

Marco Aurélio Botelho da Silva

**Modelos Neuro-Evolucionários de Redes Neurais Spiking
Aplicados ao Pré-Diagnóstico de Envelhecimento Vocal**

Tese de Doutorado

Tese apresentada como requisito parcial para
obtenção do título de Doutor pelo Programa de
Pós-Graduação em Engenharia Elétrica da PUC-
Rio.

Orientadores: Prof. Marley Maria Bernardes Rebuzzi Vellasco
Co-orientador: Prof. Edson Luiz Cataldo Ferreira

Rio de Janeiro
Setembro de 2014

Marco Aurélio Botelho da Silva

**Modelos Neuro-Evolucionários de Redes Neurais Spiking
Aplicados ao Pré-Diagnóstico de Envelhecimento Vocal**

Tese apresentada como requisito parcial para obtenção do grau de Doutor pelo Programa de Pós-Graduação em Engenharia Elétrica do Departamento de Engenharia Elétrica do Centro Técnico Científico da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Profa. Marley Maria Bernardes Rebuzzi Vellasco

Orientador

Departamento de Engenharia Elétrica - PUC-Rio

Prof. Edson Luiz Cataldo Ferreira

Co-Orientador

Universidade Federal Fluminense - UFF-Rio

Prof. André Vargas Abs da Cruz

Universidade Estadual da Zona Oeste - UEZO

Prof. Jorge Carlos Lucero

Universidade Nacional de Brasília - UNB

Prof. Jorge Luís Machado do Amaral

Universidade do Estado do Rio de Janeiro - UERJ

Dr. Leonardo Alfredo Forero Mendoza

Departamento de Engenharia Elétrica - PUC-Rio

Profa. Karla Tereza Figueiredo Leite

Departamento de Engenharia Elétrica - PUC-Rio

José Eugênio Leal

Coordenador Setorial do Centro

Técnico Científico - PUC-Rio

Rio de Janeiro, 11 de setembro de 2014

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Marco Aurélio Botelho da Silva

Graduou-se em Engenharia Eletrônica na Universidade do Estado do Rio de Janeiro. Mestrado na área de Sistemas de Telecomunicações Programa de Mestrado em Engenharia de Telecomunicações na Universidade Federal Fluminense.

Ficha Catalográfica

Silva, Marco Aurélio Botelho

Modelos Neuro-Evolucionários de Redes Neurais Spiking Aplicados ao Pré-Diagnóstico de Envelhecimento Vocal / Marco Aurélio Botelho da Silva; Orientadores: Marley Maria Bernardes Rebuzzi Vellasco; Edson Luiz Cataldo Ferreira. Rio de Janeiro: PUC-Rio, Departamento de Engenharia Elétrica, 2014.

111 f. : il. (color.) ; 30 cm

Tese (doutorado) - Pontífica Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica, 2014

Incluí referências bibliográficas.

1. Engenharia Elétrica - Tese. 2. Rede Neural Spiking. 3. Algoritmos Neuro-Evolutivos. 4. Envelhecimento Vocal. I. Vellasco, Marley M. B. R.. II. Cataldo, Edson. III. Pontífica Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. IV. Título.

CDD: 621.3

Dedico este trabalho a: Beralda (esposa), Guilherme (filho), Kako (in memoriam), Kika (in memoriam) e Sunny, Mãe (Isabel), Pai (Amadeu - in memoriam), Goreti, Rafael, Tio Zé, Vovô Mário (in memoriam) e Vovó Zica.

Agradecimentos

Ao CNPq e à PUC-Rio pelos auxílios durante o curso.

Aos meus orientadores Dra. Marley M. B. R. Vellasco e Dr. Edson Cataldo pelo estímulo e parceria na realização deste trabalho.

Aos meus familiares pelo estímulo.

À minha esposa Beralda e meu filho Guilherme pelo amor incondicional, pelo apoio, por aceitar minha ausência e por não me deixar esmorecer.

A meu pai (in memoriam).

Aos meus amigos Adriano Koshiyama, Harold Dias, Edwin Maldonado, Iury Steiner e Leonardo Forero pela força ao longo do curso.

Aos meus amigos dos laboratórios ICA e LIRA, e a equipe da secretaria do DEE.

Aos meus amigos de longa data: Landi Hudson, Beto, André, Geraldo, Jairo, Daniel, Nelci (Pipia), Tomaz.

Resumo

Silva, Marco Aurélio Botelho; Vellasco, Marley Maria Bernardes Rebuzzi; Ferreira, Edson Luiz Cataldo. **Modelos Neuro-Evolucionários de Redes Neurais Spiking Aplicados ao Pré-Diagnóstico de Envelhecimento Vocal.** Rio de Janeiro, 2014. 111p. Tese de Doutorado - Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

O envelhecimento da voz, conhecido como presbifonia, é um processo natural que pode causar grande modificação na qualidade vocal do indivíduo. A sua identificação precoce pode trazer benefícios, buscando tratamentos que possam prevenir o seu avanço. Esse trabalho tem como motivação a identificação de vozes com sinais de envelhecimento através de redes neurais do tipo *Spiking* (SNN). O objetivo principal é o de construir dois novos modelos, denominados híbridos, utilizando SNN para problemas de agrupamento, onde os atributos de entrada e os parâmetros que configuram a SNN são otimizados por algoritmos evolutivos. Mais especificamente, os modelos neuro-evolucionários propostos são utilizados com o propósito de configurar corretamente a SNN, e selecionar os atributos mais relevantes para a formação dos grupos. Os algoritmos evolutivos utilizados foram o Algoritmo Evolutivo com Inspiração Quântica com representação Binário-Real (AEIQ-BR) e o *Optimization by Genetic Programming* (OGP). Os modelos resultantes foram nomeados *Quantum-Inspired Evolution of Spiking Neural Networks with Binary-Real* (QbrSNN) e *Spiking Neural Network Optimization by Genetic Programming* (SNN-OGP). Foram utilizadas oito bases benchmark e duas bases de voz, masculinas e femininas, a fim de caracterizar o envelhecimento. Para uma análise funcional da SNN, as bases benchmark foram testadas com uma abordagem clássica de agrupamento (*kmeans*) e com uma SNN sem evolução. Os modelos propostos foram comparados com uma abordagem clássica de Algoritmo Genético (AG). Os resultados mostraram a viabilidade do uso das SNNs para agrupamento de vozes envelhecidas.

Palavras-chave

Redes Neurais *Spiking*; Agrupamento; Algoritmos Neuro-Evolutivos; Envelhecimento Vocal.

Abstract

Silva, Marco Aurélio Botelho; Vellasco, Marley Maria Bernardes Rebuzzi; Ferreira, Edson Luiz Cataldo. **Neuro-Evolutionary of Spiking Neural Networks Applied to Pre-Diagnosis of Vocal Aging**. Rio de Janeiro, 2014. 111p. PhD Thesis - Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

The aging of the voice, known as presbyphonia, is a natural process that can cause great change in vocal quality of the individual. Its early identification can benefit, seeking treatments that could prevent their advance. This work is motivated by the identification of voices with signs of aging through neural networks of spiking type (SNN). The main objective is to build two new models, called hybrids, using SNN for clustering problems where the input attributes and parameters that configure the SNN are optimized by evolutionary algorithms. More specifically, the proposed neuro-evolutionary models are used in order to properly configure the SNN, and select the most relevant attributes for the formation of groups. Evolutionary algorithms used were the Evolutionary Algorithm with Quantum Inspiration with representation Binary-Real (AEIQ-BR) and the Optimization by Genetic Programming (OGP). The resulting models were named Quantum-Inspired Spiking Neural Evolution of Networks with Binary-Real (QbrSNN) and Spiking Neural Network Optimization by Genetic Programming (SNN-OGP). Eight bases were used, and two voice benchmark bases, male and female, in order to characterize aging. NNS for functional analysis, the tested benchmark base form with a classical clustering approach (kmeans) and a SNN without change. The proposed models were compared with a classical approach of Genetic Algorithm (GA). The results showed the feasibility of using the SNN to agrupamentode aged voices.

Keywords

Spiking Neural Networks; Clustering; Neuro-Evolutionary Algorithms; Voice Aging.

Sumário

1 Introdução	16
1.1 Motivação	16
1.2 Objetivos	19
1.3 Descrição do trabalho	19
1.4 Organização do trabalho	21
2 Conceitos Fundamentais	22
2.1 Redes Neurais <i>Spiking</i>	22
2.1.1 Introdução	22
2.1.2 Neurônios biológicos	23
2.1.3 Modelos de neurônios <i>spiking</i>	25
2.1.3.1 Modelos baseados em condutância	26
2.1.3.2 Modelos baseados em limiar de disparo	26
2.1.3.3 Comparativos com modelos tradicionais	29
2.1.4 Arquitetura de uma SNN	30
2.1.4.1 Arquiteturas com desdobramento de sinapses	31
2.1.5 Codificação em pulsos dos dados de entrada	32
2.1.6 Aprendizado em SNN	33
2.1.6.1 Aprendizado não supervisionado	34
2.2 Algoritmos Neuro-Evolucionários	35
2.2.1 Algoritmos Evolutivos com Inspiração Quântica com representação Binária-Real - AEIQ-BR.	37
2.2.2 Otimização por Programação Genética – OGP	47
3 Modelos Neuro-Evolucionários para <i>Spiking Neural Networks</i> não supervisionadas	55
3.1 Modelagem da SNN	56
3.2 Primeiro Modelo: <i>Quantum-Inspired Evolution of Spiking Neural Networks with Binary-Real representation (QbrSNN)</i>	59
3.4 Segundo Modelo: <i>Spiking Neural Network Optimization by Genetic</i>	

<i>Programming (SNN-OGP).</i>	66
4 Estudo de Casos <i>Benchmark</i>	70
5 Bases com sinais de voz	77
5.1 Parâmetros extraídos dos sinais de voz	79
5.2 Parâmetros extraídos do sinal glotal	80
5.3 SNN evoluída aplicada nas bases de vozes para identificação de envelhecimento	83
6 Conclusões e Trabalhos Futuros	93
6.1 Trabalhos Futuros	95
7 Referências Bibliográficas	97
8 Apêndice A	105

Lista de figuras

Figura 2.1: Potencial da membrana de um neurônio onde ϑ_j representa o limiar de ativação e μ_j o potencial da membrana: (a) sinais pré-sinápticos excitatórios chegando com tempos espaçados, não permitindo o disparo; (b) sinais recebidos em intervalos pequenos se sobrepõem, chegando ao valor limite e permitindo o disparo do neurônio. Fonte (Alexandre06).	24
Figura 2.2: Potencial da membrana. Fonte (Maass99).	24
Figura 2.3: Classificação dos modelos de neurônios. Fonte (Alexandre06).	25
Figura 2.4: Arquitetura da rede <i>spiking</i> . Fonte (Alexandre06).	30
Figura 2.5: Exemplo de arquitetura de rede com múltiplas sinapses. Fonte (Bohte03).	31
Figura 2.6: Codificação utilizando campos receptivos gaussianos.	32
Figura 2.7: Gráfico da função de aprendizado. Fonte (Bohte03).	35
Figura 2.8: Representação gráfica de um círculo unitário com amplitude de probabilidade de se observar os valores 0 e 1 para um q-bit qualquer. Fonte (Pinho09).	39
Figura 2.9: Pseudo-código do AEIQ-B. Fonte (Pinho09).	41
Figura 2.10: Algoritmo com inspiração quântica com representação real – AEIQ-R. Fonte (Cruz07).	42
Figura 2.11: Exemplo de gene quântico do AEIQ-R. Fonte (Cruz07).	43
Figura 2.12: Diagrama do sistema evolutivo com Inspiração Quântica. Fonte (Cruz07).	46
Figura 2.13: Pseudocódigo do algoritmo AEIQ-BR. Fonte (Pinho09).	47
Figura 2.14: Pseudocódigo do algoritmo OGP. Fonte (Koshiyama13).	49
Figura 2.15: Representação de: $x2 + y$ (a) Representação sob a forma de árvore; (b) forma pré-fixada. Fonte (Koshiyama13).	49
Figura 2.16: Operador de Reprodução Direta. Fonte (Koshiyama13).	50
Figura 2.17: Operador de Cruzamento. Fonte (Koshiyama13).	51
Figura 2.18: Operador de Mutação. Fonte (Koshiyama13).	51
Figura 2.19: Exemplo de um indivíduo multigênico. Fonte (Koshiyama13).	52
Figura 2.20: Operação de Cruzamento de baixo nível. Fonte (Koshiyama13).	51

Figura 2.21: Operação de Mutação. Fonte (Koshiyama13).	53
Figura 2.22: Operação de cruzamento de alto nível. Fonte (Koshiyama13).	53
Figura 3.1: Estrutura da SNN utilizada nesta tese SNN.	57
Figura 3.2: Adaptação dos pesos sinápticos através do algoritmo de aprendizado. (a) Inicialização dos pesos com o mesmo valor. (b) Pesos foram atualizados e o próximo pulso é selecionado Fonte (Alexandre06).	58
Figura 3.3: Cromossomo usado para otimizar, simultaneamente, os parâmetros e os atributos de entrada de uma SNN.	60
Figura 3.4: Representação gráfica dos indivíduos quânticos.	61
Figura 3.5: Exemplo da Silhueta (Si) para a base Iris .	64
Figura 3.6: Pseudocódigo do QbrSNN.	65
Figura 3.7: Arquitetura do QbrSNN.	66
Figura 3.8: Cromossomo com base em árvore para otimizar simultaneamente parâmetros e atributos de entrada de uma SNN.	67
Figura 3.9: Pseudocódigo do SNN-OGP.	68
Figura 3.10: Arquitetura do SNN-OGP.	69
Figura 5.1: Aparelho Fonador. Fonte (Silva10).	78
Figura 5.2: (a) Parâmetros do sinal glotal; (b) Sinal glotal. Fonte (Silva10).	81
Figura 5.3: Processo para caracterização de uma voz envelhecida.	82
Figura 5.4: Cromossomo com a descrição dos parâmetros para a SNN.	84
Figura 5.5: Curvas de evolução dos modelos para as bases de voz Masculino e Feminino.	85
Figura 5.6: Resultado do cromossomo final utilizando o QbrSNN.	88
Figura 5.7: Resultado do cromossomo final utilizando o SNN-OGP.	89
Figura 5.8: Variações dos parâmetros referentes à base do sexo Masculino.	90
Figura 5.9: Variações dos parâmetros referentes à base do sexo Feminino.	91
Figura 8.1: Modelo de neurônios. Fonte (Izhikevich04).	107
Figura 8.2: Espectro	107
Figura 8.3: Plano z	109
Figura 8.4: Diagrama de fase	109
Figura 8.5: Trato vocal	110
Figura 8.6: Programa <i>Praat</i> .	111

Lista de tabelas

Tabela 2.1: Probabilidades de observação de cada um dos possíveis estados do indivíduo quântico apresentado na equação 32.	40
Tabela 2.2: Parâmetros do AEIQ-BR (Pinho09).	47
Tabela 2.3: Parâmetros de controle de um algoritmo PG.	48
Tabela 3.1: Valores e intervalos dos parâmetros da parte real do cromossomo.	60
Tabela 3.2: Valores da Silhueta (Si).	64
Tabela 3.3: Descrição dos parâmetros utilizados no SNN-OGP.	69
Tabela 4.1: Tabela das bases de dados benchmark.	71
Tabela 4.2: Resultados obtidos para bases benchmark com Si como função de avaliação e o cR para o melhor indivíduo.	72
Tabela 4.3: Valores dos parâmetros utilizados nos modelos: QbrSNN, SNN-OGP e AG.	73
Tabela 4.4: Resultados obtidos para bases benchmark com Si como função de avaliação e o cR para o melhor indivíduo.	73
Tabela 4.5: Valores dos parâmetros (média).	75
Tabela 4.6: Número de atributos selecionados (média).	75
Tabela 5.1: Resultados obtidos para as bases de voz.	84
Tabela 5.2: Resultados obtidos para as bases de voz para diferentes valores de avaliações.	86
Tabela 5.3: Valores dos parâmetros para SNN (média).	86
Tabela 5.4: Intervalos dos atributos na base de dados Feminino.	89
Tabela 5.5: Intervalos dos atributos na base de dado Masculino.	89
Tabela 5.6: Valores dos parâmetros dos locutores na base de dado Feminino.	88
Tabela 5.7: Valores dos parâmetros dos locutores na base de dado Masculino.	92

Tabela 5.8: Valores dos parâmetros dos locutores na base de dados Masculino

90

Acrônimos

RNA – Redes Neurais Artificiais.

SNN – Spiking Neural Networks.

RNP – Rede Neural Pulsada.

LSM – Liquid State Machine.

eSNN – Evolving Spiking Neural Networks.

QiPSO–Evolving Spiking Neural Network Quantum-inspired Particle Swarm Optimization.

QbrSNN – Quantum-Inspired Evolution of Spiking Neural Network with Binary-Real.

SNN-OGP – Spiking Neural Network Optimization by Genetic Programming.

AEIQ-BR – Algoritmo Evolutivo com Inspiração Quântica com representação Binária-Real.

OGP – Otimização por Programação Genética.

AEIQ – Algoritmo Evolutivo por Inspiração Quântica.

EPSP–Excitatory Post Synaptic Potential.

IPSP – Inhibitory Post Synaptic Potential.

HH – Hodgkin-Huxley.

I&F – Integrate and Fire.

SRM – Spiking Response Model.

RBF – Radial Basis Function.

PSP – Post Synaptic Potential.

AE – Algoritmos Evolutivos.

AG – Algoritmos Genéticos.

PG – Programação Genética.

AEIQ-B – Algoritmo Evolutivo com Inspiração Quântica com representação Binária.

AEIQ-R – Algoritmo Evolutivo com Inspiração Quântica com representação Real.

PGMG–Programação Genética Multigênica.

Si – Silhueta.

cR – Correct Rand.

1

Introdução

1.1

Motivação

O envelhecimento da voz é um processo natural que sofre influência hormonal e emocional. Mas, também, com o passar do tempo, o relaxamento das cordas vocais, alteração no movimento das articulações, atrofia da musculatura da laringe, e outras, vão afetando a qualidade vocal e o processo de envelhecimento vocal, conhecido como presbifonia, ocorre.

A qualidade vocal envolve dados anatômicos das estruturas do aparelho fonador e do uso que o indivíduo faz dessas estruturas na emissão da voz. A qualidade vocal é analisada pelos parâmetros do tipo de voz, sistema de ressonância, características da emissão, articulação, pronúncia, ritmo e velocidade da fala; além de resistência vocal, frequência e intensidade (Titze94; Santos05).

Em uma voz envelhecida é possível encontrar diversas alterações como: fendas glóticas, espaços entre as pregas vocais dificultando o fechamento completo; assimetria das pregas vocais, redução na amplitude de vibração causada pela atrofia muscular, voz rouca, tremor na voz, menor sustentação na frequência fundamental, volume diminuído, sopro, pressão subglótica reduzida, menor intensidade, redução do tempo fonatório, redução na velocidade de fala, pausas articulatórias e outras alterações (Santos05).

Uma motivação para essa pesquisa é caracterizar vozes envelhecidas utilizando os parâmetros extraídos dos sinais de voz, através de técnicas computacionais denominadas redes neurais *spiking*.

As Redes Neurais artificiais (RNAs) surgiram por volta de 1940 com o objetivo de modelar processos que ocorrem no cérebro humano, tentando reproduzir aspectos interessantes de seu processamento, de forma computacional, para a solução de problemas.

Do ponto de vista biológico, os modelos tradicionais de redes neurais podem ser considerados biologicamente inspirados, com alto grau de abstração. A construção das RNAs é baseada na transmissão de informações entre

neurônios, que são codificadas através de taxas de pulsos recebidos em um curto intervalo de tempo, utilizando variáveis contínuas para modelar processos que ocorrem através de sequências de pulsos temporais, para sinais variantes no tempo.

