

MINISTÉRIO DA DEFESA
EXÉRCITO BRASILEIRO
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA
INSTITUTO MILITAR DE ENGENHARIA
CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

CARLOS ALBERTO VILLOTE DE OLIVEIRA

PAULO EDUARDO ALTHOFF

RENAN GEMIGNANI

EMPREGO DE REDES NEURAIIS PARA O *POKER TEXAS HOLD'EM*

RIO DE JANEIRO

2014

INSTITUTO MILITAR DE ENGENHARIA

CARLOS ALBERTO VILLOTE DE OLIVEIRA

PAULO EDUARDO ALTHOFF

RENAN GEMIGNANI

EMPREGO DE REDES NEURAIIS PARA O *POKER TEXAS HOLD'EM*

Projeto de Final de Curso apresentado ao
Curso de Engenharia de Computação do In-
stituto Militar (IME) de Engenharia, como
parte das exigências do IME.

Orientador: Prof. Marcos Veloso Peixoto,
D.I.

RIO DE JANEIRO

2014

c2013

INSTITUTO MILITAR DE ENGENHARIA

Praça General Tibúrcio, 80 Praia Vermelha

Rio de Janeiro - RJ CEP: 22290- 270

Este exemplar é de propriedade do Instituto Militar de Engenharia, que poderá incluí-lo em base de dados, armazenar em computador, microfilmar ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es) e do(s) orientador(es).

----- Althoff, Paulo Eduardo/ Gemignani, Renan/ Oliveira, Carlos Alberto Villote de
Emprego de Redes Neurais para o *Poker Texas Hold'em*, no Rio de Janeiro /
Carlos Alberto Villote de Oliveira, Paulo Eduardo Althoff, Renan Genignani
- Rio de Janeiro : Instituto Militar de Engenharia, 2013.

n f. : il., graf., tab. : - cm.

Dissertação (graduação) - Instituto Militar de Engenharia, 2014.

1. Formação.

INSTITUTO MILITAR DE ENGENHARIA

CARLOS ALBERTO VILLOTE DE OLIVEIRA

PAULO EDUARDO ALTHOFF

RENAN GEMIGNANI

EMPREGO DE REDES NEURAIIS PARA O *POKER TEXAS HOLD'EM*

Projeto de Final de Curso apresentado ao Curso de Engenharia de Computação do Instituto Militar (IME) de Engenharia, como parte das exigências do IME.

Aprovado em __/__/____ pela seguinte Banca Examinadora:

Orientador: Marcos Veloso Peixoto, D.I.

Alex Garcia, D.Sc.

Ricardo Choren Noya, D.Sc.

RIO DE JANEIRO

2014

A primeira coisa a entender é que você não entende.
SOREN AABYE KIERKEGAARD

SUMÁRIO

SUMÁRIO	6
LISTA DE ILUSTRAÇÕES	8
LISTA DE TABELAS	10
1 INTRODUÇÃO	14
1.1 MOTIVAÇÃO	15
1.2 OBJETIVO	15
1.3 JUSTIFICATIVA	16
1.4 METODOLOGIA	16
1.5 ESTRUTURA	17
2 REDES NEURAIS	18
2.1 HISTÓRIC	18
2.2 CÉREBRO HUMANO	19
2.3 COMPONENTES DAS REDES NEURAIS COMPUTACIONAIS	20
2.3.1 CONEXÕES	21
2.3.2 NÚCLEO DE PROCESSAMENTO	22
2.3.3 FUNÇÕES DE LIMAR	22
2.3.4 PADRÕES	25
2.4 CLASSIFICAÇÕES DE REDES NEURAIS	26
2.5 ADALINE	27
2.6 <i>BACKPROPAGATION</i>	27
3 O PÔQUER	29
3.1 BARALHO	29
3.2 RANKING DE COMBINAÇÕES DE CARTAS	30
3.3 AÇÕES	32
4 CONSTRUÇÃO E TESTES DA API BASE DE REDE NEURAL	34
5 ESTRUTURA DA SOLUÇÃO	35
5.1 ESTRUTURA GERAL	35

5.2	MODELAGEM DO Oponente	36
5.2.1	ESTIMAÇÃO DAS PROBABILIDADES	36
5.3	REDE PRINCIPAL E CLASSIFICAÇÃO EM PERFIS	37
6	SELEÇÃO DE JOGADORES PARA TREINAMENTO DA REDE	40
6.1	IRC POKER DATABASE	40
6.2	SELEÇÃO DOS DADOS	42
7	CONCLUSÃO	44
8	CRONOGRAMA DE ATIVIDADES	45
9	REFERÊNCIAS	46

LISTA DE ILUSTRAÇÕES

2.1	Esquema de um neurônio	20
2.2	Esquema de um neurônio artificial	21
2.3	RNA na forma de grafo	22
2.4	Função Linear	23
2.5	Função Step	23
2.6	Função Rampa	24
2.7	Função Gaussiana	24
2.8	Função sigmóide	25
2.9	Conexão feed-forward	26
2.10	Topologia Backpropagation	28
3.1	Ouros, copas, paus e espadas	29
3.2	Ordem decrescente de valor	29
3.3	Par de dez	30
3.4	Dois pares de sete com quatro	30
3.5	Trinca de cinco	30
3.6	Sequência de dez a ás	31
3.7	<i>Flush</i> de espadas	31
3.8	<i>Full House</i> de dama com dois	31
3.9	Quadra de rei	31
3.10	<i>Straight Flush</i>	31

3.11	<i>Royal Straight Flush</i>	32
5.1	Fluxo de dados da rede neural principal	38
5.2	Diagrama das classes utilizadas no aprendizado	39
6.1	Diagrama relacional do banco de dados utilizado	40
6.2	Diagrama de classe para as classes básicas	41
6.3	Diagrama de classe para a fachada	42

LISTA DE TABELAS

5.1	Parâmetros de entrada da RN utilizada para modelar o oponente	37
5.2	Parâmetros de saída da RN utilizada para modelar o oponente	37
5.3	Parâmetros de entrada da RN utilizada para classificar o oponente para cada uma das jogadas em uma janela deslizando.	38
5.4	Parâmetros de saída da RN utilizada para classificar o oponente	38

LISTA DE SIGLAS

RN	Rede neural
RNA	Rede Neural Artificial

RESUMO

Redes neurais vem sendo utilizadas para a resolução de problemas de decisão com grande sucesso, especialmente problemas de informação imperfeita ou ruidosa, dos quais o pôquer é um exemplo. Neste trabalho, orientado ao jogo do pôquer *Texas Hold'em*, é proposta a utilização de várias redes neurais na modelagem de oponentes heurística-mente selecionados para permitir uma melhor análise de perfis de jogadores, resultando em formulações implícitas de contra-estratégias pela rede. A adequação de redes neurais nesses casos já foi comprovada em diversos trabalhos.

Palavras-chave: redes neurais, aprendizado de máquina, pôquer.

ABSTRACT

Neural networks have been used for solving decision problems to great success, especially problems with imperfect/noise-laden information, of which poker is a prime example. Here, we analyze Texas Hold'em poker, proposing the usage of multiple topologically-arranged neural networks to model heuristically selected opponents to allow for player profile analyses and subsequently formulate counterstrategies with the neural network.

Keywords: Neural networks, machine learning, poker.

1 INTRODUÇÃO

No meio militar, são inúmeras as situações onde práticas de inteligência artificial e aprendizado de máquina são utilizadas, como por exemplo decisões em combate militar, reconhecimento de alvos, robótica, sensoriamento remoto, biometria, etc.

Na área de decisões estratégicas, uma forma de simular resultados de aprendizados é sua aplicação em jogos, por serem situações que promovem interações entre adversários para comparar escolhas que busquem um objetivo comum. Dentre os jogos mais utilizados nessas simulações, o pôquer é um dos mais presentes, por possuir uma extensa bibliografia de uso em aprendizado. Além disso este jogo possui situações interessantes do ponto de vista de simulações militares, por conter informações ocultas do adversário, e a decisão ser influenciada por blefes e jogadas do oponente.

A maior parte dos estudos na área advém do grupo de inteligência artificial aplicada ao pôquer da Universidade de Alberta. Uma parte dos estudos e implementações de software de pôquer faz uso de estratégias e contra-estratégias baseadas na teoria de equilíbrio de Nash [9], com algoritmos de avaliação sendo utilizados, principalmente, com o objetivo de selecionar a estratégia mais interessante para a situação atual conforme resposta do oponente. Deste modo, almeja-se manter a menor quantidade de tempo possível em uma estratégia percebida como não-ideal. Esta abordagem é baseada mais fortemente no fundamento matemático da teoria do jogo do que no melhoramento da estratégia em si.

