Replicating Super Mario as a String

Joey Parker

Original Paper

- Title - Super Mario as a String: Platformer Level Generation Via LSTMs
- Authors - Adam Summerville and Michael Mateas
- Year - 2016
- Published at - https://doi.org/10.48550/arXiv.1603.00930

THE PROBLEM
(With human-designed levels)

- Time
- Design Skills
- Creativity
- Time

CHALLENGES
(of generating levels)

Limited Data

Most games have a maximum of hundreds of levels, but even more are far below that.

Long Levels

Each level is thousands of characters long.

Patterns

Levels are made up of complicated design patterns,

MOTIVATION

T

TIME

Level Design takes a significant amount of time.

S

SKILL-INTENSITY

Due to the number of skills needed to design a level, it is difficult to learn.

C

COMPLEXITY

The design of a level is but one tiny part of a massive program.

L

LEARNING

While this project may not directly help any current game development projects, what I learned from this can be used to make impactful models for future projects

Related Works/ Approaches

RELATED WORK/APPROACHES 1

Title: Online Level Generation in Super Mario Bros via Learning Constructive Primitives
Author: Peizhi Shi and Ke Chen
Year: 2016
Published at: 2016 IEEE Conference on Computational Intelligence and Games (CIG)

Approach:

- Learning-based procedural content generation (LBPCG)
- CURE clustering algorithm
- Weighted random forests

Pros:

- High degree of control
- High Playable Rate

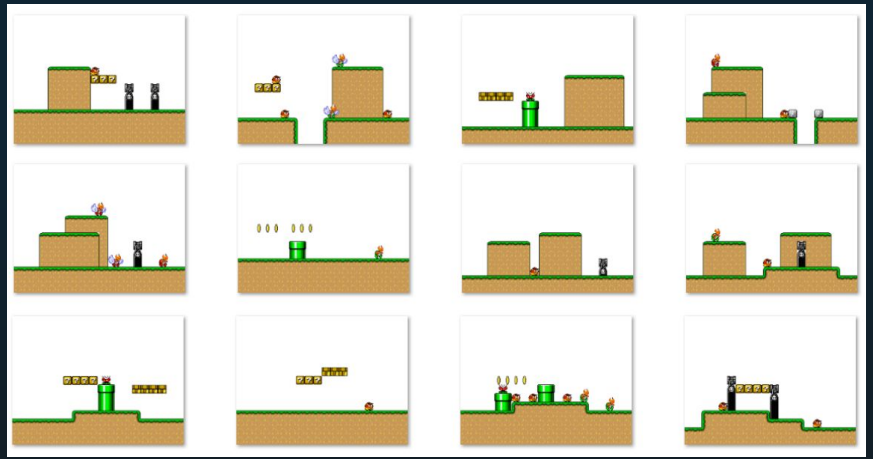


Figure 1:
Sampled level segments, Peizhi Shi and Ke Chen

Cons:

- High degree of manual labor

RELATED WORK/APPROACHES 2

Title: MarioGPT: Open-Ended Text2Level Generation through Large Language Models
Author: Shyam Sudhakaran, Miguel González-Duque, Claire Glanois, Matthias Freiberger, Elias Najarro, and Sebastian Risi
Year: 2023
Published at: arXiv

Approach:

- Transformers
- LLMs
- DistilGPT2
- Repeated Slicing

Pros:

- User-friendly
- High degree of control
- Multiple Paths

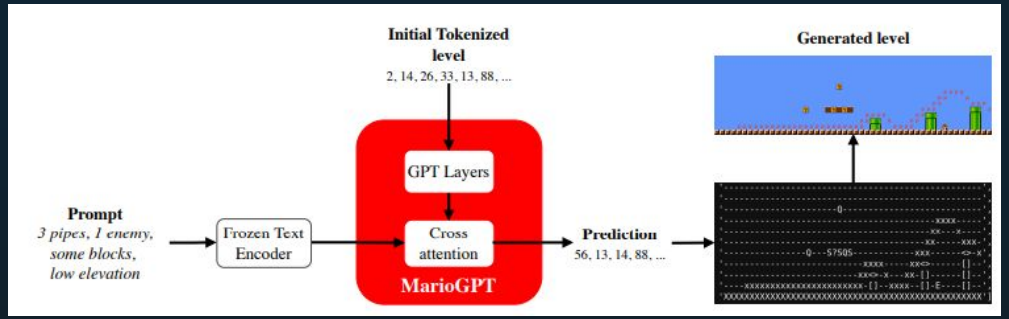


Figure 2:
Sampled level segments, Sudhakaran, Shyam, et al

Cons:

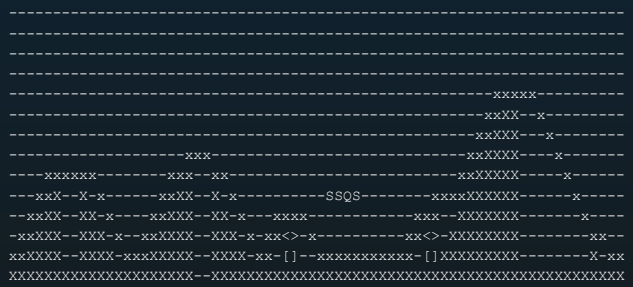
- Tradeoff between uniqueness and quality
- Enemy problems

My Approach/ Method

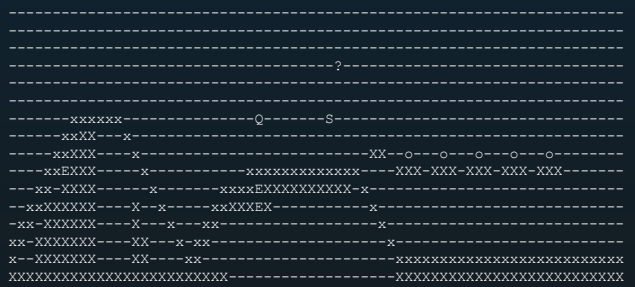
METHOD OVERVIEW

In a single sentence: My method involves using a Long Short-Term recurrent neural network (LSTM) to generate Super Mario Brothers levels from a dataset of original levels

Train on a lot of this



To generate a lot of this



Quick Note on contributions

For my project, I heavily modified Zhihan Yang's attempt of replicating the same paper.

Yang's GitHub:
https://github.com/zhihanyang2022

The Project Itself:
https://github.com/zhihanyang2022/super_mario_as_a_string

What is **Not Original**

- The Original Mario Level Data
- *Most* of the Data Pre-processing
- Creating the seed for level generation
- A tool that turns text levels into pngs

What is **Original**

- Everything else, including:
- The Model
- The Training
- The Generation
- Snaking, Pathing, and Column Depth
- Metric Calculation and Evaluation

Pre-Processing

1

Combine Levels Into One File

2

Modify each Level for Snaking/ Pathing/ Column Depth

3

Get the number of unique characters in the levels

4

Divide each level string into overlapping sequences of 200 characters

5

Split sequences into Inputs and Targets

6

One-Hot Encode the Inputs

Input and Target Sequences

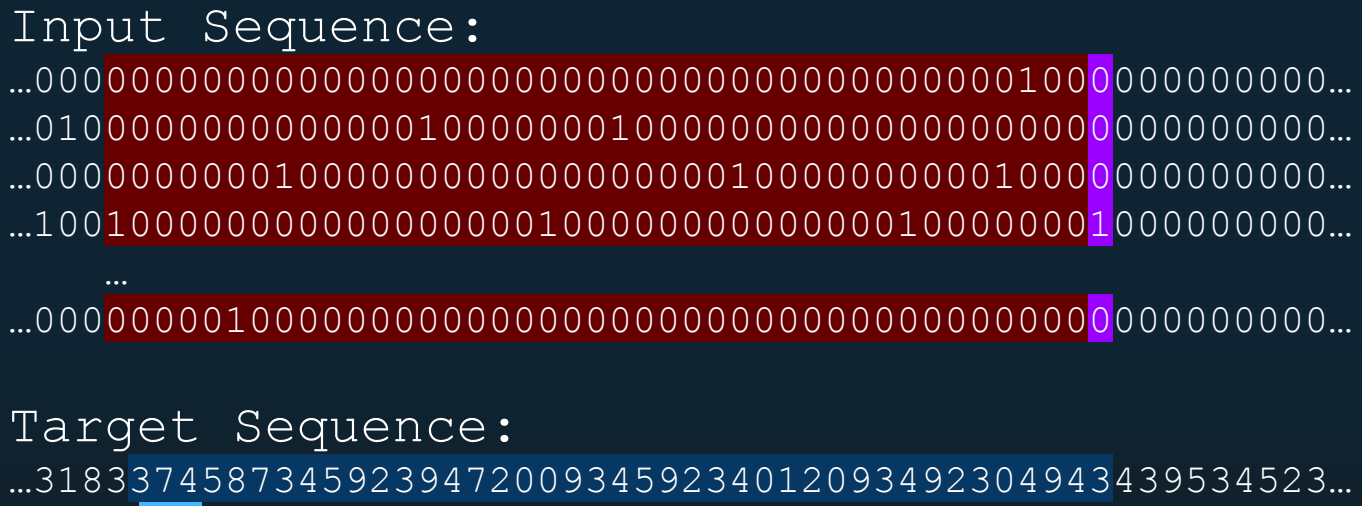
Input Sequence:

…asldhfuoasdufhao8sdufhnlfawoe8iufhapodufhbasdsadfasdfa…

Target Sequence:

…asldhfuoasdufhao8sdufhnlfawoe8iufhapodufhbasdsadfasdfa…

Input and Target Sequences



So what are LSTMs?

LSTMs are a type of artificial neural network designed for sequential data processing. LSTMs address the vanishing gradient problem in traditional recurrent neural networks (RNNs) by introducing specialized memory cells with gating mechanisms. These gates enable LSTMs to selectively store, update, and retrieve information, allowing them to capture long-range dependencies. Effectively, LSTMs create sequences that have good global and local coherence.

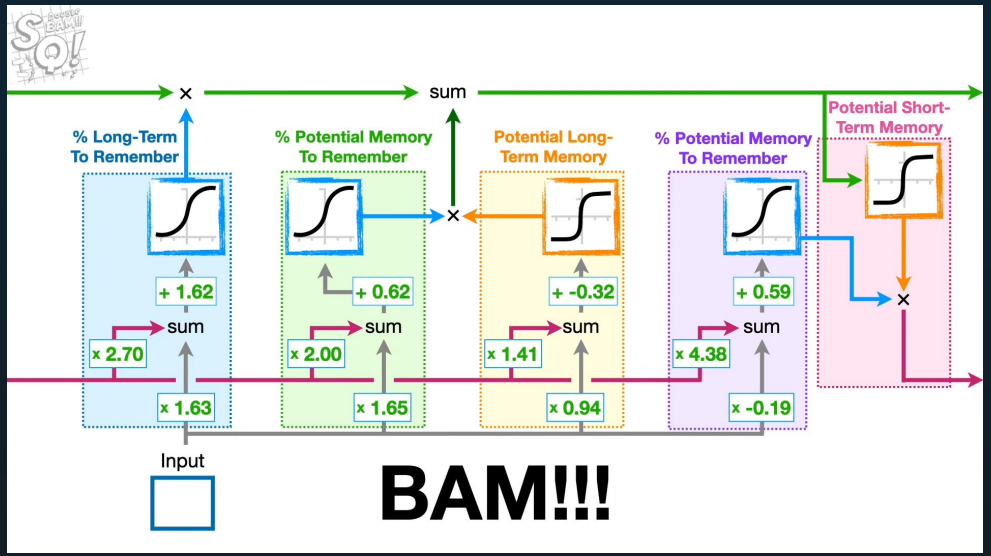


Figure 3:
LSTM Diagram, Josh Starmer

THE MODEL

Model Parameters:

