**Machine Learning for Video Games**

MEL\*\*629T DISSERTATION

Submitted in partial fulfillment of the requirements of the

M.Tech Data Analytics

by

**P. Sivakishore**

**2017HD12517**

Under the supervision of

**Pratibha Mishra**, **Director**

Dissertation work carried out at

Altran technologies, Bangalore

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE

Pilani (Rajasthan) INDIA

March 2020

DISSERTATION

(**Machine Learning for Video Games**)

Submitted in partial fulfillment of the requirements of the

**M.Tech Data Analytics**

by

**P. Sivakishore**

**2017HD12517**

Under the supervision of

**Pratibha Mishra**, **Director**

Dissertation work carried out at

Altran technologies, Bangalore

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE

PILANI (RAJASTHAN)

(March 2020)

**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI**

**CERTIFICATE**

This is to certify that the Dissertation entitled **Machine Learning for Video Games**

is carried out and submitted by Mr. **P. Sivakishore** ID.No. **2017HD12517** in partial fulfillment of the requirements of MEL ZG629T Dissertation, embodies the work done by him

under my supervision.

Place: Bangalore Signature of the Supervisor

Name: **Pratibha Mishra**

Date: Designation: **Director**

**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI**

**FIRST SEMESTER 2019-2020**

MELZG629T **DISSERTATION**

Dissertation Title : Machine Learning for Video Games

Name of Supervisor : Pratibha Mishra

Name of Student : P Sivakishore

ID No. of Student : 2017HD12517

# Abstract

The issue of Video Game industry is to put lot of manpower to test AAA Games to small Indie Games. Therefore, we intended to aid in testing the game using Machine Learning that involves concepts like Neural Networks and Deep Learning. This can be achieved by training Artificial Intelligence using data set generated while playing the game. Then the data set is processed through deep learning and neural networks to generate datasets. Once the Neural Network dataset is generated, this dataset can be feed to Game player so that it can played by itself for maximum profit. By this we can play the game number of times and look for crash and bugs in the gameplay that can be fixed in development stage and thereby reducing the development cost and manpower required for making the game a GOLD BUILD. From this project, we hope to build an alternative way to test Video Game for Game Industry using Machine Learning. For proof of concept we are developing a Tetris game in python and implement Deep Learning and Neural Network that generates the data set that helps the Game to play itself automatically and helps to reproduce the bugs in the game. The project also involves in concept like serialization and deserialization of data.

**Key Words: Neural Network, Machine Learning, Deep Learning**

# List of Symbols & Abbreviations used

|  |  |
| --- | --- |
| AI | Artificial Intelligence |
| GOLD BUILD | Gaming industry term for final release |
| CNN | Convolution Neural Network |
| ML | Machine Learning |

# List of Figures

[Figure 5.4‑1 Tetris Symbols 9](#_Toc40051832)

[Figure 5.4‑2 Full Game 9](#_Toc40051833)

[Figure 7.1‑1 Program screen to plot state of Tetris 10](#_Toc40051834)

[Figure 7.1‑2 Game state plotted as Game Data 10](#_Toc40051835)

[Figure 7.2‑1 Different rotations of a Tetris symbol 11](#_Toc40051836)

[Figure 7.3‑1 Pseudo CNN Representation 12](#_Toc40051837)

[Figure 7.4‑1 20x10 state image to 20x20 state image 13](#_Toc40051838)

[Figure 7.6‑1 Convolutional layer diagram showing two adjacent layers (left and right) separated by the 18](#_Toc40051839)

# Table of Contents

[2 INTRODUCTION 9](#_Toc40039663)

[2.1 PROBLEM STATEMENT 9](#_Toc40039664)

[2.2 THE EXISTING SYSTEM 9](#_Toc40039665)

[2.3 OBJECTIVE 9](#_Toc40039666)

[2.4 SCOPE OF DISSERTATION 9](#_Toc40039667)

[*3* BUILD TETRIS GAME 10](#_Toc40039668)

[4USE CONVOLUTION NEURAL NETWORK 11](#_Toc40039669)

[4.2 THE INPUT 11](#_Toc40039670)

