Integrating Automated Speech Recognition with Reinforcement Learning

Austin Ray [1 2] **Do-Hyoung Park** [2] **Vignesh Venkataraman** [1]

1. Introduction

The goal of our project is to explore the integration of Automated Speech Recognition (ASR) with Reinforcement Learning (RL), for both theoretical and practical purposes. This is a particularly interesting problem because it examines whether using indirect rewards and RL techniques is effective in training an ASR model, and would have similar implications for the field to those of the Atari DeepMind paper.

We will start with the simple OpenAI Gym FrozenLake-v0-4x4-Deterministic environment. We will train an agent to process speech-based features that describe the state of the system (e.g. the spoken digits of the state) and output the Q values of that state – similar to the Atari Deep Q Network (DQN), but with auditory instead of visual features.

After implementing our auditory DQN, we have several other experiments we'd like to try, including: (1) Using both visual and auditory features with the Atari Pong DQN from the homework; (2) A FrozenLake agent whose objectives depend on vocal commands; (3) Tuning a FrozenLake agent to a real human's reward function via vocal reactions (rewards). Essentially, we hope to investigate what stages/inputs of the RL process we can supplant with auditory inputs, and compare how doing so on different elements of the agent (e.g. states, delayed rewards, immediate rewards) might affect overall model efficacy and speed of learning, both in offline, research-focused settings and in online, real-world settings. Additionally, if the auditory DQN doesn't end up producing a viable FrozenLake agent, we are interested in exploring a two-component system similar to Baseline 4 (see below) that takes the uncertainty of the ASR component into account by modeling FrozenLake as an Hidden Markov Model. We flesh out these planned experiments in our "Future Work" section below.

All that said, it's important to note that our main goal is to implement the auditory DQN – all other experiments are auxiliary goals and all baselines discussed below explicitly apply to the auditory DQN.

We realize that Q-Learning might not end up being the best learning choice for RL agents with ASR components, and we're excited to try other learning methods such as SARSA, Policy Gradient, and model-based methods to see what works best. We also plan to experiment with more recent RL techniques, such as double Q-learning and dueling networks.

One way we will evaluate our agent is by comparing it to our agent from Assignment 1 in terms of overall performance, time-complexity of training, and space-complexity of training on the FrozenLake environment. If we complete our ensemble DQN introduced above, we will compare the performance and requirements of our agent to that of the Atari DeepMind DQN.

We will also test the viability of transitioning our trained ASR DQN into a stand-alone spoken digit recognizer. For this second evaluation task, we will force the network to output its theories (probabilities) as to which state it is in as an intermediate layer, from which we can compose a spoken digit recognizer by taking an argmax. Specifically, we will use classification accuracy on our spoken digits dataset as the evaluation metric here. This task might involve some neural net surgery + transfer learning in an official digit classification setting.

2. Completed Work

2.1. Baselines (Already Completed)

2.1.1. Baseline 1: Q-Learning in Isolation

Our baseline for FrozenLake performance is a Q-Learning agent trained in isolation (i.e. no ASR component). On both the deterministic and stochastic 4x4 FrozenLake environments, our Q-learning agent, using an epsilon-greedy exploration strategy in order to ensure an exhaustive exploration of the world, learns an optimal policy within 10,000 episodes. With an optimal policy for the stochastic environment, the agent reaches the goal state (succeeds) in $> 78\%$ of episodes. With an optimal policy for the deterministic environment, the agent reaches the goal state (succeeds) in

[1]CS 224S, Stanford University, Stanford, CA [2]CS 234, Stanford University, Stanford, CA. Correspondence to: Do-Hyoung Park <dpark027@stanford.edu>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, 2017. JMLR: W&CP.

100% of episodes – it simply takes the shortest path to the goal state. These are the main benchmarks we will be using for evaluating the performance of our auditory DQN on the deterministic and stochastic 4x4 FrozenLake environments.

