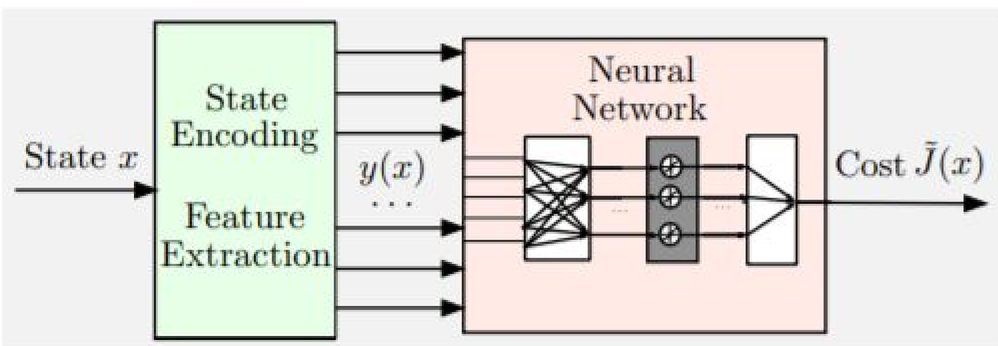
The state encoding operation that transforms x into the neural network input y(x) can be instrumental in the success of the approximation scheme. Examples of possible state encodings are components of the state x, numerical representations of qualitative characteristics of x, and more generally features of x, i.e., functions of x that aim to capture “important nonlinearities” of the optimal cost-to-go function.

We take the fidelity vector as an encoded state corresponding to current quantum state and the action (actions : 1) Use GRAPE control 2) Use free evolution) . For a given initial state and final state, the vector corresponding to the fidelity between the intermediates states and the final state, is represented as y(x). This y(x) will summarize the state properties till that point. (ie the path)



Algorithm:

1) Find the Hamilonian for a Unitary operation using the GRAPE algorithm (This is the heuristic)
2) Lets say the above algorithm computes the Dynamic Decoupling Sequence for the unitary in time T
    a) Then divide the time frames into T/N slots (N is an input parameter) t0, t1, t2 … tn (where t0 =0 and tn = T)
    b) Compute the state the initial state has evolved into during the time period t0 to t1 under the unitary operator Hamilonian (H_u) (found using grape algorithm)
    c) Compute the state the initial state has evolved into during the time period t0 to t1 Under free evolution hamiltonian(H_f)

    d Find “k” intermediate states from t0 to t1 (ie evolution path) under both hamiltonian. Conver the states into two 1-D array. Let the vector corresponding to H_u be S_u and H_f be S_f

    e) Convert the S_u and S_f into fidelity vector F_u and F_f with respect to the initial state in the next time slot t2.

f) Use the neural network trained for fidelity vector cost minimization to find the cost to go function, Accordingly select the right action.
g) Choose either S_u or S_f as the state for the slot t0-t1. Choosing S_u is the action equivalent to selecting the control sequence generated by GRAPE. Choosing S_f is equivalent to choosing free evolution over GRAPE as the action

h) Continue this process for all time slots

Rollout
In the previous step we only used free evolution as an alternative to choosing GRAPE. In the rollout the solution for above step will be the heuristic updation and will introduce other actions such as pulse for sigmax(). We continue these steps to add more operations.

Ultimately the objective is to have a path with minimal cost (based on the fidelity variations) which can be converted to a decoupling sequence.