[4.3 THE OUPUT 12](#_Toc40039671)

[4.4 BUILD CNN MODEL 13](#_Toc40039672)

[4.5 TRAIN DATASET 14](#_Toc40039673)

[4.6 THE GAME DATA 15](#_Toc40039674)

[4.7 TRAIN MODEL 15](#_Toc40039675)

[5CONCLUSIONS 16](#_Toc40039676)

[6DIRECTIONS FOR FUTURE WORK 17](#_Toc40039677)

# INTRODUCTION

## PROBLEM STATEMENT

Game Industry is investing lot of time and money for developing an AI that can play as equivalent to human to make a game efficient. The efficient the game, that much immersive the game would be. Game Testing also involves lot of effort financial and time wise. We need a solution that helps to cut cost the amount of time and money spend on this aspect and same time give the gamer a sub immersive experience while playing the game. This way we can help game industry with huge profit margins, along with faster game development time and testing time. If we take all AAA game, most of the game missing their release data due to lack of a solution to address the problem.

## THE EXISTING SYSTEM

The existing system involves manual algorithm to develop an efficient AI to play against or play with the Human player. The testing process also involves manual testing where the game is played repeatedly. Both are time consuming and cost effective.

## OBJECTIVE

The objective is to provide an industrial solution and implement a proof of concept that provides the solution to the above problem statement.

## SCOPE OF DISSERTATION

Develop proof of concept

1. Develop a Tetris game
2. Develop an AI with cost effective way using CNN

The objective of this dissertation is to explain the way a data science concept like Neural Network can be used on development of Games in testing and creating AI bots to help immersive experience.

The document involves all the steps to build Neural Network to build AI Tetris Bot

# BUILD TETRIS GAME

The Tetris game is developed from scratch using pygame, a python game library. The game

holds a 20x10 NumPy array as state of the board where it uses +ve number and 0 to represent the state of the Tetris, the +ve is existence of block and 0 represents the lack of block in the game. The symbols as shown in below figure are represented in 7 NumPy arrays and each NumPy array contains 4 states of individual symbol as transformation of the symbol as rotation. A random symbol generator is used with some logic behind it to fix drought of symbol generated. A random symbol is dropped from the top of the screen with speed to 1 row per 1 sec. The pygame collision detection method is used to detect the boundaries and other symbol collision detection. Once the collision is detected, the game copies the respective symbol to 20x10 NumPy array. If the 20x10 NumPy array contains +ve number in a row, then the row is removed from the array and new row with zeros is added to the top of the NumPy array. Certain numbers are used to represent the symbol for purpose of coloring the symbol to look good.

|  |  |
| --- | --- |
| Figure ‑ Tetris Symbols | A picture containing clock, ball, meter  Description automatically generated  Figure ‑ Full Game |

# USE CONVOLUTION NEURAL NETWORK

## THE INPUT

If we step back once and get a close look at the game state NumPy 20x10, convert as show in figure 3 and plot using it as MATLAB python library, it represents an image as show in below figure 4. So, we can represent each state of Tetris game into an image 20x10 image as below figure 4. Now if we pass each state of Tetris in below represent and pass it through a CNN and expect as classification of image as an output will result in solving the problem of build an AI Bot for the game.

A screen shot of a computer

Description automatically generated

Figure ‑ Program screen to plot state of Tetris

**A picture containing screenshot

Description automatically generated**

Figure ‑ Game state plotted as Game Data

## THE OUPUT

The output of the CNN would be the best action for give state of game. The output can be representing as an action to be taken for give state of the Tetris game. Generally, a typical Tetris will have a move and rotation input as action to place a symbol in proper location. If can we represent both rotation and move into a single action, we can label that action as label for that particular state image of the Tetris game. So how to generate an action out of rotation and move is as shown in below figure each symbol will have 4 different states and any of this state can be placed in all 11 positions in the game. So, we can represent action as 11x4 44 states. 11 position is due to use of 2d NumPy array for symbols, few symbols use padding to place it correctly.

A screenshot of a cell phone

Description automatically generated

Figure ‑ Different rotations of a Tetris symbol

If we look at above fig, not every symbol has 4 possible rotation, but in the game, we add duplicate rotation to avoid crash in the game if the model predicts wrong rotation.

