Generating Bulbasaurs using DDPM

Project 4 - Statistical deep learning MT7042

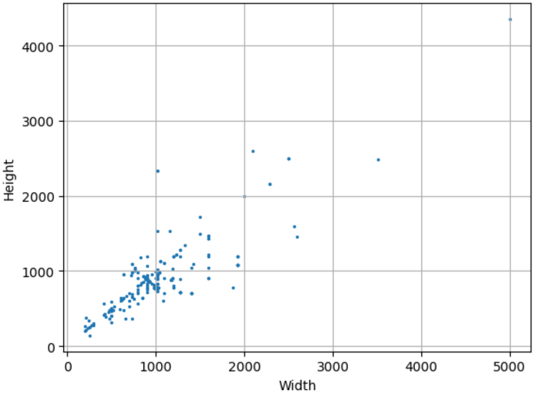
Florence Hugh & August Jonasson

2025-01-17

Introduction

Exploratory analysis

- $\approx$ 200 images of varying resolution and #channels
- pre-processing required
- data augumentation



Decay of the forward process

Transition density of the forward process

$$q(x_t|x_0) := \mathcal{N}(x_t; \sqrt{\overline{\alpha}_t}x_0, \sqrt{1-\overline{\alpha}_t}\,\mathrm{I}),$$

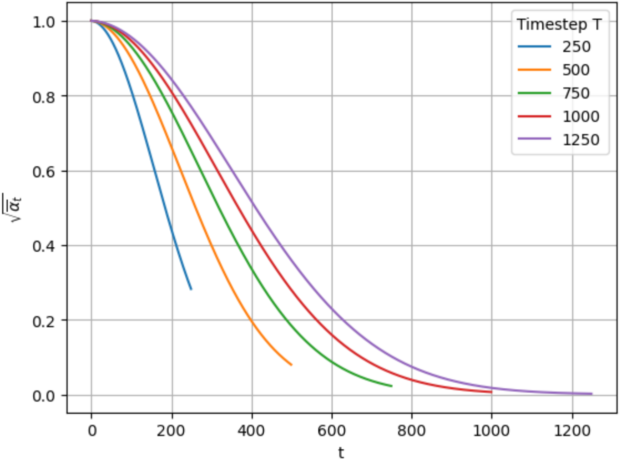
where x_t is the image at timestep t, x_0 is the starting image, and

$$\overline{\alpha}_t = \prod_{i=1}^{t} \alpha_i.$$

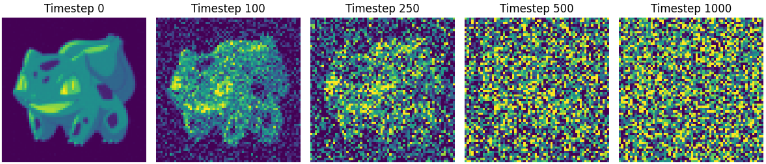
- What we need: $q(x_T|x_0) \rightarrow \mathcal{N}(x_T; 0, I), \quad \text{as} \quad T \rightarrow \infty$
- Crucial to control the decay of $\sqrt{\overline{\alpha}_t}$
  - by choosing $\{\alpha_t\}$ and T

Decay of the forward process

With linearly decaying $\{\alpha_t\}$, the decay of $\sqrt{\overline{\alpha}_t}$ for different T

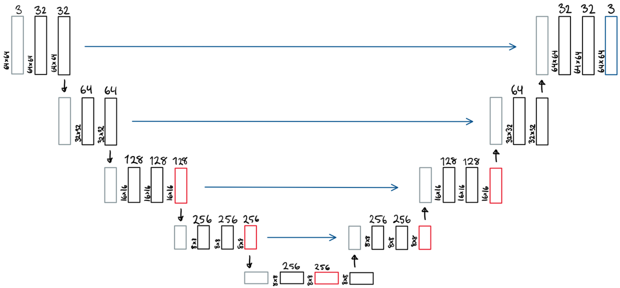


Forward noising process applied to a pre-processed image



Architecture

- U-Net
  - Capable of handling image segmentation



Optimization problem

- Equivalent to a least-squares regression problem
  - Response: Random generated noise ϵ
  - Predictor: Time-step t & image x_t
- Hence, minimization of

$$||\epsilon - \epsilon_\theta(\sqrt{\overline{\alpha}_t}x_0 + \sqrt{1-\overline{\alpha}_t}\epsilon, t)||_2^2, \quad \forall t \in \{1, \ldots, T\}.$$

Generating images

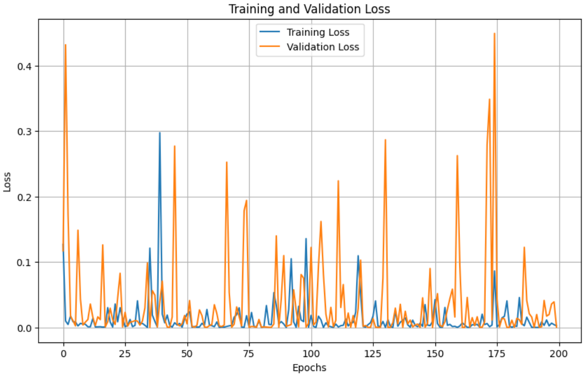
- 2000 epochs with Adam optimizer and MSE loss function

Generating images

- 2000 epochs with Adam optimizer and MSE loss function

Examining the training

- 80/20 split, different from the model that generated images
- What do we expect a healthy training to look like?



Alternative methods of validation

- Plausible vs novelty
- Quality of generated images

Fréchet inception distance

$$d_F(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\mu', \Sigma'))^2 = ||\mu - \mu'||_2^2 + \mathrm{tr}\left(\Sigma + \Sigma' - 2(\Sigma\Sigma')^{\frac{1}{2}}\right),$$

Where $\mathcal{N}(\mu, \Sigma)$ is the distribution of the original image set and $\mathcal{N}(\mu', \Sigma')$ of the generated images.

Possible improvements

- Monitoring the training
  - 90/10 split
  - Leave-one-out cross-validation
- How to handle overfitting
  - Add data augumentation
  - Dropout
  - L_1 (Lasso) and L_2 (Ridge) penalty