O surgimento dessas evidências impulsionou pesquisas para a criação de modelos que não fossem somente biologicamente inspirados, mas também biologicamente plausíveis, permitindo o surgimento de outros tipos de codificação dos dados, novos modelos de neurônios e adaptação dos métodos de treinamento existentes utilizando os pulsos produzidos por um neurônio.

Redes neurais que codificam a informação nos pulsos produzidos pelos neurônios são denominadas Redes Neurais *Spiking* (*Spiking Neural Network* - SNN), e são apresentadas como um novo paradigma conexionista criando uma terceira geração de modelos de redes neurais (Maass97). O processamento da informação das SNNs baseia-se no momento da emissão do pulso criando uma vantagem direta que é o processamento em tempo real da informação, possibilitando modelos que envolvam sinais variantes no tempo.

SNNs têm sido principalmente aplicadas utilizando aprendizagem supervisionada. Há, porém, alguns trabalhos que se referem a problemas de agrupamento e aprendizagem não supervisionada (Bohte03; Adibi05).

Estudos anteriores têm mostrado que as redes neurais que utilizam neurônios *spiking* são, computacionalmente, superiores as redes neurais tradicionais que utilizam neurônios com funções de ativação sigmoideal (Maass97; Gerstner02), obtendo resultados promissores na solução de problemas importantes do mundo real.

Antonio (Antonio04) utiliza SNN em reconhecimento de locutor; Verstraeten *et al.* (Verstraeten05) utiliza SNN em reconhecimento de palavras isoladas utilizando um modelo computacional denominado *Liquid State Machine* (LSM); Knoblauch (Knoblauch05) sugere possíveis soluções que empregam SNN em memória associativa comparando com redes neurais associativas tradicionais na forma do modelo Willshaw clássica; Iannella *et al.* (Iannella05) utilizou SNN para aproximação de funções, com o objetivo de encontrar as componentes que compõem as funções a serem aproximadas; Booij *et al.* (Booij05) desenvolveu uma extensão do algoritmo *SpikeProp* (Bohte03) com base em retropropagação do erro (*error-backpropagation*) para um neurônio que emite múltiplos pulsos (o algoritmo *SpikeProp* foi desenvolvido para classificação de padrões codificados em pulsos individuais); Mohammed *et al.* (Mohammed12) propôs um algoritmo denominado SPAN (*Spike Pattern Association Neuron*)

onde os trens de pulsos são transformados em sinais analógicos durante a fase de aprendizagem, para que operações matemáticas possam ser efetuadas; Kasabov *et al.* (Kasabov11) propôs a construção de novos modelos computacionais probabilísticos neurogenéticos utilizando a doença de Alzheimer como estudo de casos.

Na construção de uma SNN, vários parâmetros que configuram a rede precisam ser estipulados, necessitando de uma combinação correta para que a rede neural tenha um bom desempenho. Isso se torna um problema por não existir regras claras ou específicas de como estes parâmetros devam ser ajustados, e o ajuste manual dos parâmetros de uma SNN torna-se uma tarefa desafiadora.

Semelhante às RNAs, o ajuste dos parâmetros tem uma influência significativa sobre o desempenho final da SNN, sendo esse o motivo de utilizar algoritmos evolutivos (AE) para otimizar essas informações, apresentando-se como uma importante ferramenta na evolução das SNNs.

Na literatura, encontram-se várias propostas que permitem o tratamento de dados espaço-temporal com recursos de otimização de parâmetros de uma SNN (eSNN), com o intuito de conseguir uma melhor precisão em problemas de classificação e previsão. Estudos recentes mostram aplicações de eSNN em reconhecimento de padrões de audiovisual (Wysoski10), reconhecimento de sabores (Soltic08) e reconhecimento visual (Wysoski08), modelagem e reconhecimento de padrões de sinais cerebrais (Kasabov12), estudos referente a problemas ecológicos (Schliebs09) e o desenvolvimento de novos modelos probabilísticos de neurônios *spike* (kasabov10).

Assim como o desempenho de uma SNN está ligado diretamente à combinação correta dos parâmetros da rede, a utilização de um grande número de atributos na entrada não necessariamente traduz em uma maior precisão nos resultados da rede. Em alguns casos, ter menos atributos de entrada pode ajudar a reduzir o tempo de processamento e a aumentar o desempenho da rede. Assim, a otimização dos atributos se apresenta como uma ferramenta essencial de pré-processamento, removendo informações que não contribuem para um bom resultado.

Diversos estudos foram desenvolvidos utilizando otimização dos parâmetros da SNN e atributos da entrada, mais especificamente para tarefas de classificação: Hamed *et al.* (Hamed09) propôs um método para otimizar atributos e parâmetros de uma eSNN utilizando inspiração quântica por enxames de partículas (*Envolving Spiking Neural Network Quantum-inspired Particle Swarm*

Optimization – QiPSO); Hamed *et al.* (Hamed11) propôs uma eSNN utilizando inspiração quântica por exames de partículas para problemas de classificação; Schliebs *et al.* (Schliebs11) criou um modelo baseado em *reservoir* evoluindo uma eSNN para aprendizagem *on-line* de padrão espaço temporal. Schliebs *et al.* (Schliebs13) fornece uma ampla pesquisa bibliográfica sobre redes neurais *spiking* evoluídas; Kasabov *et al.* (Kasabov13) apresenta um novo modelo de SNN evoluída que utiliza aprendizagem *rank-order* e SDSP (*Spike Driven Synaptic Plasticity*) para tipos de aprendizado supervisionado e semi-supervisionado; Bodyanskiy *et al.* (Bodyanskiy14) introduziu diversas alterações na aprendizagem de uma rede *spiking Fuzzy*; Kasabov (Kasabov14) apresentou uma nova arquitetura de uma SNN, chamada NeuCube, para a criação de modelos para mapear, aprender e compreender respostas do cérebro a estímulos externos.

1.2 Objetivos

Baseado na discussão anterior. O objetivo principal desta tese é propor novas abordagens neuro-evolucionárias, com base na SNN utilizada em (Ruff98, Bohte03), onde os parâmetros da SNN são otimizados e os atributos relevantes são automaticamente selecionados para resolver problemas de agrupamento. A utilização de algoritmos evolucionários para a seleção de atributos e determinação dos parâmetros para aprendizado não supervisionado em SNNs é inédito, assim como a sua aplicação em agrupamento de vozes com características de envelhecimento.

1.3 Descrição do trabalho

Dois novos modelos neuro-evolucionários foram propostos denominados: *Quantum-Inspired Evolution of Spiking Neural Networks with Binary-Real representation* (QbrSNN) e *Spiking Neural Network Optimization by Genetic Programming* (SNN-OGP). Esses modelos baseiam-se em dois modelos de algoritmos evolutivos: o Algoritmo Evolutivo com Inspiração Quântica com representação Binário-Real (AEIQ-BR) (Pinho09) e Otimização por Programação Genética (OGP) (Koshiyama13) respectivamente.

Os algoritmos evolutivos de inspiração quântica (AEIQ) têm sido utilizados em problemas de otimização combinatória, podendo ser aplicado na solução onde o espaço de soluções tem uma alta dimensionalidade com desempenho melhor do que os algoritmos convencionais. Além de muitas outras propriedades importantes, AEIQ tem a capacidade de encontrar uma boa solução, mais rapidamente, usando menos indivíduos. Esse recurso reduz drasticamente o número de avaliações necessárias e é um fator de desempenho importante quando o modelo está sendo usado em problemas onde a otimização deve ter um custo computacional (Cruz07).

Programação Genética (PG) é um método pertencente à Computação Evolutiva, biologicamente inspirado e que sintetiza programas de computador para desenvolver alguma tarefa. Tipicamente, a PG utiliza uma população de indivíduos em que cada um é denotado por uma estrutura em árvore, onde cada indivíduo representa um conjunto de expressões matemáticas, de modo que cada uma seja combinada para satisfazer a função objetivo.

Esta tese utiliza uma variante da PG denominada de *Optimization by Genetic Programming* (OGP). O OGP é um algoritmo de otimização baseada em Programação Genética Multigênica (PGMG), no qual um indivíduo é representado como um complexo de estruturas em árvores, sendo uma abordagem clássica com base na teoria da evolução de Darwin (Hinchliffe96; Searson07). É um algoritmo que busca soluções satisfatórias em um grande espaço de busca com uma representação mista, de forma a sintetizar uma solução para atender critérios pré-estabelecidos.

O OGP propicia uma maior facilidade na varredura do espaço de busca, pois as soluções residem no espaço das funções e as mudanças nas operações matemáticas permitem que a solução alterada seja bem diferente da anterior. Os operadores de recombinação são similares ao de um AG clássico, diferente do AEIQ-BR que usa operadores baseados em alterações na função distribuição de probabilidade.

As diferenças entre os modelos QbrSNN e SNN-OGP tendem a não trafegar pelo espaço de busca de maneira similar, e portanto, mesmo com condições iniciais semelhantes, o desempenho final dos algoritmos como por exemplo: melhor solução encontrada e velocidade de convergência devem variar substancialmente.

Os modelos desenvolvidos foram validados através de:

- i. Testes em bases *benchmark* de diferentes níveis de dificuldade.
- ii. Teste para formação de grupos em uma base de dados de voz, utilizando parâmetros acústicos extraídos da voz humana para agrupar vozes envelhecidas.
- iii. Comparação dos resultados dos modelos propostos com uma SNN padrão sem otimização de parâmetros e seleção de atributos. Os resultados da base de voz foram comparados também com uma abordagem clássica de agrupamento *Kmeans* (Kanungo02).
- iv. Comparações das duas novas abordagens neuro-evolucionárias com relação ao custo computacional.
- v. Apresentação de testes estatísticos para evidenciar a eficácia dos modelos.

1.4

Organização do trabalho

Este trabalho está dividido da seguinte forma:

- i. No capítulo 2 será apresentado um resumo dos fundamentos teóricos necessários para uma compreensão das SNNs e dos algoritmos neuro-evolucionários utilizados nesse trabalho.
- ii. No capítulo 3 serão apresentados os modelos neuro-evolucionários propostos.
- iii. No capítulo 4 serão apresentados os resultados obtidos com os novos modelos, em diversas bases de dados *benchmark*.
- iv. No capítulo 5 será apresentado um estudo detalhado dos parâmetros extraídos do sinal de voz, e os resultados obtidos pelos modelos para a aplicação em envelhecimento de voz.
- v. Conclusões e trabalhos futuros serão apresentados no capítulo 6.

2

Conceitos Fundamentais

2.1

Redes Neurais *Spiking*

2.1.1

Introdução

Diversos experimentos e estudos realizados por (Gerstner99) assumem que a codificação da informação no cérebro ocorre na forma de taxa de pulsos. Através desse tipo de codificação, diversas propriedades de neurônios foram descobertas, possibilitando descrever: a estrutura molecular do axônio, a comunicação e a interconexão entre as células nervosas e as ações motoras (Kandel00).

Uma compreensão mais detalhada sobre o funcionamento do neurônio biológico, forma de transmissão das informações e a utilização do tempo do pulso de forma eficiente para processar a informação, permite uma variável adicional no processamento de informações de uma SNN (Nat98; Bohte03).

A classificação dos modelos de Redes Neurais é feita de acordo com suas unidades computacionais, obtendo-se três diferentes gerações (Maass97). A primeira é baseada nos neurônios McCulloch-Pitts como unidades computacionais, tendo como característica principal o fato de o neurônio funcionar de forma binária, ou seja, em um dado instante o neurônio pode estar ativado ou não. A rede neural é constituída por linhas direcionadas, sem pesos, ligando os neurônios como se fossem sinapses, podendo ser excitatórias ou inibitórias. A segunda geração é baseada em unidades computacionais, aplicando funções de ativação com um conjunto de valores de saídas possíveis para uma soma ponderada das entradas. As SNNs são referenciadas como a terceira geração por utilizarem codificação temporal da informação, modelando precisamente interações sinápticas entre os neurônios em função do tempo de emissão dos pulsos [Hélène06].

Redes neurais que utilizam neurônios *spiking* descrevem a entrada da rede em termos de pulsos individuais e como essa entrada leva à geração de

pulsos na saída. São modelos matemáticos que têm por finalidade tentar simular o funcionamento dos neurônios biológicos e suas conexões.

Para o entendimento das SNNs, faz-se necessário aprofundar os conhecimentos dos neurônios biológicos e entender alguns conceitos básicos relacionados com seu funcionamento, fazendo uma analogia entre a biologia e as RNAs.

2.1.2 Neurônios biológicos

Os neurônios são as células básicas que formam o sistema nervoso e podem ser encontrados com formatos variados. Mesmo não havendo um único tipo de neurônio biológico, todos funcionam de forma semelhante respondendo a estímulos provenientes de entradas sensoriais ou de outros neurônios.

A seguir as características comuns entre os neurônios:

- Dendritos: são estruturas responsáveis pela recepção dos pulsos provenientes de outros neurônios e de entradas sensoriais. Os vários dendritos presentes em um neurônio formam a árvore dendrítica.
- Soma: soma ou corpo celular é a parte do neurônio onde efetivamente ocorre o processamento dos pulsos. Com base nas entradas recebidas a soma estabelece se o neurônio vai produzir ou não um sinal de saída.
- Axônio: responsável por propagar o pulso a outros neurônios ou a outras partes do corpo humano, que estejam conectados ao neurônio.

Os pulsos gerados são devidos à variação do potencial da membrana celular do neurônio, devido a estímulos provenientes de outros neurônios. Esses estímulos são chamados de sinais pré-sinápticos e alteram o potencial da membrana celular por um tempo finito, sendo de forma positiva caso sejam provenientes de sinapses excitatórias (EPSP – *Excitatory Post Synaptic Potential*), ou negativa caso sejam provenientes de sinapses inibitórias (IPSP – *Inhibitory Post Synaptic Potential*).

Para que ocorra a emissão de um pulso, o neurônio deve receber estímulos suficientes em intervalos próximos, alterando o potencial da membrana de forma que atinja o valor limite para disparo. Estímulos recebidos

em intervalos de tempo muito espaçados permitem que o potencial da membrana retorne ao seu estado de equilíbrio. A figura 2.1 ilustra o comportamento do potencial da membrana.

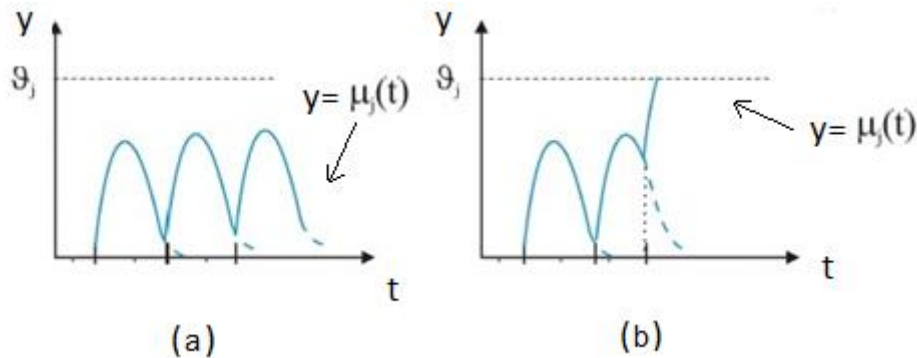


Figura 2.1: Potencial da membrana de um neurônio onde ϑ_j representa o limiar de ativação e μ_j o potencial da membrana: (a) sinais pré-sinápticos excitatórios chegando com tempos espaçados, não permitindo o disparo; (b) sinais recebidos em intervalos pequenos se sobrepõem, chegando ao valor limite e permitindo o disparo do neurônio. Fonte (Alexandre06).

O disparo do neurônio ocorre quando o potencial da membrana μ_j atinge o potencial de ação ϑ_j . O valor do potencial da membrana pode ser alterado por dois processos distintos: estímulos pré-sinápticos e refração.

Estímulos pré-sinápticos foram vistos anteriormente, e o processo de refração se deve a uma limitação que existe após a geração de um pulso. Tal limitação existe porque após a geração do pulso de saída, o neurônio fica impossibilitado de gerar outro pulso por um tempo determinado, mesmo que receba estímulos de entrada (Gerstner02). Os processos são ilustrados na figura 2.2.

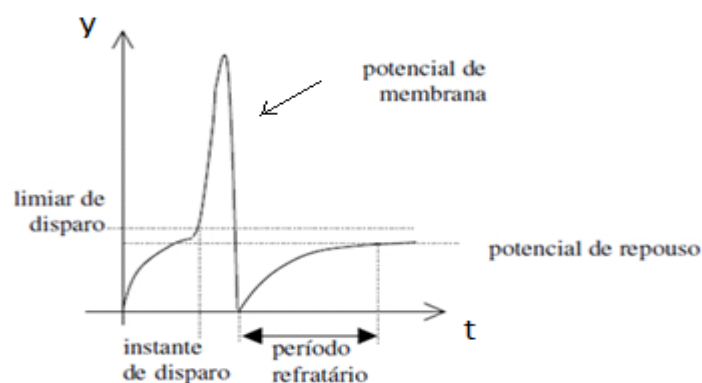


Figura 2.2: Potencial da membrana. Fonte (Maass99).

2.1.3 Modelos de neurônios *spiking*

A escolha de qual modelo de neurônio *spiking* utilizar é baseada em que tipo de fenômeno ou comportamento neural deseja-se simular. Maass (Maass97) classificou os modelos de redes neurais existentes em três gerações conforme visto anteriormente: a primeira geração é baseada nos neurônios McCulloch-Pitts como unidades computacionais, a segunda geração é baseada em unidades computacionais nas quais se aplicam funções de ativação e a terceira geração são modelos de neurônios que, de alguma forma, incorporam informações temporais na computação da ativação do neurônio.

A figura 2.3 apresenta uma classificação desses modelos segundo sua forma mais comum de uso. As seções a seguir apresentam um estudo mais detalhado desses modelos: o modelo Hodgkin-Huxley, o modelo Integra-e-dispara e o modelo SRM (Spike Response Model).

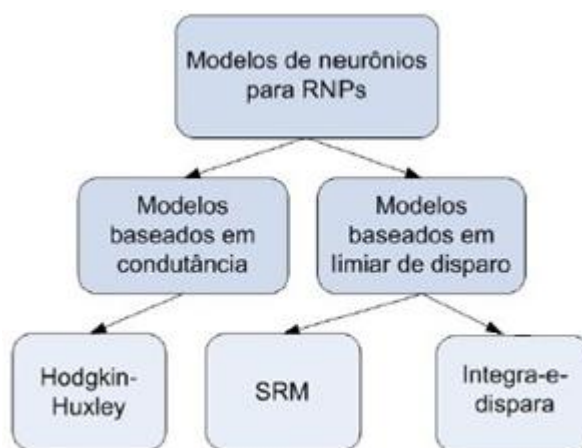


Figura 2.3: Classificação dos modelos de neurônios. Fonte (Alexandre06).

Como existem diferentes linhas de pesquisa envolvendo neurônios *spiking*, diversos modelos de neurônios são encontrados na literatura, os quais podem ser divididos em dois grandes grupos: modelos baseados em condutância e modelos baseados em limiar de disparo. As próximas seções apresentam com mais detalhes os grupos de neurônios.

2.1.3.1

Modelos baseados em condutância

Modelos baseados em condutância são modelos matemáticos que descrevem o funcionamento dos neurônios mais detalhadamente, e são utilizados em simulações nas quais o tempo de processamento não é importante. Tais modelos conseguem reproduzir grande parte das características encontradas nos neurônios biológicos permitindo investigar questões relacionadas com a integração sináptica, dendrítica, correntes iônicas e outras questões relacionadas à dinâmica da célula (Izhikevich04).

As variações entre os modelos desta classe se devem principalmente a utilização dos canais iônicos e os parâmetros das equações diferenciais que modelam o fenômeno do corpo axonal.