## BUILD CNN MODEL

For typical image classification, convolution neural network is used to train a neural network. A convolution neural network consists of convolution layer that helps to extract information from the image and reduce the size of reduce without losing the property of the image and flatten the image to generate a predication using linear layer. Below is the representation of CNN used for the project and its configuration and its visual representation.

1. INPUT of 20x20x1 (state of the game at given time)
2. Convolution layer with Filters 32, kernel size = 5, Strides = 5 with tanh activation
3. Convolution layer with Filters 64, kernel size = 3, Strides = 2 with tanh activation
4. Convolution layer with Filters 128, kernel size = 3, Strides = 1 with tanh activation
5. Linear layer with Filters 512 with tanh activation
6. Linear layer with Filters 256 with tanh activation
7. Linear layer with Filters 44 with SoftMax activation

To compile the model, I used the following parameters:

1. Optimizer: Adam with a learning rate of 0.0005
2. Loss Function: Categorical Cross entropy
3. Evaluation Metric: Accuracy

A picture containing text, map

Description automatically generated

Figure ‑ Pseudo CNN Representation

## TRAIN DATASET

To train the model, we modify the input of as the convolution neural networks work better if the input contains 1 and 0. So we convert the 20x10 NumPy array to 1 and 0 from +ve numbers and 0’s. And also, the convolution neural network works better if we provide image with size width and height i.e. 20x20. To achieve this, we need pad 10 more columns to the NumPy array. We add zero which doesn’t affect the input when provided to neural network. The 5 columns of zeros are padded left and right as below image.

A picture containing drawing

Description automatically generated

Figure ‑ 20x10 state image to 20x20 state image

As show in above figure the symbol is copied to the state 20x20 array in different locations. This is separate the symbols for better classification. For train purpose the 20x20 NumPy is serialized along with action for every symbol down and is saved. The action label is predication of 44 action possible in the game. For example, if a rotation of 2 and move of 5 is used to place the symbol in proper place, the action is calculated as move + rot \* 11 i.e. is 5 + 2\*11 = 27 and in array of 27 action at index 27 , the value is saved as 1. This way every symbol drop is serialized and saved as game data.

## THE GAME DATA

We need at least 50000 game data sets to train the model for prefect results. The game is distributed to Aricent employee to play the Tetris game which has an ability to save the serialized data as game data. The game data is collected from them on daily bases.

## TRAIN MODEL

To build CNN, we used TensorFlow library which is a google library for Neural network development. A typical Neural Network trains faster on a setup with Nvidia GPU with CUDA support. But Nvidia GPU with CUDA support are expensive to buy. The alternative way is to buy cloud with GPU support for knew days. But this also cost a bit. The best way is to train the model day by day and save it on daily bases. This way it is cost efficient, but it takes lot of time to train.

# CONCLUSIONS

The goal of this project was to create an **artificial intelligence (AI)** that learns to play **Tetris** using a **convolutional neural network (CNN)**.

It wasn’t an easy task, and the biggest challenge was how to generate a high-quality dataset to train the network for playing Tetris.

Furthermore, to teach the network as many skills as possible, I also needed a large amount of data.

Now I can say that most of the time, I spent not on programming but on collecting and pre-processing data.

In the end, though, I was able to train CNN with a pretty good dataset.

However, there is still no guarantee that the predictions from such a trained model will always be 100% accurate.

Due to the variability of the real world, the network doesn’t know how to handle all situations. It then makes stupid mistakes that ultimately lead to the loss of the game.

Anyhow, the predictions in this game are generally 90% accurate and quite acceptable.

To conclude, with a large and high-quality dataset, we can teach a convolutional neural network to play Tetris pretty well.

# DIRECTIONS FOR FUTURE WORK

The future work on this project would be adding reinforcement learning. This way we can reduction train of data. The Tetris automatically learn from the reward function.

