Training of NN's models for animation parsing and processing

Vladimir de la Cruz Borjas

A Thesis

in

The Department

of

Computer Science

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Computer Science (Computer Science) at

Concordia University

Montréal, Québec, Canada

September 2018

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Vladimir de la Cruz Borjas**

Entitled: **Training of NN's models for animation parsing and processing**

and submitted in partial fulfillment of the requirements for the degree of

Master of Computer Science (Computer Science)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

____________________ Chair
Dr. Name of the Chair

____________________ External Examiner
Dr. Name of External Examiner

____________________ Examiner
Dr. Name of Examiner One

____________________ Supervisor
Dr. Tiberiu Popa

Approved by ____________________
Sudhir Mudur, Chair
Department of Computer Science

__________ 2018 ____________________
Amir Asif, Dean
Faculty of Engineering and Computer Science

Abstract

Training of NN's models for animation parsing and processing

Vladimir de la Cruz Borjas

This research proposes the utilisation of four different techiniques for animation storage: axis-angle, euler angles, rotation matrices and quaternions, in order to determine via the use of autoencoders, the advantage and disadvantage of each of this techiniques for preserving and producing reconstructed data parsed throught a convolutional neural network. The experiment achieves a record-breaking accuracy, higher than the result of 98.04NN competition. The recall of the experiment is nn

Acknowledgments

My greatest appreciations to my parents, for bringing me into this world, and all my family for their constant support and love in my endeavors throughout my life. I am grateful to my supervisor, Dr. Tiberiu Popa for initiating this research study as well as for his continued technical guidance during the completion of this thesis work. I would also wish to acknowledge all my friends who have volunteered their help through every step of my thesis work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Chapter 1 provides a brief summary of the elements to be used in this thesis. A comprehensive literature review on the topic of Neural Networks and Autoencoders is presented in Chapter 2, showcasing from introductory concepts to more advanced techniques and their applications.

Deep Learning is an area of application of the computer science that has been growing more and more throughout the years. The possibilities of applications, the adaptability and the support for different types of problems is something that quite remarkably had not been able to be achieved with other types of algorithms or techniques in the area of Machine Learning.

In comparison with other techniques such a Reinforcement Learning (RL), Neural Networks and more specifically deep neural networks, are a source of new problem solving that was impossible to achieve before, due the intrincany of works and number of variables of the models that could be worked with other learning techniques. The adpatability of DL to solve from problems of classification, to some others of such as prediction is what have developed an interest in this kind technology with passing of years.

According to Zeng (2008) a standard neural network (NN) consists of many simple, connected processors called neurons, each producing a sequence of real-valued activations. Input neurons get activated through sensors perceiving the environment, other neurons get activated through weighted connections from previously active neurons. Some neurons may influence the environment by triggering actions. Learning or credit assignment is about finding weights that make the NN exhibit desired behavior, such as driving a car. Depending on the problem and how the neurons are connected, such behavior may require long causal chains of computational stages, where each stage transforms (often in a non-linear way) the aggregate activation of the network.

Holden (2014) worked in a project with similar motion data definition, he specified that motion is typically represented as a time-series where each frame represents some pose of a character. Poses of a character are usually parametrized by the character joint angles, or joint positions. Also, he believe that this representation is excellent for data processing, and that valid human motion only exists in a small subspace of this representation.

In this research is provided an analysis of a series of storing techniques for animations that after being parsed by convulutional neural networks will provide a reconstruction of its input. The main format utilized in this research to store the animation is BVH, due the simplicity of use, and the database defined to use this is the CMU. Further details of the implementation and the structure will be explained in Chapter 3, as well as the indicators of comparison between all of them, using as main indicators the possible loss and the MSE validation error between the original and the reconstructions.