Modelo Hodgkin-Huxley (HH)

O modelo matemático de Hodgkin-Huxley (1952) foi desenvolvido com base em experimentos realizados em axônios gigantes de lulas, por ser uma das maiores células de animais conhecidas. Por possuir um alto custo computacional o modelo não é utilizado para criação de aplicações práticas, observando-se uma maior utilização para estudos do comportamento neural direcionado para pesquisa. Basicamente, do modelo HH derivou-se todos os modelos existentes, com ampliações ou simplificações (Maass97).

O modelo é definido por um conjunto de equações diferenciais descrevendo a conservação das cargas elétricas em um pedaço da membrana celular sob a influência de algumas correntes elétricas (Alexandre06).

Por possuir um alto custo operacional, o modelo não é utilizado em aplicações práticas. Geralmente sua utilização é direcionada para o estudo do comportamento neural por parte de pesquisadores do campo da neurociência.

2.1.3.2

Modelos baseados em limiar de disparo

Os modelos baseados em limiar de disparo não descrevem fielmente o funcionamento neural, por ser um modelo simplificado do modelo Hodgkin-Huxley, mas possui uma maior plausibilidade biológica. Nestes modelos, o neurônio de saída gera um pulso quando os valores dos estímulos de entrada

permitem que o potencial de ação da membrana, do neurônio de saída, atinja um determinado valor de *threshold* (limiar de disparo).

Modelos pertencentes a esta classe não consideram características como a estrutura espacial dos neurônios, ou o funcionamento exato dos mecanismos iônicos que atuam nas células nervosas (Hélène06).

Izhikevich (Izhikevich04) apresentou um estudo sobre características que são observadas em neurônios biológicos, comparando diversos modelos de neurônios *spiking* de acordo com a sua capacidade de reproduzir ou não tais características. Ele sugere que modelos baseados em equações diferenciais possam vir a ser uma simplificação do modelo HH (Maass99). Maiores detalhes sobre este estudo ver apêndice A.

Dentre os modelos de limiar de disparo serão apresentados os dois modelos mais encontrados na literatura.

Modelo de Integra-e-Dispara

O modelo de Integra-e-Dispara é o mais utilizado dentre os modelos de limiar de disparo, sobretudo devido a sua extrema simplicidade. Para que ocorra o disparo, o potencial da membrana deve chegar a um valor limiar pré-estabelecido. Quando o potencial chega nesse limiar o neurônio dispara e o período refratário é resetado, fazendo com que o neurônio esteja apto a disparar novamente (Gerstner02).

Modelo de Resposta Impulsiva

Esse modelo é construído a partir de uma abordagem diferente do modelo Integra-e-Dispara, sendo mais genérico. O potencial da membrana pode ser alterado em função de dois diferentes eventos: a chegada de um pulso pré-sináptico ou quando o valor se aproxima do limiar e o neurônio dispara ocorrendo a refração.

O primeiro evento é em função dos pulsos pré-sinápticos que o neurônio recebe de outros neurônios, e que alteram o potencial da membrana por um período de tempo específico tanto positivamente, se forem sinais excitatórios, como negativamente, se forem sinais inibitórios. A equação 1 expressa o comportamento de neurônios com relação aos pulsos pré-sinápticos (Maass99).

$$\varepsilon_{ij}(s) = \left[\exp\left(-\frac{s - \delta_{ij}}{\tau_m}\right) - \exp\left(-\frac{s - \delta_{ij}}{\tau_s}\right) \right] H(s - \delta_{ij}) \quad (1)$$

onde $\varepsilon_{ij}(s)$ é a função que modela o comportamento do neurônio de saída j em função dos pulsos recebidos pela i -ésima sinapse; δ_{ij} é o atraso na propagação axonal da conexão ij ; τ_m, τ_s são respectivamente as constante de tempo de descida e subida da função; $H(s)$ é uma função degrau de *Heaviside*, a qual retorna 0 para valores negativos e 1 para valores maiores ou igual a 0.

O segundo evento capaz de alterar o potencial da membrana é a refração, que ocorre logo após o disparo de um pulso pelo neurônio. A função de refração é modelada segundo a equação 2:

$$\gamma_j(s) = -\mathcal{G}_j \exp\left(-\frac{s}{\tau_r}\right) H(s) \quad (2)$$

onde $\gamma_j(s)$ é a função que modela a refração do neurônio j de saída; \mathcal{G}_j é o limiar do neurônio de saída j ; τ_r é uma constante de tempo; $H(s)$ é a função degrau de *Heaviside*.

Uma vez modelados os eventos que alteram o potencial da membrana $u_j(t)$, pode-se apresentar a equação que descreve o modelo de resposta impulsiva em função do potencial da membrana (Maass99) (equação 3):

$$u_j(t) = \sum_{t_j^f \in T_j} \gamma_j(t - t_j^f) + \sum_{i \in \Gamma_j} \sum_{t_i^f} w_{ij} \varepsilon_{ij}(t - t_i^f) \quad (3)$$

(1)
(2)

onde γ_j é a função que modela a refração do neurônio j de saída; w_{ij} é o peso sináptico associado à i -ésima sinapse do neurônio j ; ε_{ij} é a função que modela o comportamento do neurônio de saída j em função dos pulsos recebidos pela i -ésima sinapse; t é a variável tempo; t_i^f é o momento do f -ésimo disparo da sinapse i ; t_j^f é o momento do f -ésimo disparo do neurônio j .

A equação (1) modela a refração do neurônio j para todo disparo de j , e (2) modela o conjunto de todos os pulsos pré-sinápticos recebidos pelo neurônio de saída j de todos os neurônios pré-sinápticos a ele. Esse modelo de neurônio é denominado *Spiking Response Model* (SRM).

O modelo apresentado nesta seção será utilizado no decorrer deste trabalho.

2.1.3.3 Comparativos com modelos tradicionais

Dentre os modelos apresentados, os que utilizam limiar de disparo são os mais adequados para implementações similares às RNAs tradicionais devido ao menor número de parâmetros e seu baixo custo computacional. Os neurônios *spiking* funcionam como uma unidade que soma os sinais de entrada e dispara caso esta soma alcance um limiar pré-estabelecido. Pode-se fazer uma analogia com o neurônio de McCulloch-Pitts (equação 4).

$$y = \begin{cases} 1 \rightarrow \sum_{k=1}^n w_k x_k \geq \mathcal{G} \\ 0 \rightarrow cc \end{cases} \quad (4)$$

onde y é a saída do neurônio; n é o número de entradas do neurônio; w_k é o peso sináptico associado a k -ésima; x_k é o valor presente na k -ésima entrada do neurônio; \mathcal{G} o limiar de disparo do neurônio de saída.

A saída da rede McCulloch-Pitts apresenta a saída igual a 1 pela soma ponderada dos pesos, e um neurônio *spiking* dispara um pulso quando a soma dos pulsos pré-sinápticos atingem o valor limiar.

É também possível fazer comparação de um neurônio *spiking* com um neurônio RBF (*Radial Basis Function*) proposto por Hopfield (Hopfield95). Esse modelo de neurônio tem como característica principal a ocorrência de funções de ativação, geralmente gaussianas, sintonizadas em torno de uma região. A resposta de um neurônio RBF é incrementada à medida que o vetor de entrada se aproxima do vetor que representa o centro da função de ativação.

O tempo de disparo, de um neurônio *spiking*, representa a distância do sinal de entrada recebido do centro de uma função de base radial. Para que isso ocorra, as conexões sinápticas da rede têm que possuir unidades de atrasos associados. Quando os atrasos das sinapses, que chegam a um determinado

neurônio de saída, estão sintonizados, isto faz com que o neurônio dispare. Quanto menor a diferença entre os atrasos, mais próximo do centro da função RBF se encontra a entrada associada. Caso essa diferença aumente, o neurônio vai demorar a disparar (Hopfield95).

Existem alguns trabalhos que as SNNs utilizam os neurônios de sua última camada como unidades RBF.

A seguir, apresenta-se a arquitetura básica da rede de forma similar à proposta por (Hopfield95), a arquitetura com desdobramento de sinapses (Nat01), e a codificação dos dados de entrada utilizando funções gaussianas proposto por (Bohte03).

2.1.4

Arquitetura de uma SNN

A arquitetura de uma rede neural descreve a disposição dos neurônios e o padrão de conexão entre eles. De maneira semelhante às redes artificiais tradicionais, as redes *spiking* têm uma arquitetura adequada ao problema que se deseja abordar. As redes *spiking* propostas por (Nat98) têm a mesma estrutura das redes tradicionais.

Nas conexões sinápticas das redes *spiking* existem pesos sinápticos que influenciam no sinal transmitido, alterando a intensidade do pulso recebido pelo neurônio.

Uma arquitetura básica consiste de neurônios totalmente conectados como o ilustrado na figura 2.4, formando duas camadas denominadas entrada *I* e saída *J*. Os pulsos são transmitidos da camada de entrada para os neurônios de saída (Alexandre06).

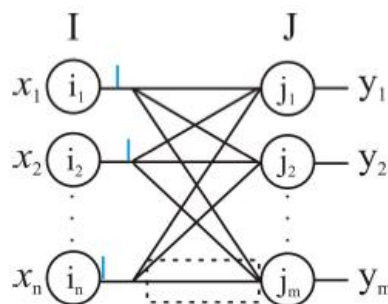


Figura 2.4: Arquitetura da rede *spiking*. Fonte (Alexandre06).

Uma diferença que pode ser encontrada nas sinapses que compõem alguns tipos de redes *spiking*, é a existência de unidades de atrasos, que fazem

com que a propagação dos pulsos seja retardada por um determinado período de tempo estabelecido, podendo este atraso ser diferente para cada sinapse.

Em algumas arquiteturas foram inseridas o conceito de múltiplas sinapses, também chamado de desdobramento sináptico, que será detalhado a seguir.

2.1.4.1

Arquiteturas com desdobramento de sinapses

Com o intuito de melhorar o desempenho da rede, foi proposto por (Nat01) uma modificação na arquitetura aumentando o número de conexões sinápticas entre os neurônios, chamado de desdobramento sináptico. Neste tipo de arquitetura, uma conexão sináptica entre dois neurônios, que compõem a rede neural, é formada por múltiplas sinapses sendo cada conexão com sua unidade de atraso e peso associado, conforme mostrado na figura 2.5.

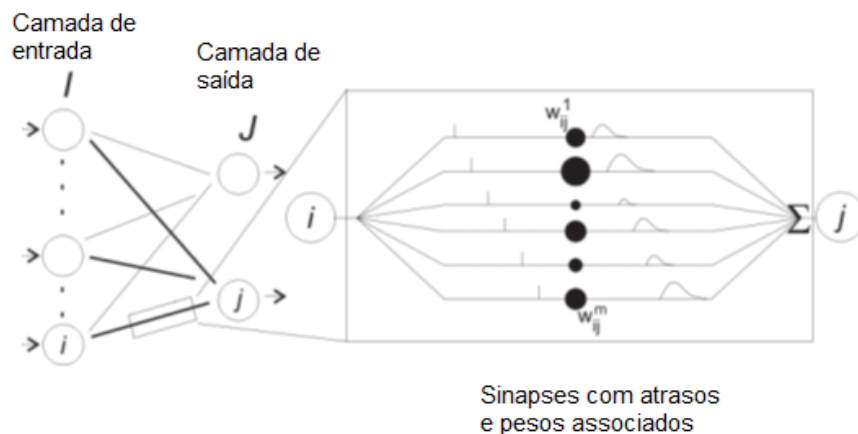


Figura 2.5: Exemplo de arquitetura de rede com múltiplas sinapses.

Fonte (Bohte03).

É possível observar que, uma conexão entre dois neurônios i e j (com $i \in I$ e $j \in J$) existem m sinapses com diferentes atrasos de propagação e pesos (w) sinápticos associados.

Na rede utilizada neste trabalho, foram utilizados neurônios de saída modelados como SRM (*Spike Response Model*), operando como neurônios RBF (*Radial Basis Function*). A resposta de um neurônio RBF é tipicamente incrementada à medida que o vetor de entrada $\vec{X} = (x_1, x_2, \dots, x_n)$ se aproxima do vetor que representa o centro da função de ativação, que neste caso é o vetor de atrasos axonais $\vec{\Delta} = (\delta_1, \delta_2, \dots, \delta_n)$. O vetor de atrasos pode assumir valores conforme mostrado na equação 5, onde m é a quantidade de sinapses.

$$\vec{\Delta} = (\delta_{ij}^1, \delta_{ij}^2, \dots, \delta_{ij}^m) = (0, 1, \dots, m-1) \quad (5)$$

A característica principal das redes *spiking*, que utilizam múltiplas conexões sinápticas de forma semelhante às redes tradicionais, é que somente os pesos são atualizados durante o processo de treinamento e não os atrasos associados às sinapses.

2.1.5 Codificação em pulsos dos dados de entrada

Uma questão importante quando se trabalha com redes *spiking* é como os neurônios codificam a informação de entrada em um trem de pulsos. Apesar da existência de vários tipos de codificação, será detalhada nesta seção a codificação proposta por Bohte (Bohte03), que utiliza campos receptivos (*receptive fields*) gaussianos como forma de codificar dados de entrada, originando um grande número de disparos defasados no tempo.

Cada atributo do vetor de entrada de dimensão n é aplicado a um conjunto de g gaussianas cobrindo toda a extensão dos dados de entrada. A figura 2.6 ilustra a codificação de um atributo do vetor de entrada \vec{X} igual a 0,3. Neste exemplo, a dimensão do vetor de entrada é de -1 a 2, utilizando 6 gaussianas com atrasos variando de 0 a 10 ms.

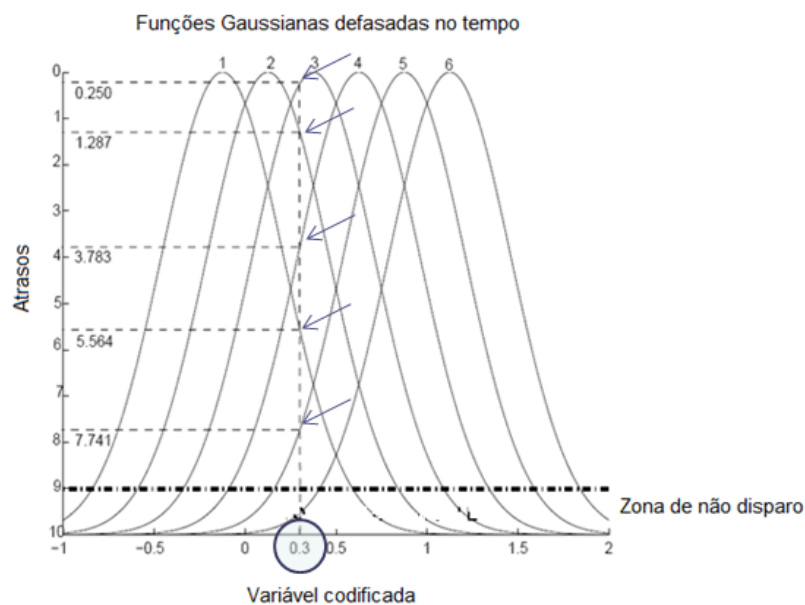


Figura 2.6: Codificação utilizando campos receptivos gaussianos.

O atributo de entrada 0,3 foi codificado em 6 neurônios com os respectivos atrasos: neurônio 1 (5,564ms), neurônio 2 (1,287ms), neurônio 3 (0,250ms), neurônio 4 (3,783ms), neurônio 5 (7,741ms). O neurônio 6 não dispara porque a interseção entre a projeção do atributo com a gaussiana 6 se encontra na zona de não disparo (neurônios menos estimulados).

O centro e a largura das gaussianas são determinados da seguinte forma: considerando uma variável de entrada com dimensão n : $[I_{min}, I_{max}]$, com m gaussianas utilizadas no campo receptivo. Para o i -ésimo neurônio codificando a variável n , o centro (C_i) e a largura (σ) das gaussianas são definidos nas equações 6 e 7 respectivamente:

$$C_i = I_{min}^n + \left(\frac{2i-3}{2} \right) \left(\frac{I_{max}^n - I_{min}^n}{m-2} \right) \quad m > 2 \quad (6)$$

$$\sigma = \frac{1}{\gamma} \left(\frac{I_{max}^n - I_{min}^n}{m-2} \right) \quad m > 2 \quad (7)$$

Assim, seja um vetor de entrada de dimensão n aplicada a um conjunto de m gaussianas, o número de neurônios na camada de entrada será composto por $n.m$ neurônios, isto é, pelos números de atributos de entrada e o número de gaussianas utilizadas para a codificação.

2.1.6 Aprendizado em SNN

Para que as redes *spiking* aprendam e possam realizar tarefas específicas, é essencial a aplicação de algoritmos de treinamento como nas RNAs tradicionais. A propriedade mais importante das redes neurais é a habilidade de aprender em seu ambiente, e com isso generalizar o seu conhecimento. Isso é feito através de um processo iterativo de ajustes aplicado a seus pesos.

Para redes *spiking* os mesmo tipos de aprendizado de máquina são utilizados:

- **Aprendizado supervisionado** - a rede é treinada através do fornecimento dos valores de entrada e de seus respectivos valores desejados de saída. Os parâmetros da rede são determinados através da minimização do erro. O mais conhecido é o **SpikeProp**

desenvolvido por Bohte (Bohte03), que utiliza os mesmos procedimentos baseados em gradiente.

- **Aprendizado não supervisionado** - a rede determina seus parâmetros sem o conhecimento da saída desejada.
- **Aprendizado por reforço** – neste caso utiliza-se uma função heurística para descrever a qualidade da resposta da rede a certa entrada. A qualidade será representada por recompensa ou penalização. Os parâmetros são determinados através da maximização (ou minimização) da recompensa ou penalização.

Como os modelos propostos nesta tese são para agrupamento, somente o aprendizado não supervisionado será descrito neste trabalho.

2.1.6.1

Aprendizado não supervisionado

Um modelo de aprendizado específico para um neurônio SRM implementando função de base radial foi proposto por Gerstner (Gerstner96) e, posteriormente, um modelo de aprendizado não supervisionado para uma rede com esse tipo de neurônio, foi apresentado por Nat (Nat98). Bohte (Bohte03) ampliou a capacidade e a precisão dessas redes para a utilização de dados reais.

O objetivo do aprendizado não supervisionado de uma rede *spiking*, lembrando que no algoritmo não supervisionado a rede determina seus grupos sem o conhecimento prévio da saída desejada, é fazer com que os neurônios RBF armazenem os centros de cada grupo, fazendo com que dispare o neurônio da camada de saída cujo centro esteja mais próximo do padrão de pulso recebidos na camada de entrada (Bohte03).

Os pesos são inicializados de forma aleatória, e após análise dos vetores de entrada na rede, o neurônio da camada de saída que disparar primeiro será considerado neurônio vencedor (*Winner takes all*).

A atualização dos pesos só ocorre no neurônio RBF vencedor, de acordo com a expressão $\eta L(\Delta t)$, onde η (taxa de aprendizado) e L uma função Hebbiana, mostrada na equação 8:

$$L(\Delta t) = (1 - b)e^{-\frac{(\Delta t - c)^2}{\beta^2}} + b \quad (8)$$

onde b determina o valor com que os pesos sinápticos são incrementados e decrementados, c determina o centro da curva de aprendizado, Δt a diferença do tempo de recebimento de um pulso e o disparo do neurônio em questão. O gráfico da função Hebbiana é ilustrado na figura 2.7.

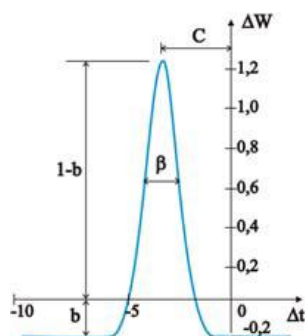


Figura 2.7: Gráfico da função de aprendizado. Fonte (Bohte03).

