Generating Video from Images

Geoffrey Penington, Mae Teo, Chao Wang
geoffp, maehwee, cwang15@stanford.edu

Introduction

- The problem: Generate realistic videos from a given initial and final image.
- The challenge: Dimension of space of possible videos is frames × height × width × colors. Very large even for short, low-resolution videos.
- Approaches: Conditional variational autoencoders (CVAEs) and conditional generative adversarial networks (CGANs), using deep convolutional neural networks (CNNs).

Dataset

- 'Moments in Time' [1]: 1,000,000 short videos, divided into 339 categories.
- Selected two categories: “Erupting” and “Skiing”.
- Downsized to 30 frames, 64 × 64 pixels each frame.

CVAE

- Training Objective:
  \[
  \min \mathbb{E}_{x \sim p(x)} [\log D(x; w_D)] + \mathbb{E}_{z \sim p(z)} [\log (1 - D(G(z; w_G); w_D))] + \mathbb{E}_{x \sim p(x)} [\lambda ||c(G(z; w_G)) - c(x)||_2^2]
  \]
  \[
  \text{total } = \text{video } + \text{last frame } + \text{first frame } = 600 + 1000 + 1000
  \]

Baseline Model

- Linear interpolation between initial and frame image.

CGAN

- Training Objective:
  \[
  \min \max_{w_G, w_D} \mathbb{E}_{x \sim p(x)} [\log D(x; w_D)] + \mathbb{E}_{z \sim p(z)} [\log (1 - D(G(z; w_G); w_D))] + \mathbb{E}_{x \sim p(x)} [\lambda ||c(G(z; w_G)) - c(x)||_2^2]
  \]

Results (See our live demos!)

- Used Adam optimizer. Used “one-sided label smoothing” for CGAN.
- For “erupting”: CVAE trained on 2000 videos, CGAN trained on 120 videos.
- For “skiing”: CVAE trained on 1800 videos.
- Qualitative metric: Look at Videos!
- Quantitative metric: \( L_2 \) distance between generated and real videos (test set of 100 videos)

<table>
<thead>
<tr>
<th></th>
<th>( L_2 ) distance</th>
<th>Interpolation</th>
<th>CVAE</th>
<th>CGAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Erupting</td>
<td>62 ± 58</td>
<td>180 ± 110</td>
<td>223 ± 96</td>
<td></td>
</tr>
<tr>
<td>Skiing</td>
<td>153 ± 89</td>
<td>197 ± 80</td>
<td>N/A</td>
<td></td>
</tr>
</tbody>
</table>

Future

- More time and GPU resources to train on larger datasets, and more carefully tune the hyperparameters.
- Use deeper and more complex architectures, e.g. recurrent neural networks (RNNs).

References