# BIBLIOGRAPHY AND REFERENCES

<https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-1-introduction/>

<https://en.wikipedia.org/wiki/Tetris>

<http://news.mit.edu/2017/explained-neural-networks-deep-learning-0414>

<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

<https://www.tensorflow.org/>

<https://pytorch.org/>

<https://www.youtube.com/watch?v=gZmobeGL0Yg&list=PLZbbT5o_s2xq7LwI2y8_QtvuXZedL6tQU>

<https://www.pygame.org/news>

# APPENDICES

Convolutional layer: it is the core layer in the architecture of CNN. A set of learnable filters (kernels or feature detectors) are convolved with the overlapped subsets of the input across the width and height. The output feature maps (or activation maps) corresponding to the convolutional layer are generated by adding a bias and applying a non-linear activation function (e.g., sigmoid, tanh, rectified linear unit (ReLU), etc.). The detailed expression for the i-th feature map at layer k, denoted as Z k i , can be expressed as: Z k i = J( nXk−1 j=1 w k ji ∗ Z k j − 1 + b k i ), (1) where i and j are, respectively, the feature map index at layer k and layer k−1 , J is the non-linear activation function (ReLU activation function– max(0, x) is applied in this paper), w k ji denotes the kernel shared in the i-th feature map at layer k , and b k i is the corresponding bias of the i-th feature map at layer k. Notice that each feature map shares the same parameterization (weights and bias) for different receptive field, as shown in Fig S2, thus, it can effectively detect features regardless of their position and reduce the number of parameters to be learned in the training stage. The total number of learnable parameters is (h∗h∗r+1)∗nk, and the output size of feature map is denoted by (W − h)/S + 1, where S is the stride (in Fig S2, stride S = 1). The depth for the output feature map is decided by the number of total filters.

A close up of a map

Description automatically generated

Figure ‑1 Convolutional layer diagram showing two adjacent layers (left and right) separated by the

the sketch of the specific convolution process

**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI**

**II SEMESTER 2019- 2020**

**DISSERTATION**

**Dissertation Outline**

**BITS ID No. 2017HD12517 Name of Student: P Sivakishore**

**Name of Supervisor:** Pratibha Mishra

**Designation of Supervisor**: Director

**Qualification and Experience: M. Tech and 20 Years 9 Months**

**E- mail ID of Supervisor:** [pratibha.mishra@altran.com](https://social.intra.aricent.com/People/_layouts/15/EnterpriseProfile/Profile.aspx?accountname=AD\BGH07534)

**Topic of Dissertation**: Machine Learning for Video Games

**Name of First Examiner:** Anil Kumar Jain

**Designation of First Examiner**: Project Manager

**Qualification and Experience: M. Tech and 20 Years**

**E- mail ID of First Examiner:** [anil.jain@altran.com](https://social.intra.aricent.com/People/_layouts/15/EnterpriseProfile/Profile.aspx?accountname=AD\BGH32650)

(Signature of Student) (Signature of Supervisor)

Date:------------ Date:--------------

**Dissertation Outline should contain the following:**

* **Dissertation Topic: Machine Learning for Video Games**
* **Dissertation Title:**
* Design a Neural Network that generates data sets for player inputs to play game automatically.
* **Objectives**:
* Develop a Game from scratch.
* Add serialization of gameplay data to the game
* Generate Datasets by feeding deep learning and Neural networks
* Add functionality to game that can play game by itself from the Neural network generated dataset
* **Scope of work**:
* Understand the Neural Network and Deep Learning concepts to generate a valid game input dataset to play the game automatically
* **Background:**
* Only had background knowledge of game development and python which is used in this project.
* **Methodology**:
* The game is developed in python using game library.

**Detailed Plan of Work** (according to the semester calendar)

|  |  |  |  |
| --- | --- | --- | --- |
| Serial Number of Task | **Tasks or subtasks to be done** (be precise and specific) | Expected date or week of completion | Specific Deliverable in terms of the project |
| 1)  2)  3)  4)  5)  6) | Understanding the concept like Machine Learning and Neural Networks.  Create a git repo and write a game using python  Mid report  Complete game data serialization and finish the training the data set using neural network  Add logic to play game using neural network dataset.  Complete Report preparation | 1-Feb-2020  2-March-2020  11-March-2020  31-march-2020  20 - April-2020  30 – April -2020 | Report.doc  A playable game  Report.doc  Neural Network class in python and executable code  Executable code  Final\_report.doc |

**Supervisor’s Rating of the Technical Quality of this Dissertation Outline**

EXCELLENT / GOOD / FAIR/ POOR (Please specify): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Supervisor’s suggestions and remarks about the outline (if applicable).**

Date\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Signature of Supervisor)

Note: The above 02 pages pertaining to the Dissertation Outline are NOT to be included in the Final Report. They have been included only to make the document a comprehensive one.