Redes Neurais consistem uma rede de unidades de processamento que através da comunicação frequente, tenta dar características de aprendizado à máquina, e são utilizadas nas mais diversas áreas e pesquisas. E são amplamente utilizadas em problemas de informação imperfeita. Mesmo softwares baseados em estratégias fixas se valem, em alguns casos, da utilização de algoritmos de aprendizado de máquina baseados em redes neurais. Este tem o objetivo de realizar, ao menos, a modelagem dos oponentes [10] de modo a prever as probabilidades de o oponente responder de certa forma dado seu histórico de jogadas. As máquinas descritas em [1] exemplificam isso, na qual a abordagem utilizada para construção do agente-fim não é descrita. Entretanto, redes neurais são usadas para prever as jogadas do oponente, iniciando-se com uma rede limpa que é consequentemente adaptada à medida que o oponente age. Uma desvantagem dessa utilização da rede neural é a possível infactibilidade de se treinar a rede satisfatoriamente em tempo real em um ambiente com agentes humanos. Mesmo com essa desvantagem, a modelagem de redes neurais se mostra mais promissora que uma modelagem

baseada puramente em pesos relativos a cada tupla ação-mão possível, tal como em [2].

Uma abordagem em que o agente objetivo-fim do estudo é de fato baseado em redes neurais está descrita em [5], porém sem tomar em consideração as especificidades do modo de jogo do oponente, levando a uma abordagem relativamente simples no que concerne a modelagem da rede em si, e focando na evolução dos agentes pré-determinados via combinação dos mesmos e utilização de heurísticas co-evolucionárias, como o algoritmo halls da fama, por exemplo, que realiza a seleção dos agentes que mais se adaptam, descartando modelos menos eficientes.

1.1 MOTIVAÇÃO

No emprego militar da guerra, mesmo com esforços de inteligência e contra-inteligência, a situação do campo de batalha é quase sempre uma representação incompleta da realidade, o que impossibilita a tomada de decisões pela simples determinação de todos os resultados possíveis por simulação. Isto torna métodos de aprendizado de máquina importantes para o emprego militar [8]. Deste modo, cria-se a necessidade de uma verdadeira inteligência artificial que tenha capacidade de tomar decisões baseada em informação incompleta.

Devido às características de informação incompleta do jogo, o pôquer é frequentemente estudado na área de aprendizado de máquina tanto como objetivo fim quanto como um campo de testes de abordagens de aprendizado[9, 10].

1.2 OBJETIVO

No presente trabalho, objetiva-se melhorar o modo com que uma análise prévia do oponente pode ser utilizada como entrada adicional para uma rede neural, de forma a conseguir classificá-lo em um, dentre uma série de perfis distintos pré-gerados, treinados e também mod-

elados utilizando redes neurais. Tal informação será utilizada como parâmetro de entrada pela rede para gerar uma contra-estratégia específica para determinado perfil de oponente.

1.3 JUSTIFICATIVA

Redes neurais tem se mostrado ferramentas úteis tanto para a modelagem de oponentes no pôquer, quanto para a resolução de problemas de decisão, como já foi mostrado em trabalhos correlatos[5, 10]. Contudo, as implicações decorrentes do uso de uma rede neural como agente *stand-alone*, capaz de jogar competitivamente, ainda não foram completamente exploradas. O *Texas Hold'em*, a princípio, não seria adequado ao modelo necessário para a utilização de redes neurais, uma vez que dada uma mesma entrada não existe uma resposta 'correta', já que para qualquer ação, deve-se ser levada em conta a subsequente ação tomada pelo adversário. Desta forma, faz-se necessário utilizar alguma técnica que preveja as respostas mais prováveis do oponente a cada ação tomada, podendo assim julgar a adequação do modelo e realizar uma análise mais acurada. Neste caso o uso de redes neurais se enquadra, pois o jogo passa ser um problema com grande quantidade de tempo e dados disponíveis para realizar um treinamento.

1.4 METODOLOGIA

Na fase de estudos teóricos, objetivou-se o levantamento de dados sobre redes neurais como técnica de aprendizado de máquina, além do estudo do jogo do pôquer em si. No planejamento, foram construídas a topologia da rede a ser utilizada para modelar o oponente, e a diagramação de classes do sistema. Foram coletados dados para o futuro treinamento da rede em um banco disponibilizado pela Universidade de Alberta.

Com base na pesquisa teórica e nos exemplos conhecidos de redes desenvolvidas, foi mode-

lado a RNA para o sistema de treinamento, já posteriormente desenvolvida e o reconhecedor de perfis, a ser desenvolvido na próxima etapa.

Após o término do desenvolvimento e treino da RNA, testes serão feitos quanto ao desempenho da abordagem proposta nesse trabalho, para possíveis comparações com abordagens já conhecidas.

1.5 ESTRUTURA

No capítulo 2, será apresentada a fundamentação teórica acerca de redes neurais mínima necessária para o entendimento do trabalho. No capítulo 3, um breve resumo das regras do pôquer *Texas Hold'em* destinado ao leitor que não é familiarizado com o mesmo. No capítulo 4, é mostrada a topologia a ser utilizada pela rede de simulação do oponente. No capítulo 5, é descrita a interface com o banco de dados de mãos e as heurísticas utilizadas para selecionar os jogadores utilizados para compor o conjunto de treinamento de cada perfil, bem como os jogadores utilizados para treinar a RN objetivo-fim.

2 REDES NEURAIS

Uma importante técnica de aprendizado de máquina atualmente, é a estrutura conhecida como rede neural, que consiste em formar uma rede de unidades de processamento que enviam mensagens umas as outras, tentando imitar o funcionamento do cérebro humano. Por esse motivo, é a técnica que mais auxilia no entendimento da inteligência humana, e que se espera obter uma inteligência artificial mais próxima daquela. O exposto abaixo foi baseado no exposto em [7].

2.1 HISTÓRIC

As primeiras informações sobre neurocomputação surgiram em 1943, em artigos de McCulloch & Pitts , que inauguraram o estudo de redes neurais. Os autores fizeram uma analogia entre células nervosas vivas e o processo eletrônico, simulado o comportamento do neurônio natural, no qual o neurônio possuía uma saída que era uma função da soma de valor de suas diversas entradas. Eles introduziram o cálculo de pesos e unidades limiares.

Em 1948, o livro *Cybernetics* de Wiener descreve alguns conceitos sobre controle, comunicação e processamento estatístico de sinais.

Em 1950, por meio do livro *The Organization of Behavior*, Hebb descreveu um sistema de aprendizado por correlação dos neurônios que acabou dando origem a Regra de Aprendizagem de Hebb. No ano seguinte, Minsky construiu o primeiro neuro computador.

Em 1956, a Inteligência Artificial foi oficialmente criada por McCarthy, Minsky, Shannon, Newell & Simon, Samuel e outros. Em Dartmouth College nasceram os dois paradigmas da IA: a simbólica, que tenta simular o comportamento inteligente humano desconsiderando os mecanismos responsáveis por tal, e a conexionista, que acredita que construindo-se um sistema que simule a estrutura do cérebro, este sistema apresentará inteligência.

No ano seguinte, Rosenblatt criou o Perceptron, um modelo cognitivo que consistia de unidades sensoriais conectadas a uma única camada de neurônios de McCulloch & Pitts,

demonstrando que se fossem acrescentadas sinapses ajustáveis, as redes neurais de McCulloch & Pitts podem ser treinadas para classificar padrões em classes linearmente separáveis, convergindo em um número limitado de passos.

Em 1969, Minsky e Papert publicam *Perceptron*, que expõe as limitações do modelo de Rosenblatt, provando que tais redes não são capazes de resolver uma ampla classe de problemas devido às restrições de representam, tendo impacto devastador no interesse dos pesquisadores sobre redes neurais.

Na década de 1970 não houve avanços devido a pouca credibilidade dos recentes estudos na área.

Em 1982, a introdução do modelo conexionista permitiu esclarecer boa parte das dúvidas até então existentes em relação ao processo dinâmico executado por certas redes neurais. Em meados da década de 1980 surgiu a descrição do algoritmo de treinamento *backpropagation*, que se baseia na retropropagação dos erros para realizar os ajustes de pesos das camadas intermediárias. A área de redes neurais teve um desenvolvimento explosivo após a publicação em 1986 do *Parallel Distributed Processing* editado por Rummerhart & McClelland, com a multiplicação exponencial de teses e artigos científicos.