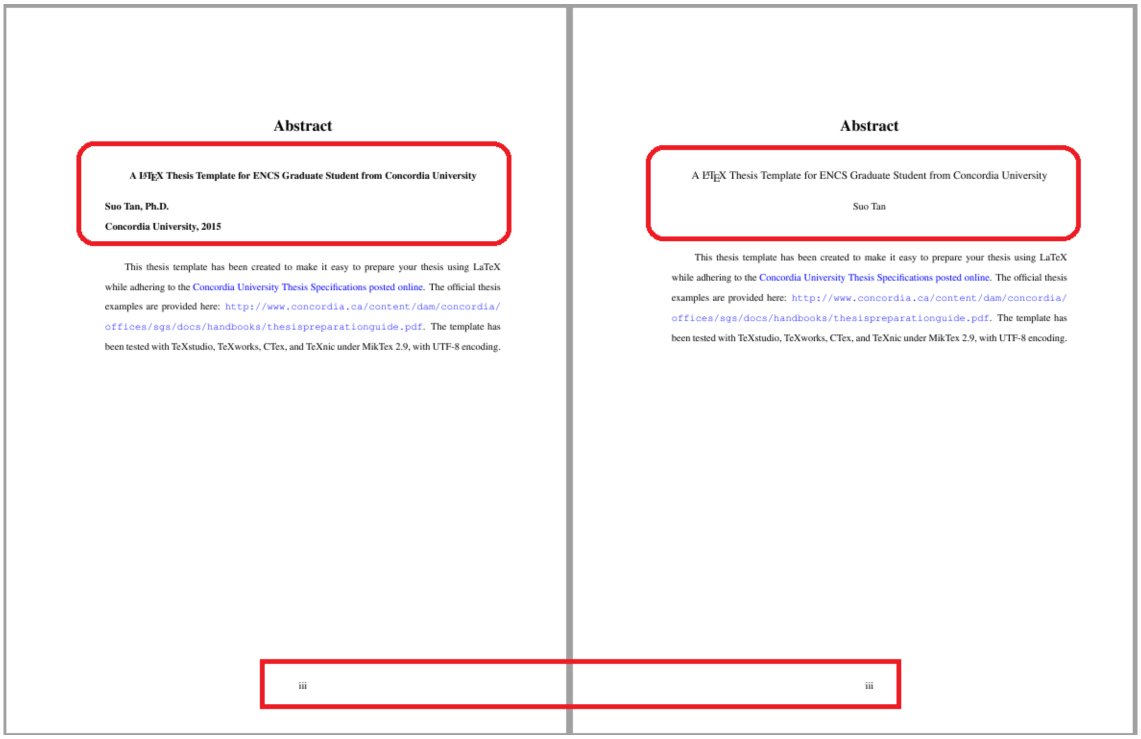
In this research is provided an analysis of a series of storing techniques for animations that after being parsed by convulutional neural networks will provide a reconstruction of its input. The main format utilized in this research to store the animation is BVH, due the simplicity of use, and the database defined to use this is the CMU. Further details of the implementation and the structure will be explained in Chapter 3, as well as the indicators of comparison between all of them, using as main indicators the possible loss and the MSE validation error between the original and the reconstructions.

1.1 Figure and Table

Text body of the Section 1.1.

1.1.1 Figure

A figure example is shown in Figure 1.1.


Abstract

A LaTeX Thesis Template for ENCS Graduate Student from Concordia University

Suo Tan, Ph.D.

Concordia University, 2015

This thesis template has been created to make it easy to prepare your thesis using LaTeX while adhering to the Concordia University Thesis Specifications posted online. The official thesis examples are provided here: http://www.concordia.ca/content/dam/concordia/offices/sgs/docs/handbooks/thesispreparationguide.pdf. The template has been tested with TeXstudio, TeXworks, CTex, and TeXnic under MikTex 2.9, with UTF-8 encoding.

iii

Abstract

A LaTeX Thesis Template for ENCS Graduate Student from Concordia University

Suo Tan

This thesis template has been created to make it easy to prepare your thesis using LaTeX while adhering to the Concordia University Thesis Specifications posted online. The official thesis examples are provided here: http://www.concordia.ca/content/dam/concordia/offices/sgs/docs/handbooks/thesispreparationguide.pdf. The template has been tested with TeXstudio, TeXworks, CTex, and TeXnic under MikTex 2.9, with UTF-8 encoding.

iii

Figure 1.1: An illustration of requirement compliance.

1.1.2 Table

Table 1.1 illustrates a very complex table with figures in its cells.

1.2 Itemized examples using list structures in LaTeX

Item list using "`itemize`" structure are given below:

- Use bold/italic for emphasis, but keep its use to a minimum. Avoid using underlining in your paper.
- Use a consistent spelling style throughout the paper (US or UK).
  - Use double quotes.
  - Use %, not percent.
  - Do not use ampersands (&) except as part of the official name of an organization or company.

Table 1.1: Elements defined for the ROM (Zeng, 2008).

