Neural Network for Playing Blackjack

Jeremy David & Kenny Groszman

NML 502

April 20, 2017

What is blackjack?

Description:

- Blackjack is the most popular casino game in the world [1]
- Object is to beat the dealer by having a higher score without exceeding 21
  - Face cards are worth 10
  - Aces are worth 1 or 11 (player's choice)

Strategy:

- Players may double down or split at the beginning of the hand
- Players then hit (draw a card) or stay (accept their sum)
- Dealer must follow pre-specified hit-stay rules (hit up to 16 and soft 17)
- Well-defined "optimal strategy" gives casino 0.5% edge in game [2]

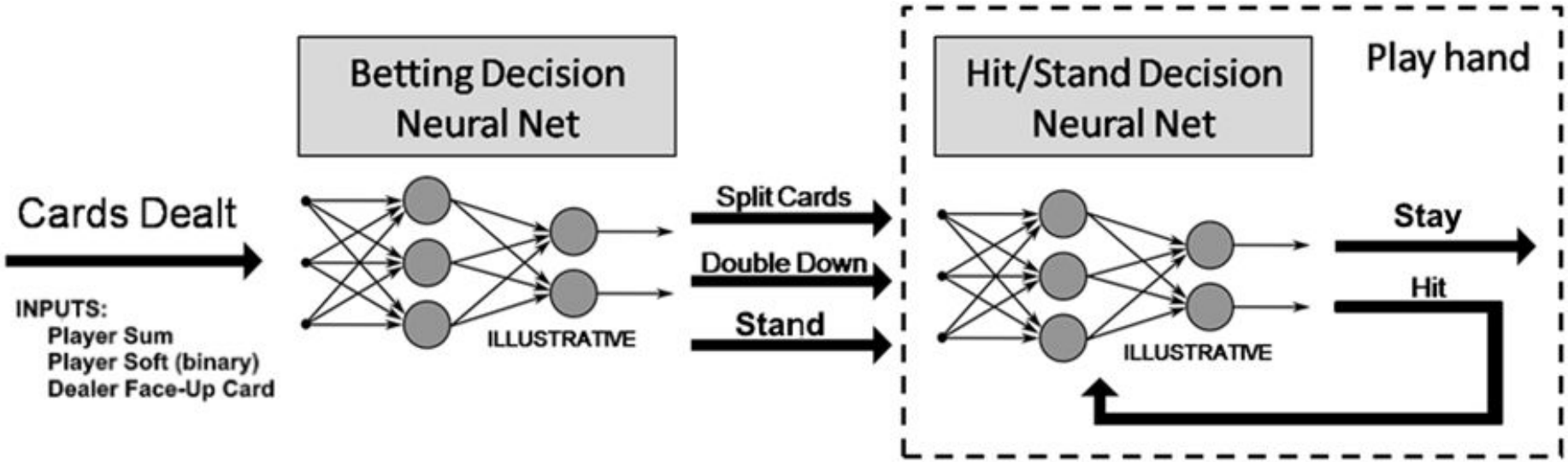
[1] *Scarne's New Complete Guide to Gambling*, p. 342,
[2] https://www.blackjackapprenticeship.com/resources/blackjack-strategy-charts/

Objectives

1. Test behavior of backpropagation neural network with stochastic data
    - Error comes from multiple identical input samples with different desired outputs
    - Attempt to reduce error associated with stochastic data

2. Train neural network to make decisions that are as profitable as possible
    - Compare profitability of neural network decisions vs. pre-defined “optimal strategy”

Network Scheme Structure



Data Generation

For each data set

1. Randomly generate 6 decks of cards
2. Deal cards to dealer and player
3. Play dealer's hand according to pre-specified dealer rules
    - Hit until sum is either 16 or soft 17
4. Play player's hand from omniscient point of view
    - Play all possible strategies and choose best-payout strategy as optimal
    - In case of tied payout, choose optimal strategy based on logic rules (next slide)
5. Use optimal strategy and dealt cards to generate neural network input
6. Repeat Steps 1-5 until satisfied with sample size

Simulating the 2 Different Decisions

Split/Double-Down/Stand Decision

Split only allowed if first 2 dealt cards have same value

- Double bet, cards split into 2 hands

Double down doubles bet and allows only one more hit

Stand keeps bet and play hand

Optimal strategy is the one that, if the player plays perfectly, earns the player the biggest payout

