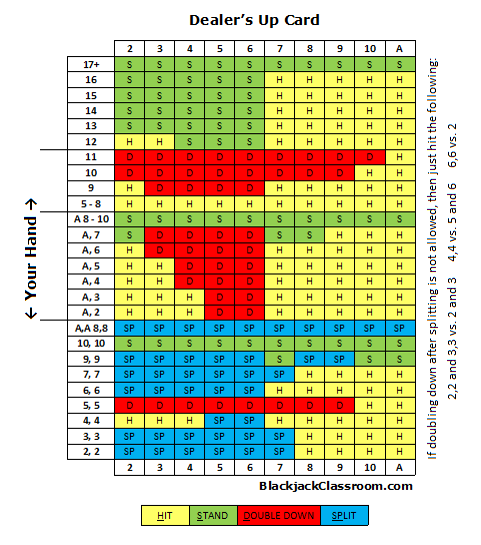
**NML 502 Final Project Proposal**

Jeremy David, Ken Groszman

3/20/2017

Statement of Problem:

Blackjack, sometimes called twenty-one, is the world’s most popular casino game [1- Scarne Complete Guide to Gambling]. It is also commonly cited as the casino game where the player has the best odds of winning, with the most widely accepted “optimal strategy” (below) giving the player 49-51 odds of winning in the long-run. In this project, we will teach an artificial neural network how to play blackjack and see what strategy it suggests to maximize profitability in different blackjack situations (varying the number of decks, number of players at the table, etc.). As an extension, we will see if we can simulate teaching our network how to “count cards”, a strategy used by professional gamblers to increase the probabilities of winning. We hope that through this project, we will either confirm or debunk the common blackjack “optimal strategy” and gauge the effect of a variety of factors on a player’s profitability in a blackjack game.



**Figure 1:** Simple optimal blackjack strategy [2- DroidPoker.com]

Data Description:

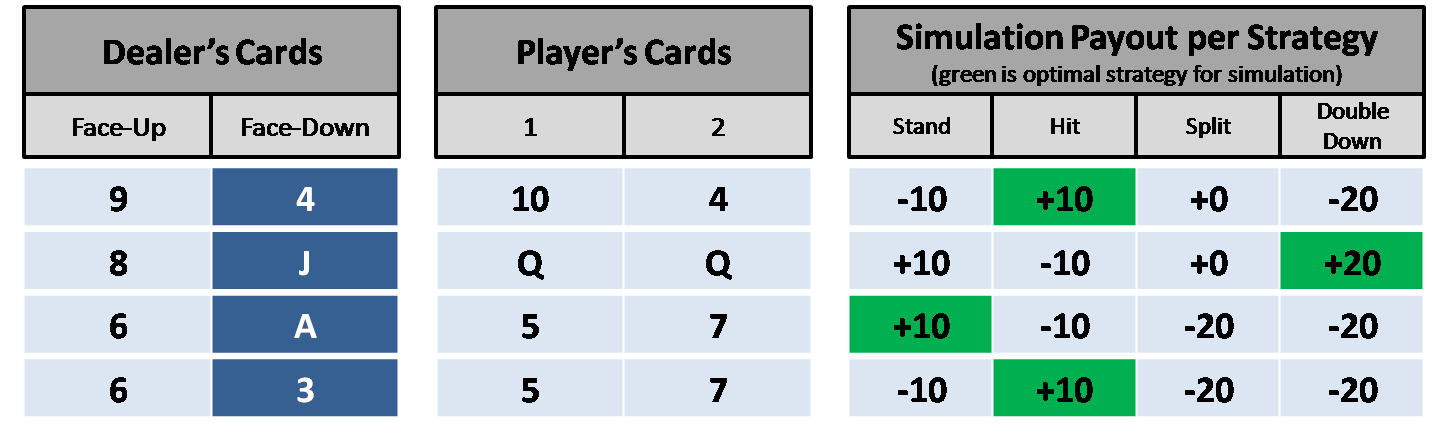
In order to accomplish the goals of this project, we will generate training and testing data simulating the results of real hands of blackjack in different situations, including:

* The number of decks being used to play
* The number of players at the table (and by extension, the number of cards visible to the player)
* Different rules of play (blackjack pays out 3-2 vs. 6-5)
* Multiple hands played from the same set of decks (i.e. the ability to “count cards”)

*Generating Data*:

We will use MATLAB to create a vector of cards corresponding to the number of decks which will then be randomly permuted to simulate shuffling. Cards will be “drawn” from this deck as needed in order to simulate a round of the game. The dealer’s strategy will be simulated, according to common casino rules (hit anything less than 16 or a soft 17). We will then simulate all of the player’s different possible strategies (hit, stand, split, double down) to reveal which strategy would have given the greatest payout.

An example of this sort of simulation for one player is below. The green box indicates which strategy would have given the optimal payout in this simulation. Notice that the simulation is stochastic and the same situation would not always give the same payout (as in rows 3 and 4 below). We can train over an arbitrarily large number of simulations, over which some strategy should prevail as optimal for that situation.



**Figure 2:** Four examples of simulated blackjack hands, with the green box indicating what would be the optimal strategy.

Testing data can be generated in the same manner, with the network selecting which strategy it think would be optimal. The efficacy of the network can be measured as the net “payout” after a certain number of simulated hands. This makes the results of our network consequential and transferrable to a real life situation.