Type		Graphic Representation	Description
Object	Object	$\boxed{\text{O}}$	Everything in the universe is an object
	Compound Object	$\boxed{\boxed{\text{O}}}$	It is an object that includes at least two objects in it
Relation	Constraint Relation	$\bullet\!\!-\boxed{\xi}\!\rightarrow$	It is a descriptive, limiting, or particularizing relation of one object to another
	Connection	$- -\boxed{\iota}- \rightarrow$	It is to connect two objects that do not constrain each other
	Predicate Relation	$-\boxed{\rho}\rightarrow$	It describes an act of an object on another or that describes the states of an object

- Keep hyphenation to a minimum. Do not hyphenate 'coordinate' or 'non' words, such as 'nonlinear'.

The following are using "`enumerate`" structure:

(1) For complete or near complete sentences, begin with a capital letter and end with a full stop.

(2) For short phrases, start with lower case letters and end with semicolons.

1.3 Algorithm

The pseudo code shown in Algorithm 1 describes the proposed algorithm.

Algorithm 1 Calculate the probability of G

Require: $p \in [0, 1]$, G
Ensure: None
for $i = 0 \to 2^d - 1$ **do** ▷ d is an integer
  if $n(\nu_i) = 0$ **then**
    if $x < p$ **then** ▷ x is a normal distribution number in the range of $[0, 1]$
      Occupy v_i site with probability p
    end if
  end if
end for

1.4 Equation

An equation example is shown in Eq. 5.

$$f(ENC) = \int_0^1 (e^x + x^2) \tag{1}$$

1.5 Quotations

> "It was easier in the beginning when there was only the RED-camera, but now, after RED, it just continuous. And all the different manufacturers, they cannot agree upon what is the standard file format, codec, or compression algorithms, and so on. It is a jungle."

CEO, Full Name (Company A)

1.6 Citations

It is suggested that you choose "**citet**" and/or "**citep**" to cite references. The "**citet**{**key**}" gives you a format of "**Name (1990)**", whileas "**citep**{**key**}" delivers a format of "**(Name, 1990)**". For example, Wang and Zeng (2009) extended their research from (Zeng, 2008).

Chapter 2

Literature Review

In this chapter it will be referenced previous literature and related research to this thesis. It's worth noticing that neural networks have been developed as a medium of defining problem solving in a defined fashion similar to the structure of the human brain.

Neural networks have emerged as a ubiquitous model in machine learning, due their flexibility to adapt to very different types of problems, and handling a numerous amount of variables. From prediction models to pattern recognition, image processing, or generative models, NN and DL are techniques that are defined to learn from data, allowing them to do data-driven decisions. Learning from experience, machines therefore can solve problems without the definition of a predefined model, but in doing so, at least with a classical implementation, such as CNN, the definition of the model is then encoded within the a set of neurons, called hidden layer, makes it more difficult to easily interpret or change the model without re-training the neural network with other data that could adapt the output of the model.

For the setup of the neural networks and tools to be used in the implementation the following components were used:

2.0.1 Keras

Keras (2018) is a framework for easy and fast protyping of neural networks that can run in CPU or GPU's, excelent for deep learning. It runs using Python and works as an abstraction layer for another backend suchs as Tensorflow and Theano.

Keras contains numerous functions for building blocks such as layers, objective functions, activation functions, optimizers, and a set of tools to facilitate the work with image and text.

2.0.2 Tensorflow

As defined in their github repository Tensorflow (2018) ”TensorFlow is an open source software library for numerical computation using data flow graphs. The graph nodes represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) that flow between them. This flexible architecture enables you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device without rewriting code. TensorFlow also includes TensorBoard, a data visualization toolkit.”

2.0.3 Autoencoders

One of the several applications of neural networks is the definition of an architecture defined as Autoencoders. Autoencoders are structures that are compromise of at least an enconding layer, a number of hidden layers and a decoder, where the hidden space is decoded as output.

This type of algorithms have several applications from denoising, to reconstruction, and definition of manifolds depending of the input of the raw data provided.

2.0.4 Neural Networks

2.0.5 Convolutional Networks

As Rashid (2016) NN’s emerged from a drive for biologically inspired computers, similar to human brains. Specifically Ian Goodfellow and Courville (2016) mentions that Convolutional Neural Networks (CNN’s), is the more classic configuration for neural networks, that is specialized for processing data that has a known grid-like topology. Examples can be time-series data, prediction series, and image data. The name convolutional neuralnetwork indicates that the network employs a mathematical operation called convolution, that consist of a linear operation over the nodes with a rectifier function on the output of each operation that works as an input to another neuron.

