

Generating Hands with Reinforcement Learning

Written by Chase Mutzig

February 3, 2024

Abstract

From their inception, AI has been able to do incredible things, from modern art, to web searching and creation, to faking an entire person into existence. However, something that has always been difficult for machines are hands, specifically of the human variety. Always too many fingers, too many segments, or no hand at all, it is incredibly rare that a hand is properly formed via a machine. In this paper, we will continue an experiment that used a variational auto encoder to generate hands by adding reinforcement learning. The goal is to generate a human hand that is anatomically accurate 70-80% of the time. To do this, we will use a dataset of around 11k hands in specific orientations and positions and feed them to the modified neural network. This paper will summarize the end results gained from using a VAE, explain how reinforcement learning is projected to help accuracy, and the expected implementation of reinforcement learning to the VAE.

Introduction

AI mimicry in the form of vocals, photos, and videos is growing in popularity. With this growth in popularity, it is also becoming increasingly more difficult to differentiate between what is fake and real. Take a person's face as an example. In the early generations, an AI would have all the parts of the face; eyes, nose, ears, eyebrows, mouth, etc. However, these parts may have had mismatched sizes, locations, or other impossibilities (Kalpokas and Kalpokiene, 2022). In current times, these mistakes have been ironed out and it can often be very difficult to spot the malformed details. If you take a human or any humanoid figure, one of the most reliably messed up details would be the hands(Meg, 2023).

Hands are typically attached via a wrist, have a palm, and five digits. Each of these digits are made up of sections that have fingernails, fingerprints, and joints. In AI generated material, there are often incorrect numbers of such details. Currently, the main issue relates to quantity of features, rather than quality of features, as many current generators are able to make good quality and high fidelity images. In an effort to correctly generate hands 70-80% of the time, we are going to be modifying a neural network specialized in focusing on these easy-to-miss details by adding reinforcement learning. The reinforcement learning agent will work to improve the baseline understanding of the neural network in an effort to improve the results enough as to reach the 70-80% accuracy goal.

Background

To summarize the preceding experiments, a variational auto encoder is a neural network structure that forms a latent understanding of the input, then tries to recreate the input from the created understanding. In this case, the input is roughly 11,000 hands all in similar positions, and the output is a *strictly new* hand that was not previously in the input data. This newly generated hand should be visibly recognizable as a hand, and should be anatomically correct. This means that the hand should have a palm, four fingers, and a thumb. Each digit also must have the correct amount of segments, if visible.

A variational auto encoder was chosen specifically due to the nature of this task. VAEs generate probabilistic latent spaces, allowing them to generate many different samples and are thus well suited for tasks where we want to interpolate between different data; in this case, the skin tones, positions, and structure of the input hands(Pu, Y, et. al. 2016).

A standard auto encoder has a deterministic latent space, meaning they do not capture variations in the input data well(Bank, et. al., 2023).

As for GANs, or generative adversarial networks, they implicitly learn a latent space through a generator network which may not be as understandable or controllable as a VAE latent space (El-Kaddoury, et. al., 2019).

Previous Results

The results from the VAE section will be considered the preceding and baseline results for this experiment.



Figure. 1: Best baseline results from pure VAE training.

The minimum goal will be to generate results better than

these preceding results, and the overall goal will be to generate anatomically correct hands 70-80% of the time. To clarify, if a batch of 20 hands are created, then roughly 15 of those hands must be anatomically correct.

Related Work

Specifically with references to hands, there has been work mainly on reading hand motions and gestures. In one such case, Stanford University has made “[a] new AI learning scheme combined with a spray-on smart skin [that] can decipher the movements of human hands to recognize typing, sign language, and even the shape of simple familiar objects” (Patel, 2023).

In another similar project, people are using AI to learn and read gestures in an effort to make a hands-free, no external devices blackboard, similar in style to Iron-Man’s holo-desk. By reading hand gestures, a person can write numbers and letters or create more blackboards, lead meetings, and many other applications (Soroni et al., 2021).

Lastly, although there are many more examples of AI using hand reading, there is a physics simulation that tries to map hand movements to a virtual environment, similar to the aforementioned project. However, this project also attempts *accurate* haptic feedback by using reinforcement and imitation learning. This means they don’t want any clipping of fingers into objects, and for the person doing the manipulation to feel the opposing forces physically, from virtual objects (Garcia-Hernando, et. al., 2021).