Num_layers = 3

Vocab_size = 14-16

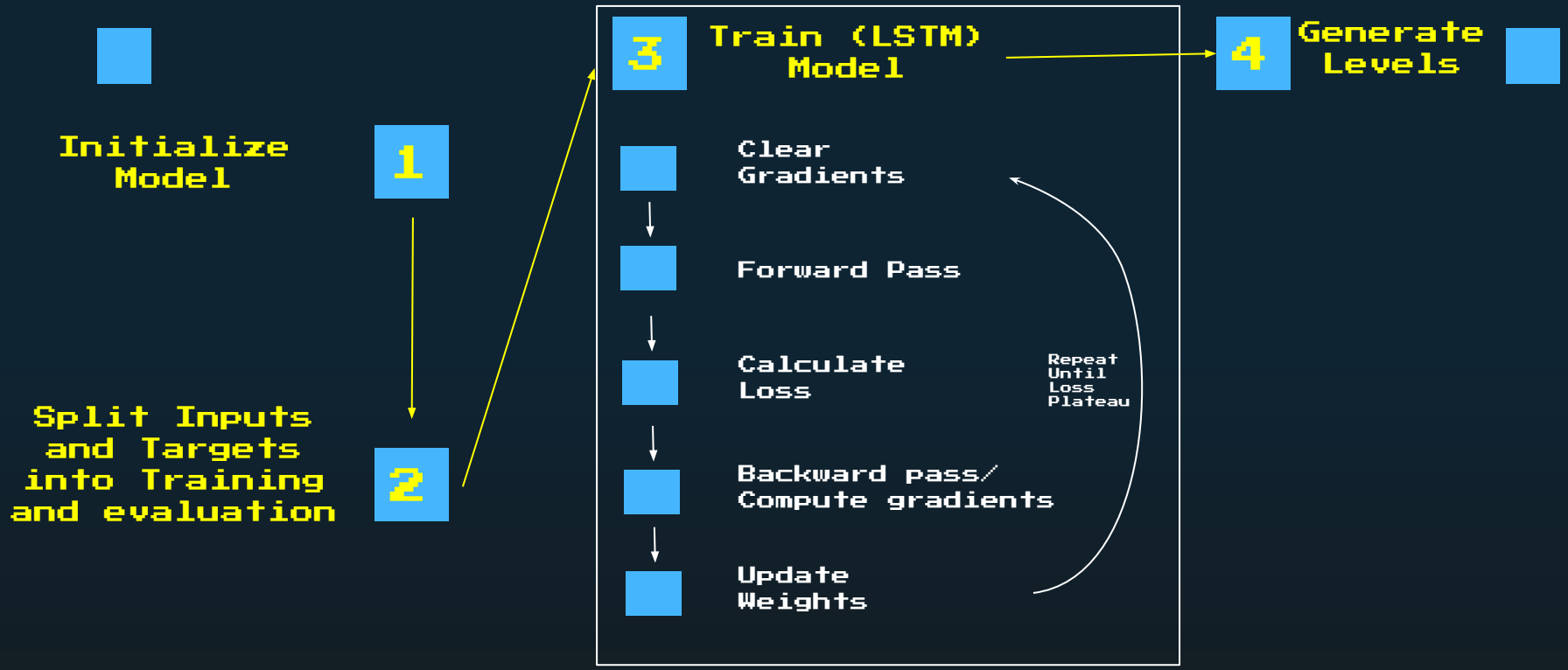
Hidden_size = 516

Dropout = 0.5

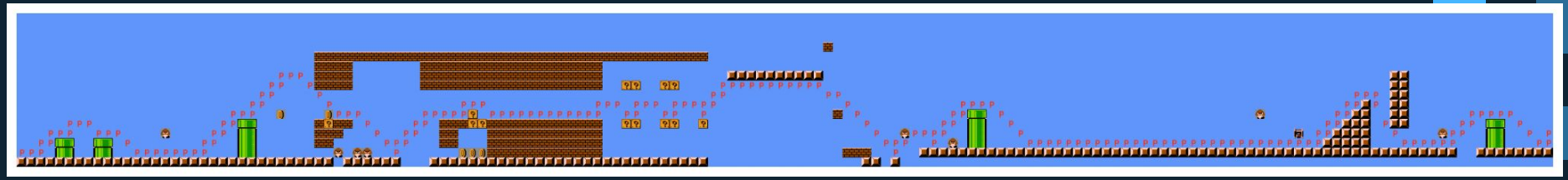
This project's LSTM Model

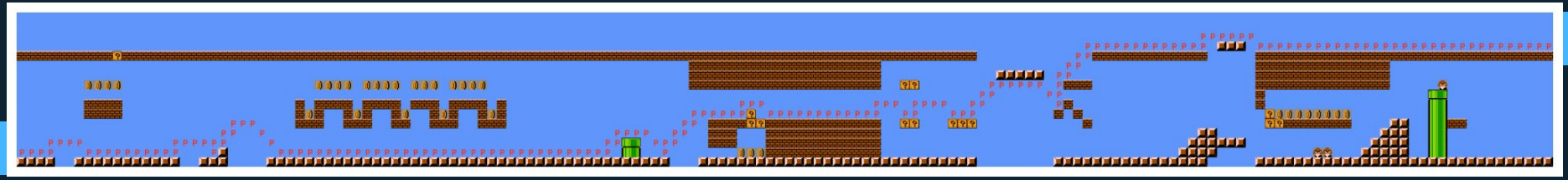
```python
import torch
import torch.nn as nn
import torch.nn.functional as F
class LSTMModel(nn.Module):
    def __init__(self, vocab_size, hidden_size, num_layers, dropout):
        super().__init__()

        self.hidden_size = hidden_size
        self.num_layers = num_layers

        self.lstm = nn.LSTM(vocab_size, hidden_size, num_layers, dropout=dropout, batch_first=True)
        self.fc = nn.Linear(hidden_size, vocab_size)

    #The reason I have the h0 and c0 is so I can predict the next char in a sequence later
    #This isn't used during training
    def forward(self, x, h0=None, c0=None):
        if h0 is None or c0 is None:
            h0 = torch.zeros(self.num_layers, x.size(0), self.hidden_size).requires_grad_().to(x.device)
            c0 = torch.zeros(self.num_layers, x.size(0), self.hidden_size).requires_grad_().to(x.device)

        #We need to detach as we are doing truncated backpropagation through time (BPTT)
        #If we don't, we'll backprop all the way to the start even after going through another batch
        out, (hn, cn) = self.lstm(x, (h0.detach(), c0.detach()))

        # Index hidden state of last time step
        out = self.fc(out)

        return out, (hn,cn)
```

