RoboPianist

Improvements to learning a robotic dexterity challenge.

Matt Smith [1] Eric Ye [2]

[1]Computer Science, Stanford [2]Electrical Engineering, Stanford

Introduction

RoboPianist is a robotic dexterity benchmark implemented in Mujoco by Zakka et al[2]. The environment (pictured below) tasks two robotic hands with performing a piece of music on a piano. The environment is challenging due to the high dimensionality of the action space and the sparse rewards. The reference implementation is trained using soft actor-critic (SAC) using a 3-layer MLP encoder for the actor and the critic, and trained an actor and critic from scratch for each piece it performed. Our goal was to train an agent that was able to generalize well to new pieces with little-to-no finetuning.

Methods & Experiments

Pretraining

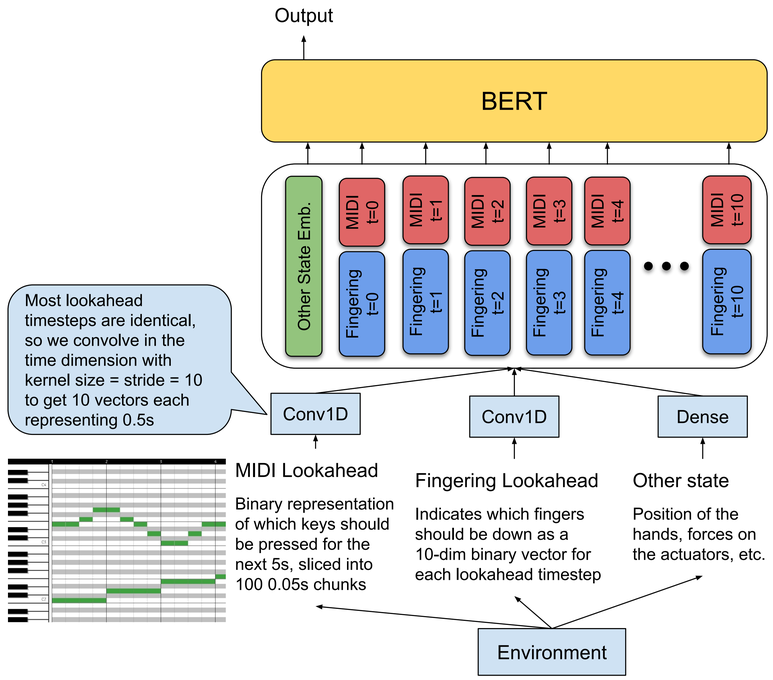
The original RoboPianist Suite came with two major scale environments for debugging, C Major and D Major. We augment this with the remainder of the chromatic scale with full fingering information, and train the agent to play these scales prior to training on the target piece. We also explore pretraining on 138 pieces from the PIG dataset [1], a dataset of Midi files with annotated fingering information, and evaluted on the robopianist-etudes-12 dataset from Kazza et al.

Hindsight relabelling

Hindsight relabelling is a reinforcement learning technique that helps especially in scenarios with sparse rewards such as this. We implement this by rolling out a trajectory based on the current agent, recording which keys were actually played during the episode, finding the closest finger to each played key and creating a new task for the same agent based on those recorded keys.

Transformer-Based Agent

We replaced the multilayer perceptron from the original model-free agent with a transformer-based agent based on BERT. To do this, we unravel the horizon of goal states into a sequence and we append the current state to each time step in the sequence. We also increase the midi lookahead from 0.5s (10 steps) to 5s (100 steps) and add fingering information to the lookahead.



Results

Pretraining We found that while scales pretraining improved for some songs, such as French Suite No. 1 Allemande, it did not improve the performance of the agent in general for the etude songs. We also found that pretraining on other songs in the dataset did not help. We saw that the pretraining on other songs was not converging so we stopped it before letting it run to completion to get results.

Approach	F1 Score after 1M steps
Zakka et al. (no pretraining)	0.538 ± 0.122
Scale pretraining	0.524 ± 0.092

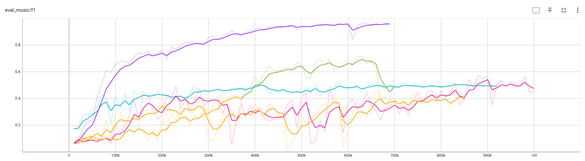
Transformers

We trained our transformer model on French Suite No 1 Allemande. We also ported the increased lookahead length, 1D convolutions and fingering lookahead to the MLP model, which we report as MLP+Conv.

	100k Steps		1M Steps	
	F1	Time Elapsed	F1	Time Elapsed
MLP	0.050	20 min	0.543	3.8h
Transformer	0.154	39 min	0.173	6.5h
MLP+Conv	0.156	20 min	0.122	3.9h

The transformer and MLP+Conv encoders increase training velocity initially, but stop making progress much earlier than the original MLP algorithm.

Hindsight relabelling Hindsight relabelling did not improve learning rate or learning over a non-relabelling baseline and actually performed worse. We think this is because the nature of relabelled tasks is significantly different from the nature of the songs the agent is trying to learn. The relabelled task might have no notes, or have a single finger pressing multiple keys at once which is challenging for the agent to reproduce.



F1 without hindsight (purple) compared to F1 with hindsight relabelling with various reward weightings. Some training runs were prematurely cancelled to save time.

Conclusion

While we found that Pretraining on individual scales can help learning, pretraining on multiple scales or pieces did not help the agent generalize to new pieces better. Architecture changes (Transformer and MLP+Conv) yielded initial improvements to F1 score but stopped learning much earlier than the MLP model. We hope to continue working on hindsight relabelling until the final project report and make a positive improvement with one of the approaches.

We would like to thank Kevin Zakka for his help with the RoboPianist code.

[1] Eita Nakamura, Yasuyuki Saito, and Kazuyoshi Yoshii. Statistical learning and estimation of piano fingering. *CoRR*, abs/1904.10237, 2019.

[2] Kevin Zakka, Laura Smith, Nimrod Gileadi, Taylor Howell, Xue Bin Peng, Sumeet Singh, Yuval Tassa, Pete Florence, Andy Zeng, and Pieter Abbeel. Robopianist: A benchmark for high-dimensional robot control, 2023.

matt@mjksmith.com, ericye16@gmail.com