Na década de 1990 ocorreu a integração da pesquisa de redes neurais à corrente principal da inteligência artificial.

Nos anos 2000 as redes neurais foram vistas como um das técnicas de aprendizado de máquina, e não necessariamente como a melhor técnica para muitos problemas.

2.2 CÉREBRO HUMANO

O cérebro humano é composto por unidades chamadas neurônios, cerca de 100 bilhões dessas unidades trabalham em paralelo, enviando sinais eletroquímicos uns aos outros. Apesar de cada neurônio trabalhar na faixa de menos de 1 kHz (uma taxa muito baixa se comparado a um computador moderno que trabalha em GHz), seu poder de processamento é muito elevado devido às múltiplas conexões entre eles e pela comunicação rápida, que permite que os 10^{11} neurônios trabalhem ao mesmo tempo.

A estrutura de um neurônio pode ser dividida em 3 partes: dendritos, corpo celular, axônio (2.1).

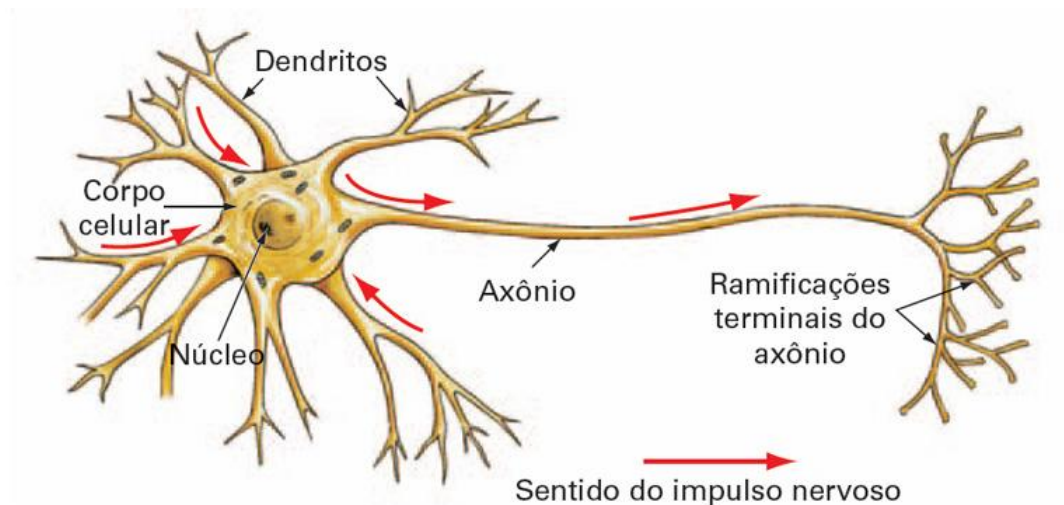


Figura 2.1: Esquema de um neurônio

Os dendritos recebem sinais elétricos dos outros neurônios através das sinapses, que são impulsos nervosos (estímulos) enviados através de ativadores químicos. O corpo celular processa as informações, definindo pesos para as conexões e enviando uma resposta para outro neurônio através do axônio.

Conjuntos de redes de neurônios, são agrupados em áreas de responsabilidades como cerebelo, hipotálamo, córtex, etc.

2.3 COMPONENTES DAS REDES NEURAIIS COMPUTACIONAIS

As redes neurais computacionais, ou redes neurais artificiais (RNA), são feitas a partir de vários neurônios artificiais, que é um modelo eletrônico do neurônio biológico, como representado na figura 2.2.

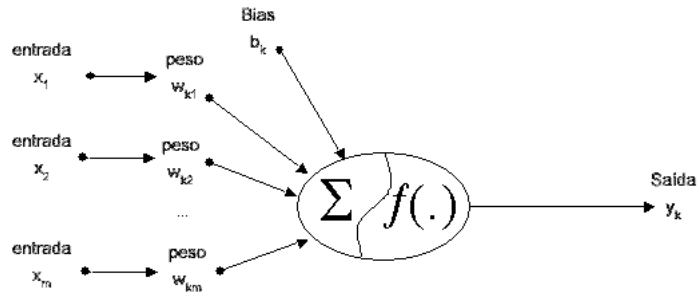


Figura 2.2: Esquema de um neurônio artificial

É possível destrinchar esse funcionamento do neurônio artificial em 4 componentes principais: as conexões, o núcleo de processamento, as funções de limiar e os padrões.

2.3.1 CONEXÕES

Imaginando a RNA como um grafo monodirecionado, tem-se as conexões entre os nós fazendo o papel dos axiônios (ao enviar a informação de saída de um neurônio) e dos dendritos, (associando um peso a cada entrada). Esse peso nada mais é que um valor que será ajustado durante o treinamento da rede neural, podendo ser positivos (excitam a entrada) ou negativos (inibem a entrada). A informação percorre esse grafo por essas conexões para chegar aos núcleos de processamento para ser analisada.

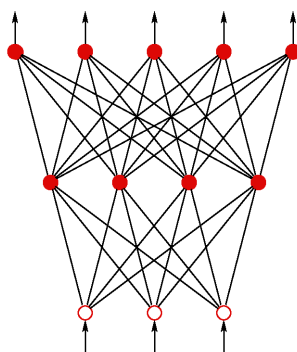


Figura 2.3: RNA na forma de grafo

2.3.2 NÚCLEO DE PROCESSAMENTO

O núcleo de processamento é a parte da rede neural onde é realizado o processamento das entradas vindas das conexões. Na analogia com o grafo, um núcleo de processamento seria um nó com várias entradas e uma única saída, obtida pelo processamento das informações de entrada, geralmente uma combinação linear destas ou uma análise comparativa dessa combinação. Essa saída só pode depender dos elementos de entrada, sem influência de dados externos e deve ser única.

As diversas funções que podem ser utilizadas para estabelecer essa transferência entre neurônios, são chamadas de funções de limiar e são abordadas no item a seguir.

2.3.3 FUNÇÕES DE LIMIAR

As funções de limiar, ou funções para transferência de sinais entre neurônios, são aquelas que determinam a intensidade e a forma que os valores são alterados para serem transmitidos entre eles. Destacam-se cinco tipos: a linear, a *step*, a em rampa, a gaussiana e a sigmoid.

- A função linear é uma equação linear da forma $f = a.x + b$, onde a é a inclinação da reta.

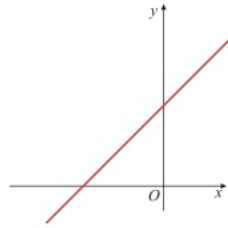


Figura 2.4: Função Linear

- A *step*, é uma função que possui um resultado binário, ou seja, de apenas dois valores possíveis. Assim ela tem a forma $f(x) = p + q.u(x - j)$, onde u é a função de *Heaviside*. Essa função também é conhecida como *hard-limiter*. Uma desvantagem do uso dessa função é que quaisquer ajustes feitos nos pesos que alimentam os neurônios com o uso desta função terão efeitos não tão eficientes e limitados, devido a sua descontinuidade acentuada.

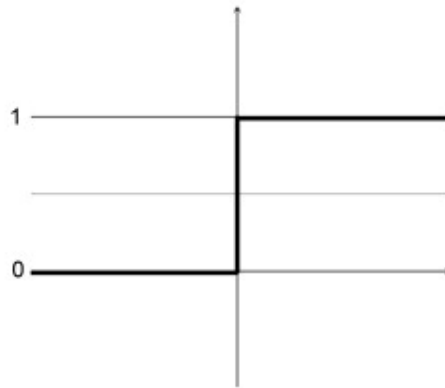


Figura 2.5: Função Step

- A função de limiar em rampa é semelhante à função *step*, mas a transição entre um valor e outro não é instantânea, devendo haver uma fase de transição entre 2 valores,

atenuando o problema citado acima. Sendo $f(x) = \gamma$ para $x > z$, $f(x) = -\gamma$ para $x < z$ e para valores no intervalo $(-z, z)$ $f(x)$ varia entre $-\gamma$ e γ , como mostra a figura 2.6.

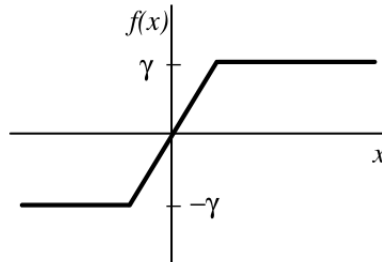


Figura 2.6: Função Rampa

- A gaussiana é uma função que se baseia em um ponto médio e uma variância pré-definida $f(x) = e(x^2/v)$, onde v é a variância. Seu gráfico é apresentado na figura 2.7.