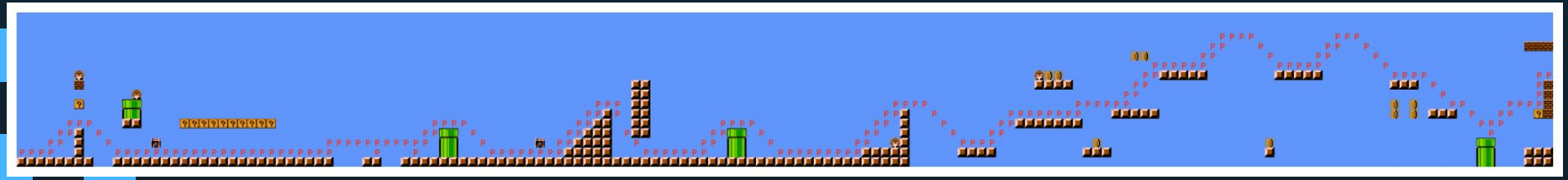
Approach Steps

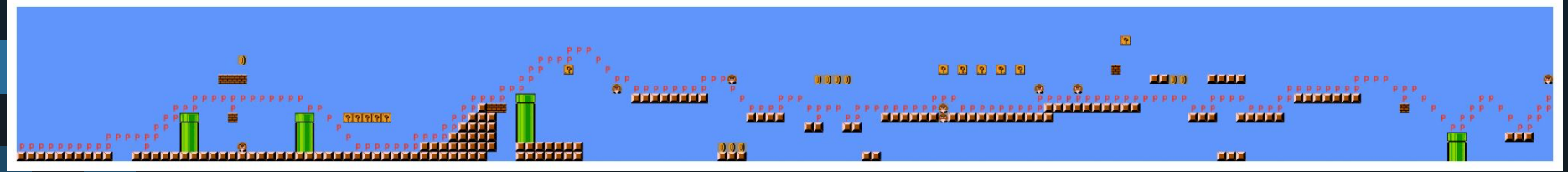


Generated Levels Results



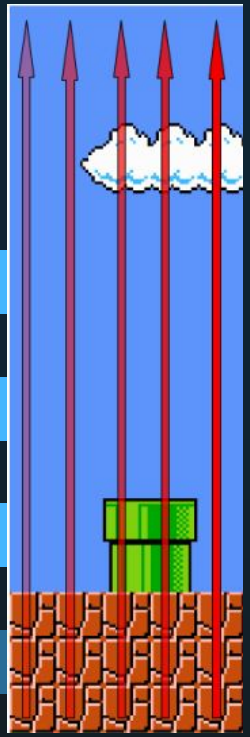




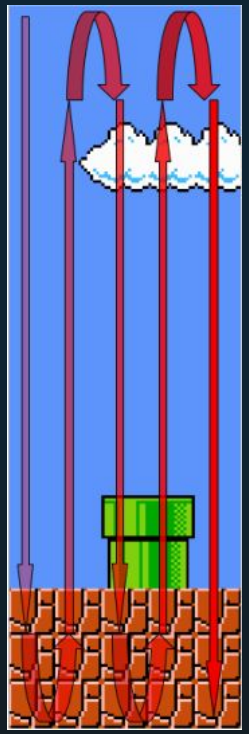


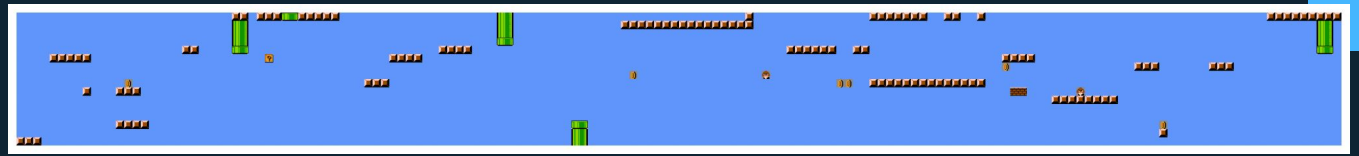
Snaking Results

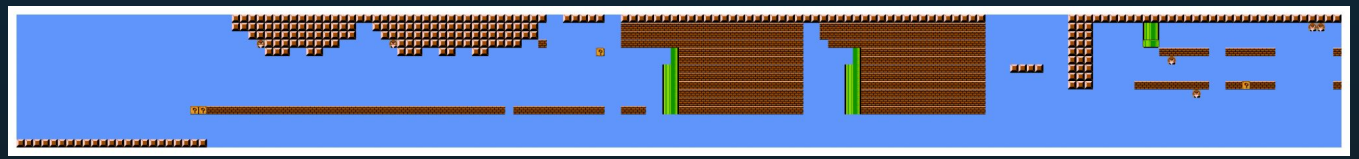
Normal



Snaking







Main Observations

- The LSTM model can successfully generate playable levels at a reasonable rate
- Snaking the input is **Not** recommended, as it can cause the levels to flip around and create strange artifacting
- It takes an average of 6.5 seconds to generate each level. This means the model successfully fixes our problem of time taken to design and build levels

Closing Remarks

Thank you for your time and attention.

The Project Github can be found at
https://github.com/NinjajkI/ITCS5156Project

Suggestion for Future Work

- Test different hyperparameter combinations
- Attempt using a LLM instead of a LSTM
- Expand dataset to include more original mario levels

References

Shi, Peizhi, and Ke Chen. “Online level generation in Super Mario Bros via learning constructive primitives.” *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, Sept. 2016, https://doi.org/10.1109/cig.2016.7860397.

Sudhakaran, Shyam, et al. “MarioGPT: Open-Ended Text2Level Generation through Large Language Models”, *arXiv*, Feb. 2023, https://doi.org/10.48550/arXiv.2302.05981.

Summerville, Adam, and Michael Mateas. “Super Mario as a String: Platformer Level Generation Via LSTMs”, *arXiv*, Nov. 2016, https://doi.org/10.48550/arXiv.1603.00930.

Starmer, J. (2023, January 24). Long short-term memory with pytorch + lightning. YouTube. https://www.youtube.com/watch?v=RHGiXPuo_pI