Classification Model for Poker Hands

Zachariah Ly
BSc. Computer Science
Mount Royal University
Calgary, AB
zly859@mtroyal.ca

Abstract

Poker hand classification could be done as an algorithm. However, recognition of poker hands in a variety of settings (whether computed in game engine or visually in a physical environment) is another feat that benefits from being learned. Such a classification model is potentially an accessory to existing poker-related projects as a way to recognize card(s) potential and poker hands. The report explains the casus actio for the model and further steps for the project. The project code can be found here: finalProjML.ipynb

1 Introduction

The classification model for learning and identifying properties of poker hands was done as a final project for COMP 4630. The primary dataset simply consists of 11 attributes, with 5 referring to card suit and another 5 referring to card rank of 5 cards within a row. The last row is a class label stating the 5-card's poker hand rank. This model would augment a poker bot's aspect of game-state, in particular play with realistic yet calculated behaviour in an active poker match (eg. flush flop, player(s) betting after certain card(s) are shown, etc.). Learning what a poker hand is not limited to the poker hands themselves, but also the potential of other cards upcoming and possible cards in another player's hand. The value in this is that with poker being an imperfect information game [1], thus making decisions with non-absolute details.

The original project scope would've been to create a model that can operate from start to finish of a single poker round. However, constraint of time, complexity, and technical skills has reduced the scope to the basis classification model of recognizing poker hands. The optimal project would finish with a model that can take information from 7 cards and determine the best poker hand made from a "7 cards pick 5 cards" scenario. One step beyond a reasonable project scope (but critical for building an effective poker bot) for rational decision making and inductive logic. On a player's action, such a model would allow recognition of shown cards and their potential that would lead to better poker hands that a bot would play towards. This could be what the bot can see from pocket cards and community cards, or how other players/bots use their action as cards are revealed [2]. From another final project from a prior year [3], the students have made gains in their approach of predicting poker hand strength based on betting and checking behaviours. This model would be accessory to recognition of hands and complement a further step of predicting these hands, whether potential upcoming hands, what another player has based on action, or both at the same time.

For this project, only the backup goal of having a basic classifier for 5-card poker hands was achieved. One issue was the necessity of creating an entire dataset post-training which the dataset will include 2 more cards to total 7-cards per row in a dataset. As well, time allocated to the project was compromised by a hasty leave from the province and handling death & funeral processions until I could return to finishing this project. At this point, I am just happy to have been able to submit something somewhat sensible.

2 Dataset and Preprocessing

This 25,000 row dataset from Kaggle represents poker hands within 11 features: 10 attributes and a numerical class ranging from 0 to 9. There are two categories for the attributes. The first being the card suit (spades, hearts, diamonds, clubs) which have no numerical rank. The second attribute is the rank of the card (from 1-13), which is obviously ranked with 2-10 carrying their regular values. 11, 12, and 13 represent the Jack, Queen, and King face cards while 1 represents Aces. There are five cards per row, and each has an assigned suit and rank.

With the composition of poker hands, suits correspond to flush-related hands (five cards of the same suit). Rank of card determines most hands, with pairs, three/four of a kind, straight-related, and full house hands. If the model works correctly, there will be a heavier emphasis on the card rank, rather than suits which a model should basically determine if all cards in the hand consist of the same suit.

The main issue with this is that numerically, Ace is the highest rank and can fit at both the top and bottom of rank. Otherwise the dataset attributes are straightforward and complete. No handling of missing value was necessary for this database. With better setup, pairing of suit and rank in preprocessing would've been an asset for training the data in regards to correlations for both 5-card and 7-card scenarios of predicting poker hands.

Table 1: Poker Hand Ranks

Rank	Poker Hand
0	High Card
1	One Pair
2	Two Pair
3	Three of a Kind
4	Straight
5	Flush
6	Full House
7	Four of a Kind
8	Straight Flush
9	Royal Flush

3 Model Architecture and Training

This model is simple, with the input layer taking the shape of the 10 attribute parameters that make up the 5 cards per row. Then running them all through two dense layers, the first one being 128 neurons, then the second layer reduced to half of the neurons. Initially, a single layer or leading with a smaller layer first had slow increases to accuracy and relatively high loss. The output layer devolves into nine neurons that are mapped to poker hands and their rank in table 1. Regardless, the 2-to-1 ratio of the inner layers has decent results of 97% accuracy and 10% loss after training for 100 epochs and testing on the provided .csv dataset available for testing in this database repository.

Albeit, this is a somewhat straightforward dataset that needed further expansion to the project (as mentioned in previous discussions of intended iterations of the project). This model simply recognizes what poker hands are made up of what properties involving the suit, rank, and the combination of both. In hindsight, this model likely will fail at some differentiations such as royal flushes and straight flushes, since they're relatively the same with the exception of royal flushes being a straight flush of all the top cards.

4 Discussion and Conclusion

If I'm being honest, I severely missed the mark for what this project is supposed to be. At this point, I am just submitting what I have in order to attempt to pass the course considering my current circumstances. One issue was identifying the ability to code a database of 7-card hands would've been time consuming, but possible in the timeframe. This was recognized late, though the approach of hard-coding trying each combination of the 7 pick 5 cards scenario was one case that could've been furthered.

As mentioned, the continuation of this project would be adhering to a functional poker bot, with this model being as a foundational step for building a poker bot or anything poker-related. With learning poker hands, the missing iteration of the final project to be up to specification would involve determining the best five cards of seven possible cards. However, if this was completed, this would provide a model that can be expanded to recognize potential poker hands based on 2-4 cards instead of all 5. This is critical for a poker bot to perform poker actions effectively to:

I. Increase stakes by betting or raising, especially depending on potential of current cards.
II. Predict what other players have based on their actions from inductive logic for actions, such as a player raising based on an Ace on the table.

Given better circumstances, the more appropriate project would've been training a basic model, then applying the model to handle 7-cards, first by testing each and every hand possible, followed by having the model take the 7-cards as input, and identify the best hand while mapping the selected 5-cards. This would've made for an appropriate project to scope.

References

[1] Martin, Z. & Michael, J. *Regret Minimization in Games with Incomplete Information*. Edmonton: https://poker.cs.ualberta.ca/publications/NIPS07-cfr.pdf

[2] Oscar, B. *Applying Deep Reinforcement Learning to Poker (2019)*.
https://www.adaltas.com/en/2019/01/09/applying-deep-reinforcement-learning-poker/

[3] James, B., Alex E., Maxwell W. *Predicting Texas Holdem Hand Strength*. Stanford: https://cs229.stanford.edu/proj2013/finalpaper.pdf