Controllable Level Blending between Games using Variational Autoencoders

Anurag Sarkar¹, Zhihan Yang² and Seth Cooper¹

¹Northeastern University
²Carleton College
(Towards) Controllable Level Blending between Games using Variational Autoencoders

Anurag Sarkar\textsuperscript{1}, Zhihan Yang\textsuperscript{2} and Seth Cooper\textsuperscript{1}

\textsuperscript{1}Northeastern University
\textsuperscript{2}Carleton College
(Towards) Controllable Level Blending between Games using Variational Autoencoders

Still no playability!

Promising results and future directions!
Motivation

- Past work on training models on existing levels to generate new levels
  - Sequence prediction using LSTMs
  - Conceptual blending using graphical models

Summerville and Mateas, 2016

Guzdial and Riedl, 2016
Motivation

• Past work on training models on existing levels to generate new levels
  • Sequence prediction using LSTMs
  • Conceptual blending using graphical models

• Gow and Corneli proposed generating new games by blending entire games

VGDL Frogger + VGDL Zelda = Frolda
Motivation

• Past work on training models on existing levels to generate new levels
  • Sequence prediction using LSTMs
  • Conceptual blending using graphical models
• Gow and Corneli proposed generating new games by blending entire games

IDEA: PCGML techniques + Game Blending
Blending Levels using LSTMs

- Trained LSTMs on levels of *Super Mario Bros.* and *Kid Icarus*

- Sampled from trained models to generate levels containing properties of both games

- Parametrized generator with weights to control approximate percentage of each game in blended level

(SMB=0.2, KI=0.8)

(SMB=0.8, KI=0.2)
Drawbacks

• Blended levels by taking turns between *Super Mario Bros.* and *Kid Icarus*

• Allowed control of proportion of each game in blended level but no control over more fine-grained tile-based properties
Solution: Variational Autoencoder (VAE)

• Enables more holistic blending of level properties by capturing latent space across both games

• Allows generation of segments satisfying specific properties

• More conducive to co-creative level design
Variational Autoencoder

- Autoencoders are neural nets that learn lower-dimensional data representations
  - Encoder $\rightarrow$ input data to latent space
  - Decoder $\rightarrow$ latent space to reconstructed data

Vanilla Autoencoder
Variational Autoencoder

- Autoencoders are neural nets that learn lower-dimensional data representations
  - Encoder $\rightarrow$ input data to latent space
  - Decoder $\rightarrow$ latent space to reconstructed data

- VAEs make latent space model a probability distribution (e.g. Gaussian)
  - Allows learning continuous latent spaces
  - Enables generative abilities similar to those of GANs
Motivation for VAE

• Past work in using autoencoders for Mario level generation
  • Autoencoders for Level Generation, Repair and Recognition, Jain et al. (2016)
  • Explainable PCGML via Design Patterns, Guzdial et al. (2018)

(a) Original  (b) Unplayable  (c) Repaired

Guzdial et al. (2016)

Guzdial et al. (2018)
Motivation for VAE

• Past work in using autoencoders for Mario level generation
  • Autoencoders for Level Generation, Repair and Recognition, Jain et al. (2016)
  • Explainable PCGML via Design Patterns, Guzdial et al. (2018)

• Evolving Mario Levels in the Latent Space of a DCGAN (i.e. MarioGAN), Volz et al. (2018)
Motivation for VAE

• Past work in using autoencoders for Mario level generation
  • Autoencoders for Level Generation, Repair and Recognition, Jain et al. (2016)
  • Explainable PCGML via Design Patterns, Guzdial et al. (2018)

• Evolving Mario Levels in the Latent Space of a DCGAN (i.e. MarioGAN), Volz et al. (2018)

• Use MarioGAN-based approach to capture latent space of 2 games instead of 1

Volz et al. (2018)
Why VAE over GAN?

- VAE architecture more conducive to co-creative level design
  - Designers don’t have to directly use latent space vectors
  - More explicit control in defining inputs to the system
  - More useful to blend/interpolate between known segments rather than latent vectors
VAE vs GAN vs VAE-GAN

- Trained a GAN and a VAE-GAN in addition to the VAE to compare generative capabilities in a level blending context.

- VAE-GAN is a hybrid generative model
  - Combines VAE and GAN by collapsing VAE decoder into a GAN generator.
Dataset and Training

- Trained on a level each from SMB (Level 1-1) and KI (Level 5) taken from the Video Game Level Corpus (VGLC)

- Each level is a 2D character array

- Each tile type was encoded using an integer and then with one-hot encoding for training
Dataset and Training

• To account for orientation, used 16x16 sliding window

• 187 segments of SMB + 191 segments of KL = 378 total segments

• Models learned to generate 16x16 blended level segments

• VAE, GAN and VAE-GAN all trained using same number of segments and with similar training conditions
Generation

• Trained models generate 16x16 segments in combined SMB-KI latent level design space

• Generation involves feeding a latent vector into the VAE’s decoder which outputs a one-hot encoded array which is converted to the 16x16 level segment

• Two generation methods
  • Like GANs, use random latent vectors or evolve optimal vectors using search
  • Unlike GANs, generate segments based on input segments
Evaluation

• Used four metrics for evaluation
  • Density
  • Difficulty
  • Non-Linearity
  • SMB Proportion
Evaluation

- Used four metrics for evaluation
  - Density
  - Difficulty
  - Non-Linearity
  - SMB Proportion

- Compared generative performance of VAE with that of GAN and VAE-GAN
  - How well models capture latent space spanning both games → computed above metrics for 10K random latent vectors
  - Accuracy of evolving desired segments using CMA-ES → evolved 100 segments with target values of 0%, 25%, 50%, 75%, 100% for each metric
• VAE does best at generating segments that are a mix of either game while GAN and VAE-GAN generate segments with mostly SMB or mostly KI elements
Results

• VAE does best at generating segments that are a mix of either game while GAN and VAE-GAN generate segment with mostly SMB or mostly KI elements

• VAE is better at capturing the latent space spanning both games as well as the space in between
  • 18% of VAE segments have elements of both games
  • 8% for GAN
  • 5% for VAE-GAN
Results

- GAN does better than VAE only for 100% Density and 75% and 100% Difficulty
Results

• GAN does better than VAE only for 100% Density and 75% and 100% Difficulty

• Ignore structures in training levels since actual segments would not be 100% solid nor have 16 enemies and hazards
Results

• No model does particularly well in blending desired SMB and KI proportions but VAE does well for the 50% case

• With similar training, VAE learns a latent space that is more representative while having more variation to enable better blending
Application in Co-Creative Design
Application in Co-Creative Design

- Interpolation between games
Application in Co-Creative Design

- Alternate connections between segments
Application in Co-Creative Design

- Generating segments satisfying specific properties

KI Hazards  SMB ?-Marks  SMB Enemies  KI Doors  KI Platforms
Application in Co-Creative Design

- Generating segments with desired proportions of different games
Future Work

• Playability
Future Work

• Playability

• Vector math in level design space
Future Work

- Playability
- Vector math in level design space
- Co-Creative Level Design Tool
Future Work

- Playability
- Vector math in level design space
- Co-Creative Level Design Tool
- Multiple Games and Genres
Future Work

• Playability

• Vector math in level design space

• Co-Creative Level Design Tool

• Multiple Games and Genres

Contact

Anurag Sarkar
Northeastern University
sarkar.an@husky.neu.edu