Methodology

For this experiment, we want to change the latent space of the VAE by adjusting latent variables to improve the understanding of what a hand is, and thus improve the generated hands.

Due to its suitability, the Soft Actor-Critic (SAC) algorithm will be used. SAC is an algorithm that is well suited for continuous action spaces, which the RL agent will be dealing with since the base of this model is a VAE that generates a continuous latent space (Haarnoja, et. al., 2019). Continuing with the SAC algorithm, the agent must also have clearly defined states, actions, and rewards.

The states represent information about the environment that the agent will use to make decisions regarding any available actions it may take. Each state will be the collection of latent variables that encode information on the latent space, or understanding of the model. There will be one state per episode.

The actions will be adjustments and transformations applied to the latent variables in the VAE latent space. The agent will use SAC to learn a policy that maps the states to the actions, allowing it to modify the latent variables and hopefully improve the latent understanding of what a hand may be.

The reward for the RL agent will be based on the change in loss after each training epoch. If the loss decreases between epochs, then the agent is rewarded a positive reward in proportion to the change. If the loss instead increases, the

proportional reward will be negative. The algorithm used is as follows:

$R_t = -\lambda * \Delta L_t$ if $t > 1$, where R_t is the reward at epoch t , ΔL_t is the change in loss from the previous epoch, defined as $L_t = L_t - L_{t-1}$, and λ is a positive constant scaling factor. The use of λ allows for the control of reward scaling, making rewards scale more or less depending on the loss change. λ is not strictly necessary for this algorithm to work, yet is included due to the flexibility in experimentation it provides.

SAC Algorithm

The Soft Actor-Critic algorithm has four main parts, each of which are responsible for a major portion of what makes the SAC algorithm good (Derman, et. al., 2018).

Critic Update The critic update evaluates the quality of the actions taken by the agent. This update ensures an accurate estimation of the cumulative reward, incorporating the reward, discounted future rewards, and the entropy-adjusted policy. The critic update is given as the following:

$$\mathcal{L}_Q(\phi) = E_{(s,a,r,s') \sim \mathcal{D}} \left[\frac{1}{2} (Q_\phi(s, a) - \right]$$

$$(r + \gamma \min_{a'} Q_{\phi'}(s', a') - \alpha \log \pi_\psi(a'|s'))^2 \quad (1)$$

- s : Current state
- s' : Next state
- a : Taken action
- a' : Next taken action
- r : Reward
- \mathcal{D} : Data distribution representing experiences collected from the environment
- γ : Discount factor for future rewards
- α : Temperature parameter
- $\pi_\psi(a' | s')$: Policy representing the probability of taking action a' in state s'

Value Function Update The value function update is responsible for estimating the cumulative reward and aligning the policy’s entropy-adjusted distribution with a target entropy. This encourages a balance between trying new things (exploration) or doing something already done (exploitation), leading to diverse yet high-quality policies. The VFU is given as:

$$\mathcal{L}_V(\theta) = E_{s \sim \mathcal{D}} \left[\text{KL} \left[\pi_\psi(\cdot | s) \middle\| \exp \left(\frac{Q_\phi(s, \cdot)}{\alpha} \right) \right] \right] \quad (2)$$

- $\mathcal{L}_V(\theta)$: Loss function for the value function update.
- $E_{s \sim \mathcal{D}}$: Expectation over the data distribution \mathcal{D}
- KL: Kullback-Leibler divergence
- $\pi_\psi(\cdot | s)$: Policy representing the probability distribution of actions given state s
- $Q_\phi(s, \cdot)$: Q-function representing the expected cumulative reward given state s and various actions
- α : Entropy temperature parameter

Policy Update The policy update updates the policy to maximize the expected reward along with the entropy term. Doing this encourages the agent to explore its environment and to take actions that yield high rewards. The policy update is given as:

$$\mathcal{L}_\pi(\psi) = E_{s \sim \mathcal{D}} \left[E_{a \sim \pi_\psi(\cdot|s)} \left[\alpha \log(\pi_\psi(a|s)) - Q_\phi(s, a) \right] \right] \quad (3)$$