A aplicação deste tipo de aprendizado apresenta algumas características (Alexandre06):

- **Antecipação do disparo:** o treinamento da rede aumenta a sensibilidade dos neurônios, fazendo-os disparar. Quando isso ocorre, os neurônios RBF passam a disparar mais cedo com as sucessivas rerepresentações dos padrões de treinamento (Nat98).
- **Deslocamento do vetor de pesos:** a ideia básica do processo de treinamento é que o neurônio tem seu peso incrementado antes do disparo. Os pesos sofrem um deslocamento nas sinapses com a evolução do treinamento (Alexandre06).

A seguir será apresentado uma breve introdução aos algoritmos Neuro-Evolutivos.

2.2 Algoritmos Neuro-Evolucionários

Algoritmos evolutivos (AE) é um ramo da inteligência computacional que procura imitar a evolução biológica, com o intuito de resolver problemas de buscas e otimização baseado no processo genético. Esses algoritmos trabalham com uma população de indivíduos para encontrar soluções viáveis para um determinado problema.

Esses algoritmos têm o seguinte procedimento: codifica-se cada possível solução de um problema em uma estrutura denominada “cromossomo” que é composta por uma cadeia de bits, caracteres ou números reais. Esses cromossomos representam indivíduos, que são evoluídos ao longo de várias gerações (Mitchell92).

O processo de evolução começa com a criação aleatória dos indivíduos que formam a população inicial. A partir de um processo de seleção baseado na aptidão de cada indivíduo, são escolhidos indivíduos para a nova fase de reprodução, que cria novas soluções. Para criar novas soluções, são utilizados operadores genéticos.

Os operadores mais conhecidos são: Seleção, Cruzamento e Mutação. Seleção, conforme mencionado acima é o processo de escolha de cromossomos da população, baseado na sua aptidão, para a fase de reprodução, de novos indivíduos. Cruzamento é um operador baseado na troca de partes dos cromossomos, combinando suas características para formar soluções. Mutação é a troca aleatória do valor contido em um gene de um cromossomo por outro valor possível (Mitchell92). Esses operadores promovem modificações nos indivíduos da população, os quais serão avaliados a cada ciclo por uma função de avaliação.

Diversos tipos de AE podem ser encontrados na literatura, dentre os quais pode-se citar:

- Algoritmos Genéticos (AG);
- Programação Genética (PG);
- Evolução Diferencial;
- Algoritmos Culturais;
- Programação Evolutiva.

AE são frequentemente utilizados para a evolução de RNAs (Cruz07), pois permitem solucionar alguns problemas encontrados na modelagem através de algoritmos tradicionais. Problemas de supertreinamento, ocasionado pelo elevado número de épocas de treinamento e/ou elevado número de neurônios na camada escondida, e problemas de convergência prematura a um mínimo local são constantemente encontrados na modelagem de RNAs pelo método do gradiente decrescente e retropropagação do erro (Pinho09). Assim, o uso de AE para determinação automática da estrutura e dos valores dos pesos sinápticos de uma RNA, permite a criação de redes mais eficientes.

Nas duas próximas seções serão apresentados os algoritmos AEIQ-BR (Algoritmo Evolutivo com Inspiração Quântica com representação Binário-Real) e o OGP (*Optimization by Genetic Programming*), utilizados na composição desta tese. A utilização desses algoritmos se deve ao AEIQ-BR tem a capacidade de encontrar uma boa solução mais rapidamente utilizando menos indivíduos, reduzindo o número de avaliações (Pinho09). O OGP é um algoritmo que explora conceitos da Programação Genética e da Programação Genética Multigênica (PGMG), de forma a codificar uma expressão matemática em cada árvore presente em um indivíduo da população, de modo que cada uma seja combinada para satisfazer a função objetivo. A representação é feita por uma estrutura em árvore, com base em princípios de programação genética, buscando soluções satisfatórias em um grande espaço de busca com uma representação mista (Koshiyama13), ideal para o problema abordado nesta tese.

2.2.1

Algoritmos Evolutivos com Inspiração Quântica com representação Binária-Real - AEIQ-BR.

O termo “computação quântica” é utilizado para descrever processos computacionais que se baseiam em fenômenos específicos da Mecânica Quântica, capazes de diminuir o esforço e a complexidade na solução de determinados problemas da computação clássica.

Em Mecânica Quântica, é possível que uma partícula esteja em dois ou mais estados ao mesmo tempo. Esta capacidade de estar simultaneamente em vários estados denomina-se *superposição*. Esta propriedade de superposição de estados faz com que todo o espaço de busca de um problema seja considerado simultaneamente, mesmo que certos estados não venham a ser utilizados, tornando dispositivos quânticos exponencialmente mais eficientes que um dispositivo clássico.

O uso de sistemas clássicos inspirados nesses fenômenos pode produzir um ganho substancial em relação ao tempo de processamento e à qualidade da solução (Han02; Cruz07).

Computadores clássicos interpretam impulsos elétricos como *bits* (*binary digit*), sendo o *bit* a menor unidade de informação em computadores clássicos. Em computadores quânticos, a menor unidade de informação é o *qbit* (*quantum bit*), que pode representar um estado 0, 1 ou uma superposição de ambos estados (Zhang09).

Os primeiros AE com inspiração quântica foram criados com representação binária, onde cada gene representa um *qbit*. Entretanto, problemas de otimização numérica, como é o caso da determinação automática dos pesos de uma rede neural, são mais eficientemente tratados com cromossomos com representação real (Cruz07). Assim, o modelo AE com inspiração quântica original (Han02) foi estendido para representação real pura e híbrida (binária-real).

A seguir serão descritos os algoritmos evolutivos com Inspiração Quântica com representação Binária e Real, já que o AE mais adequado para a proposição de neuro-evolução envolve os dois tipos de representação.

Algoritmo Evolutivo com Inspiração Quântica com Representação Binária (AEIQ-B).

O algoritmo evolutivo com inspiração quântica usando representação binária foi proposto inicialmente por Han (Han02). Este algoritmo é caracterizado por um cromossomo, uma função de avaliação e uma dinâmica populacional. A representação binária utiliza um cromossomo formado por qbits conforme equação 9.

$$[\alpha, \beta]^T \quad (9)$$

onde os números α e β , que são números reais, satisfazem à condição de normalização (equação 10):

$$|\alpha|^2 + |\beta|^2 = 1 \quad (10)$$

Um estado ψ de um *qbit* pode ser representado por (equação 11).

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle \quad (11)$$

Ou seja, o *bit* é substituído pelo *qbit*, e os valores 0 e 1 do *bit* são substituídos pelos vetores $|0\rangle$ e $|1\rangle$, representado pela equação 12.

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ e } |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (12)$$

O valor $|\alpha|^2$ indica a probabilidade de que o *qbit* terá valor 0 quando observado, e o valor $|\beta|^2$ indica a probabilidade do *qbit* ter valor 1 quando observado. É possível visualizar na figura 2.8 a relação entre α e β .

No momento em que uma solução é observada em um destes sub-espacos de soluções, todos os outros estados e possíveis soluções deixam de existir (Pinho09).

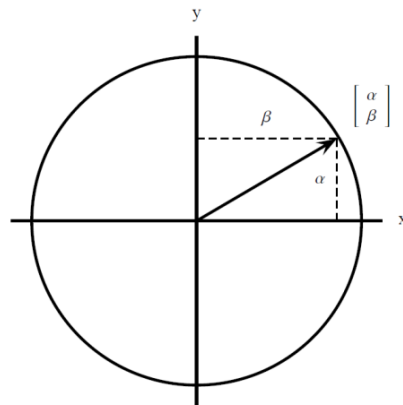


Figura 2.8: Representação gráfica de um círculo unitário com amplitude de probabilidade de se observar os valores 0 e 1 para um *qbit* qualquer. Fonte (Pinho09).

Deste modo cada indivíduo quântico representa uma superposição de indivíduos formados por m genes. Um indivíduo formado por m *qbits* com valores mostrados na equação 13, pode representar 2^m possíveis padrões simultâneos, devido à representação probabilística e superposição de todos os estados.

$$q = \begin{bmatrix} \alpha_1 \\ \beta_1 \end{bmatrix} \begin{bmatrix} \alpha_2 \\ \beta_2 \end{bmatrix} \dots \begin{bmatrix} \alpha_m \\ \beta_m \end{bmatrix} \text{ onde: } |\alpha|^2 + |\beta|^2 = 1, i = 1, 2, \dots, m. \quad (13)$$

Por exemplo, um indivíduo quântico formado por 3 *qbits*, pode ser descrito conforme a equação 14.

$$q = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix} \quad (14)$$

Para se calcular a amplitude de probabilidade do estado $|000\rangle$ (o módulo da amplitude de probabilidade ao quadrado é igual à probabilidade de se observar o estado), multiplicam-se as amplitudes de probabilidade de se observar os estados 0 em cada um dos *qbits* $(\alpha_1, \alpha_2, \alpha_3)$. Para o cálculo da amplitude do estado $|001\rangle$ observam-se os *qbits* $(\alpha_1, \alpha_2, \beta_3)$, e assim sucessivamente. Assim, a possibilidade de observação dos estados pode ser representada por:

$$\frac{1}{4}|000\rangle + \frac{\sqrt{3}}{4}|001\rangle + \frac{1}{4}|010\rangle + \frac{\sqrt{3}}{4}|011\rangle + \frac{1}{4}|100\rangle + \frac{\sqrt{3}}{4}|101\rangle + \frac{1}{4}|110\rangle + \frac{\sqrt{3}}{4}|111\rangle$$

É possível montar a tabela 2.1 com os possíveis estados desse indivíduo, com as probabilidades de se observar todos os possíveis estados.

Estado	Probabilidade
$ 000\rangle$	$(1/\sqrt{2})^2 \cdot (1/\sqrt{2})^2 \cdot (1/2)^2 = 1/16$
$ 001\rangle$	$(1/\sqrt{2})^2 \cdot (1/\sqrt{2})^2 \cdot (\sqrt{3}/2)^2 = 3/16$
$ 010\rangle$	$(1/\sqrt{2})^2 \cdot (1/\sqrt{2})^2 \cdot (1/2)^2 = 1/16$
$ 011\rangle$	$(1/\sqrt{2})^2 \cdot (1/\sqrt{2})^2 \cdot (\sqrt{3}/2)^2 = 3/16$
$ 100\rangle$	$(1/\sqrt{2})^2 \cdot (1/\sqrt{2})^2 \cdot (1/2)^2 = 1/16$
$ 101\rangle$	$(1/\sqrt{2})^2 \cdot (1/\sqrt{2})^2 \cdot (\sqrt{3}/2)^2 = 3/16$
$ 110\rangle$	$(1/\sqrt{2})^2 \cdot (1/\sqrt{2})^2 \cdot (1/2)^2 = 1/16$
$ 111\rangle$	$(1/\sqrt{2})^2 \cdot (1/\sqrt{2})^2 \cdot (\sqrt{3}/2)^2 = 3/16$

Tabela 2.1: Probabilidades de observação de cada um dos possíveis estados do indivíduo quântico apresentado na equação 32.

Em cada passo evolutivo, um *qbit* de um indivíduo quântico é atualizado de tal forma a convergir para uma probabilidade máxima igual a 1 para os estados $|0\rangle$ ou $|1\rangle$ do melhor indivíduo possível. Esse processo de convergência é feito através de operadores quânticos e funciona como uma atualização de cada gene dos indivíduos.

No algoritmo quântico binário, Han (Han00) utilizou o operador porta de Rotação (θ é o ângulo de rotação), ou *q-gate*, conforme visto na equação 15.

$$U(\theta) = \begin{bmatrix} \cos(\theta) & (+ \text{ ou } -)\sin(\theta) \\ (+ \text{ ou } -)\sin(\theta) & \cos(\theta) \end{bmatrix} \quad (15)$$

Essa matriz é utilizada na atualização da população, multiplicando cada coluna do indivíduo quântico (equação 16).

$$\begin{bmatrix} \alpha' \\ \beta' \end{bmatrix} = U(\theta) \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} \quad \text{onde } i=1, 2, \dots, j \quad (16)$$

Os valores de θ são definidos através de uma tabela, de modo que a matriz de rotação seja capaz de modificar os valores de $(\alpha_i \text{ e } \beta_i)$, aumentando às chances de observação dos melhores indivíduos (Pinho09).

O AEIQ-B é definido como na figura 2.9.

```

iniciar
   $t \leftarrow 0$ ;
  inicializa  $Q(t)$ 
  gera  $P(t)$  observando estados de  $Q(t)$ 
  avalia  $P(t)$ 
  armazena as melhores soluções de  $P(t)$  em  $B(t)$ 
  enquanto não ocorrer condição de parada
     $t \leftarrow t + 1$ 
    gera  $P(t)$  observando estados de  $Q(t - 1)$ 
    avalia  $P(t)$ 
    atualiza  $Q(t)$  usando  $q\text{-gate}$ 
    armazena as melhores soluções de  $B(t - 1)$  e  $P(t)$  em  $B(t)$ 
    armazena a melhor solução  $b$  de  $B(t)$ 
  fim
fim

```

Figura 2.9: Pseudo-código do AEIQ-B. Fonte (Pinho09).

Algoritmo Evolutivo com Inspiração Quântica com Representação Real (AEIQ-R).

O algoritmo quântico com representação binária não é o mais adequado quando se deseja proceder a uma otimização numérica. Para contornar esse problema, Cruz (Cruz07) apresentou um modelo com inspiração quântica que utiliza representação baseada em números reais, sendo mais adequado para

problemas de otimização numérica (a variável aleatória deixa de ser discreta e passa a ser contínua). Este modelo foi denominado AEIQ-R.

A figura 2.10 mostra a listagem completa do algoritmo evolutivo com inspiração quântica com representação real – AEIQ-R.

De forma idêntica ao AEIQ-B, o algoritmo com representação real necessita de uma população de indivíduos que representem a superposição dos possíveis estados que o indivíduo clássico pode assumir ao ser observado (Cruz07).

```

iniciar
1.   $t \leftarrow 1$ 
2.  Gerar população quântica  $Q(t)$  com  $N$  indivíduos com  $G$  genes
3.  enquanto ( $t \leq T$ )
4.     $E(t) \leftarrow$  gerar indivíduos clássicos observando indivíduos quânticos
5.    se ( $t=1$ ) então
6.       $C(t) \leftarrow E(t)$ 
7.    senão
8.       $E(t) \leftarrow$  recombinação entre  $E(t)$  e  $C(t)$ 
9.      avaliar  $E(t)$ 
10.      $C(t) \leftarrow K$  melhores indivíduos de  $[E(t) \cup C(t)]$ 
11.   fim se
12.    $Q(t+1) \leftarrow$  Atualiza  $Q(t)$  usando os  $N$  melhores indivíduos de  $C(t)$ 
13.    $t \leftarrow t + 1$ 
14. fim enquanto
fim

```

Figura 2.10: Algoritmo com inspiração quântica com representação real – AEIQ-R. Fonte (Cruz07).

Esta representação deve respeitar a condição de que o conjunto de estados observáveis deve ser contínuo e não discreto como no caso do AEIQ-B. Assim, uma população quântica $Q(t)$, em um instante t qualquer do processo evolutivo é formada por um conjunto de N indivíduos quânticos $q_i (i = 1, 2, 3, \dots, N)$. Cada indivíduo quântico q_i desta população é formado por G genes $g_{ij} (j = 1, 2, 3, \dots, G)$ que, por fim, são formados por funções que representam uma densidade de probabilidade. Os indivíduos quânticos podem ser descrito pela equação 17 (Cruz07).

$$q_i = [g_{i1} = p_{i1}(x), g_{i2} = p_{i2}(x), \dots, g_{iG} = p_{iG}(x)] \quad (17)$$

onde as funções p_{ij} representam as funções densidade de probabilidade com $(i = 1, 2, 3, \dots, N)$ e $(j = 1, 2, 3, \dots, G)$. Esta função densidade de probabilidade é usada pelo AEIQ-R para gerar os valores para os genes dos indivíduos clássicos, ou seja, a função $p_{ij}(x)$ deve ser integrável na região do domínio dentro do qual as variáveis que se deseja otimizar podem assumir valores. Isto permite calcular a distribuição cumulativa de probabilidade e assim usar a distribuição de probabilidade para gerar valores para os genes clássicos por meio de processo de observação (amostragem) dos indivíduos quânticos (Cruz07).

A função de densidade de probabilidade escolhida por Cruz foi à da variável aleatória uniforme. Um exemplo é ilustrado na figura 2.11 onde a variável x , que se deseja otimizar, está no intervalo $[-1, 1]$. Esta curva representa a superposição de todos os estados possíveis para a variável x . Este conjunto de distribuição de cada variável forma uma superposição para cada uma das variáveis do problema.

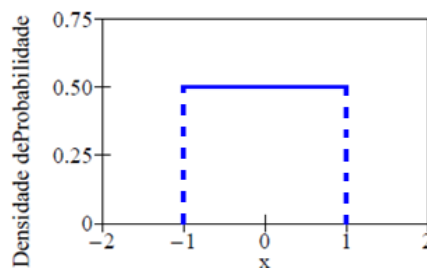


Figura 2.11: Exemplo de gene quântico do AEIQ-R. Fonte (Cruz07).

O gene quântico pode ser representado pelos valores dos limites inferior e superior, ou o valor do centro(μ) e a largura do pulso(σ). A partir dessa distribuição, sorteia-se um conjunto de n pontos que formam a população.

O passo seguinte do algoritmo realiza uma operação de recombinação entre os indivíduos da população nova (geração atual) e da população antiga (geração anterior), com a finalidade de permitir o aproveitamento do material genético, já presente na população clássica, melhorando a capacidade do algoritmo explorar regiões próximas dentro do espaço de busca.

Com a nova população clássica gerada, deve-se tomar a decisão sobre como substituir os indivíduos de uma eventual população pré-existente. Existem algumas abordagens possíveis para lidar com esta etapa do algoritmo:

- Substituir todos os indivíduos da população antiga pelos indivíduos da população nova;
- Substituir todos os indivíduos da população antiga pelos indivíduos da população nova, preservando o melhor indivíduo da população nova;
- Substituir os piores indivíduos da população antiga pelos melhores indivíduos da população nova;
- Substituir os piores indivíduos da população antiga por indivíduos quaisquer da população nova.

Finalmente, após a geração dos indivíduos clássicos, é necessário atualizar os indivíduos da população quântica. Este procedimento depende do tipo de função densidade de probabilidade que se definiu para usar nos genes quânticos. Em geral, este processo deve:

- Reduzir o espaço de busca da função que se quer otimizar. No AEIQ-R isto é feito reduzindo-se o tamanho da região onde a função densidade de probabilidade (genes quânticos) tem probabilidade diferente de 0;
- Mapear as regiões mais promissoras do espaço de busca. Isto deve ser feito aumentando-se a probabilidade de se observar um determinado conjunto de valores para o gene clássico nas proximidades dos indivíduos mais bem-sucedidos da população clássica.

Na primeira geração, o tamanho do domínio para a função que se deseja otimizar deve ser o maior possível para que todas as regiões do espaço de busca seja mapeadas. Os genes de todos os indivíduos quânticos devem estar definidos de tal forma que, todos os valores possíveis dentro do intervalo (domínio) tenham chance de serem observados. Obviamente que, no caso de se ter informações *a priori* sobre regiões mais promissoras, as funções densidade de probabilidade podem ser redefinidas de tal forma que os indivíduos quânticos ofereçam maiores chances de que genes na região mais promissora do espaço de busca sejam observados com mais frequência do que em outras regiões.

Utilizando a função de densidade de probabilidade da variável aleatória uniforme, para a atualização dos genes quânticos durante o processo evolutivo, podem-se utilizar os seguintes processos:

- Modificar a largura da curva, alterando os limites: inferior e superior, de modo a reduzir gradativamente o espaço de busca;
- Modificar a posição da curva de modo que o ponto central fique mais próximo do valor do indivíduo clássico.