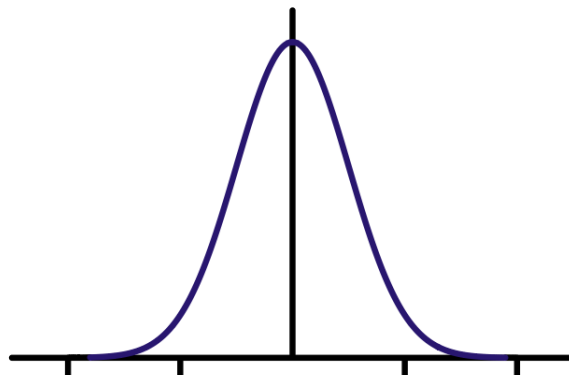


Figura 2.7: Função Gaussiana

- A função *sigmoid* é uma variante da função em rampa, com a vantagem de diminuir mais ainda o problema da função step, deixando-a totalmente contínua. Essas características permitem uma transição gradual e não linear entre dois estados. Este tipo de função é utilizada também em outras áreas de estudo como sociologia e química.

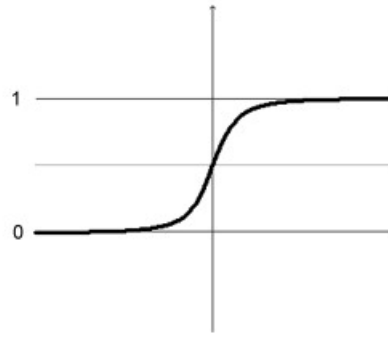


Figura 2.8: Função sigmóide

Neste trabalho será utilizada a função de limiar sigmóide, com uma otimização que é muito utilizada no ramo das redes neurais, que é a mudança dos limites da função para não trabalhar com números negativos. Serão usados os valores da sigmoid variando entre 0 e 1, sendo aceito e não aceito respectivamente. A principal utilidade da sigmóide, no que tange a uma rede neural, é que sua derivada é expressa de uma forma bastante simples: $f'(x) = f(x) * (1 - f(x))$. Isto será útil ao implementar o algoritmo de *backpropagation*, que depende desta derivada. Com o cálculo da mesma rapidamente efetuado, a iteração do algoritmo se torna mais rápida.

2.3.4 PADRÕES

Os padrões são os dados que alimentam a RNA. Podem ser valores numéricos, imagens, caracteres, entre outros, que serão posteriormente transformados em números. A um determinado padrão de entrada corresponde um sinal de saída, sendo que o tipo do padrão de entrada pode ser diferente do padrão de saída. Nestes casos torna-se essencial um pré-processamento das informações, ou seja, a seleção e representação correta das características do padrão a ser aplicado pode afetar a performance da rede. Assim, para um bom desempenho da rede neural, deve-se optar pelo melhor conjunto de características para os padrões.

2.4 CLASSIFICAÇÕES DE REDES NEURAIS

Não há um modo oficial de se classificar uma rede neural, mas existem classificações já consagradas quanto a 3 de suas características: tipo de aprendizado, tipo de entradas e forma de conexão.

Tipo de aprendizado identifica se na rede existe uma definição para o que se espera na saída, assim sendo utilizada para ajustar os pesos das entradas, neste caso a RNA é de aprendizado supervisionado, caso contrário é de auto-aprendizado.

Os tipos de entrada podem ser intervalar, onde as entradas são de forma contínua, ou seja, aceitam qualquer valor como entrada, ou podem ser discretas, como por exemplo uma entrada binária.

Existem três tipos de forma de conexão, que é a forma como os neurônios estão conectados e como é a transmissão de dados. Na alimentação competitiva, os neurônios de uma mesma vizinhança recebem as mesmas entradas e competem entre si para ser a única saída.

Na alimentação à frente, a rede é dividida em camadas onde as saídas de uma servem como entradas para a próxima camada.

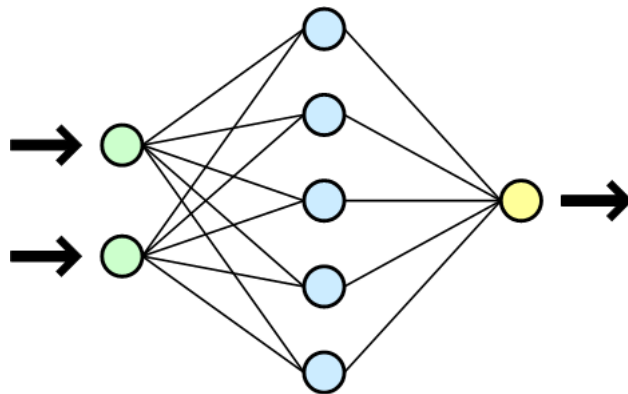


Figura 2.9: Conexão feed-forward

Na retro-alimentação, os sinais de saída são utilizados para alimentar camadas anteriores.

Neste trabalho será utilizado um modelo de de alimentação à frente, com aprendizado supervisionado e entrada intervalar, conhecido como backpropagation. Esse modelo é baseado em outro modelo mais simples chamado adaline, e ambos os modelos estão explicitados nos itens subsequentes.

2.5 ADALINE

Adaline (Adaptative Linear Network), como seu nome diz utiliza um combinador adaptativo linear, onde a saída de cada neurônio é a combinação linear das entradas multiplicadas por um peso, seguindo a fórmula $S(t) = \sum_{i=0}^{n-1} p_i(t)e_i(t)$, onde S é a saída, e_i são os elementos do vetor de entradas e p_i seus respectivos pesos.

A regra de adaptação de pesos utilizada no adaline pode ser o algoritmo perceptron ou o $\alpha - LMS$, a diferença entre eles está na fórmula do peso e do erro, onde o perceptron utiliza as fórmulas de pesos $p(t+1) = p(t) + \alpha(\varepsilon(t)/2)e(t)$ e erro $\varepsilon(t) = d(t)f_{limiar}(S(t))$. Já o $\alpha - LMS$ utiliza pesos como sendo $p(t+1) = p(t) + \alpha(\varepsilon(t)e(t)/(|e(t)|^2))$ e para o cálculo do erro utiliza $\varepsilon(t) = d(t)S(t)$. Onde α é o coeficiente de aprendizado, ε o erro e d é a saída esperada.

A rede adaline tem como vantagem sua simplicidade, pois a entrada não necessita de um pré-processamento complexo e são poucos parâmetros a serem ajustados, mas ela possui a limitação de conseguir apenas reconhecer padrões mais simples, necessitando de adaptações para análises mais complexas.

2.6 BACKPROPAGATION

O *Backpropagation* é um algoritmo de aprendizado utilizado em redes neurais de três ou mais camadas de adalines interconectados, que utilizam como função limiar a sigmóide.

A atualização dos pesos neste modelo, é feita iniciando-se pelo erro da saída da última camada, e a partir daí são calculados os erros das camadas anteriores, dando o nome do algoritmo (retropropagação).

O erro da última saída é dado por $\varepsilon_i(t) = S(t)(1-S(t))(d_i(t)-S(t))$ e com ele são ajustados os pesos da última camada por $p_i(t+1) = p_i(t) + \alpha\varepsilon_i(t)e_n(t)$. O parâmetro α , denominado taxa de aprendizado, tem a seguinte importância: quanto menor, mais lenta e precisa é a convergência do algoritmo. Com isso, os erros das camadas anteriores podem ser calculados

por $\varepsilon_i(t) = e_n(t)(1-e_n(t))\sum_k \varepsilon_k p_{ik}(t)$, onde k são os neurônios conectados à seguir ao nó atual e p_{ik} é o peso associado a esta ligação. E com este erro recalcula-se os pesos das camadas intermediárias $p_i(t+1) = p_i(t) + \alpha \varepsilon_i(t) e_n(t) + \mu(p_i(t) - p_i(t-1))$. O coeficiente μ é denominado momento, e determina a importância relativa da última diferença de peso em relação à atual. A calibração dos parâmetros α e μ é normalmente feita de modo empírico, de acordo com os dados possuídos e a topologia da rede.

A topologia do backpropagation é um adaline expandido com camadas intermediárias como na figura 2.10.

