; et al. *Deep learning*, volume 1. Citeseer, 2017, ISBN 0262035618, 166–485 pp.

[3] Krishtopa. What Are Neural Networks, Why They Are So Popular And What Problems Can Solve. 2016. Available from: `https://steemit.com/academia/@krishtopa/what-are-neural-networks-why-they-are-so-popular-and-what-problems-can-solve`

[4] Rosenblatt, F. The Perceptron: A Probabilistic ModelFor Information Storage And Organization In The Brain. *Psychological Re-view*, 1958: p. 2047, doi:0.1037/h0042519.

[5] Nielsen, M. A. *Neural Networks and Deep Learning*. Determination Press, 2015.

[6] Rojas, R. *Neural networks: a systematic introduction*. Springer Science & Business Media, 2013, ISBN 9783642610684, 37–99 pp.

[7] Leskovec, J.; Rajaraman, A.; et al. *Mining of massive data sets*. Cambridge university press, 2020, ISBN 9781108476348, 523–569 pp.

[8] Maladkar, K. 6 Types of Artificial Neural Networks Currently Being Used in ML. Available from: `https://analyticsindiamag.com/6-types-of-artificial-neural-networks-currently-being-used-in-todays-technology/`

[9] Lipton, Z. C.; Berkowitz, J.; et al. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015:

pp. 5–25, ISSN 2331-8422. Available from: https://arxiv.org/pdf/1506.00019.pdf

[10] Maladkar, K. 6 Types of Artificial Neural Networks Currently Being Used in ML. Available from: https://analyticsindiamag.com/6-types-of-artificial-neural-networks-currently-being-used-in-todays-technology/

[11] Team, T. A. Main Types of Neural Networks and its Applications-Tutorial. Aug 2020. Available from: https://medium.com/towards-artificial-intelligence/main-types-of-neural-networks-and-its-applications-tutorial-734480d7ec8e

[12] Backpropagation. Available from: https://brilliant.org/wiki/backpropagation/

[13] Goodfellow, I.; Bengio, Y.; et al. *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[14] Convolutional Neural Network. Available from: https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html

[15] A Comprehensive Guide to Convolutional Neural Networks-the ELI5 way. Dec 2018. Available from: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[16] Recurrent Neural Networks. Jun 2019. Available from: https://towardsdatascience.com/recurrent-neural-networks-d4642c9bc7ce

[17] Understanding Recurrent Neural Networks in 6 Minutes. Sep 2019. Available from: https://medium.com/x8-the-ai-community/understanding-recurrent-neural-networks-in-6-minutes-967ab51b94fe

Appendix **A**

Acronyms

GUI Graphical user interface

XML Extensible markup language

Appendix B

Contents of enclosed CD

- `readme.txt` .... the file with CD contents description
- `exe` .... the directory with executables
- `src` .... the directory of source codes
  - `wbdcm` .... implementation sources
  - `thesis` .... the directory of LaTeX source codes of the thesis
- `text` .... the thesis text directory
  - `thesis.pdf` .... the thesis text in PDF format
  - `thesis.ps` .... the thesis text in PS format