Para se realizar os processos descritos anteriormente, podem-se usar algumas estratégias. Para o primeiro processo, pode-se utilizar um decaimento exponencial da largura da curva, um decaimento linear ou, ainda, um processo similar ao usado em algoritmos de estratégia evolutiva chamada “regra do 1/5”. Para o segundo processo, pode-se definir em termos de quais indivíduos da população clássica serão usados para atualizar os pulsos da população quântica. Podem ser escolhidos indivíduos mais aptos, os indivíduos menos aptos, um conjunto aleatório de indivíduos ou usar um processo de roleta para selecionar quais os indivíduos que farão parte do grupo de indivíduos clássicos que serão usados para atualizar os indivíduos quânticos. Maiores detalhes, sobre as estratégias citadas acima, podem ser vistos em Cruz (Cruz07).

Após a escolha dos indivíduos que serão usados (o número de indivíduos clássicos escolhidos deve ser igual ao número de indivíduos quânticos e, portanto, o número de indivíduos clássicos não pode ser menor do que o número de indivíduos quânticos), o centro dos pulsos de cada indivíduo quântico deve ser modificado em relação ao valor de cada gene quântico. Em geral, pode-se usar um cálculo que desloque o centro dos pulsos na direção do ponto indicado pelo gene do indivíduo clássico. A figura 2.12 ilustra o diagrama completo do sistema evolutivo do algoritmo.

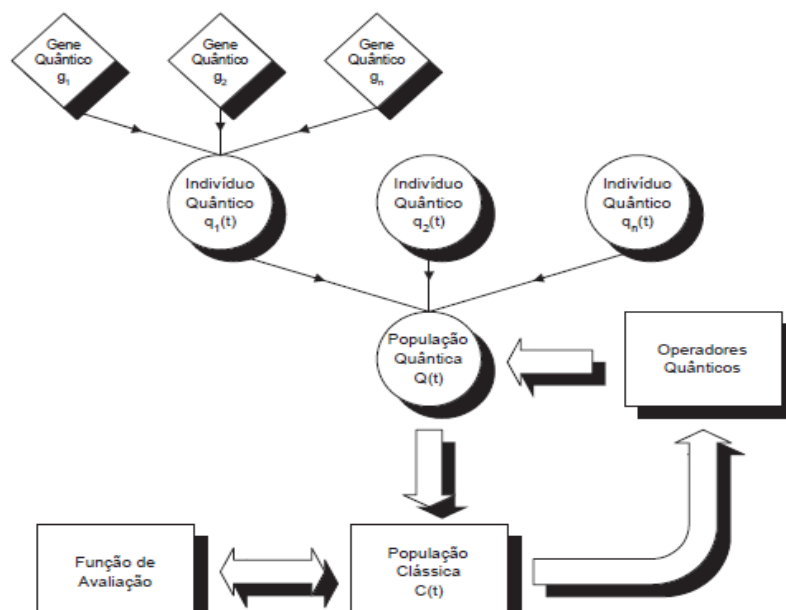


Figura 2.12: Diagrama do sistema evolutivo com Inspiração Quântica. Fonte (Cruz07).

Algoritmo Evolutivo com Inspiração Quântica com Representação Binária-Real – AEIQ-BR:

Muitos problemas do mundo real podem ser representados através de decisões do tipo combinatória e numérica simultaneamente. O AEIQ-BR se apresenta como um algoritmo com representação mista, numérica e combinatória, sendo esse um diferencial para otimização, herdando as principais características como:

- Capacidade de otimização global de um problema;
- Representação probabilística do espaço de busca representando alta diversidade populacional em cada indivíduo quântico;
- Capacidade de exploração com poucos indivíduos na população, aproveitando o espaço de soluções.

A figura 2.13 ilustra os passos do pseudocódigo do algoritmo AEIQ-BR. O desenvolvimento do AEIQ-BR e os resultados de aplicação deste algoritmo são apresentados em Pinho (Pinho09).

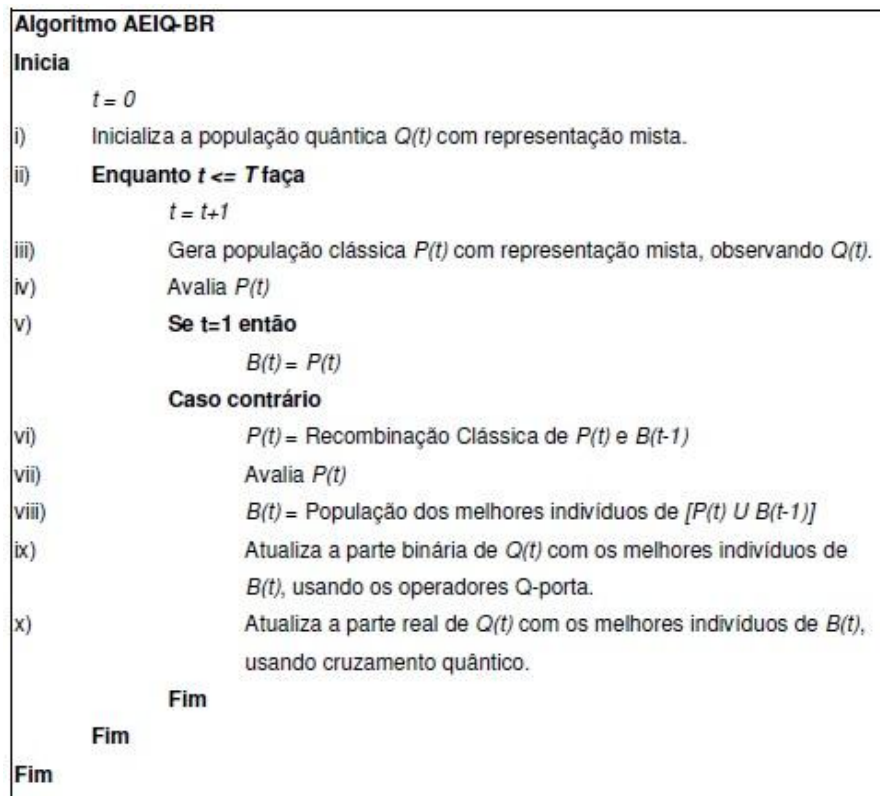


Figura 2.13: Pseudocódigo do algoritmo AEIQ-BR. Fonte (Pinho09).

Os parâmetros que configuram o algoritmo AEIQ-BR estão disponibilizados na tabela 2.2:

QbrSNN	Descrição dos parâmetros
NQ	Número de indivíduos quânticos
NC	Número de indivíduos clássicos
T	Número de gerações ou evoluções
Ccb	Probabilidade de crossover clássico binário
Ccr	Probabilidade de crossover clássico real
$\Delta\theta$	Ângulo de rotação
Cq	Taxa de crossover quântico
$updatesT$	Intervalo entre geração até que os operadores sejam executados

Tabela 2.2: Parâmetros do AEIQ-BR (Pinho09).

A seguir será apresentado o algoritmo de otimização por Programação Genética utilizada nesta tese.

2.2.2 Otimização por Programação Genética – OGP

Otimização por Programação Genética explora conceitos da Programação Genética (PG), que se apresenta como uma técnica da computação evolutiva

inspirada nos conceitos de seleção natural e recombinação genética. Esta permite sintetizar programas de computador que solucionem problemas sem a necessidade de serem programados para tal, onde a representação é feita por um complexo de estruturas em árvores, com base em princípios de programação genética (Koza92; Poli08; Langdon10).

O conceito geral da PG é uma meta-heurística baseada no conceito de população de soluções. O espaço de busca é formado a partir das possíveis combinações entre os Terminais e as Funções ou Operações Matemáticas descritas a seguir (Koshiyama13):

- **Terminais:** variáveis de entrada e constantes.
- **Funções ou Operações Matemáticas:** unárias, binárias, ternárias, além de outras definidas pelo usuário.

Para iniciar uma rotina de PG clássica, o usuário deve definir os parâmetros citados anteriormente e um conjunto de parâmetros básicos de acordo com a tabela 2.3:

Tamanho da População
Método de Seleção
Taxa de Mutação
Número de Gerações
Taxa de Elitismo
Taxa de Cruzamento
Altura da árvore
Taxa de Reprodução Direta
Taxa de Edição
Método de Geração da População Inicial
Taxa de Destruição
Taxa de Encapsulamento

Tabela 2.3: Parâmetros de controle de um algoritmo PG.

Na tabela 2.2 pode-se observar que as 6 primeiras linhas apresentam parâmetros também presentes nos Algoritmos Genéticos (AG) clássicos, sendo os últimos parâmetros específicos para o método PG.

A figura 2.14 ilustra o processo de execução da PG.

Algorithm 1 Programação Genética

1: Passos Iniciais:

- Definição dos Terminais e Operações Matemáticas
- Configurar Demais Parâmetros
- Estabelecer a Função de Avaliação

2: Inicialização da População

3: Avaliação

4: **if** Critério de Parada foi Atingido **then** Retornar a População

5: **else**

6: **while** Critério de Parada não for Atingido **do**

7: Seleção

8: Recombinação

9: Avaliação

10: **end while**

11: **end if**

12: Retornar a Última População

Figura 2.14: Pseudocódigo do algoritmo OGP. Fonte (Koshiyama13).

Primeiramente gera-se uma população inicial aleatória, tal que cada indivíduo seja formado por Terminais e Operações Matemáticas escolhidos de forma aleatória. Estes são avaliados e os de maior aptidão são selecionados por alguma heurística para a aplicação dos operadores de recombinação. Portanto, os novos indivíduos passam pelo critério de parada e, caso o critério não seja atendido, o número de gerações é incrementado e as etapas são repetidas.

Representação e Forma da Solução:

Na PG cada indivíduo é denotado por uma expressão matemática codificada através de uma estrutura em árvore (figura 2.15).

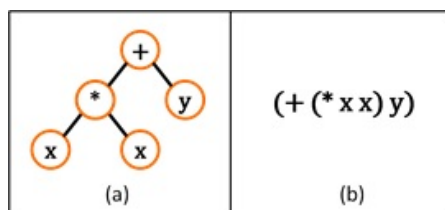


Figura 2.15: Representação de: $x^2 + y$ (a) Representação sob a forma de árvore; (b) forma pré-fixada. Fonte (Koshiyama13).

Conforme ilustrado na figura 2.15, as operações matemáticas aparecem nos vértices internos da árvore, e os Terminais nos vértices externos (folhas) da árvore.

Avaliação e Seleção:

Da mesma forma que outros modelos evolucionários, o processo de avaliação avalia a qualidade da solução decodificada em cada indivíduo, sendo necessário que a Função de Avaliação estabeleça uma relação direta com o problema específico (Koshiyama13).

Recombinação:

Nesta etapa os indivíduos selecionados para a recombinação sofrem a ação dos operadores, sendo estes aplicados uma única vez nos indivíduos. Os operadores de recombinação mais usados são apresentados a seguir (Koshiyama13):

- **Reprodução direta:** o indivíduo selecionado é copiado sem qualquer alteração para a próxima geração (figura 2.16) (Koshiyama13).

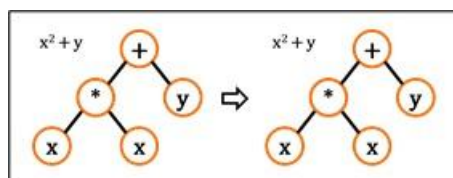


Figura 2.16: Operador de Reprodução Direta. Fonte (Koshiyama13).

- **Cruzamento:** dois indivíduos são selecionados e um ponto de corte é escolhido aleatoriamente em cada um dos genitores. As sub-árvores resultantes são permutadas formando novos programas para a próxima geração (figura 2.17) (Koshiyama13).

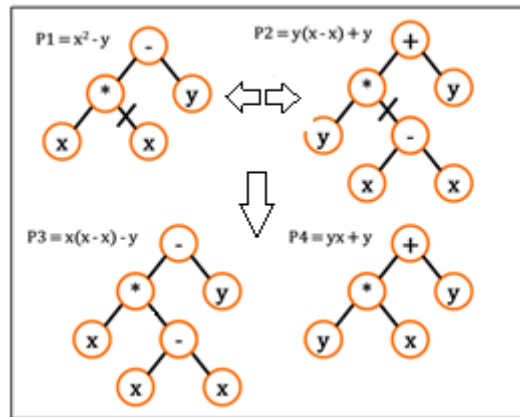


Figura 2.17: Operador de Cruzamento. Fonte (Koshiyama13).

- **Mutação:** seleciona um ponto interno ou externo, aleatoriamente, da árvore de um programa. Remove-se esta sub-árvore inserindo uma nova aleatória (figura 2.18) (Koshiyama13).

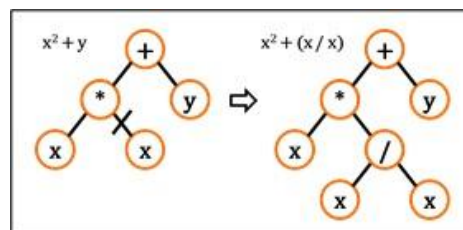


Figura 2.18: Operador de Mutação. Fonte (Koshiyama13).

Programação Genética Multigênica:

A Programação Genética Multigênica (PGMG) é uma generalização da PG tradicional, pois denota um indivíduo como um complexo de estrutura em árvore, também chamados de genes que, da mesma forma que a PG, recebe um conjunto de Terminais X_{ij} , buscando uma saída Y_i (Hinchliffe96; Searson07).

A representação da PGMG é semelhante ao de uma PG, no que diz respeito à estrutura em árvore, mas um indivíduo é representado por um complexo de estruturas em árvore conforme figura 2.19. Cada árvore desta estrutura é uma solução parcial para o problema, sendo que a saída final pode ser um resultado da combinação linear.

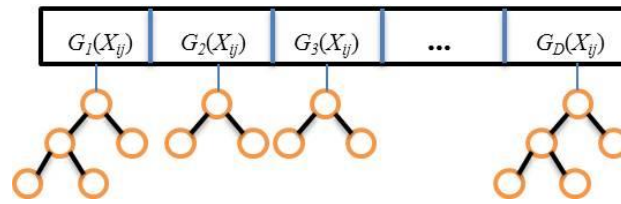


Figura 2.19: Exemplo de um indivíduo multigênico. Fonte (Koshiyama13).

O processo de avaliação e seleção é efetuado de forma similar à PG clássica. Em relação aos operadores de recombinação, a operação de Mutação na PGMG é similar à efetuada na PG clássica, e a operação de Cruzamento necessita de uma distinção no nível de operação a ser realizada, sendo possível aplicar o cruzamento nos níveis alto e baixo.

O baixo nível é o espaço onde é possível manipular as estruturas (Terminais e Operações Matemáticas) das equações presentes em um indivíduo. Na PGMG, tanto a mutação como o cruzamento de baixo nível são semelhantes ao modo efetuado na PG clássica.

A figura 2.20 apresenta um indivíduo multigênico com cinco genes, efetuando uma operação de cruzamento de baixo nível, e a figura 2.21 a operação de mutação.

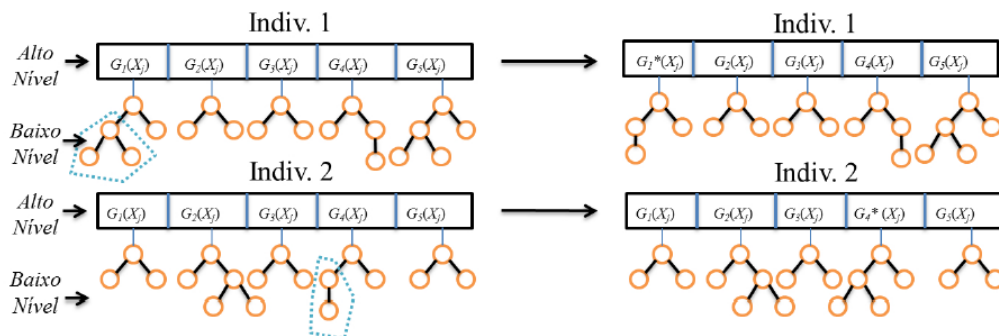


Figura 2.20: Operação de Cruzamento de baixo nível. Fonte (Koshiyama13).

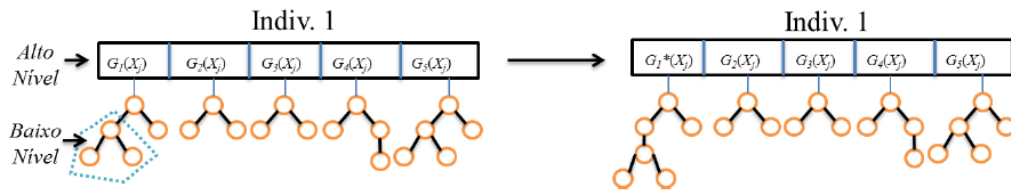


Figura 2.21: Operação de Mutação. Fonte (Koshiyama13).

A figura 2.22 ilustra um cruzamento de alto nível para um indivíduo de 5 genes. O alto nível é o espaço em que se manipulam de forma macro as equações presentes no indivíduo. Assim, a partir de dois pontos aleatórios, são permutadas equações de um indivíduo para outro, afetando substancialmente a saída resultante.

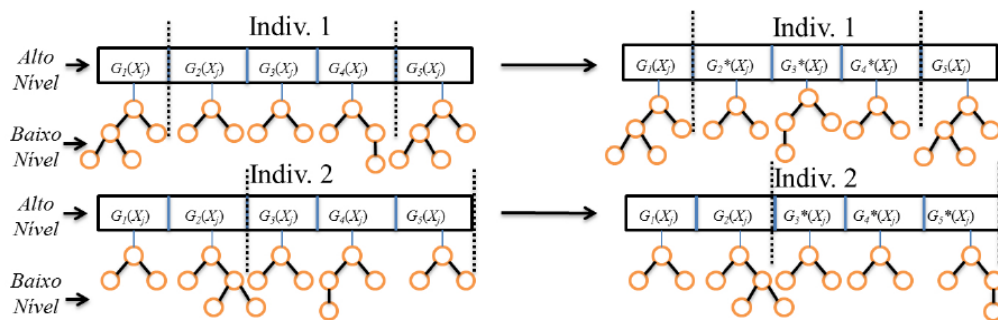


Figura 2.22: Operação de cruzamento de alto nível. Fonte (Koshiyama13).

A reprodução direta, onde o indivíduo selecionado é copiado sem qualquer alteração para a próxima geração, é idêntica ao da PG clássica.

Em linhas gerais, o procedimento evolutivo da PGMG se diferencia da PG clássica pela adição de dois parâmetros: número máximo de árvores por indivíduo e taxa de cruzamento de alto nível. No número máximo de árvores por indivíduo, sempre se utiliza um valor elevado para que não haja problemas no processo de síntese da solução. Já a taxa de cruzamento de alto nível é um parâmetro que deve ser ajustado nas configurações do algoritmo durante as execuções.

As funções aparecem nos vértices internos da árvore e podem ser: operações aritméticas (soma, subtração, multiplicação e divisão), funções matemáticas (seno, cosseno e tangente), operações booleanas (*and*, *or* ou *not*),

operadores condicionais (*if* ou *else*), funções de iteração (*for* ou *while*), até específicas do problema (Koshiyama13).

Próximo capítulo apresentará a proposição de dois modelos neuro-evolucionários para redes *spiking* baseados nestes modelos evolucionários descritos nesse capítulo, AEIQ-BR e o OGP.

3

Modelos Neuro-Evolucionários para *Spiking Neural Networks* não supervisionado

Modelos neuro-evolucionários exploram o potencial de duas diferentes áreas: Redes Neurais Artificiais (RNA) e Algoritmos Evolucionários (AE). Esses modelos buscam a melhor configuração para uma rede neural utilizando algoritmos evolutivos, de forma a melhorar o processo de treinamento e configuração da rede neural.

O interesse principal deste trabalho é desenvolver novos modelos híbridos para problemas de agrupamento em que os atributos de entrada e os parâmetros de uma rede neural *spiking* (SNN) são otimizados utilizando algoritmos evolutivos. Foram utilizadas duas abordagens evolucionárias, o algoritmo evolutivo com inspiração quântica com representação binária-real (AEIQ-BR) (Pinho09) e a otimização por programação genética (OGP) (Koshiyama13).