Hit/Stay Decision

Hit choose to add another card to your hand, which is added to your score

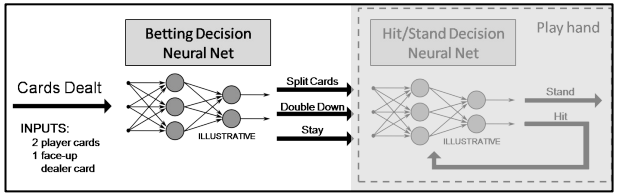
- However, if you exceed 21, you lose

Stay choose to add another card to your hand, which is added to your score

- However, if you exceed 21, you lose

Optimal strategy is the one that gives you the best chance to beat the dealer

Network 1: Split/Double-Down/Stand

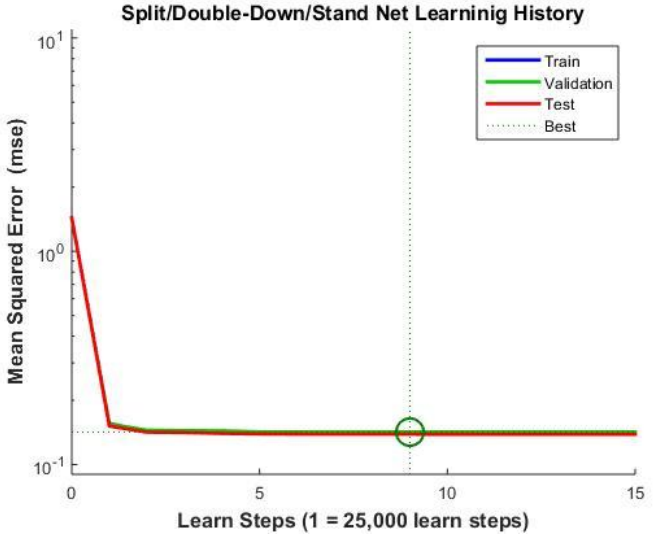


Inputs:

1. Player Sum
2. Player Soft
   - Binary
3. Dealer Card
4. Splitting allowed
   - Binary

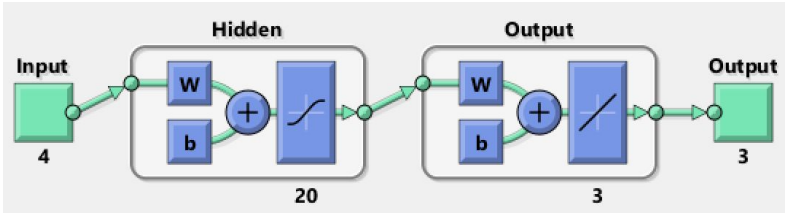
Outputs:

1. One-in-c encoded optimal strategies (3 classes)

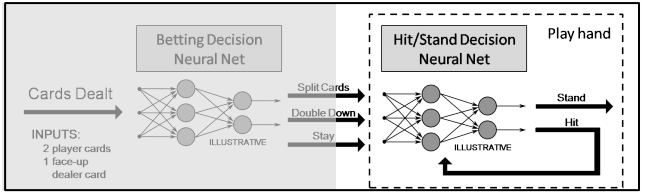


Selected Network Information:

- MATLAB *Neural Network Toolbox* LM backpropagation network
- 4 (+1) - 20 (+1) - 3
- Online training
- MSE error measure
- Decaying learning rate



Network 2: Hit/Stay Decision

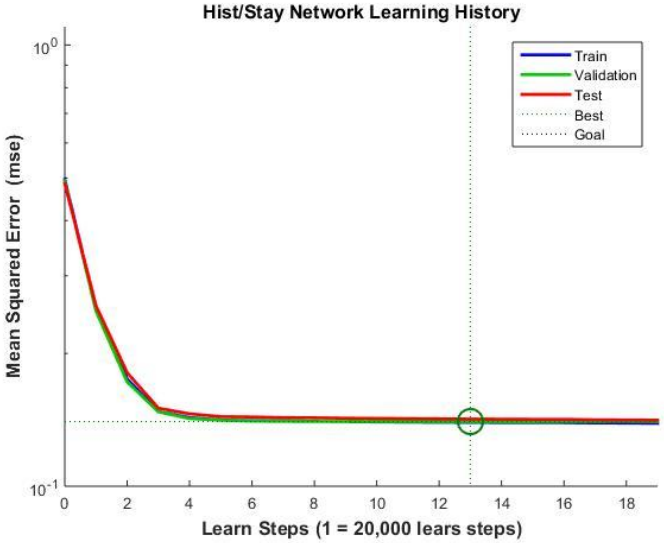


<u>Inputs</u>:

1. Player Sum
2. Player Soft
   - Binary
3. Dealer Card

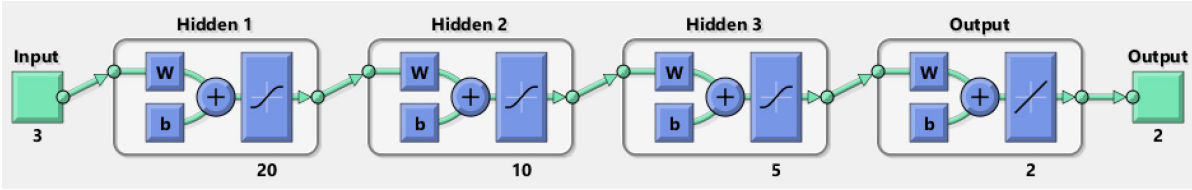
<u>Outputs:</u>

1. One-in-c encoded optimal strategies (2 classes)



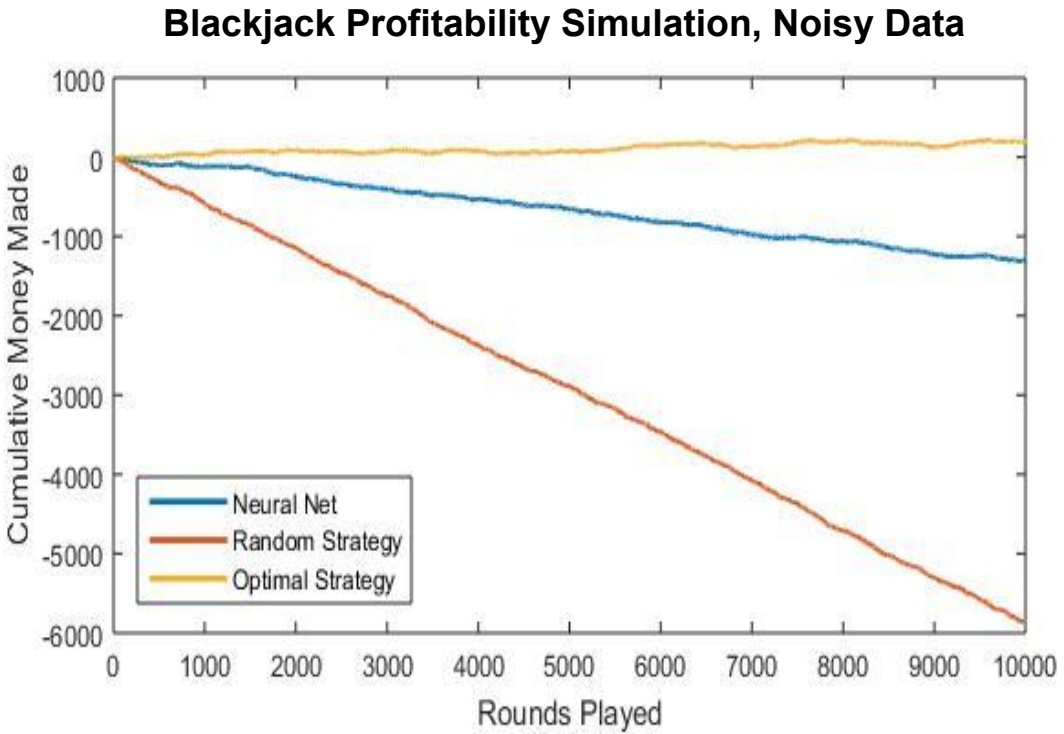
<u>Selected Network Information:</u>

- MATLAB *Neural Network Toolbox* LM backpropagation network
- 3 (+1) - 20 (+1) - 10 (+1) - 5 (+1) - 2
- Online training
- MSE error measure
- Decaying learning rate



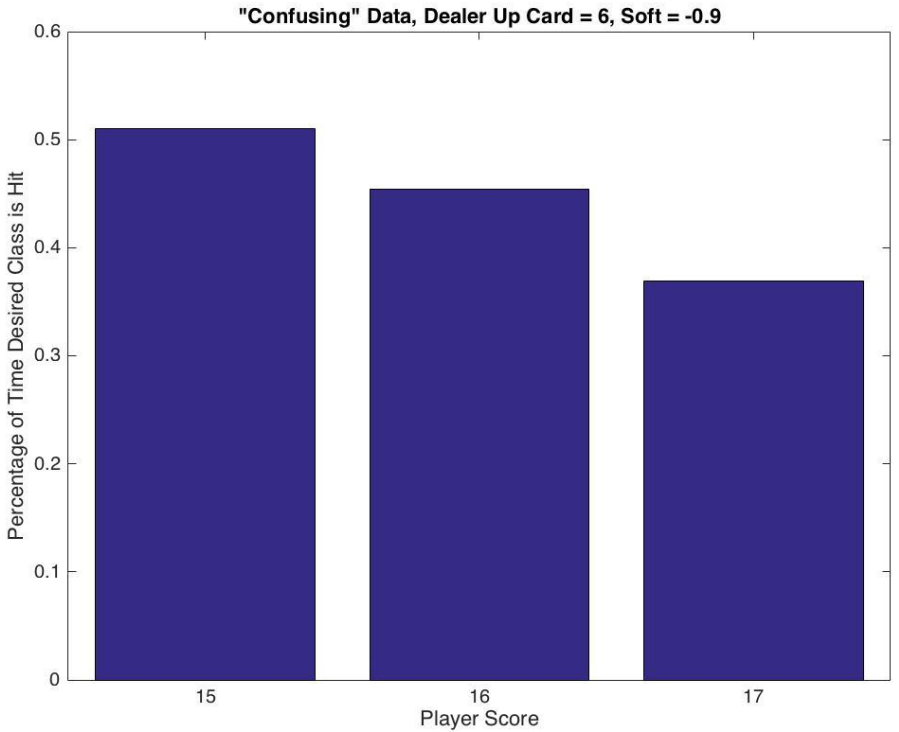
Results #1: Network Confused by Noisy Input Data