2.1.2. Baseline 2: Trained Digit Recognizer in Isolation

Our baseline for digit classification performance is a digit recognizer trained in isolation. The model we use for this classifier is a recurrent single-depth neural network with a single GRU cell and an affine transform layer for classification. Trained in isolation to convergence, our digit recognizer achieves 94.9% accuracy on the training set and 95.5% classification accuracy on the validation set - both sets are subsets of the TIDIGITS data corpus. After further tuning, it is not infeasible to imagine sub-1% classification error, as has been shown in several papers.

2.1.3. Baseline 3: Q-Learning Agent Trained in Isolation; Tested in Tandem with Untrained Digit Recognizer

Another baseline we implemented for FrozenLake performance was to take the trained Q Learning agent from Baseline 1 and combine it with an untrained digit recognizer. While the FrozenLake Q-learning agent had previously always been provided wholly accurate information about its state within the world by the OpenAI environment, we integrated the ASR component by, upon arrival at a new state, sampling a digit audio recording from the TIDIGITS dataset corresponding to the ID of the agents state (from a collection of 30 different recordings of the same digits) and feeding it to an untrained digit recognizer to tell the RL agent what state it was currently in. In other words, the agent received unreliable state information at test time. The purpose of this baseline was mainly to ensure that the ASR and Q-Learning models would interface together without any issues, but also to show that giving an agent unreliable state information leads to almost certain failure. Note that the combination of a trained FrozenLake agent and a digit recognizer is different from the end-to-end auditory DQN we plan to implement - we simply explored this setup as a baseline.

Out of 100 episodes, the agent given state information by the untrained digit recognizer made it to the goal state zero times. This makes sense, as the agent cannot make optimal actions if it doesnt know what state it is in. This baseline is therefore equivalent to using a pretrained digit recognizer with a random policy - even if the agent knows what state it is in, it cannot make optimal actions without knowing which actions are optimal.

2.1.4. Baseline 4: Q-Learning Agent Trained in Isolation; Tested in Tandem with Trained Digit Recognizer

Our fourth baseline is like our third baseline, but with a pretrained digit recognizer that was trained on a modified dataset of digit recordings that only included sequences of two digits, in which we likely overfit the training data, with the understanding that we would never give our system an input sequence of longer than two digits. After 500 epochs of training on the full dataset, our model consistently achieved a training set error of < 0.02 and an average training set edit distance of < 0.01 in the digit recognition task alone. To be clear, we trained the Q-Learning and digit recognizer models separately and achieved optimal isolated models for each, and then tested them together.

This process was basically equivalent to adding noise to the state representation for the Q-Learning agent during testing, with the digit recognizer being the noise. The key difference between this setup and that in Baseline 3 is that the noise is drastically reduced, leading us to believe the agent would be able to reach the goal state at least sometimes.

When tested on the Stochastic 4x4 FrozenLake environment, this model achieved a 100-trial success rate of 32% (benchmark is 78% from Baseline 1), while it achieved a much better 100-trial success rate of 76% in the Deterministic 4x4 FrozenLake environment (benchmark is 100% from Baseline 1). These results make sense; while our trained model is quite successful at identifying the state from the speech inputs, it's not perfect; and even in a deterministic world, where in theory we should be able to reach the goal state every time via our learned optimal policy, our agent fails nearly a quarter of the time due to rare misidentifications of the state by the digit recognizer, which often tend to lead the agent into a hole. In the stochastic world, the "noise" presented by these occasional misidentifications adds to the noise inherent in the stochasticity of the world to combine for a model that is only successful at solving the FrozenLake problem a third of the time. We are unsure whether the auditory DQN proposed below will perform better than this baseline on the deterministic and stochastic environments and are quite interested to see the results.

2.2. End-to-End Integrated Model (Current Work)

Ultimately, the model we are currently working on is an end-to-end ASR/RL integrated model in the spirit of the Atari DeepMind paper, where deep Q-learning with experience replay was used to train an agent to play the game of Pong. Instead of using a visual input encoded using a convolutional neural network to teach the agent its state, as the DeepMind team does, we hope to use our digit recordings from the TIDIGITS dataset as the input at each step

of the FrozenLake environment to our neural network that will serve both as a digit recognizer and a Q-function approximator, which will return a tensor of Q-values corresponding to each action from that state. The model will then use an ε-greedy exploration technique to explore the world in each episode and conduct a gradient descent step on the Q-function after each episode using the experience replay sampling technique and a target network to stabilize the gradient descent.