Por serem mais promissores devido à capacidade de encontrar uma boa solução com um menor número de avaliações, e utilizando uma representação com uma estrutura em árvore, com base em princípios de programação genética. Essas abordagens foram desenvolvidas de forma a avaliar o seu desempenho na otimização dos parâmetros das redes neurais *spiking*, e verificar qual o mais adequado para o problema de agrupamento.

Na construção de uma SNN, uma combinação correta dos parâmetros de configuração da rede determina um bom desempenho da rede. Seleção de atributos é também uma importante ferramenta de pré-processamento para remover informações que não contribuem para o aprendizado não supervisionado. Um maior número de atributos não necessariamente se traduz em maior precisão de classificação ou agrupamento. Em alguns casos, um valor menor de atributos ajuda a reduzir o tempo de processamento e a produzir melhores resultados, por considerar apenas os atributos relevantes para o problema.

Este capítulo apresenta os modelos híbridos propostos para ajustar automaticamente as SNNs para problemas de agrupamento. Os modelos neuro-evolutivos propostos foram denominados: *Quantum-Inspired Evolution of Spiking*

Neural Networks with Binary-Real representation (QbrSNN) e Spiking Neural Network Optimization by Genetic Programming (SNN-OGP).

3.1

Modelagem da SNN

O modelo de SNN utilizado neste estudo é baseado no modelo proposto por Hopfield (Hopfield95), e ampliado por Nat (Nat98). Este modelo codifica os padrões de entrada em pulsos defasados no tempo, utilizando múltiplas sinapses e neurônios de saída modelados como SRM (*Spike Response Model*), operando como neurônios RBF (*Radial Basis Function*).

A codificação de informação utilizada foi com base na codificação proposta por (Bohte03), de tal forma que os atributos são codificados em um grande número de disparos defasados no tempo, utilizando funções gaussianas sobrepostas defasadas.

A arquitetura proposta é semelhante a uma rede neural tradicional, com os neurônios da primeira camada totalmente ligados aos neurônios da camada seguinte. A primeira camada é composta pelos neurônios codificados em função do número de atributos de entrada e o número de gaussianas utilizadas, a segunda camada é constituída por neurônios SRM operando como neurônios RBF (*Radial Basic Function*) (Bohte03).

Cada vetor de entrada $\vec{X} = (x_1, x_2, \dots, x_n)$ é codificado em um trem de pulsos formando os neurônios da primeira camada. O número de neurônios na camada de entrada será composto pelas quantidades de atributos de cada vetor e a quantidade de gaussianas utilizadas na codificação.

Os neurônios de saída representam a formação dos grupos compostos por $\vec{Y} = (y_1, y_2, \dots, y_n)$. Os neurônios da primeira camada estão ligados à camada de saída através de múltiplas sinapses associadas com diferentes atrasos e pesos. A figura 3.1 apresenta um exemplo da estrutura da SNN utilizada nesta tese.

O algoritmo de aprendizado não supervisionado implementa o método Hebbiano de aprendizado através do algoritmo do neurônio vencedor – *winner-takes-all*. O processo do algoritmo de aprendizado consiste em adaptar os atrasos axonais das múltiplas sinapses, $\vec{\Delta} = (\delta_1, \delta_2, \dots, \delta_n)$, de modo que, cada neurônio de saída dispare à medida que o vetor de entrada $\vec{X} = (x_1, x_2, \dots, x_n)$ se aproxime do vetor que representa o centro do grupo.

O centro do neurônio RBF de saída é dado por um vetor $\vec{c}_j = (c_{j1}, c_{j2}, \dots, c_{jn})$, e a rede funciona de forma que cada neurônio de saída

dispare somente uma vez. Sinapses inibitórias interligam os neurônios da camada de saída para desativar após o primeiro disparo. O objetivo dessas sinapses inibitórias é desativar todos os neurônios após o primeiro disparo, ou seja, quando um neurônio dispara os outros são desativados.

Quando o vetor de entrada, codificado no tempo, se aproxima do vetor C_j o neurônio de saída dispara, significando que o valor do limiar de disparo do neurônio de saída foi alcançado (Nat98). A distância entre os vetores de entrada e saída é calculada pela distância euclidiana ($\|\vec{X} - \vec{C}_j\|$) (Hopfield95).

A figura 3.1 apresenta um exemplo de estrutura da SNN utilizada nesta tese.

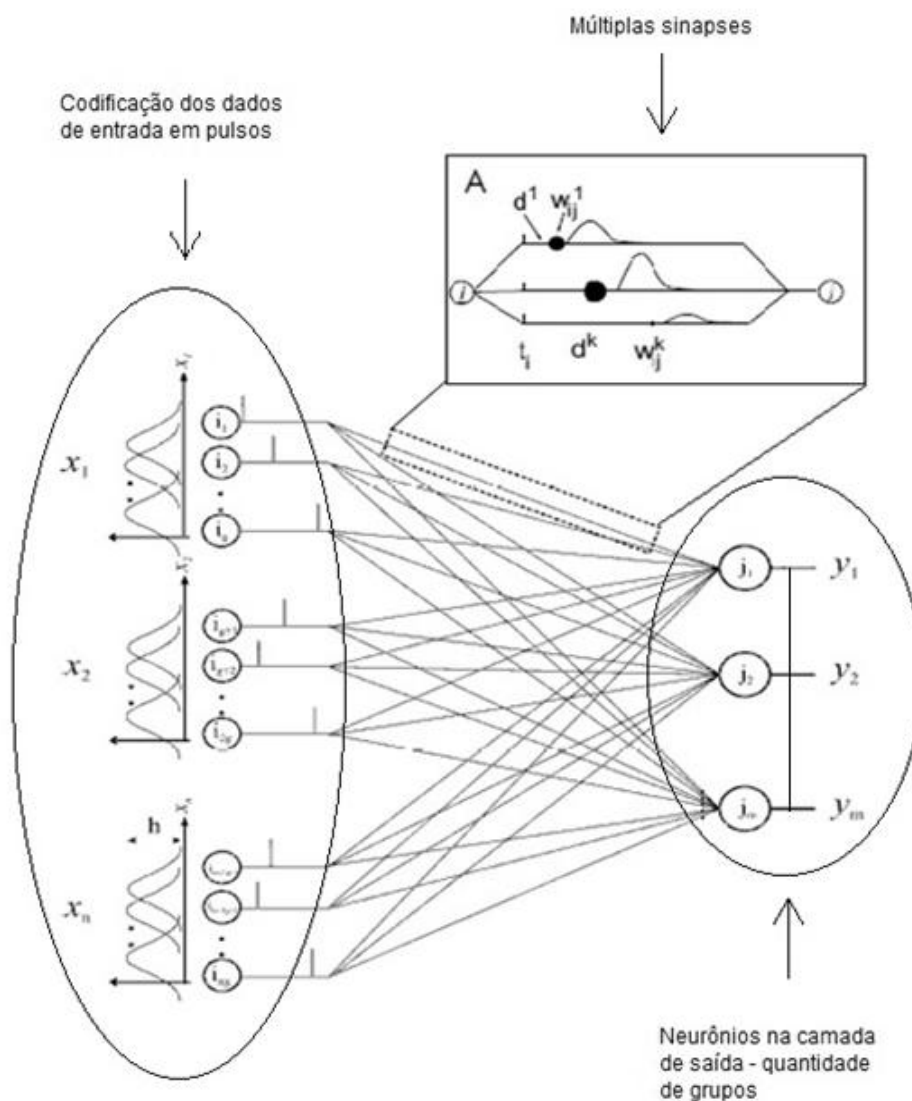


Figura 3.1: Estrutura da SNN utilizada nesta tese SNN.

Figura 3.2: Adaptação dos pesos sinápticos através do algoritmo de aprendizado. (a) Inicialização dos pesos com o mesmo valor. (b) Pesos foram atualizados e o próximo pulso é selecionado Fonte (Alexandre06).

Os pesos das sinapses são inicializados com valores iguais a 1. O pulso mais próximo do neurônio j é selecionado, e o algoritmo de aprendizado identifica o pulso que antecede o pulso selecionado. O peso sináptico do pulso selecionado é incrementado e os demais são decrementados. O processo se

repete até que o tempo de disparo de neurônio de saída esteja próximo ao tempo do pulso do neurônio de entrada. Logo se pode afirmar que esse neurônio de entrada contribuiu para que o neurônio de saída disparasse, sendo ele eleito o neurônio vencedor.

A seguir serão apresentados os modelos híbridos propostos para o problema de agrupamento, onde os atributos de entrada e os parâmetros que configuram a SNN serão otimizados por algoritmos evolutivos.

3.2

Primeiro Modelo: *Quantum-Inspired Evolution of Spiking Neural Networks with Binary-Real representation* (QbrSNN)

Conforme discutido nos capítulos anteriores, muitos problemas podem ser representados através de decisões como: quais variáveis utilizar para uma modelagem; ou como obter os parâmetros adequados para o bom desempenho dos modelos ou algoritmos a serem utilizados. Algoritmos Evolutivos (AE) tem sido uma ótima ferramenta na evolução das RNAs, como por exemplo, na otimização dos pesos sinápticos que, quando bem ajustados, evitam problemas de convergência prematura a um mínimo local e supertreinamento das redes. Nos AE não existe uma regra de aprendizado da rede neural, mas sim um processo evolutivo de adaptação ao problema (Cruz07). Além de obter pesos para o funcionamento da rede, outros fatores podem ser interessantes na utilização de AE, como: seleção dos atributos mais relevantes para o treinamento da rede e a determinação da topologia neural ótima.

A escolha do modelo AEIQ-BR na implementação do primeiro modelo híbrido neuro-evolutivo de redes neurais *spiking* para aprendizado não supervisionado, deve-se à possibilidade de utilizar representação mista, numérica e binária, no cromossoma do algoritmo evolutivo. As principais vantagens do QbrSNN são: capacidade de otimização global da rede neural *spiking*; representação probabilística do espaço de busca com alta diversidade populacional em cada indivíduo; poucos indivíduos populacionais com grande capacidade de exploração e aproveitamento do espaço de soluções, diminuindo o tempo computacional; e a seleção dos atributos de entrada relevantes para serem utilizadas com entrada para a rede neural.

O modelo QbrSNN representa a ligação entre o AEIQ-BR (Pinho09) e uma SNN para aprendizado não supervisionado. Os parâmetros que configuram a SNN e o número de neurônios na saída (quantidade de grupos formados)

definem a parte real do cromossomo com 10 genes (nG , Δ , η , τ , $tmax$, b , c , ϑ , β e $nout$). A tabela 3.1 ilustra a descrição e os intervalos utilizados para os parâmetros da parte real do cromossomo:

Intervalos dos parâmetros da parte real do cromossomo		
nG	Número de Gaussianas	8 - 20
Δ	<i>Delay</i>	8 - 20
η	Taxa de Aprendizado	0,1 - 1
τ	Tempo da curva do potencial da membrana	3 - 7
$tmax$	Tempo do cálculo do neurônio vencedor	3 - 7
b	Valor de incremento e decremento dos pesos	0,1 - 1
c	Valor do pico da curva de aprendizado	3 - 7
ϑ	Valor de <i>threshold</i> (limiar)	5 - 15
β	Valor de largura da curva de aprendizado (vizinhança)	1 - 4
$nout$	Número de neurônios na saída	2 - 10

Tabela 3.1: Valores e intervalos dos parâmetros da parte real do cromossomo.

A parte binária é formada por tantos genes quantos forem os atributos de entrada de cada base de dados a ser agrupada. Por exemplo, no caso de uma base de dados com 5 atributos de entrada, o número de genes da parte binária é igual a 5, onde o valor 1 no gene j indica que a j -ésima atributo deve ser considerado no modelo e o valor 0 caso contrário.

A figura 3.3 ilustra o cromossomo completo genérico do modelo QbrSNN.

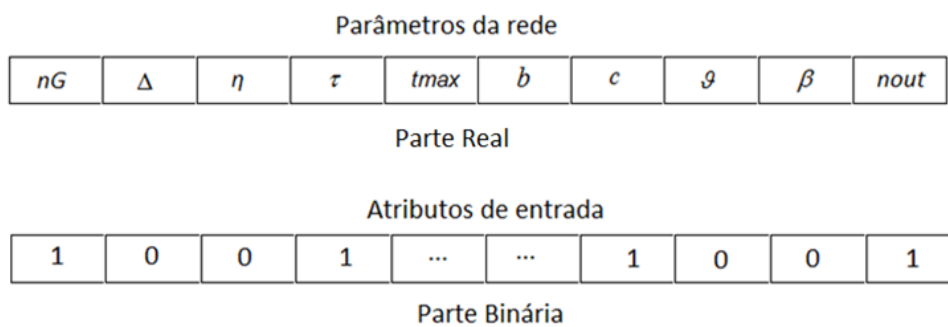


Figura 3.3: Cromossomo usado para otimizar, simultaneamente, os parâmetros e os atributos de entrada de uma SNN.

O QbrSNN inicializa aleatoriamente a população quântica de acordo com as estratégias apresentadas no capítulo anterior para o AEIQ-R e o AEIQ-B, isto é: os genes da parte binária são inicializados com valores $\alpha = \beta = 1/\sqrt{2}$, de

forma que os bits 0 e 1 tenham igual probabilidade de serem observados; e os genes da parte real são inicializados com funções densidade de probabilidade uniforme na faixa de valores possíveis de cada parâmetro. A tabela 3.3 ilustra os intervalos utilizados para os parâmetros da parte real do cromossomo:

A figura 3.4 mostra um exemplo das funções densidade de probabilidade ($p_{ij}(x)$) dentro dos intervalos escolhidos para os parâmetros, utilizando valor do centro (μ) e a largura do pulso (σ) para cada intervalo utilizado:

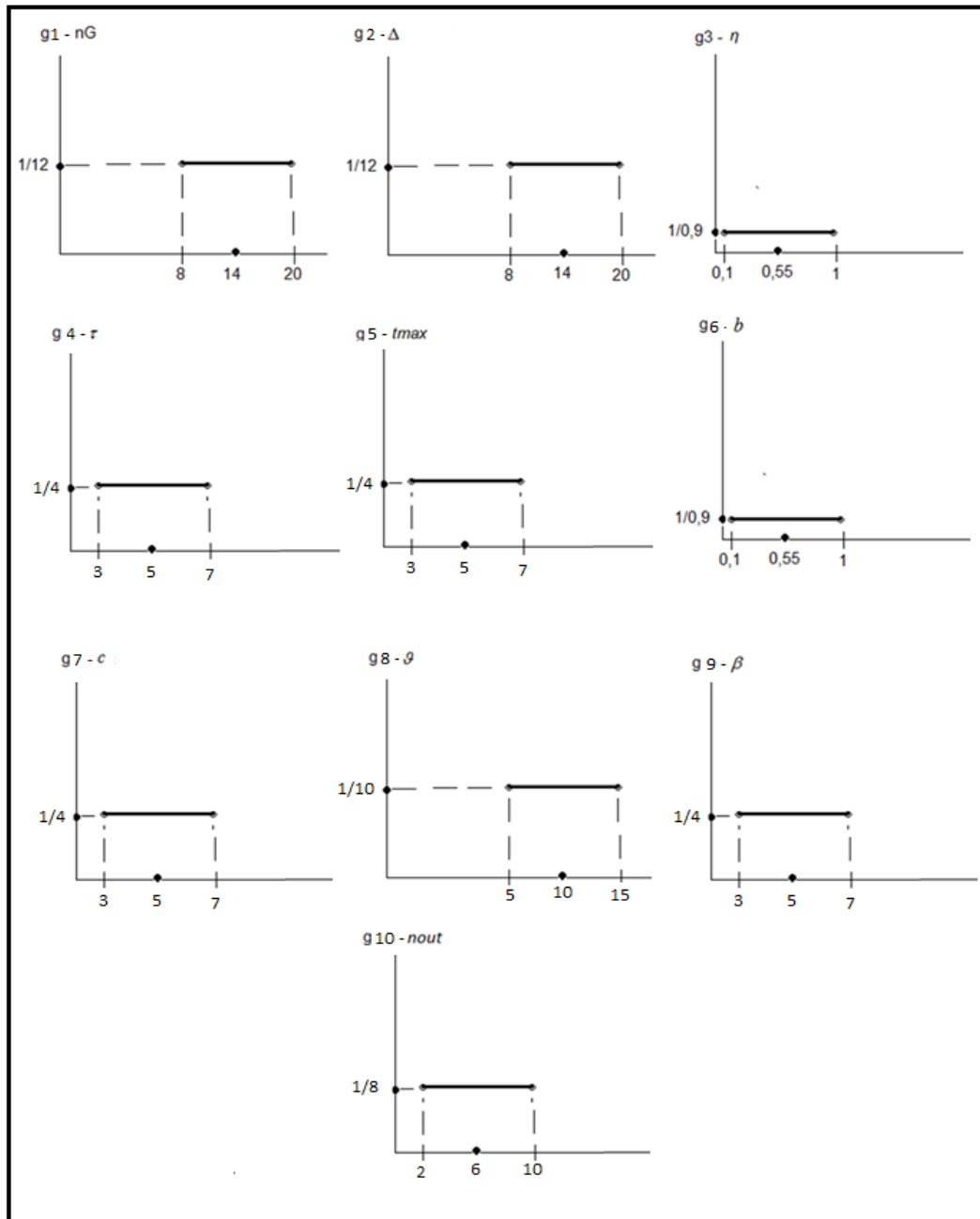


Figura 3.4: Representação gráfica dos indivíduos quânticos.

Na equação 18 é possível visualizar um indivíduo capaz de representar uma superposição de todas as possíveis soluções do domínio, no primeiro passo

do algoritmo QbrSNN. Para a parte do indivíduo com representação real têm-se os genes formados pelos valores: do centro (μ) e a largura do pulso (σ) para cada intervalo, e $1/\sqrt{2}$ para que os bits 0 e 1 tenham igual probabilidade de serem observados:

$$Q(t) = \left[\begin{array}{c} [14; 14; 0,55; 5; 5; 0,55; 5; 10; 5; 6] \\ [12; 12; 0,9; 4; 4; 0,9; 4; 10; 4; 8] \end{array} \right]_{real} \left[\begin{array}{c} [1/\sqrt{2}; \dots; 1/\sqrt{2}] \\ [1/\sqrt{2}; \dots; 1/\sqrt{2}] \end{array} \right]_{binário} \quad (18)$$

A partir do segundo passo do algoritmo iniciará a evolução e assim se manterá até que atinja o número máximo de gerações.

Em seguida, os indivíduos quânticos são observados (cada indivíduo quântico gera n indivíduos clássicos, $n \in N$) e cada indivíduo clássico configura uma SNN, a qual é treinada, com base nessa configuração, para realizar o agrupamento dos dados. Após a realização do treinamento, o agrupamento resultante é então avaliado, de acordo com a função de avaliação escolhida.

A função de avaliação desempenha um papel crucial para a validação do algoritmo. Com base na função de aptidão definida, cada indivíduo clássico é avaliado e essa avaliação é utilizada para selecionar os indivíduos clássicos que serão utilizados para atualizar os indivíduos quânticos.

Para que o algoritmo tenha alguma vantagem evolutiva, é recomendável que os parâmetros quânticos sejam ajustados de tal forma que estes passem a representar em um passo posterior, ou seja, outra geração, espaços de possíveis soluções mais promissoras para a solução de um problema. A atualização dos genes quânticos binários é realizada pelo operador q -gate, o qual é definido como a matriz de rotação dada pela equação 19:

$$\begin{bmatrix} \alpha' \\ \beta' \end{bmatrix} = \begin{bmatrix} \cos \theta & (+ \text{ ou } -) \sin \theta \\ (+ \text{ ou } -) \sin \theta & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad (19)$$

onde $i = 1, 2, 3, \dots, j$ e j é o comprimento do indivíduo quântico, e valores de θ (ângulo de rotação) são definidos pelo usuário. Os valores de θ devem ser capaz de modificar os valores de α e β de modo que aumente as chances de que os indivíduos, com melhores avaliações, sejam observados (Pinho09).