- All training data from 20,000 data generation simulations used
- Network plays optimal strategy in 80% of situations
- Input data “confused” network during training

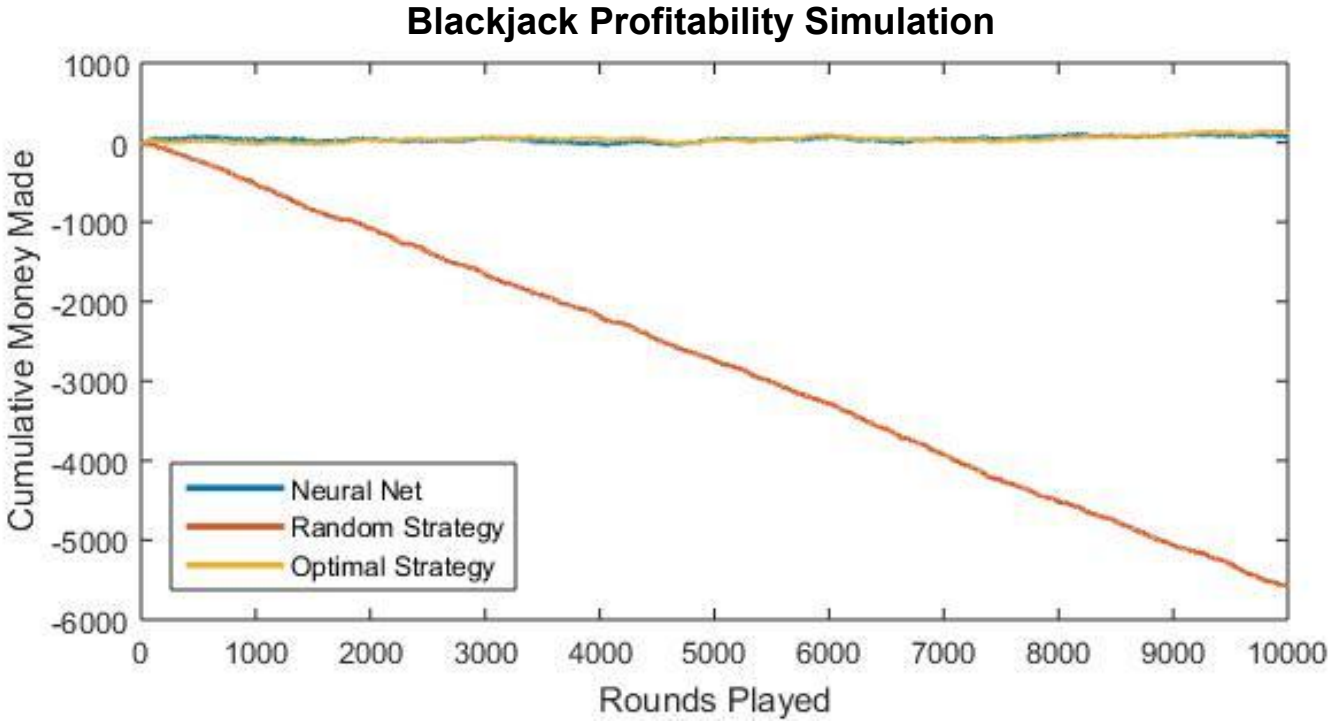


Results #1: Network Confused by Noisy Input Data

- Inconsistent outputs for the same input data confuses network
- Network is able to make predictions, but struggles with more confusing inputs
- Our solution: remove noise by removing duplicates and using most probable output



Results #2: Optimal Strategy Replicated



Conclusions/Future Work

1. BP Paradigm is not good with noisy or contradicting data

2. After noise removal, BP Paradigm was able to play blackjack on par with the optimal strategy

3. Future work investigating probabilistic networks, and adding a “counting cards” feature would likely yield better results