brew.ai Fall Term Overview

Aravind Parasurama, Connor Yates, Cody Holliday

Introduction

- Purposes and Goals
- Problems Encountered
- Design
  - Learning Algorithm
  - Hardware Design
  - User Interface Design
- Progress Review
- Current Status

Purpose

- brew.ai is a hardware and software solution for automated brewing of mead or beer
- Enables amateurs to brew good tasting beer or mead
- Enables professionals to automate the brewing process

Goals

- Have a physical, polished device that has the needed electronics and software to control the brewing process
- Have an easy-to-use user interface
- Have a learning service that improves brewing over time
- Create a reliable and maintainable product

Problems Encountered

- Decision between temperature control devices
- Thermistor vs. thermocouple for temperature measurement
- Method of communication between project components
- User interface method
- Connection Between controller and User Interface

Overall Design

3 Components

- Hardware
  - Microcontroller
- Learning Algorithm
  - Raspberry Pi
- User Interface
  - Android Device

Learning Algorithm

Three main parts:

1. Q-Learning Algorithm
2. Neural Networks as Q-Value Approximators
3. Overall Agent Structure

To learn by associating available actions (temperature control, stirring speed) with the user's' responses about a batch.

Q-Learning

Q learning associates a given state (temperature, CO_2 level, specific gravity, time) with the best known action to take at this state.

Calculates Q-Values to give a ranking to each action (stirring, temperature control, and brewing termination) at this given state.

By picking the maximum Q-Value, the agent takes the best action.

Q-Values are updated over time based on prior knowledge, past Q-Values, and the reward from the user.

Neural Networks as Q-Value Approximators

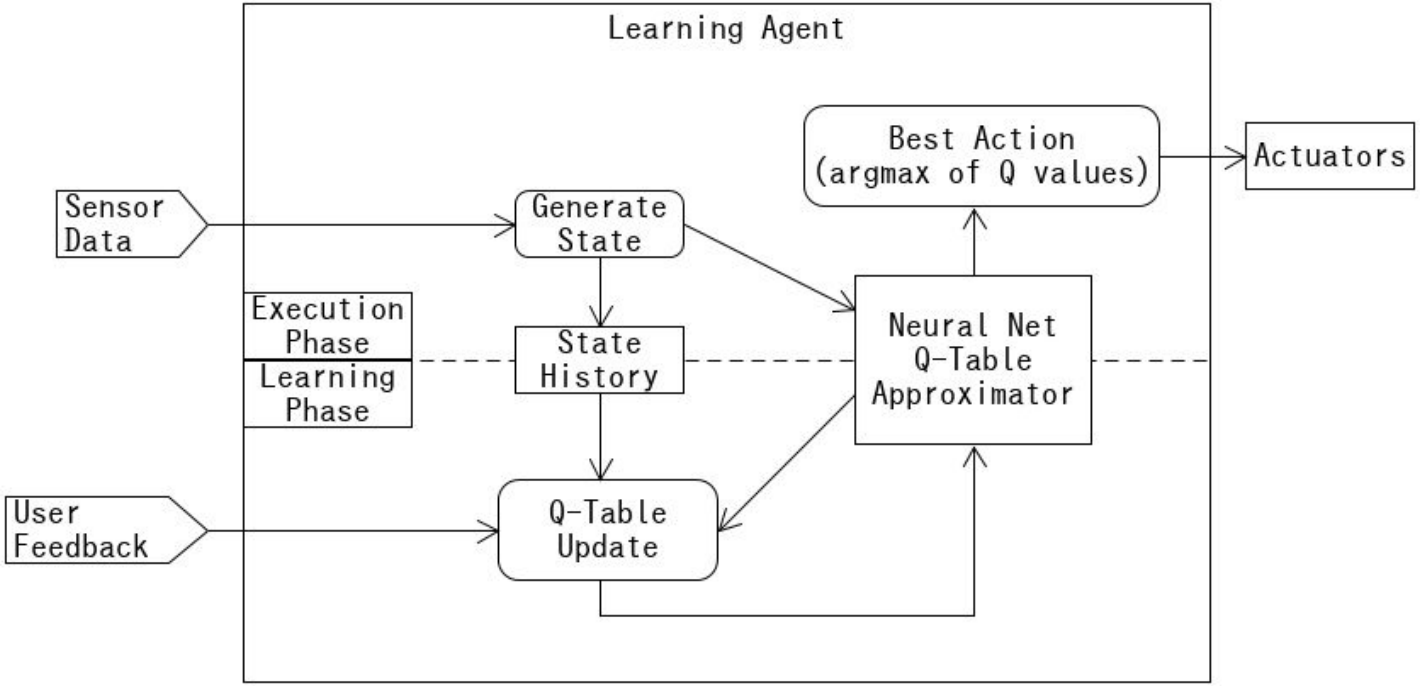
Q-Values are traditionally stored in finite tables, but our problem space is continuous...

Neural Networks can approximate the Q-table.

Input : State, Output : Q-Value for action i

Increases complexity of learning

Agent Structure



Hardware

- Device is a bucket lid with electronic components securely housed within
- Powerful heating element
- Temperature, carbon dioxide sensors
- Digital hydrometer

User Interface

3 Separate Parts:

- Android Device
- Bluetooth connection
- Data processing on device.

Android Device

- Android App simple to develop
  - Lots of support from Google
- Cheap devices available
- Independent Interface
- Opens the door for App release

Bluetooth Connection

- Connection to Controller
- Universal Interface
- Bluetooth v4 offers LE mode
- TCP connection
- App release

Data Processing on Device

- Spreads out processing load
- Android Graphing library -- Graphview
  - Rendering and Scaling
- JSON data format

Progress Review: Weeks 3-5

- Meeting with Dale, assessing his expectations
- Set-up source control
  - Github, waffle.io
- Wrote problem statement
  - Revised after Dale’s feedback

Progress Review: Weeks 6-8

- Brewing
  - Manually brewing mead
- Tech review
  - Determined who is responsible for what technology
- Requirements document
  - Finalized initial project requirements

Progress Review: Weeks 9-11

- Design document
  - Finalized initial project design
- Progress report

Current Status

- One batch of mead
- First Design Complete
- No metrics for mead quality
- No progress on developing a business
- Already have the Controller and Microcontroller

Appendix

Q Learning Update Function:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \cdot \left(r_{t+1} + \gamma \cdot \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right)$$

s := state, a := action, alpha := learning rate, r := reward, gamma := discount factor