Da mesma forma, é necessária a atualização dos genes quânticos reais, isto é, modificar a largura da curva alterando os limites (inferior e superior) de modo a reduzir gradativamente o espaço de busca.

Para atualizar o intervalo da curva foi utilizado o decaimento exponencial com velocidade de atualização, dada pela amplitude de todos os indivíduos na população $B(t)$ (população dos melhores indivíduos clássicos), conforme mostrado na equação 20:

$$\sigma_{ij}^{t+1} = \sigma_{ij}^t + \lambda^t \cdot \left(\left(\max \left((x_{ij}^t)_r \right) - \min \left((x_{ij}^t)_r \right) \right) - \sigma_{ij}^t \right) \quad (20)$$

onde, σ_{ij}^t é a largura do j -ésimo gene do i -ésimo indivíduo quântico em $Q(t)$; $(x_{ij}^t)_r$ representa uma solução numa região mais promissora; λ^t é um número aleatório entre $[0,1]$ determinando a velocidade de atualização (Pinho09).

Para a atualização do centro foi utilizado uma combinação convexa dada pela equação 21:

$$\mu_{ij}^{t+1} = \mu_{ij}^t + \lambda^t (x_{ij}^t - \mu_{ij}^t) \quad (21)$$

onde, μ_{ij}^t é o centro do gene quântico na geração t ; x_{ij}^t é o valor do gene clássico; λ^t indica o percentual que se quer deslocar o centro do gene quântico na direção do gene clássico.

A função de avaliação desempenha um papel crucial para uma aplicação bem sucedida dos algoritmos. Vários métodos que avaliam a qualidade de agrupamentos podem ser aplicados, por exemplo, o método de Ward (Everitt01) e Silhueta (Si) (Kaufman90). No processo de validação do modelo foi utilizado o método de Silhueta (Si), e este foi maximizado no processo de evolução do modelo. O método de Si combina os métodos de coesão e separação dos padrões nos grupos formados, utilizando na equação 22.

$$Si = \frac{b(k) - a(k)}{\max \{a(k), b(k)\}} \quad (22)$$

onde, para um padrão de entrada (k) , calcula-se a similaridade média deste padrão entre todos os padrões do grupo que ele pertence $a(k)$, calcula-se então a similaridade média de todos os padrões dos outros grupos para encontrar o menor valor $b(k)$.

O valor do coeficiente de Si varia entre $[-1, 1]$, sendo o valor negativo não desejado, pois, corresponde ao caso em que $a(k)$, a similaridade média para os

padrões no grupo, é maior que $b(k)$. O melhor é quando o coeficiente de Si seja positivo ($a(k) < b(k)$) e que $a(k)$ seja o mais próximo de 0 possível, desde que, o coeficiente assumira valor 1 quando $a(k)=0$.

- $(S_i) \approx 1$ – o objeto é bem classificado no agrupamento.
- $(S_i) \approx 0$ – o objeto está entre os agrupamentos.
- $(S_i) \approx -1$ – o objeto é mal classificado no agrupamento.

Um exemplo de gráfico da silhueta na formação dos grupos para a base Iris é ilustrado na figura 3.5, e pode ser interpretado conforme tabela 3.2.

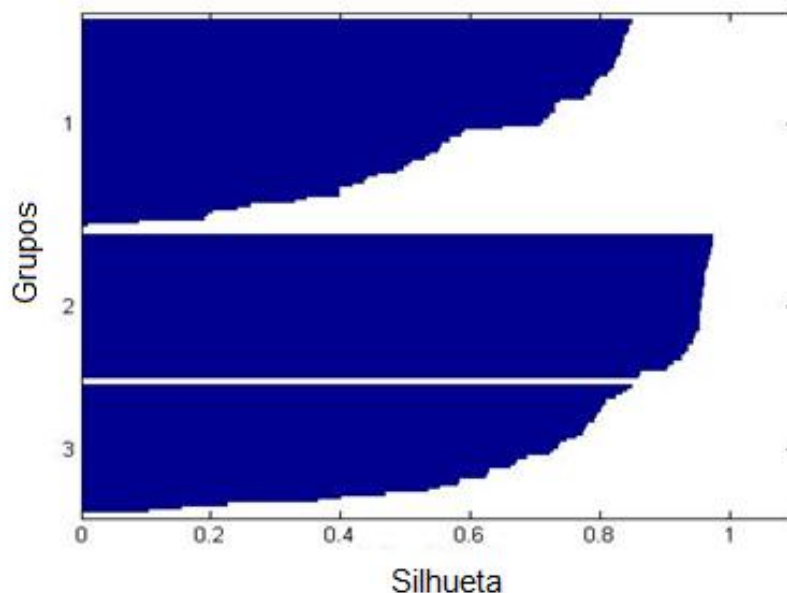


Figura 3.5: Exemplo da Silhueta (S_i) para a base Iris .

$S(i)$	Descrição
0,71 - 1,00	Uma estrutura forte foi encontrada
0,51 - 0,70	Uma estrutura razoável foi encontrada
0,26 - 0,50	A estrutura é fraca e pode ser superficial
$\leq 0,25$	Nenhuma estrutura foi encontrada

Tabela 3.2: Valores da Silhueta (S_i).

A figura 3.6 mostra a listagem completa do algoritmo evolutivo com inspiração quântica com representação real para uma SNN:

Algoritmo QbrSNN**Inicia** $t = 0$ Inicializa a população quântica $Q(t)$ com N indivíduos com representação mista.**Enquanto $t \leq T$ faça**Gera população clássica $P(t)$ com representação mista, observando $Q(t)$.**Observação do q-bits**

Gerar um número aleatório entre 0 e 1.

Se este número estiver entre 0 e α^2 então

Atribui-se 0 à observação.

Caso contrário

Atribui-se 1.

Observação do q-real (quântico real)

Gerar um número aleatório entre 0 e 1.

Atribui-se a função cumulativa de probabilidade o valor deste número aleatório.

Obtém-se o gene clássico real através da inversa da função cumulativa de probabilidade.

Configura a rede neural *spiking* com os parâmetros e os atributos de entrada selecionados.

Avalia a SNN (indivíduo) – utiliza a função de avaliação Silhueta.

Se $t = 1$ então $B(t) = P(t)$ **Caso contrário** $P(t) = \text{Recombinação Clássica de } P(t) \text{ e } B(t-1)$ Avalia $P(t)$ $B(t) = \text{População dos melhores indivíduos de } [P(t) \cup B(t-1)]$ Atualiza a parte binária de $Q(t)$ com os melhores indivíduos de $B(t)$, usando os operadores *q-gate*.Atualiza a parte real de $Q(t)$ com os melhores indivíduos de $B(t)$, usando cruzamento quântico.**Fim****Fim**

Retorna a melhor solução: parâmetros e a seleção de atributos

Fim**Figura 3.6: Pseudocódigo do QbrSNN.**

Este processo é repetido até que um critério de parada seja atendido. A figura 3.7 apresenta a arquitetura de uma QbrSNN.

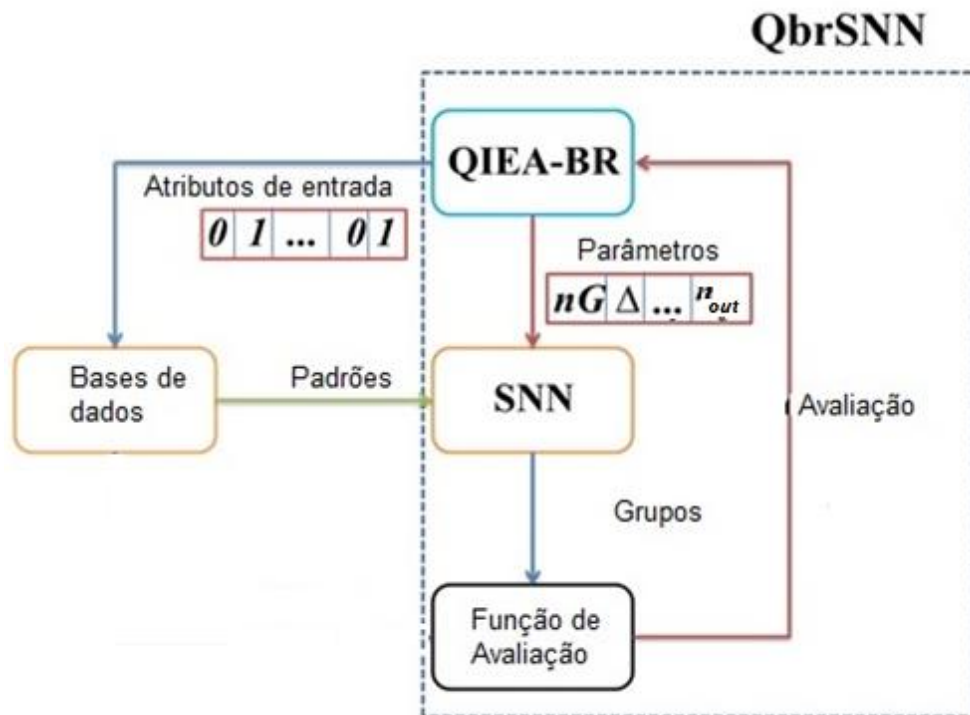


Figura 3.7: Arquitetura do QbrSNN.

3.3

Segundo Modelo: *Spiking Neural Network Optimization by Genetic Programming (SNN-OGP)*.

Outro modelo com características adequadas para a neuro-evolução de uma rede *spiking* para aprendizado não supervisionado foi desenvolvido, e a importância de sua utilização se deve à possibilidade de um indivíduo ser representado como um complexo de estruturas em árvores, buscando soluções satisfatórias em um grande espaço de busca com uma representação mista, avaliando a capacidade dessa nova meta-heurística em propor soluções no âmbito de sistemas neuro-evolutivos.

A principal vantagem é a possibilidade de buscar soluções, em um espaço grande de busca, das funções proporcionando algumas facilidades em termos de varredura, buscando os melhores indivíduos para resolverem um determinado problema. A facilidade na varredura do espaço de busca se deve as mudanças nas operações matemáticas, que proporcionam novas soluções diferentes das anteriores.

A figura 3.8 descreve o indivíduo baseado em árvore completa.

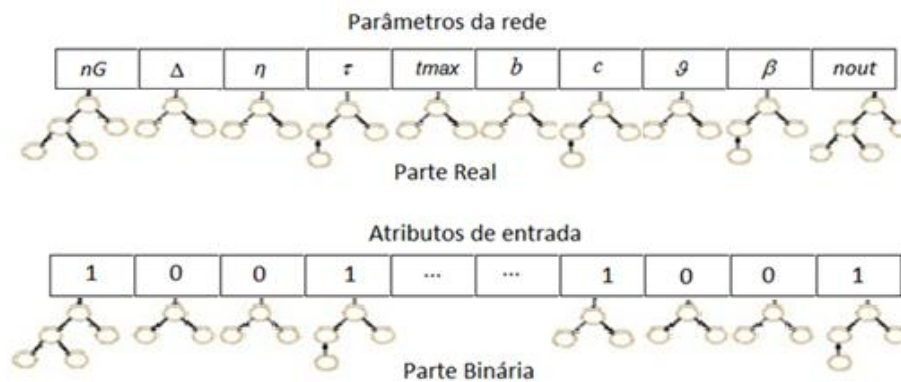


Figura 3.8: Cromossomo com base em árvore para otimizar simultaneamente parâmetros e atributos de entrada de uma SNN.

Cada valor do parâmetro, real ou binário, é uma função representada em formato de árvore. Para a parte binária, o valor será 1 se a função estiver ativa, ou seja, esta será incluída no processo de agrupamento, e 0 caso contrário.

O algoritmo de otimização OGP tem início a partir das especificações dos parâmetros populacionais (tamanho da população, número de gerações, etc...), e dos operadores genéticos (taxa de cruzamento, mutação, etc...). Na sequência, é estabelecido pelo usuário o número de funções incorporadas em um indivíduo.

A partir dessas especificações, uma população de indivíduos é gerada aleatoriamente utilizando o método *Ramped Half-Half*. Esse método é uma mescla dos métodos *Full* e *Grow*, onde, para a geração de árvores é estabelecida uma probabilidade verificando qual método será utilizado. No método *Full*, todas as árvores têm o mesmo tamanho especificado no algoritmo e no método *Grow* as árvores têm tamanho variado.

As soluções expressas por cada indivíduo são avaliadas na função objetivo do problema de otimização explorado. Após toda a população ter sido avaliada, verifica-se se o critério de parada foi atendido; se sim, retorna a população; caso contrário, o algoritmo só será interrompido caso o critério de parada seja alcançado.

Os passos do algoritmo para uma SNN é ilustrado na figura 3.9, e detalhado a seguir:

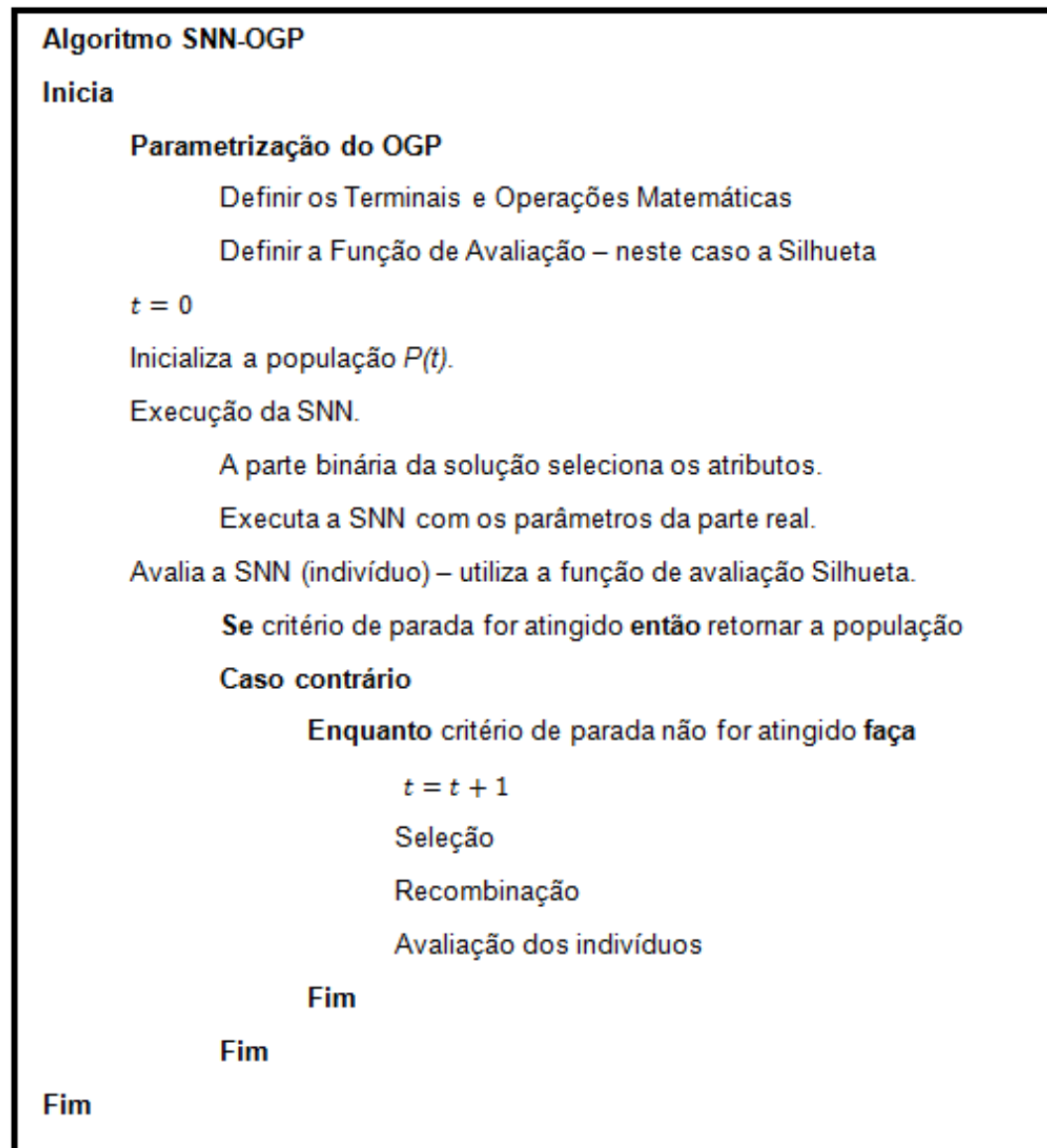


Figura 3.9: Pseudocódigo do SNN-OGP.

No processo de validação do modelo foi utilizado o método de Silhueta (Si), e este foi maximizado no processo de evolução do modelo.

Durante a execução do algoritmo, são efetuadas três operações: seleção, recombinação (aplicação dos operadores) e avaliação. A primeira consiste em escolher, a partir de algum método heurístico (roleta, torneio, etc...) a próxima população. Então, nesta nova população, são aplicados operadores de cruzamento de baixo e alto nível e mutação.

A tabela 3.5 descreve os parâmetros utilizados no algoritmo:

OGP	Descrição dos parâmetros
Operações matemáticas	Soma; Subtração; Multiplicação
Taxa de Crossover	Alto; Baixo
Taxa de Mutação	Baixo

Tabela 3.3: Descrição dos parâmetros utilizados no SNN-OGP.

Esta nova população é reavaliada, sendo novamente testado se o critério de parada do algoritmo foi atingido; se foi a população atual é retornada; caso contrário, este processo é retomado quantas vezes forem necessárias.

Após estas definições, a SNN é treinada com base nos resultados de acordo com os valores dos parâmetros e atributos de entrada selecionados. O próximo passo é avaliar cada indivíduo classificando a população com base nestes valores. Operadores de seleção e de recombinação também são aplicados, e este processo é repetido até que um critério de parada seja alcançado. A figura 3.10 ilustra a arquitetura do SNN-OGP.

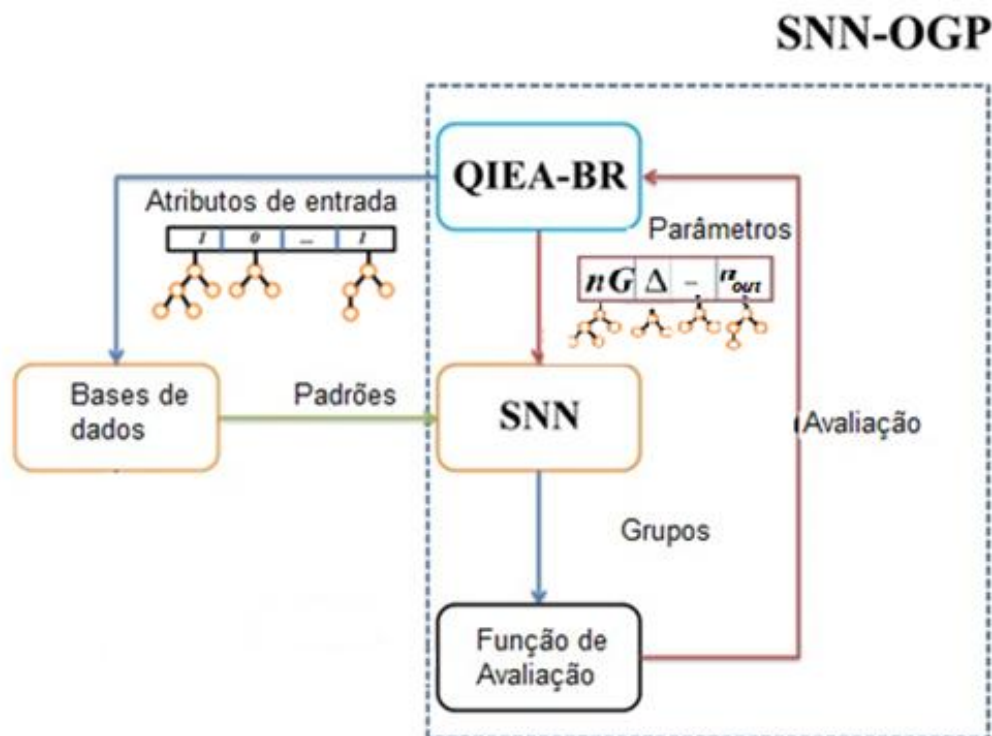


Figura 3.10: Arquitetura do SNN-OGP.

