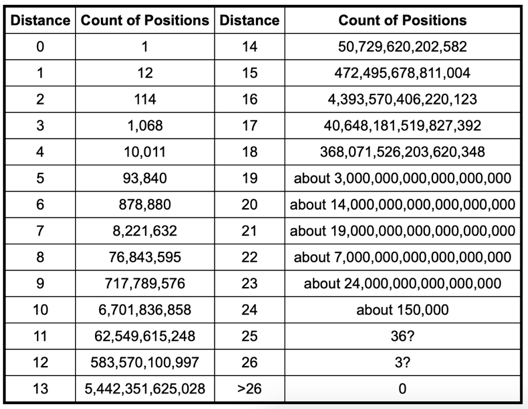
**Research Questions** - Implementing neural network approaches to solve the Rubik’s Cube have struggled to succeed without human help and have had to rely on hand-engineered features and group theory to systematically find solutions.

**Research Objective**  –

* To obtain valuable heuristic functions automatically, so that the whole problem-solving process is free of human knowledge. (***first objective***)
* To use heuristic rules of the algorithm to perform random walks back from the goal state and try to learn how far the goal has been achieved. (***seconds objective***)

To prove **first objective**, neural networks with more than one hidden layer tend to perform better than shallow ones. Therefore, for this research I considered several net architectures and parameters, all with more than 3 fully-connected hidden layers. For the estimator training, it is essential to generate examples with a given distance or the scrambled number from the objective state— data examples of mixed cubes with distance is needed to be generate. In order to do this, I scrambled it back randomly from an objective state (solved state of Rubik’s Cube). That provides a cube setup that is long, as it cannot be sure that there are no shorter walks that will produce the same result. With the distance (*up to d≈20, refer to Figure 1*), it is necessary for the number of configurations to increase exponentially with a given distance, so that automatically the probability of a configuration should be high.



*Figure 1: Histogram of cube configurations (*[*http://cube20.org/qtm/*](http://cube20.org/qtm/)*)*

Three models are trained for a Deep Neural Network (DNN) regression with the different layer and neuron numbers, over 25 scrambles bas:

: has three layers, of 70, 60 and layer of 50 neurons, respectively.

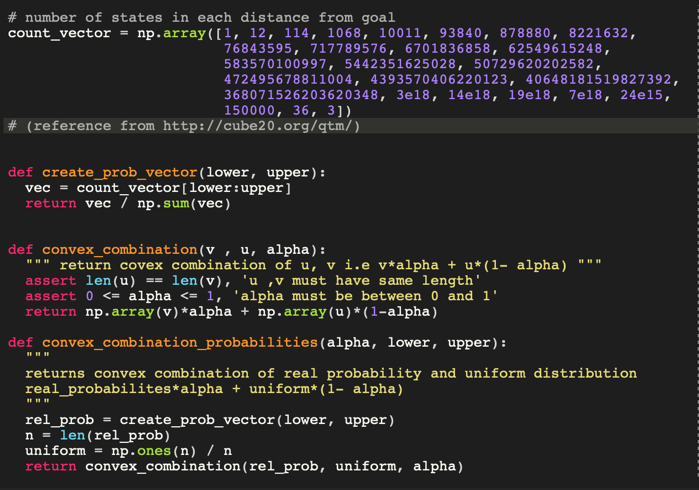
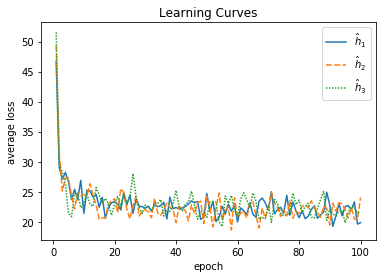
: has four layers and 50 neuron levels each layer.

: has five layers, with neurons: 50, 40, 30, 20 and 20.

All models have been trained on 100,000 labelled cube examples. Below are functions related to probability vectors (*refer to Figure 2*):

* ***create\_prob\_vector*** generates the "real" distribution of cube configurations.
* ***convex\_combination*** returns a convex combination of two vectors.
* ***convex\_combination\_probabilities*** returns a convex combination of the real distribution and the uniform one.