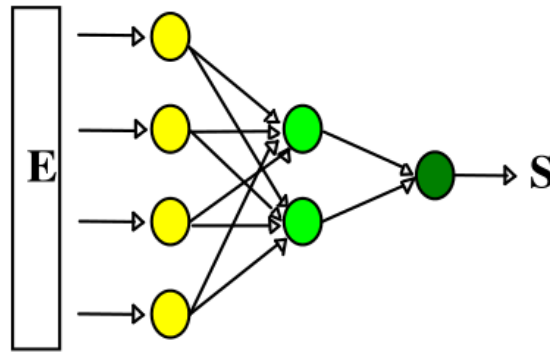


Figura 2.10: Topologia Backpropagation

A vantagem do modelo é que ele abrange uma grande quantidade de padrões que podem ser utilizados, e por este motivo foi escolhido para este trabalho.

3 O PÔQUER

O pôquer é um jogo de cartas jogado por duas ou mais pessoas, no qual os jogadores com as cartas total ou parcialmente escondidas fazem apostas em um pote central e após o fim das apostas, o pote central é atribuído ao jogador que possuir o melhor conjunto de cartas dentre os que permaneceram na jogada, ou ao único que restou caso todos os outros tenham desistido. Pôquer é um banco de ensaio interessante para pesquisas em inteligência artificial, pois é um jogo de informações imperfeitas, onde os múltiplos agentes concorrentes têm de lidar com conhecimento probabilístico, avaliação de risco e a possível fraude, assim como são as decisões feitas no mundo real. A modelagem do oponente é um outro problema difícil em aplicações de tomada de decisão, sendo essencial para alcançar alto desempenho no pôquer.

3.1 BARALHO

O pôquer é jogado com um barulho comum, de 52 cartas, divididas em 4 grupos, chamados naipes. Os naipes são identificados por nomes: ouros , copas, paus e espadas e seus símbolos, representados na figura 3.1.

Figura 3.1: Ouros, copas, paus e espadas

No pôquer os naipes possuem o mesmo valor, ou seja, o naipe da carta nunca é critério de desempate no pôquer. Cada naipe possui 13 cartas em ordem decrescente de valor: o ás (A), o rei (K), a dama(Q), o valete (J) e os números de 10 a 2, como é ilustrado na figura 3.2.



Figura 3.2: Ordem decrescente de valor

3.2 RANKING DE COMBINAÇÕES DE CARTAS

O ranking das combinações de 5 cartas no pôquer é exibido a seguir em ordem crescente de valor:

Carta maior: quando não há nenhuma combinação, para efeito de desempate, ganha o que possuir a carta de maior valor.

Par: Um par de cartas, isto é, duas cartas iguais. Se mais de um jogador tiver um par, então ganha quem tiver o par de cartas de maior valor. Caso ambos tenham o mesmo par, então aquele que possuir a carta de maior valor fora do par é o vencedor. Essa carta de desempate é chamada *kicker* (Figura 3.3).



Figura 3.3: Par de dez

Dois pares: Dois pares de cartas. Se mais de um jogador tiver um dos pares iguais, então ganha quem tiver o segundo par com o valor mais alto. Caso ambos tenham os mesmos dois pares, então aquele que tiver a quinta carta (*kicker*) de maior valor é o vencedor (Figura 3.4).



Figura 3.4: Dois pares de sete com quatro

Trinca: Três cartas iguais. Se mais de um jogador tiver esse jogo, então ganha quem tiver a trinca com uma carta de maior valor (Figura 3.5).



Figura 3.5: Trinca de cinco

Sequência (*Straight*): Cinco cartas em sequência. O ás pode ser a maior carta ao lado do rei ou a menor carta ao lado do 2. Se mais de um jogador tiver uma sequência, então ganha aquele que tiver a sequência com cartas de maior valor. Caso ambos tenham a mesma sequência, então é considerado empate (Figura 3.6).



Figura 3.6: Sequência de dez a ás

Flush: Cinco cartas do mesmo naipe, mas não em sequência. Se mais de um jogador tiver um *flush*, então ganha aquele que tiver a carta de maior valor (Figura 3.7).



Figura 3.7: *Flush* de espadas

Full House: Um par junto com uma trinca de cartas. Se mais de um jogador tiver um *full house*, então ganha aquele que tiver a trinca com cartas de maior valor (Figura 3.8).



Figura 3.8: *Full House* de dama com dois

Quadra (*Four*): Quatro cartas iguais. Se mais de um jogador tiver uma quadra, então ganha aquele que tiver a quadra com uma carta de maior valor (Figura 3.9).



Figura 3.9: Quadra de rei

Straight Flush: Sequência de cinco cartas do mesmo naipe. Se mais de um jogador tiver um *straight flush*, então ganha aquele que tiver a sequência com cartas de maior valor (Figura 3.10).



Figura 3.10: *Straight Flush*

Royal Straight Flush: As cinco cartas de maior valor do mesmo naipe. Um Royal Straight Flush é simplesmente um *straight flush* com um Ás como carta mais alta. É a combinação mais forte do pôquer (Figura 3.11).



Figura 3.11: *Royal Straight Flush*

3.3 AÇÕES

No pôquer, os jogadores possuem disponíveis as seguintes ações: passar (*check*), desistir (*fold*), apostar (*bet*), pagar (*call*) e aumentar (*raise*).

Um jogador ao passar está apenas passando a vez a outro jogador, e não renunciando ao pote central. Se ainda não houver apostas na rodada atual, o jogador pode passar a vez, e se todos os jogadores passarem na rodada, então a rodada é considerada completa.

Um jogador pode desistir da jogada, renunciando o interesse ao pote central.

Caso nenhuma aposta tenha sido feita ainda na rodada atual, um jogador pode apostar. Se um jogador apostar, o jogador adjacente (no sentido horário) tem três opções: desistir, pagar ou aumentar. A rodada só termina depois que todos os jogadores tenham igualado suas apostas, ou caso nenhum jogador pague a aposta feita por outro jogador, que se torna o vencedor da rodada.

Um jogador paga quando iguala a aposta feita por outro jogador na rodada.

Quando um jogador não só iguala a aposta de outro jogador, mas também acrescenta, ele está aumentando a aposta. Todos os jogadores subsequentes deverão pagar o maior valor apostado, ou aumentar ainda mais, se quiserem continuar na jogada.

Quando a última aposta ou aumento na rodada final for igualada, ocorre a abertura de cartas. É nesse momento que se determina o vencedor do pote, uma vez que as combinações de cartas de cada jogador são mostradas uma a uma.

Existem diversas variantes do pôquer, dentre as quais pode-se categorizar livremente em: pôquer aberto, pôquer fechado e *community card poker*, dentre outras. O estilo de pôquer conhecido como Texas-Hold'em é um tipo de *community card poker*, onde existem

cartas abertas para uso de todos os jogadores. Ele é atualmente o pôquer mais conhecido e jogado nos cassinos e torneios, e conseqüentemente com maior quantidade de estudo acerca do mesmo, fator pelo qual será utilizado como base no presente trabalho.

No *Texas Hold'em* os jogadores recebem duas cartas fechadas e são dispostas cinco cartas fechadas sobre a mesa. Para vencer, o jogador deve combinar suas duas cartas fechadas com três das cinco abertas ou a melhor combinação, podendo usar as cinco cartas comunitárias. Ao contrário do pôquer aberto tradicional, as cartas abertas são as mesmas para todos os participantes. Existem um total de quatro rodadas de apostas. Antes do início das rodadas de apostas, os dois jogadores à esquerda do crupiê (*dealer*) devem fazer apostas obrigatórias, conhecidas como *Big Blind* e *Small Blind*. O *Small Blind* paga metade do *Big Blind*.

Na primeira rodada cada jogador recebe duas cartas fechadas e é feita a primeira rodada de apostas. O jogador à esquerda do *Big Blind* poderá desistir, pagar a aposta do *Big Blind* para entrar na jogada, ou aumentar a aposta. Os outros jogadores também optam por umas destas três ações.

Ao final da primeira rodada de apostas, o crupiê apresenta o *Flop*, isto é, abre as três primeiras cartas na mesa e inicia-se a segunda rodada de apostas pelo jogador à esquerda do crupiê.

Em seguida o crupiê apresenta o *Turn*, isto é, abre a quarta carta na mesa, e inicia-se a terceira rodada de apostas.

Na última rodada, o crupiê apresenta o *River*, isto é, abre a quinta e última carta fechada na mesa, e inicia-se a quarta e última rodada de apostas. Após o término da última rodada de apostas, os jogadores mostram seus jogos (*showdown*) a começar pelo último que apostou ou que aumentou alguma aposta. Somente este jogador tem a obrigação de mostrar seu jogo, os demais podem deixar suas cartas escondidas se perceberem que não vencerão.

Existem algumas variações no *Texas Hold'em* quanto ao limite máximo no valor de uma aposta. O *No Limit Hold'em* permite o chamado *all-in*, ou seja, apostar tudo que se tem. Entretanto, nenhuma aposta pode ser inferior ao valor apostado para entrar na partida.