In essence, the update rule for our neural network parameters θ can be expressed as:

$$\theta = \theta + \alpha\left(r + \gamma \max_{a' \in A} Q_{\theta^-}(s', a') - Q_\theta(s, a)\right)\nabla_\theta Q_\theta(s, a)$$

for each step of the gradient descent, where θ^- represents the target network, which is only updated sporadically and doesn't factor into the gradient being computed.

There are some notable differences between the DeepMind theory and our proposed theory; the Atari DeepMind environment is large and continuous in game space, while our game space is finite and discrete, meaning that we should expect our Q-network to converge much more quickly than did the Atari DeepMind network. The values of actions in our environment should also be easier to learn, as the rewards in the FrozenLake environment are less delayed than in the Atari environment, meaning it should be easier for our network to learn which actions in which states are optimal. (This also means that using a policy gradient model for learning optimal policies should update more frequently and thus converge more quickly as well.) Finally, the CNN used in the DeepMind paper is likely to have many more parameters than our final network, as image processing from a large pixel map required many different modeling layers, while our audio inputs come neatly packaged in MFCC encodings, which is already preprocessed and thus should require much less hefty processing by a neural network to provide outputs. Given that the DeepMind DQL was able to converge within a reasonable amount of time in the Atari game space, we should thus expect our model to learn on the FrozenLake environment relatively quickly.

3. Future Work

3.1. Visual + Auditory Atari DQN

We plan to combine the visual and auditory inputs to our network via a combined-input DQN with both auditory and visual features that is purely additive to the Atari DeepMind paper. That is, the network takes in both DQN visual features + speech-based state-describing features, with outputs remaining the same. We hypothesize this network will perform better than the network in the DQN paper, given it has a larger body of inputs that can be used to describe its state at each point in the game. This would have implications for real-world robotic agents that receive a variety of sensory data.

3.2. Dynamic Auditory Goal Inputs

Can we teach an agent to dynamically receive speech inputs from a human and use that input to adjust its optimal policy in the FrozenLake world? That is, if a human gives the command to the agent to "stay" while it is in state 7, can the agent change its objective to remain close to state 7, instead of trying to approach the goal? This would likely involve a much more robust state input, with state features corresponding not only to the ID of the current state, but also to the audio recording of the last vocal command given to the agent and the square the agent was in when it was given that command. This would likely involve training the same agent on several different environments with different reward structures corresponding to each of the desired commands, while separately training it to classify the various commands that might be given to it at runtime.

3.3. Teaching a Policy Through Spoken Human Interaction

We feel it would also be interesting to see if a policy can be taught through an agent solely through auditory reinforcement from a human. That is, either following every step in an episode or following the terminal state, having a human give feedback like "Yay!" or "Good job!" or "Bad!" or such, in an effort to convey that human's personal optimal policy to the agent. This could be in lieu of a traditional reward function or as a supplement to one. The feedback speech inputs would be pre-trained, with the primary question of interest being whether this is a viable and realistic way for training to be done in the real world.

3.4. HMM FrozenLake

Finally, if our auditory DQN doesn't produce a viable agent for the FrozenLake environment, we are interested in pursuing a model similar to that found in Baseline 4 (see above). We would improve upon Baseline 4 by modeling the FrozenLake environment as an HMM, training our Q-Learning agent to take into account the state uncertainty we get with ASR. The agent could utilize HMM-related algorithms like the Viterbi Algorithm to infer what state it is actually in given a history of noisy state information, allowing it to make more educated decisions in the environment. Note that this experiment would exist simply to explore the potential performance of combined ASR-RL systems, not to attempt to train an ASR system in an RL environment, which is the primary goal of this project.