Based on the trained data with about the same loss and training curves (*refer to Figure 3).* Learning curves are widely used in machine learning for algorithms that learn (optimize their internal parameters) incrementally over time. During the training of a machine learning model, the current state of the model at each step of the training algorithm can be evaluated. It can be evaluated on the training dataset to give an idea of how well the model is “*learning”*.

*****.*

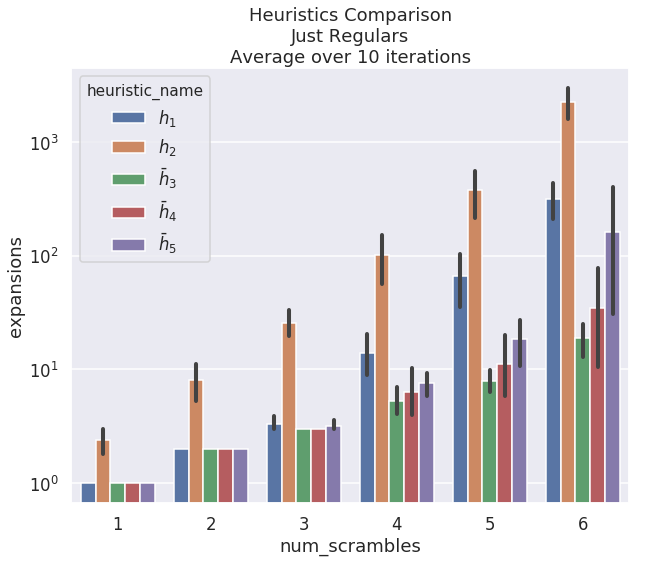
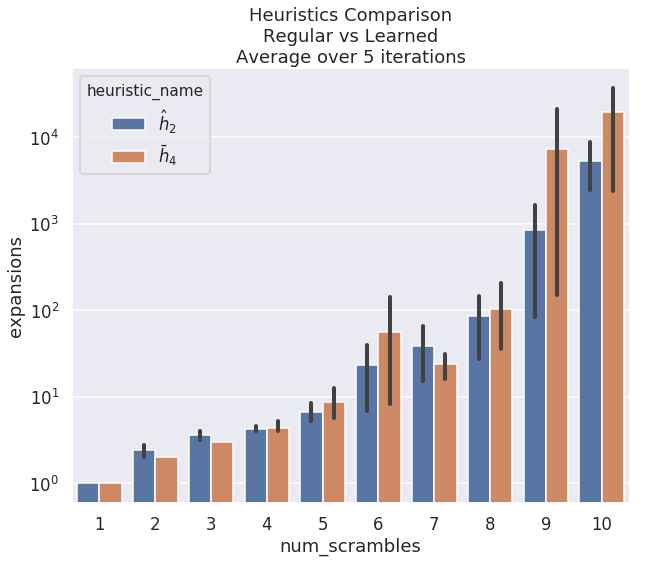
*Figure 2: Functions for probability vectors Figure 3: Learning curves of trained data.*

To prove **second objective**, two type of heuristic rules are used, Informed Search Algorithm (admissible and inadmissible).

* Two admissible heuristic rules are used, Maximum Misplaced Edge Cubes in Face of Rubik’s Cube and Minimum Misplaced Edge Cubes in Rubik’s Cube Face.
* Three admissible heuristic rules are used by finding the non-optimal path, Maximum Misplaced Edges and Corner Cubes in Face of Rubik’s Cube, Maximum and Minimum Misplaced Corner and Edge Cubes in Face of Rubik’s Cube and Maximum Misplaced Corner and Maximum Misplaced Edge Cube in Face of Rubik’s Cube.

As expected, the heuristics allowed for were too inefficient, so I only used the inadmissible values for larger scrambles () values. These actions were successful in solving up to = 25 moves. The heuristic has been successful especially with averages of 1, 500 nodes for 25 scramble movements from originally trained model. Furthermore, inadmissible heuristics did not significantly affect the optimal solution. To test this, I checked whether the returned length of the solution exceeds the number of scramble moves for this example. (*refer Figure 4*)

The learned heuristic, , is marginally better than the best person made, as shown in the results.During plotting the comparison, it was relatively accurate, and expected it to perform better. However, the results satisfied with the results given the limited training time and data. (*refer Figure 5*)

****

*Figure 4:* Heuristics Comparison Just Regulars *Figure 5:* Heuristics Comparison Regular

Versus Learned