4 CONSTRUÇÃO E TESTES DA API BASE DE REDE NEURAL

O primeiro passo tomado na construção do projeto foi o desenvolvimento de uma interface genérica básica de rede neural *feed-forward* para ser utilizada nas topologias que serão definidas nas seções subsequentes, seguindo o modelo de algoritmo delineado na seção 2. A base utilizada para todas elas é a de uma rede em três camadas (input, camada interna e output) com ativação sigmóide.

Para realizar o teste desta API (garantindo seu funcionamento básico de acordo com princípios de TDD) e de modo a obter considerações concretas no que concerne a como transformar o modelo matemático em dados que possam ser utilizados diretamente na rede com convergência adequada, foi utilizado um problema de classificação trivial, muito mais simples do que o tratado no resto deste trabalho. Os problemas utilizados foram "dado um caracter, decidir se o mesmo representa uma vogal ou não", e "dada uma reta e um dos semiplanos determinados pela mesma, determinar se o semiplano contem um ponto". Com os testes de convergência definidos a seguir para duas topologias distintas (e outros testes não mostrados aqui), chegou-se às seguintes conclusões gerais:

- O tratamento de informação discreta como sendo continuamente distribuída no intervalo $[0,1]$ tornou a convergência muito mais lenta no primeiro caso;
- A magnitude do fator de aprendizado α não influenciou no fato de a rede convergir para um valor correto em nenhum dos casos, mas acelerou esta convergência.

Estas orientações gerais foram utilizadas para a modelagem topológica das redes mostradas na próxima seção.

5 ESTRUTURA DA SOLUÇÃO

5.1 ESTRUTURA GERAL

A solução proposta para a criação de um agente *stand-alone* capaz de competir é composta de três componentes principais, que serão expostos nas seções subsequentes. São eles:

- Uma serie de redes neurais capaz de cada modelar um oponente (ou uma série de oponentes), que formam os *perfis de jogador*. Os dados que compõem cada perfil são selecionados heurísticamente;
- Uma rede neural capaz de classificar as últimas jogadas de um oponente em algum dos perfis do item acima tendo em mãos a ação simulada do perfil - esta será denominada *classificador*;
- Uma rede neural que modela um jogador que é bem-sucedido (em termos gerais) contra jogadores com os variados perfis. Esta será chamada *rede principal*.

O funcionamento básico do sistema dar-se-á da seguinte forma: ao se jogar contra um oponente, primeiro o classificador identifica qual é o perfil de jogador mais adequado ao oponente e produz uma indicação de qual perfil é mais adequado. Tendo esta informação, executa-se o perfil de jogador escolhido e são geradas as probabilidades de *fold*, *call* e *raise* dada a situação atual do jogo. Estas probabilidades são então introduzidas como parâmetros de entrada da rede neural principal de modo que a rede principal possa utilizar estas probabilidades para agir de maneira diferente, a depender da tendência do jogador e do peso atribuído à informação de perfil pela rede principal. Por fim, as informações do estado atual e do perfil modelado para os oponentes são usados como entrada para a rede principal, que determina uma ação a ser tomada (*fold*, *call* ou *raise*, com a porcentagem de *raise* no caso apropriado). Esta ação pode então ser executada no jogo.

Esta abordagem é, em termos de expressividade matemática, equivalente a uma rede neural que contivesse uma chave única representando o jogador em uma estrutura de três camadas[10]. Por meio da modularização e da classificação em perfis, tenciona-se diminuir o volume de dados necessário para se realizar o treinamento da rede até que ela possa jogar de maneira satisfatória.

5.2 MODELAGEM DO Oponente

Em primeiro lugar, será apresentada a topologia básica da rede neural utilizada na modelagem do oponente (assumindo uma mesa com tamanho máximo de 8 jogadores) nas tabelas 5.1 e 5.2. Para simplificar a modelagem matemática, foram ignorados alguns fatores que são de potencial influência sobre a ação do oponente, tal como a posição do mesmo na mesa, uma vez que a utilização da mesma implicaria em um encadeamento de previsões para prever as ações dos últimos jogadores na fila.

Outra importante simplificação a ser realizada na classificação no decorrer de um jogo é assumir que as probabilidades de melhor mão estão identicamente distribuídas dentre os jogadores cujas mãos são desconhecidas e que estão remanescentes no jogo - deste modo, mesmo quando não se sabem as cartas do agente, pode-se utilizar alguma métrica de modo a estimar as ações do oponente. Isto é especialmente relevante quando se leva em conta que a maioria dos bancos de dados de mãos não contém as informações referentes às mãos dos jogadores que realizam *fold*, levando a uma informação incompleta na construção da rede. A não-utilização desta simplificação tornaria o treinamento da rede inviável.

As saídas desta rede neural de são representativas do perfil do jogador exposto à situação atual de jogo.

5.2.1 ESTIMAÇÃO DAS PROBABILIDADES

Um dos problemas de maior complexidade encontrados ao se realizar a transformação dos dados puros para a parametrização requerida pela rede foi, dada a situação da mesa, encontrar de fato a probabilidade de a mão ser a melhor da mesa. Existem cálculos que computam a probabilidade de se atingir um determinado jogo (os chamados cálculos de outs), contudo, estas probabilidades não representam a chance de se vencer a mão de fato. Por outro lado, o número de mãos necessário para se determinar a probabilidade exata da melhor mão é intratável sem a utilização de enormes tabelas de consulta que não caberiam na memória física disponível. Decidimos, então, utilizar o método de Monte Carlo (amostragem aleatória) para determinar uma probabilidade aproximada, contando quantas vitórias ocorreram na amostragem. Tal método é tanto mais preciso do que a estimação via Outs (que não leva

em conta de forma alguma o jogo do oponente), quanto é quase tão rápido quanto o uso de tabelas de consulta sem o consequente uso excessivo de memória.

Parâmetros de entrada	Domínio
Probabilidade de a mão ser a melhor da mesa (π)	$0 \leq \pi \leq 1$
Inverso da quantidade de fichas possuída pelo jogador ($1/s$)	$0 \leq 1/s \leq 1$
Quantidade de jogadores ($n/8$)	$0 < n/8 \leq 1$
Inverso da quantidade de fichas possuída pelo i-ésimo oponente ($1/s_i$)	$0 \leq 1/s_i \leq 1$
Inverso da quantidade de fichas no pote ($1/p$)	$0 < 1/p < 1$
Jogo na fase de pré-flop? p	$p \in \{0, 1\}$
Jogo na fase de flop? f	$f \in \{0, 1\}$
Jogo na fase de turn? t	$t \in \{0, 1\}$
Jogo na fase de river? r	$r \in \{0, 1\}$

Tabela 5.1: Parâmetros de entrada da RN utilizada para modelar o oponente

Parâmetros de saída	Domínio
Fold? f_r	$f_r \in \{0, 1\}$
Call? k_r	$k_r \in \{0, 1\}$
Raise? c_r	$c_r \in \{0, 1\}$
Fração do pote a ser apostada em raise	$0 \leq r/p \leq 1$

Tabela 5.2: Parâmetros de saída da RN utilizada para modelar o oponente

5.3 REDE PRINCIPAL E CLASSIFICAÇÃO EM PERFIS

Para cada potencial oponente (até um máximo de 7) serão adicionadas as informações de perfil adquiridas nos perfis que melhor se encaixarem às informações da tabela 4.1. As jogadas utilizadas para classificar o perfil seguirão um esquema de janela deslizante, de modo a evitar que um oponente que troque de estratégia com pouca frequência seja classificado de forma incorreta devido ao peso de suas ações iniciais, que possivelmente não são mais relevantes ao estado atual do jogo. Contudo, um oponente que troque de estratégia constantemente continuaria podendo evitar uma classificação precisa. A topologia da rede neural do classificador está descrita nas tabelas 5.3 e 5.4.

A topologia da rede principal é similar à utilizada para realizar a modelagem do perfil do oponente, com duas diferenças principais: primeiro, não há necessidade de se estimar a probabilidade de haver a melhor mão, uma vez que presumimos que se sabem as cartas na mão do próprio jogador. Além disso, a entrada da rede principal é acrescida da saída da rede de perfil, da seguinte forma: a informação saída da rede que modela o perfil é acrescida à entrada da rede principal para cada jogador, conforme figura.