- $\mathcal{L}_\pi(\psi)$: Loss function for the policy update
- $E_{s \sim \mathcal{D}}$: Expectation over the data distribution \mathcal{D}
- $E_{a \sim \pi_\psi(\cdot|s)}$: Expectation over the policy's action distribution given state s .
- α : Entropy temperature parameter
- $\log(\pi_\psi(a|s))$: Log probability of the action a given state s under the policy
- $Q_\phi(s, a)$: Q-function representing the expected cumulative reward given state s and action a

Temperature Entropy Update The entropy temperature update regulates exploration vs. exploitation. This update makes sure the agent maintains the wanted amount of exploration or exploitation by changing the temperature parameter based on the policy's entropy. Mathematically, the update is given as:

$$\alpha \leftarrow \text{clip}(\alpha + \eta \cdot E_{s \sim \mathcal{D}} [-\log(\pi_\psi(a|s)) - \text{Target Entropy}, \alpha_{\min}, \alpha_{\max}]) \quad (4)$$

- α : Entropy temperature parameter (to be updated)
- $\text{clip}(\cdot)$: Clip function ensuring the value stays within a specified range
- η : Learning rate for the update
- $E_{s \sim \mathcal{D}}$: Expectation over the data distribution \mathcal{D}
- $-\log(\pi_\psi(a|s))$: Negative log probability of the action a given state s under the policy
- Target Entropy: Variable to be specified in practical application
- α_{\min} and α_{\max} : Minimum and maximum values for the entropy temperature

Evaluation

Aside from being anatomically correct, the hand must be the correct color (within any reasonable *human* skin tone range), have the correct orientation, be a clear image with little blur, etc. Essentially, the evaluation will be on a visual basis of whether the output is closer to what a hand is than the previous output. Out of the roughly 20 images per output, about 15 of them need to be highly accurate for this project to be deemed a success and reach the 70-80% benchmark.

Timeline

February 12: Proposal Paper and Presentation:
 By February 19: Code the SAC algorithm as functions
 By February 24: Add in the agent
 March 11: Finish training and consolidate results
 March 18: Present Midterm Paper, Presentation, and Demo

Conclusion

In this paper we talked about the continuation of a neural network experimentation that tried to generate images of hands and how we will work to improve it using reinforcement learning. This previous experimentation used a variational auto encoder and got poor results in color. The agent spoken of will use an algorithm known as soft actor-critic or SAC in an effort to improve the latent understanding and thus output generation.

References

- Matthias, Meg. "Why does AI art screw up hands and fingers?". Encyclopedia Britannica, 25 Aug. 2023
- Kalpokas, Ignas, and Julija Kalpokiene. "From Gans to Deepfakes: Getting the Characteristics Right." Springer-Link, Springer International Publishing, 2022
- Prachi Patel. "Spray-on Smart Skin Reads Typing and Hand Gestures." IEEE Spectrum, IEEE Spectrum, 3 Mar. 2023
- F. Soroni, S. a. Sajid, M. N. H. Bhuiyan, J. Iqbal and M. M. Khan, Hand Gesture Based Virtual Blackboard Using Webcam, 06 Dec. 2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 2021
- Guillermo Garcia-Hernando, Edward Johns, & Taekyun Kim, Physics-Based Dexterous Manipulations with Estimated Hand Poses and Residual Reinforcement Learning. IEEE Spectrum, IEEE Spectrum, 2021
- Pu, Y., Gan, Z., Henao, R., Yuan, X., Li, C., Stevens, A., & Carin, L. (n.d.). Variational Autoencoder for Deep Learning of Images, Labels, and Captions, NeurIPS Proceedings, 2016
- Bank, Dor, Noam Koenigstein, and Raja Giryes. Autoencoders. Machine learning for data science handbook: data mining and knowledge discovery handbook Feb. 2023
- El-Kaddouri, M., Mahmoudi, A., Himmi, M.M. "Deep Generative Models... and Generative Adversarial Networks", Springer, 2019.
- Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. "Soft Actor-Critic Algorithms and Applications". Cornell University arXiv, 2019
- Esther Derman, Daniel J. Mankowitz, Timothy A. Mann, and Shie Mannor. "Soft-Robust Actor-Critic Policy Gradient". Cornell University arXiv, 2018