2.0.6 BVH

The Biovision Hierarchy (BVH) character animation file format was developed by Biovision, that have a format with the addition of a hierarchical data structure representing the bones of the skeleton. The BVH file consists of two parts where the first section details the hierarchy and initial pose of the skeleton and the second section describes the channel data for each frame, thus the motion section. It's a widely accepted format that due the simplicity of its structure can be used in several 3d animation software.

2.0.7 Formats of Representation of Animation

Animation can be stored in different representation formats, each with different characteristics such as length of representation, format or complexity of the data stored.

Some of represented items are as follow:

- Axis Angle. This representation system is compromised of 4 values, 3 representing a vector and another additional one representing the angle θ that defines the number of the degrees to rotate such vector.

- Rotation Matrices

  We can consider the following set of matrices, for a given euler angles yaw, pitch and roll, each matrix represent a different rotation over a different axis.

  $$Rz(\alpha) = \begin{pmatrix} cos\alpha & -sin\alpha & 0 \\ sin\alpha & cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

  $$Ry(\beta) = \begin{pmatrix} cos\beta & 0 & sin\beta \\ 0 & 1 & 0 \\ -sin\beta & 0 & cos\beta \end{pmatrix}$$

  $$Rx(\gamma) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos\gamma & -sin\gamma \\ 0 & sin\gamma & cos\gamma \end{pmatrix}$$

Therefore the 3x3 combination of R(α,β,γ) = Rz(α)*Ry(β)*Rz(γ)

represents a rotation under a 3d space.

- Euler Angles

  Using Euler's definition any rotation or sequence of rotations of a rigidbody or coordinate system (x,y,z) about a fixed point is equivalent to a single rotation by a giving angle (θ) about a fixed angle that runs throught a fixed point.

  The main disadvantage of euler angles are the gimbal lock singularities, when two axis get alligned, applying a rotation over a third axis may yield the same transformation as if it was applied over one of the angles previously allgned.

- Quaternions

  Quaternions is a very stable number system that extends complex numbers. The fundamental algebra of quaternions indicate that:

$$i^2 = j^2 = k^2 = ijk = -1 \tag{2}$$

  They have the particular advantage over Euler angles that they avoid the problem of Gimbal lock, by definition they are normalized, and computationally they can be represented via a vector of 4 dimensions, therefore if representing a rotation can be defined that:

$$q = cos(\theta/2) + (x + y + z) * sin(\theta/2) \tag{3}$$

$$q = x + y + z + w \tag{4}$$

$$w = cos(\theta/2) \tag{5}$$

  Where w is the θ rotation angle around the axis of the quaternion.

Chapter 3

Conclusion and Future Work

3.0.1 Summary

A summary of what was found

3.0.2 Contributions

This thesis presented a number of theoretical and practical contributions, including: -THe implelmentations... -

3.0.3 Future Work

Future work

Appendix A

My Appendix

Appendix figure example is shown in A.1 below

Figure A.1: An figure example in Appendix A.

References

Holden, D. (2014). Deep learning in neural networks: An overview. , 75. Retrieved 2018-06-12, from https://arxiv.org/abs/1404.7828

Ian Goodfellow, Y. B., & Courville, A. (2016). *Deep learning.* Retrieved from http://www.deeplearningbook.org (Book in preparation for MIT Press)

Keras. (2018). Why use keras? Retrieved 2018-06-29, from https://keras.io/why-use-keras/

Rashid, T. (2016). *Make your own neural network*. CreateSpace Independent Publishing Platform.

Tensorflow. (2018). Tensorflow github. Retrieved 2018-07-05, from http://github.com/tensorflow/tensorflow

Wang, M., & Zeng, Y. (2009, April). Asking the right questions to elicit product requirements. *International Journal of Computer Integrated Manufacturing*, *22*(4), 283–298. Retrieved from http://dx.doi.org/10.1080/09511920802232902 doi: 10.1080/09511920802232902

Zeng, Y. (2008, August). Recursive object model (ROM) – Modelling of linguistic information in engineering design. *Computers in Industry*, *59*(6), 612–625. Retrieved 2015-10-21, from http://www.sciencedirect.com/science/article/pii/S0166361508000249 doi: 10.1016/j.compind.2008.03.002