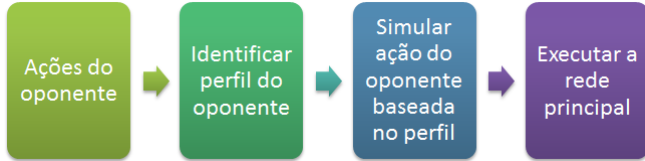


Figura 5.1: Fluxo de dados da rede neural principal

Parâmetros de entrada	Domínio
Probabilidade de a mão ser a melhor da mesa (π)	$0 \leq \pi \leq 1$
Inverso da quantidade de fichas possuída pelo jogador ($1/s$)	$0 \leq 1/s \leq 1$
Quantidade de jogadores ($n/8$)	$0 < n/8 \leq 1$
Inverso da quantidade de fichas possuída pelo i-ésimo oponente ($1/s_i$)	$0 \leq 1/s_i \leq 1$
Inverso da quantidade de fichas no pote ($1/p$)	$0 < 1/p < 1$
Jogo na fase de pré-flop? p	$p \in \{0, 1\}$
Jogo na fase de flop? f	$f \in \{0, 1\}$
Jogo na fase de turn? t	$t \in \{0, 1\}$
Jogo na fase de river? r	$r \in \{0, 1\}$
Oponente deu fold?	$f_r \in \{0, 1\}$
Oponente deu call?	$k_r \in \{0, 1\}$
Oponente deu raise?	$c_r \in \{0, 1\}$
Fração do pote que foi apostada em raise	$0 \leq r/p \leq 1$

Tabela 5.3: Parâmetros de entrada da RN utilizada para classificar o oponente para cada uma das jogadas em uma janela deslizante.

Parâmetros de saída	Domínio
Melhor perfil (i)	$0 < i/n \leq 1$

Tabela 5.4: Parâmetros de saída da RN utilizada para classificar o oponente

Para representar a estrutura da solução, utilizaram-se os seguintes diagramas de classe apresentados nas figuras 5.2, 6.2 e 6.3, descrevendo as classes utilizadas na API de aprendizado e na interface com o banco de dados de mãos. O programa final será composto do classificador e da rede principal. Os perfis são utilizados apenas para treinar estas duas redes.

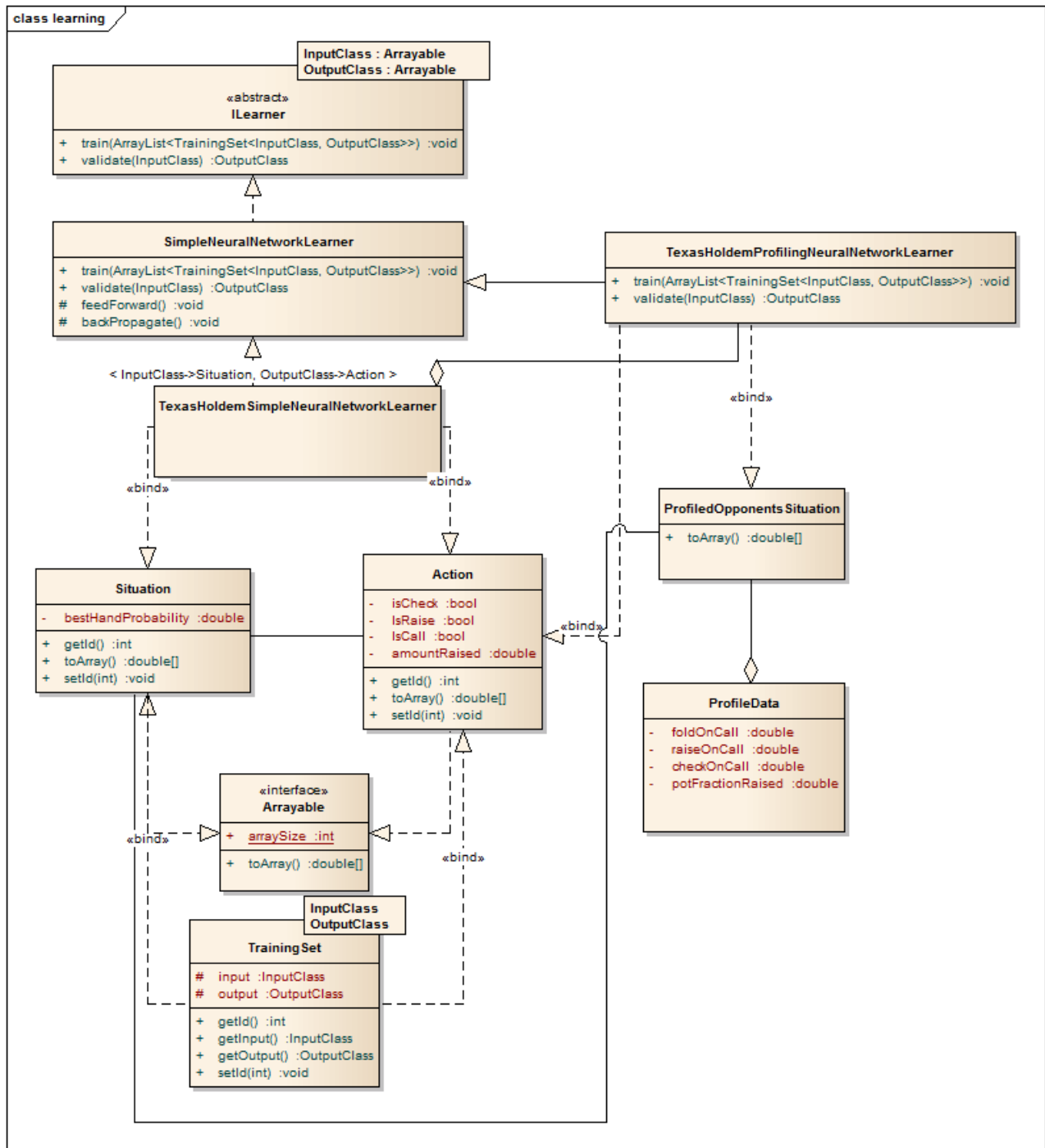


Figura 5.2: Diagrama das classes utilizadas no aprendizado

6 SELEÇÃO DE JOGADORES PARA TREINAMENTO DA REDE

Para realizar o treinamento da rede, foram coletados dados de um extenso banco chamado IRC Poker Database, que foi adaptado na estrutura de dados apresentada na figura 5.2 e inserido em um banco de dados MySQL, cujo esquema está descrito na figura

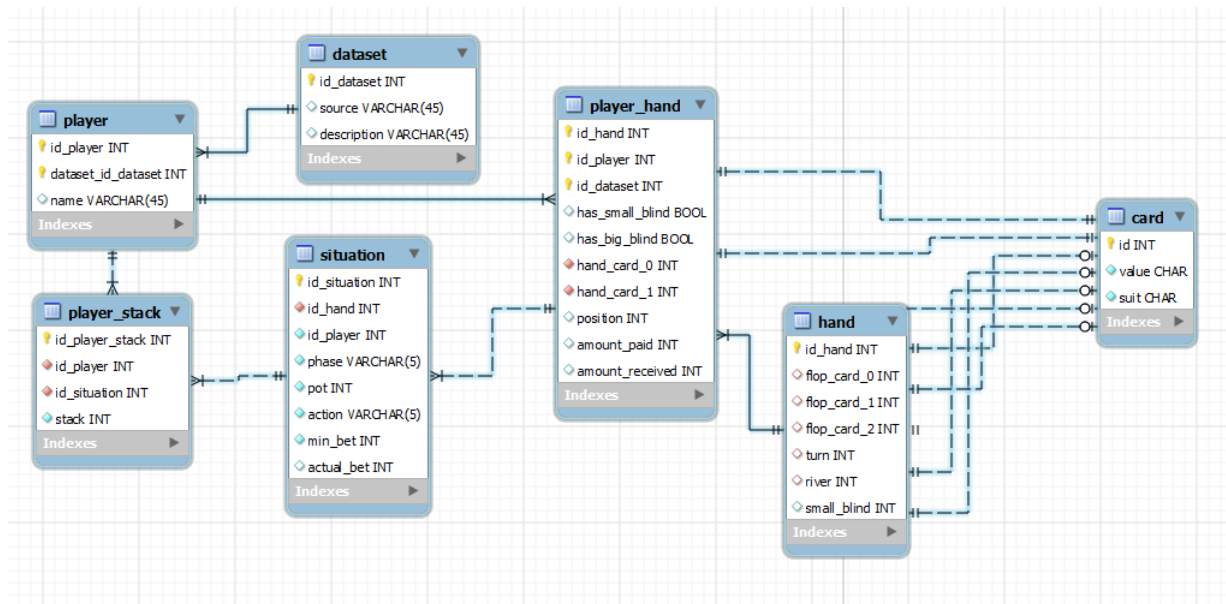


Figura 6.1: Diagrama relacional do banco de dados utilizado.

