AI Programming: Assignment 6

Eirik Vågeskar

November 24, 2015

This report describes an artificial neural network (ANN) for playing the game 2048 made in the Theano framework. It will present some of the choices made in designing the network, describe two different input representations for the network, and comment a little bit on the playing style of the network.

1 Choices Made in Implementing the ANN

This is partially commented on in the last report, but I will reiterate the points here:

- **Topology:** The topology of the network was found through some trial and error with logging. The final topology, (20, 20), yielded a hit-rate of about 78% on both the training and the validation set, which was satisfactory.
- **Activation function:** The activation function used by the network is the sigmoid function. Some trials with a hyperbolic tangent were done, but these led to results no better than guessing. Because the network already performed very well with the sigmoid function, the problem was not investigated further.
- **Learning rate:** The learning rate is adaptive, and is a function of the current epoch: $lr = \max(\frac{1}{\text{epoch number}}, lr_{min})$. This was chosen because it yielded good results in the early trials, and did not hinder results later on.
- **Momentum:** Momentum was never implemented because the networks performed well enough.

2 Representations

Two different representations were tried for the 2048 neural network. They both contain 10 extra nodes in addition to 16 input nodes (1 for each space on the board): One representing the number of merges that will result from a horizontal slide and one for the number of vertical slides, and eight for the number of free spaces in each row and column. All of these 10 extra nodes are scaled so that their values are always between 0 and 1.

What separates the two is whether the values on the board are scaled or not. The representation where the values are scaled to values between 0 and 1 clearly outperforms the unscaled version.

- **Welch test for a run of the *unscaled* representation**

```
Welch test score:  Average random tile: 100.48
Average ANN tile: 188
P-value: 0.0000143092
Number of demo points: 5/7
```

- **Welch test for a run of the *scaled* representation**

```
Mean best tile for random: 120.96
Welch test score:  Average random tile: 120.96
Average ANN tile: 369
P-value: 0.0000000000
Number of demo points: 7/7
```

3 Comments on the ANN's playing style

The ANN's training (and validation) set is a quite simple one: It is based on an AI employing a tactic of pushing the tiles up for the most part, as well as wiggling left and right to merge tiles and hopefully merge larger tiles further up later. This is a tactic that rarely achieves high tiles, like 2048, but works very well for someone playing almost only on reflexes. This suits the description of this AI very well.

As one can see from Fig. 1, the high numbered pieces quickly accumulate at the top. Already in the second picture, the highest tile is not located in the corner. This is because the AI is not very far sighted and not aware of the desirability of having a high valued piece in the corner. (Some experiments with a reward for a piece in a corner were made, but this did not improve game scores, perhaps due to the quality of the training set.)

A sequence of events characteristic of the ANN is shown in Fig. 2. The direction of the move selected for each game board is shown as a character below it (u, d, l, or r). In the first picture, the first thing that strikes an experienced player is the possibility for a good combo: Two 4s are lined up to be merged, and the result can be merged with the 8-tile in the top row. Subsequently, this can be merged with the 16 beside the 8. The AI starts this procedure, but when selecting the second move, it pushes left rather than to the right, thus leaving a space open in the corner. A high tile in the corner is desirable, but this ANN knows nothing of it.

With the board configuration in the third picture, it would most likely be beneficial to immediately push to the right and lodge a 32-tile in the top right corner. The ANN has decided that merging two 2s in the left corner to a 4-tile is more important and pushes up. Subsequently it pushes up again and merges the two 2s. At this point, it finally decides to slide to the right and place the 32 in the top right corner.

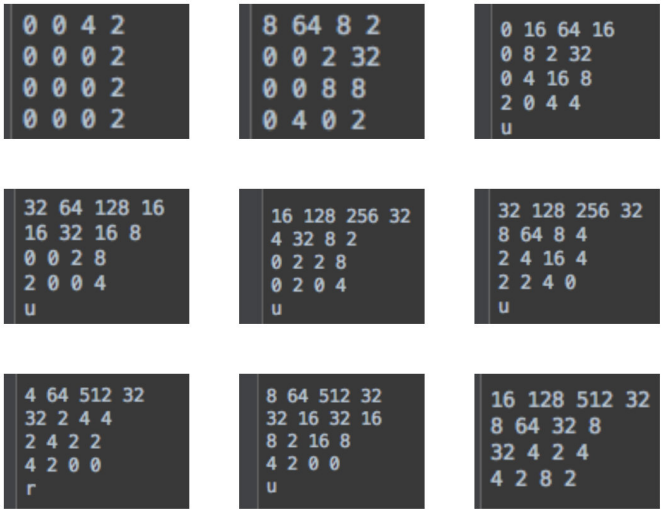

Figure 1: Various stages of a 2048 ANN game (read left to right).

As one can see, the ANN plays considerably better than a random player, but with its "risky" behaviours, it is clear how high-valued tiles end up in the middle of the top row, as seen in Fig. 1.

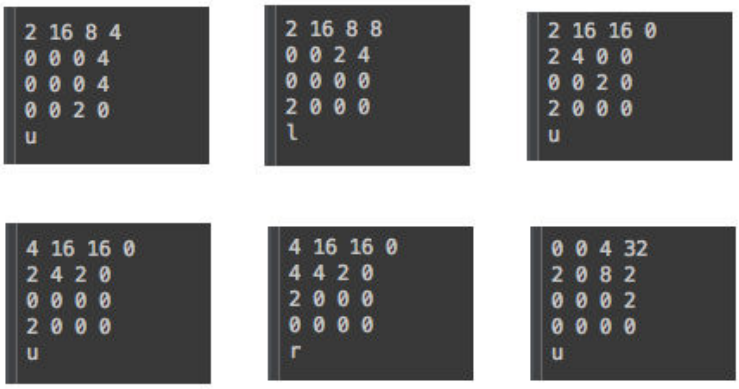

Figure 2: A typical sequence of events in the game