Brain Lesion Detection with Deep Q Learning

Motivation

The motivation for this project came from my personal experience with medical imaging. Usually when it comes to image data, the go-to approach is to use a convolutional neural network. It has been widely studied and can achieve state-of-the-art performances on various tasks, but there still exists possible drawbacks to the CNN approach. For example, training CNNs requires a large amount of annotated data, and the CNN itself lacks interpretability because of its black-box nature. Therefore, I was curious to see if we can use reinforcement learning techniques to solve similar tasks.

Problem Statement

The task I am trying to solve is brain lesion detection, inspired by the paper from J. N. Stember and H. Shalu. It involves the combination of deep learning and reinforcement learning, where an agent will be trained to navigate through brain images and locate the lesion. It will learn a policy and take actions based on what it sees in the local image region.

Dataset & Preprocessing

The brain lesion dataset in the original paper was from the BrainTS 2014 public brain tumor database. Unfortunately, I was not able to find or access the same dataset, and therefore the data used in this project came from another publicly available dataset on Kaggle at https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri/data. This dataset contains brain tumor images from 4 categories, and I manually selected 87 meningioma images for this project (62 for training and 25 for testing). All images are resized to 224x224.

Environment Setup

In order to train an agent, we first need to set up the environment. Each image is divided into 16 regions, and labels are vectorized based on which region(s) the lesion falls in. Each region then represents a state in the environment that will be used to inform the agent's decision making process.

The agent always starts in state 0 (top left) and can only take three actions at each state: stay still, move down, or move right. The reward function for the environment is as follows:

$$R = \begin{cases} -2, \text{ if the agent stays in a non-lesion state} \\ -0.5, \text{ if the agent moves into a non-lesion state} \\ +1, \text{ if the agent stays in or moves into a lesion-state} \end{cases}$$

Methodology

Two different neural networks are involved in the Deep Q-Learning algorithm, the Q-network, and the target network. These two networks share the same architecture (as shown in the figure below), but different training procedures.

- **Q-network**: Takes in a state and outputs the Q values for all possible actions. During inference, we use this network to obtain the Q values and pick an action
- **Target network:** used to generate target Q-values, not actually trained

The Q-network is trained using the experience replay mechanism. At every state during an episode, after the agent selects an optimal action and obtains the reward, the (state, action, reward) pair will be added to the agent's memory buffer. We will then randomly choose a batch of "experience" samples from the buffer and train the Q-network.

The target network is not trained. Once we iterated over all regions in an image, we will copy the Q-network weights directly over to the target network. Unlike the Q-network, which is trained multiple times per image region, the target network is only updated once per image.

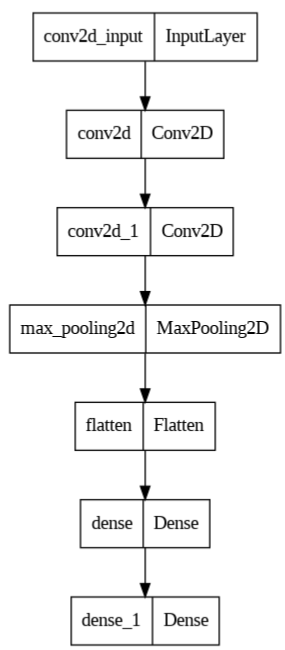

Fig. CNN architecture for the Q-network and target network

Evaluation, Results & Discussion

The agent is trained for 20 episodes using 62 training images and evaluated on 25 test images. During evaluation, the agent takes 15 steps and the state it ends up in is compared against the ground truth labels for performance assessment.

Out of the 25 test samples, the agent is only able to correctly identify the lesion on 5 occasions with an accuracy of 20%. This is significantly lower than the results achieved in the original paper. The main reason is because of the dataset itself. The dataset used in the original paper are specifically crafted for this task where the background is clear, and we have a high contrast only at the location of the lesion. The dataset I used contains raw fMRI brain images, which means that there are significant amount of noise. These images have high contrasts and noises in irrelevant positions, and the lesion itself is no longer emphasized or highlighted. Therefore, it is much more likely for the agent to make wrong decisions. If this had been trained on the original dataset, we could expect much better results.

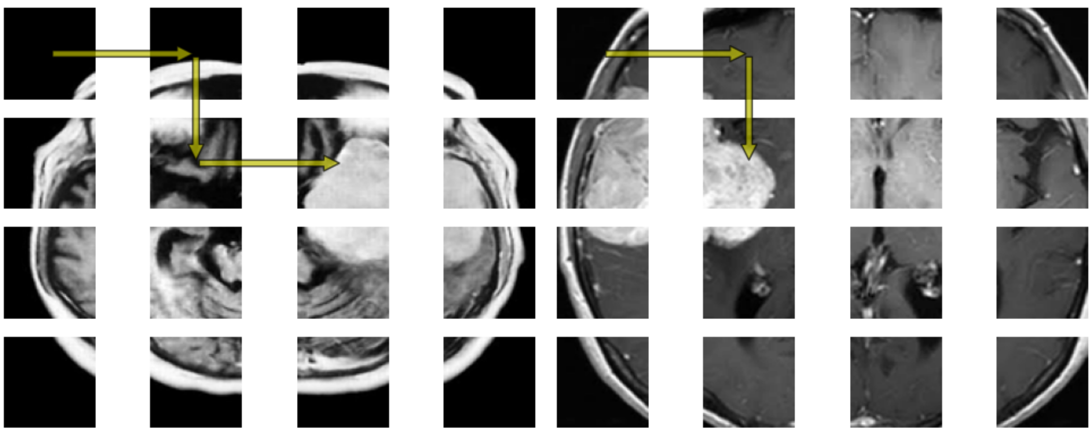

Fig. Agent paths for correctly identified lesion images

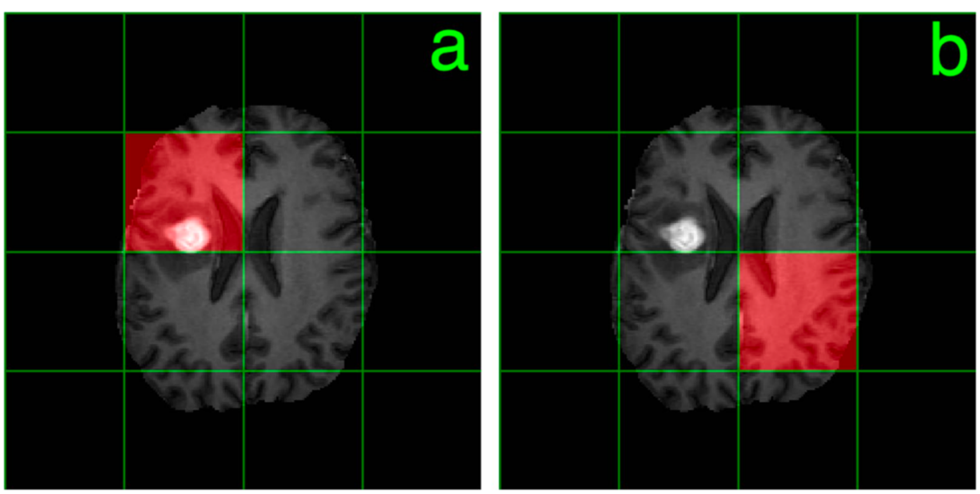

Fig. Examples of images in the original paper

Link to code on Google Drive: https://drive.google.com/drive/folders/1azw8lAjJYh6M94ux6G-mldStZ5DB4KYt?usp=sharing