Final Project – Review 1

Jacob Taylor Cassady

CECS 590: Intro to Deep Learning Algorithms

2/19/20

# Define Database

Database: Poker Hand Data Set [1]

Retrieved from: University of California Irvine Machine Learning Repository.

Attribute Description: The Poker Hand Data Set contains records of a poker hand consisting of five playing cards drawn from a 52-card poker deck. Each card is defined by two features: one describing the suit of the card and a subsequent feature describing the rank of the card. You can find descriptions of the features in Tables 1 and 2 below.

Table 1 : Attribute Descriptions

|  |  |  |  |
| --- | --- | --- | --- |
| Attributes Name | Datatype | Range | Description |
| S1 | Ordinal | 1-4 | Represents {Hearts, Spades, Diamonds, Clubs} |
| C1 | Numerical | 1-13 | Represents {Aces, 2, 3, 4, …, Jack, Queen, King} |
| S2 | Ordinal | 1-4 | Represents {Hearts, Spades, Diamonds, Clubs} |
| C2 | Numerical | 1-13 | Represents {Aces, 2, 3, 4, …, Jack, Queen, King} |
| S3 | Ordinal | 1-4 | Represents {Hearts, Spades, Diamonds, Clubs} |
| C3 | Numerical | 1-13 | Represents {Aces, 2, 3, 4, …, Jack, Queen, King} |
| S4 | Ordinal | 1-4 | Represents {Hearts, Spades, Diamonds, Clubs} |
| C4 | Numerical | 1-13 | Represents {Aces, 2, 3, 4, …, Jack, Queen, King} |
| S5 | Ordinal | 1-4 | Represents {Hearts, Spades, Diamonds, Clubs} |
| C5 | Numerical | 1-13 | Represents {Aces, 2, 3, 4, …, Jack, Queen, King} |
| Class | Ordinal | 0-9 | See Table 2. |

Table 2: Class Ordinal Descriptions

|  |  |
| --- | --- |
| Class Value | Description |
| 0 | Nothing. |
| 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. |

# Define Problem to Solve

## State of the art

In 2002, Cattral et. al published a paper on a datamining system named RAGA that uses genetic algorithms and genetic programming to extract knowledge from datasets in the form of predictive rules [2]. They applied RAGA to a poker hand dataset with 17 target classes instead of 10. They were able to achieve a training accuracy of 90.39% and a test set accuracy of 57.6% compared to their control algorithm See-5 that achieved 64.25% accuracy on the training set and 36.16% accuracy on the test set.

Several GitHub Repositories and blogs have been found that attempt to design models that accurately predict the poker hand using the dataset described in this paper. Most present the results from classical models as well as deep neural networks. Neural network architectures are typically only a few layers deep and no work has been found that preprocesses data through normalization.

## Case Study

The goal of this final project is to develop a neural network that takes in 10 features, every two features describing one card in a poker hand and outputs the predicted class value describing the hand using class descriptions from poker. This is relevant as there is a large domain of possible poker hands, 311,875,200. A neural network will be designed with goals of associating features to the output at a with a lower computational cost and smaller memory cost than searching a dictionary of 311.875,200 values would. The features of the neural network will include the following attributes: S1, C1, S2, C2, S3, C3, S4, C4, S5, and C5. The output target will be the attribute Class.

# Dataset Description

The dataset contains two sets of data. A training set containing 25,010 labeled instances out of 311,875,200 and a test set containing 1,000,00 labeled instances out of 311,875,200. Tables 3 describes the domain of possible hands. Tables 4 and 5 describe the probability distribution of the train and test datasets. It is important to note the distribution of class ordinals 7 (Four of a Kind), 8 (Straight Flush), and 9 (Royal Flush) are overrepresented in both the training and tests datasets. Histograms were generated for each dataset and can be found below each related table.

Table 3: Poker Hand Domain Class Ordinal Distribution

|  |  |  |  |
| --- | --- | --- | --- |
| Class Value | Description | Instances in Domain | Percentage of Domain |
| 0 | Nothing. | 156304800 | 50.1177394 |
| 1 | One pair. | 131788800 | 42.25690276 |
| 2 | Two pair. | 14826240 | 4.753901561 |
| 3 | Three of a kind. | 6589440 | 2.112845138 |
| 4 | Straight. | 1224000 | 0.392464678 |
| 5 | Flush. | 612960 | 0.196540155 |
| 6 | Full house. | 449280 | 0.144057623 |
| 7 | Four of a kind. | 74880 | 0.024009604 |
| 8 | Straight flush. | 4320 | 0.001385169 |
| 9 | Royal flush. | 480 | 0.000153908 |

Table 4: Training Set Class Ordinal Distribution

|  |  |  |  |
| --- | --- | --- | --- |
| Class Value | Description | Instances in Dataset | Percentage of Dataset |
| 0 | Nothing. | 12493 | 49.95201919 |
| 1 | One pair. | 10599 | 42.37904838 |
| 2 | Two pair. | 1026 | 4.102359056 |
| 3 | Three of a kind. | 513 | 2.051179528 |
| 4 | Straight. | 93 | 0.371851259 |
| 5 | Flush. | 54 | 0.215913635 |
| 6 | Full house. | 36 | 0.143942423 |
| 7 | Four of a kind. | 6 | 0.023990404 |
| 8 | Straight flush. | 5 | 0.019992003 |
| 9 | Royal flush. | 5 | 0.019992003 |

A screenshot of a cell phone

Description automatically generated

Figure 1: Train Set Target Distribution

Table 5: Test Set Class Ordinal Distribution

|  |  |  |  |
| --- | --- | --- | --- |
| Class Value | Description | Instances in Dataset | Percentage of Dataset |
| 0 | Nothing. | 1302540 | 50.1209 |
| 1 | One pair. | 1098240 | 42.2498 |
| 2 | Two pair. | 123552 | 4.7622 |
| 3 | Three of a kind. | 54912 | 2.1121 |
| 4 | Straight. | 10200 | 0.3885 |
| 5 | Flush. | 5108 | 0.1996 |
| 6 | Full house. | 3744 | 0.1424 |
| 7 | Four of a kind. | 624 | 0.023 |
| 8 | Straight flush. | 36 | 0.0012 |
| 9 | Royal flush. | 4 | 0.0003 |

A screenshot of a cell phone

Description automatically generated

Figure 2: Test Set Target Distribution

# Activity Calendar

|  |  |
| --- | --- |
| Week | Activity |
| 1 | Split data into features and target. Normalize features using min-max normalization. |
| 2 | Implement Classical Solutions (KNN, Linear SVC, Random Forest) |
| 3 | Develop tools and metrics for graphically comparing model performances. Analyze classical solutions. |
| 4 | Implement neural network model. |
| 5 | Iterate over model tuning hyperparameters, improving performance. |
| 6 | Iterate over model tuning hyperparameters, improving performance. |
| 7 | Perform analysis of neural network model’s performance with that of classical solutions. |
| 8 | Write final report and create presentation. |

# References

[1] D. Dheeru and C. Graff. *Poker Hand Data Set*, University of California, School of Information and Science. [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/Poker+Hand>

[2] Cattral, R., Oppacher, F., & Deugo, D. (2001). Evolutionary Data Mining With Automatic Rule Generalization. World Scientific and Engineering Academy and Society, Recent Advances in Computers, Computing and Communication. Crete, Greece.