6.1 IRC POKER DATABASE

Antes da existência de serviços de poker online com dinheiro real, os entusiastas do jogo podiam treinar e testar suas táticas em servidores do IRC preparados para o jogo. Essa estrutura de salas de jogos se desenvolveu a ponto de serem criados canais para diferentes modalidades e experiência no jogo.

Em 1995, Michael Maurer, criou um programa chamado "Observer", que logava nas salas de jogos, e registrava todas as mãos jogadas (exceto as mãos cobertas), resultando em um banco de dados de mais de 10 milhões de jogadas, dos mais diversos níveis de jogadores, desde principiantes até profissionais, como por exemplo Chris Ferguson, que ganhou 5 vezes o World Series of Poker, inclusive sendo campeão mundial do evento principal em 2000.

Neste trabalho serão utilizados os dados de 4 canais deste banco, consistindo em mais de

3 GB de dados, e para fazer a adaptação à estrutura citada acima, foram utilizadas rotinas em Java, Shell Script e SQL.

Para realizar a interface com o banco de dados, utilizar-se-á a estrutura de classes descritas nas figuras 6.2 e 6.3. Utilizaremos os padrões de DAO (*data access object*) e fachada para realizar este interfaceamento.

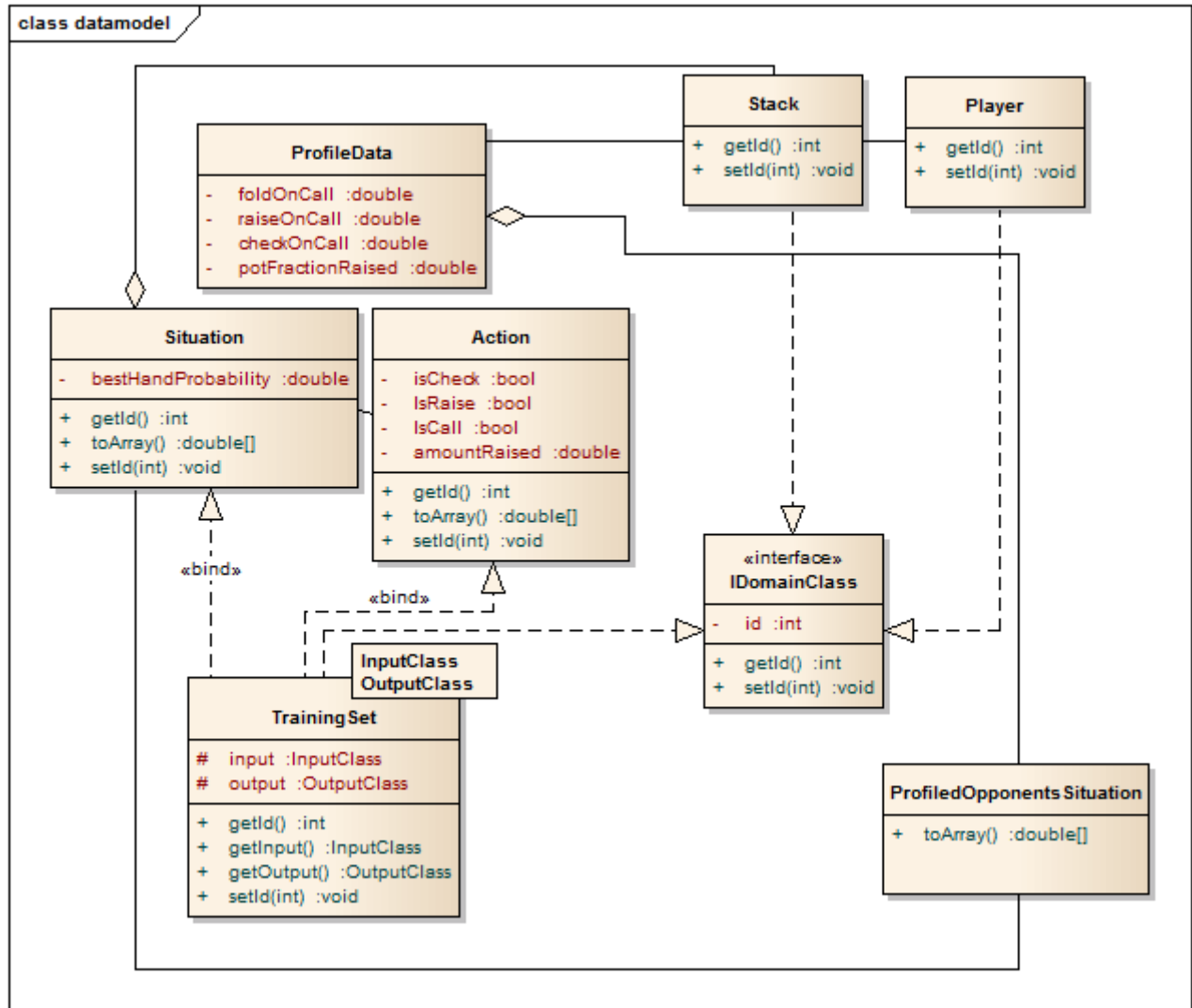


Figura 6.2: Diagrama de classe para as classes básicas

jogadores dentre *loose* (que tende a entrar em mãos mesmo quando a probabilidade inicial de possuir a melhor mão é baixa) e *tight* (que tende a não entrar em mãos a não ser que tenha uma alta probabilidade de ter a melhor mão). Uma métrica para classificar jogadores entre estes dois extremos é a quantidade de *folds* em relação ao número total de mãos. Jogadores *loose* raramente realizam fold, ao contrário de jogadores *tight*.

As faixas de corte utilizadas serão determinadas da seguinte maneira: assumindo uma distribuição normal da taxa de *fold* (mãos nas quais o jogador deu fold / total de mãos jogadas) e uma população com desvio padrão σ e média μ , define-se o jogador com média abaixo de $\mu - \sigma/2$ como *tight*, o jogador acima de $\mu + \sigma/2$ como *loose* e o jogador entre estas duas faixas como moderado.

7 CONCLUSÃO

Da pesquisa bibliográfica realizada, concluímos que é factível a utilização de redes neurais para a simulação e análise de oponentes no jogo de pôquer, dando a entender que é promissora a possibilidade de utilização de uma rede neural como agente *stand-alone* do jogo.

8 CRONOGRAMA DE ATIVIDADES

Atividades	7/13	8/13	9/13	10/13	11/13	12/13	1/14	2/14	3/14	4/14	5/14	6/14
Revisão da Literatura												
Pesquisa de métodos de aprendizado de máquina												
Pesquisa sobre Redes Neurais												
Coleta de dados para o aprendizado												
Definição e pesquisa sobre perfis de jogadores												
Implementação da RNA												
Implementação do reconhecedor de perfil e do jogador												
Treinamento, ajustes e testes												
Redação do relatório Final												

9 REFERÊNCIAS

- [1] Salim, Rohwer. **Poker Opponent Modeling** - Indiana University, 2006.
- [2] Billings, Papp et al. **Opponent Modeling in Poker** - AAAI, University of Alberta, Edmonton, Canada, 1998.
- [3] Davidson, Billings, Schaeffer, Szafron. **Improved Opponent Modeling in Poker** - University of Alberta, Edmonton, Canada, 2000.
- [4] Johnson. **Robust Strategies and Counter-Strategies: Building a Champion Level Computer Poker Player** - University of Alberta, 2007.
- [5] Nicolai, Hilderman. **No-Limit Texas Holdem Poker Agents Created with Evolutionary Neural Networks** - IEEE Symposium on Computational Intelligence and Games, 2009.
- [6] B. Beattie, G. Nicolai, D. Gerhard, and R. J. Hilderman. **Pattern classification in no-limit poker: A head-start evolutionary approach** - Canadian Conference on AI, 204215, 2007.
- [7] Russell S., Norvig P. **Artificial Intelligence: A Modern Approach (3rd Edition)**, 2010.
- [8] desJardins - **Machine Learning for Military Operations**, Rome Laboratory, Air Force Materiel Command, 1996;
- [9] Spaanenburg, ten Berg. **Modular and Hierarchical Specialization for Neural Networks** - Lund University, Department of Information Technology, 2003;
- [10] Hornik. **Approximation capabilities of multilayer feedforward networks**, Neural Networks, Volume 4, Issue 2, pg 251-257, 1991;