**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI**

**WORK INTEGRATED LEARNING PROGRAMMES DIVISION**

**II SEMESTER 2019-2020**

### MEL ZG629T DISSERTATION

**(EC-2 Mid-Semester Progress Evaluation Sheet)**

**NAME OF THE STUDENT:** P. Sivakishore

**ID NO. : 2017HD12517**

**Email Address :** **2017hd12517@wilp.bits-pilani.ac.in**

**NAME OF SUPERVISOR : Pratibha Mishra**

**PROJECT TITLE : Machine Learning for Video Games**

# EVALUATION DETAILS

|  |  |  |  |
| --- | --- | --- | --- |
| **EC No.** | **Component** | **Weightage** | **Marks Awarded** |
| 1 | Dissertation Outline | 10% |  |
| 2. | Mid-Sem Progress  Seminar  Viva  Work Progress | 10%  5%  15% |  |

|  |  |  |
| --- | --- | --- |
|  | **Supervisor** | **Additional Examiner** |
| Name |  |  |
| **Qualification** |  |  |
| **Designation & Address** |  |  |
| **Email Address** |  |  |
| **Signature** |  |  |
| **Date** |  |  |

Note: *Supervisor should announce the Mid-Semester grade to the student directly and send the completed evaluation form (along with the mid- sem report) to the BITS Coordinator on or before the due date.*

*Note: The Mid-Semester Evaluation Form is NOT to be included in the Final Report. This has been included here only to make the document a comprehensive one.*

**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI**

**WORK INTEGRATED LEARNING PROGRAMMES DIVISION**

**II SEMESTER 2019-2020**

### MEL\*\* ZG629T DISSERTATION

**(Final Evaluation Sheet)**

**NAME OF THE STUDENT:**

**ID NO. :**

**Email Address :**

**NAME OF THE SUPERVISOR:**

**PROJECT TITLE :**

***(Please put a tick ( ) mark in the appropriate box)***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.No.** | **Criteria** | **Excellent** | **Good** | **Fair** | **Poor** |
| 1 | Work Progress and Achievements |  |  |  |  |
| 2 | Technical/Professional Competence |  |  |  |  |
| 3 | Documentation and expression |  |  |  |  |
| 4 | Initiative and originality |  |  |  |  |
| 5 | Punctuality |  |  |  |  |
| 6 | Reliability |  |  |  |  |
|  | Recommended Final Grade |  |  |  |  |

# EVALUATION DETAILS

|  |  |  |  |
| --- | --- | --- | --- |
| **EC No.** | **Component** | **Weightage** | **Marks Awarded** |
| 1 | Dissertation Outline | 10% |  |
| 2 | Mid-Sem Progress  Seminar  Viva  Work Progress | 10%  5%  15% |  |
| 3 | Final Seminar/Viva | 20% |  |
| 4 | Final Report | 40% |  |
| Total out of | | 100% |  |

|  |  |  |
| --- | --- | --- |
|  | **Supervisor** | **Additional Examiner** |
| Name |  |  |
| **Qualification** |  |  |
| **Designation & Address** |  |  |
| **Email Address** |  |  |
| **Signature** |  |  |
| **Date** |  |  |

NB: Kindly ensure that recommended final grade is duly indicated in the above evaluation sheet. **POSTAL ADDRESS FOR ALL FUTURE CORRESPONDENCE. FILL IT UP NEATLY IN CAPITAL LETTER WITH PIN CODE ETC.**

**Address: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Pin Code \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

*Note: The Final Evaluation Form should NOT be bound with the report. It has to be submitted separately.*