No próximo capítulo serão apresentados os resultados dos modelos propostos com as bases *Benchmark*.

4

Estudo de Casos *Benchmark*

Neste capítulo, serão apresentados resultados utilizando os modelos neuro-evolucionários propostos para um conjunto de bases de dados *benchmark*. Com o intuito de comprovar o desempenho dos modelos evoluídos, será feita, inicialmente, uma comparação dos resultados de uma SNN padrão, sem otimização de parâmetros e seleção dos atributos de entrada, com um algoritmo clássico de agrupamento denominado *K-means* (Kanungo02).

O algoritmo *K-means* é um método de aprendizado não supervisionado, onde a formação dos grupos é feita a partir de cálculos de distâncias entre k vetores da base de dados e os centróides de cada um dos k grupos, sendo esses centróides ajustados iterativamente. A quantidade de grupos (k) que se deseja obter é informada a priori.

Para o teste inicial de desempenho dos modelos, foram utilizadas 8 bases *benchmark* (UCI; Cluster). Alguns desses conjuntos de dados são comuns na literatura e apresentam a classe para cada padrão de entrada, que, com o intuito de comparar os modelos, foram utilizadas para calcular as medidas de qualidade dos agrupamentos para o melhor indivíduo encontrado nos modelos.

Quando a classe da base de dados é disponibilizada pode-se usar o coeficiente chamado *Correct Rand* (cR) para avaliar a precisão do agrupamento (Kuncheva04).

O cR mede basicamente a relação entre a concordância e a discordância de dois grupos. Consideremos os grupos $U = \{U_1, U_2, \dots, U_R\}$ sendo a partição fornecida pelo algoritmo de agrupamento e $V = \{V_1, V_2, \dots, V_C\}$ a partição real com base no conhecimento a priori sobre os dados, pode-se definir o cR pela equação 23 (Kuncheva04).

$$cR = \frac{\sum_i^R \sum_j^C \binom{n_{ij}}{2} - \binom{n}{2}^{-1} \left[\sum_i^R \binom{n_i}{2} \sum_j^C \binom{n_j}{2} \right]}{\frac{1}{2} \left[\sum_i^R \binom{n_i}{2} \sum_j^C \binom{n_j}{2} \right] - \binom{n}{2}^{-1} \left[\sum_i^R \binom{n_i}{2} \sum_j^C \binom{n_j}{2} \right]} \quad (23)$$

onde: n_{ij} representa o número de padrões comuns aos dois grupos u_i e v_j ; n_i indica o número de padrões no grupo u_i ; n_j indica o número de padrões no grupo v_j ; n é o número total de padrões; e $\binom{a}{b}$ é o coeficiente binomial.

O índice cR pode assumir valores entre $[-1, 1]$, onde 1 denota um acordo perfeito entre partições, e valores negativos ou próximos de 0 representam algum tipo de discordância encontrada.

Todas as bases de dados são resumidas na tabela 4.1, onde n , d , e k denotam os números de padrões, de atributos e de classes, respectivamente.

Base de dados – Especificação	n	d	k
Aggregation – base de dados de figuras	788	2	7
Compound – base de dados de figuras	399	2	6
Iris – tipos de plantas	150	4	3
Wine – análise de vinhos	178	13	3
Glass – tipos de vidros	214	9	7
Yeast – localização de proteínas	1484	8	10
Breast – câncer de mama	699	10	2
Thyroid – doença da tireóide	215	5	2

Tabela 4.1: Tabela das bases de dados benchmark.

Por serem modelos para aprendizado não supervisionado todas as informações necessárias para a formação dos grupos são extraídas da base de dados. Por esse motivo, foi utilizada como função de avaliação a Silhueta, calculando-se o cR, que utiliza as classes das bases de dados para calcular as medidas de qualidade dos agrupamentos, para validar a formação dos grupos.

Para a configuração da SNN (padrão) foram utilizados os seguintes parâmetros usados por Bohte (Bohte03): 8 gaussianas defasadas com valores Δ (atrasos) de $t = 0$ a $t = 9$, para o algoritmo de aprendizado foi utilizado os valores dos parâmetros: $\tau = 3\text{ms}$, $b = -0,2$, $c = -2,85$, $\beta = 1,67$ e $\eta = 0,0025$, para a formação de 3 grupos na camada de saída. A formação dos grupos foi comparada com a classe fornecida pela base de dados Iris, obtendo um resultado satisfatório.

Os resultados da Si como função de avaliação e o cR para o melhor indivíduo, referentes às 8 bases *benchmark*, são mostrados na tabela 4.2 com destaque (cinza) para os melhores resultados.

Resultados - Si e cR para o melhor indivíduo				
Base de dados	<i>K-means</i>		SNN	
	Si	cR	Si	cR
Aggregation	0,78	0,77	0,80	0,78
Compound	0,75	0,68	0,76	0,74
Iris	0,80	0,89	0,82	0,92
Wine	0,37	0,50	0,45	0,53
Glass	0,48	0,51	0,51	0,56
Yeast	0,17	0,46	0,20	0,49
Breast	0,38	0,52	0,42	0,58
Thyroid	0,78	0,65	0,80	0,68
Média	0,56	0,62	0,60	0,66
Mediana	0,62	0,59	0,64	0,63

Tabela 4.2: Resultados obtidos para bases benchmark com Si como função de avaliação e o cR para o melhor indivíduo.

Os resultados da SNN padrão apresentaram um desempenho melhor na formação dos grupos em todas as bases de dados, obtendo o valor da Si e cR mais elevado em termos de média e mediana.

Como citado anteriormente, o desempenho de uma SNN está ligado diretamente à combinação correta dos parâmetros da rede, e à utilização de atributos de entrada relevantes. Por esse motivo, dois modelos híbridos com codificação mista (binária e real) foram propostos, de forma a sintonizar os parâmetros da rede SNN (parte real) e selecionar os atributos de entrada (parte binária). Foi também utilizado um AG clássico na evolução da SNN com a finalidade de comparar o desempenho dos modelos híbridos.

Os AG utilizam conceitos provenientes do princípio de seleção natural para abordar uma série ampla de problemas, em especial de otimização, e o processo de evolução é inspirado na teoria da evolução de Darwin. A inicialização básica de um AG clássico se resume à síntese de uma população inicial, sobre a qual serão aplicadas as ações dos passos subseqüentes do processo (Mitchell92).

A proposição dos modelos foi apresentada no capítulo anterior, de forma que a parte binária do cromossomo tem o tamanho de acordo com o número de atributos (d) de entrada de cada base de dados (tabela 4.1).

Para cada base de dados, o AG clássico e os modelos QbrSNN e SNN-OGP foram executados em 40 ensaios independentes, evoluídos em um total de 3000 avaliações. A realização de 3000 avaliações implica avaliar 3000 SNNs com configurações diferentes, sendo este valor escolhido após testes com

valores maiores de avaliações, não apresentando melhoria nos resultados, e por ser computacionalmente viável. Estas execuções foram realizadas em MATLAB (Matlab10). A tabela 4.3 define os valores dos parâmetros utilizados para os algoritmos, sendo o AG com valores *default*, o AEIQ-BR e o OGP escolhidos com base nas recomendações de (Pinho09) e (Koshiyama13) respectivamente. Esses valores foram os mesmo utilizados para todas as bases de dados.

QbrSNN	Valores dos parâmetros	SNN-OGP	Valores dos parâmetros	AG	Valores dos parâmetros
NQ	15	Tamanho da população	50	Tamanho da população	50
NC	60	Número de gerações	60	Número de gerações	60
T	50	Tamanho do Torneio	5	Tamanho do Torneio	4
Ccb	95%	Taxa de cruzamento de alto nível	50%	Taxa de Cruzamento	80%
Ccr	5%	Taxa de cruzamento de baixo nível	70%	Taxa de Mutação	20%
$\Delta\theta$	$0.050 \cdot \pi$	Taxa de Mutação	30%	Taxa de Elitismo	2
Cq	10%	Taxa de Elitismo	2%	-	-
$updatesT$	4	Altura máxima da árvore	3	-	-

Tabela 4.3: Valores dos parâmetros utilizados nos modelos: QbrSNN, SNN-OGP e AG.

A tabela 4.4 apresenta a média dos resultados de 40 ensaios dos resultados de Si e cR para cada base benchmark utilizando uma SNN padrão, um AG clássico, o QbrSNN e o SNN-OGP.

Resultados - Si e cR para o melhor indivíduo								
Base de dados	SNN		AG		QbrSNN		SNN-OGP	
	Si	cR	Si	cR	Si	cR	Si	cR
Aggregation	0,80	0,78	0,82	0,80	0,87	0,97	0,87	0,88
Compound	0,76	0,74	0,80	0,75	0,90	0,88	0,92	0,81
Iris	0,82	0,92	0,86	0,92	0,88	0,82	0,88	0,87
Wine	0,45	0,53	0,77	0,55	0,98	0,76	0,97	0,71
Glass	0,51	0,56	0,75	0,88	0,92	0,92	0,97	0,82
Yeast	0,20	0,49	0,62	0,50	0,88	0,86	0,96	0,85
Breast	0,42	0,58	0,79	0,52	0,86	0,75	0,94	0,73
Thyroid	0,80	0,68	0,88	0,77	0,96	0,80	0,97	0,89
Média	0,60	0,66	0,79	0,71	0,91	0,85	0,94	0,82
Mediana	0,64	0,63	0,80	0,76	0,89	0,84	0,95	0,84

Tabela 4.4: Resultados obtidos para bases benchmark com Si como função de avaliação e o cR para o melhor indivíduo.

O SNN padrão apresentou um desempenho pior quando comparado com os modelos neuro-evolutivos. O AG obteve o valor da Si 15% maior em média quando comparado com a SNN padrão, e os modelos QbrSNN e SNN-OGP obtiveram o valor da Si aproximadamente 35% mais elevado que a SNN padrão e 15% mais elevado que o AG clássico. Isso confirma a vantagem do ajuste automático dos parâmetros das SNN, uma vez que se torna difícil fazer esses ajustes manualmente.

O teste de Wilcoxon (Derac11) foi aplicado para comparar as medidas da Si entre as abordagens. Os modelos QbrSNN e SNN-OGP claramente superaram as abordagens sem evolução em termos da mediana (valor-p <0,01). Portanto, realizando seleção de atributos e otimização de parâmetros com poucas avaliações pode-se melhorar muito a qualidade de agrupamento.

A comparação entre a Si e o cR, entre o QbrSNN e o SNN-OGP, não apresentou uma diferença significativa (p-valor > 0,05), sendo essa diferença muito pequena. Na tabela 4.4, é possível ver que algumas bases de dados obtiveram o melhor valor da Si no modelo QbrSNN enquanto que, em outras bases, o SNN-OGP foi melhor. Nos modelos o melhor valor do cR nem sempre coincidiu com a melhor Si.

No modelo QbrSNN os melhores resultados foram encontrados mais rapidamente, apesar de ter exigido uma média de 40% a mais esforço computacional do que SNN-OGP. Portanto, o usuário deve considerar este *trade-off* ao aplicar QbrSNN para seleção de atributos e otimização de parâmetros.

Estas semelhanças também são expressas nos valores dos parâmetros ótimos finais para cada base de dados, e do número final de atributos de entrada selecionados para cada SNN (tabelas 4.5 e 4.6). Em média, 45% dos recursos de entrada não foram considerados para o agrupamento e, na maioria das bases de dados, o número de neurônios de saída (grupos formados) obtidos foram muito próximo do número de classe de cada base de dados.

É possível visualizar na tabela 4.6 que o número de atributos utilizados para a obtenção do melhor resultado de agrupamento foi diferente em algumas bases de dados. Um número maior de atributos aumenta o tempo de processamento da rede, não significando obter melhores resultados ou uma maior precisão.

Valores dos parâmetros (média aproximada)										
Base de dados	QbrSNN									
	nG	Δ	η	τ	$tmax$	b	c	ϑ	β	$nout$
Aggregation	9	19	0,1035	4	6	0,13	3	8	3	7
Compound	8	12	0,1200	4	7	0,41	4	7	5	6
Iris	8	8	0,8142	5	7	0,64	4	12	5	3
Wine	9	9	0,1247	7	7	0,98	5	8	5	3
Glass	8	9	0,1294	6	7	0,84	4	9	5	7
Yeast	8	10	0,3990	7	7	0,83	5	8	4	8
Breast	9	10	0,9902	7	6	0,12	5	12	5	2
Thyroid	8	12	0,9839	4	7	0,52	4	12	5	2
Média	8	11	0,4581	6	7	0,56	4	10	4,63	-

Valores dos parâmetros (média aproximada)										
Base de dados	SNN-OGP									
	nG	Δ	η	τ	$tmax$	b	c	ϑ	β	$nout$
Aggregation	8	20	0,8990	7	7	0,10	4	6	5	7
Compound	8	8	0,9207	4	7	0,90	5	5	5	6
Iris	8	20	0,1004	7	7	0,79	5	5	5	3
Wine	8	18	0,1006	7	7	0,88	5	5	5	3
Glass	8	15	0,1137	3	7	0,11	4	5	5	7
Yeast	8	20	0,1585	3	6	0,95	4	5	3	6
Breast	8	8	0,1058	3	7	0,79	5	6	6	2
Thyroid	8	8	0,3720	7	7	0,16	4	6	5	2
Média	8	15	0,3463	5	7	0,59	5	5	4,88	-

Tabela 4.5: Valores dos parâmetros (média).

Número de atributos (média aproximada)			
Base de dados	d	QbrSNN	SNN-OGP
Aggregation	2	2	2
Compound	2	2	2
Iris	4	3	3
Wine	13	8	7
Glass	9	4	4
Yeast	8	4	6
Breast	10	6	8
Thyroid	5	3	4

Tabela 4.6: Número de atributos selecionados (média).

É possível visualizar na tabela 4.6 que o número de atributos utilizados para a obtenção do melhor resultado de agrupamento foi diferente em algumas bases de dados. Um número maior de atributos aumenta o tempo de

processamento da rede e não significa que a rede apresente melhores resultados ou uma maior precisão.

A base de dados *Yeast* apresentou um número de grupos (n_{out}) diferentes das classes fornecidas pela base, para ambos os modelos. No modelo QbrSNN das 10 classes disponíveis 8 grupos foram formados, de forma que, as classes de “proteína da membrana” formaram um único grupo e a classe de “retículo endoplasmático” não formou grupo. No modelo SNN-OGP 6 grupos foram formados, sendo que as classes de “proteína da membrana” formaram um único grupo, a classe de “retículo endoplasmático” não formou grupo e a classe “*peroxisoma*” também não influenciou a formação dos grupos.

O próximo capítulo apresentará um estudo de caso com bases de dados composta por parâmetros extraídos do sinal de voz.

5

Bases com sinais de voz

Durante a dissertação de Mestrado do autor desta Tese de Doutorado, um extenso estudo sobre características de envelhecimento vocal foi discutido, e como os parâmetros extraídos dos sinais de voz poderiam contribuir com a identificação de vozes envelhecidas.

A voz é uma ferramenta necessária para a comunicação do ser humano, e sua produção ocorre pelo trabalho em conjunto dos sistemas respiratório, digestivo, nervoso, além de cartilagens, músculos, ligamentos e ossos. São 3 os subsistemas que atuam sucessivamente na produção da fala: o respiratório, responsável pelo fluxo de ar gerado; o laringeal, que é a parte do sistema onde se encontra a glote (espaço entre as cordas vocais) e o supralaringeal, que atua como um filtro modificando o espectro da voz para cada indivíduo (Titze94).

Devido à contração e expansão dos pulmões, um fluxo de ar é gerado criando uma diferença de pressão entre o ar dos pulmões e a pressão do ar na boca, provocando um escoamento de ar. Ao passar pelas cordas vocais, esse escoamento transforma-se em uma série de pulsos de ar, formando o sinal glotal em um sinal quase periódico, de baixa intensidade, cuja frequência fundamental é denominada “frequência fundamental da voz”.

Os pulsos de ar formados na glote sofrem influência do sistema de ressonância formado pelos tratos vocal e nasal, que funcionam como um filtro, gerando frequências de ressonância chamadas de formantes. Os formantes têm um papel importante na inteligibilidade da fala. Os diferentes sons emitidos só são possíveis devido a essa filtragem.

Os formantes são os efeitos ressonantes no trato vocal relacionado à amplificação da energia do som no subsistema supralaríngeo. Estão relacionados à anatomia específica do aparelho fonador de cada indivíduo, tendo um papel importante na inteligibilidade da fala (Titze94) (figura 5.1).

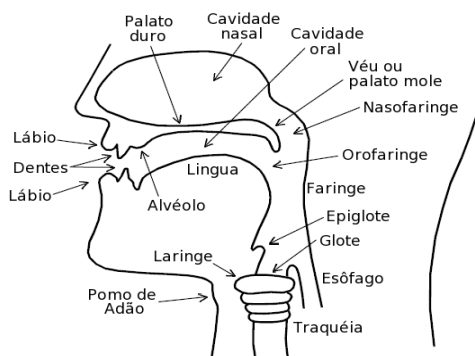


Figura 5.1: Aparelho Fonador. Fonte (Silva10).

Um dos principais objetivos desta tese é analisar o problema de envelhecimento de voz utilizando os parâmetros extraídos dos sinais de voz, com os modelos de agrupamento discutidos. Assim, os modelos propostos foram empregados na caracterização do envelhecimento da voz. Esse tipo de aplicação se apresenta como um trabalho inédito devido à dificuldade de identificação do envelhecimento vocal (Santos05).

Uma das principais formas de se caracterizar o sinal de voz é através de medições acústicas, devido estas permitirem revelar importantes características fisiológicas através de variações de seus valores. A partir daí, os parâmetros obtidos através do sinal de voz podem ser usados pela SNN para agrupar vozes envelhecidas.

O envelhecimento é um processo fisiológico, contínuo e irreversível, e grandes transformações físicas ocorrem nas pessoas idosas, mas essas transformações não se devem somente pela idade cronológica. As condições físicas e sociais que se apresentam os indivíduos, também são fatores fundamentais para a detecção do envelhecimento vocal.

Estudos sobre características de envelhecimento vocal foram utilizados em diversos trabalhos, e a utilização desses parâmetros, que são utilizados para identificar patologias diversas, mostraram-se bons discriminantes para tarefas de classificação.

Para cada registro de voz, os parâmetros foram extraídos diretamente do sinal de voz e do correspondente sinal glotal, através de filtragem inversa, usando os softwares Praat e Aparat (Praat08; Aparat05). Os seguintes parâmetros foram obtidos: F0 (frequência fundamental), *Jitter*, *Shimmer*, HNR, Ko, Ka, pp, NAQ, AQ, CIQ, OQ1, OQ2, OQa, QoQ, SQ1, SQ2, DH12 e HRF. Os parâmetros podem ser divididos em 3 grupos: parâmetros no domínio do tempo, no domínio da frequência e os que representam variações da frequência fundamental (Pulaka05).

A seguir uma breve descrição dos parâmetros extraídos do sinal de voz e do sinal glotal.

5.1

Parâmetros extraídos dos sinais de voz

Através de algumas medidas acústicas da voz, informações sobre a saúde vocal podem ser obtidas, distinguindo aquelas que ocorrem naturalmente com a idade daquelas associadas a algum tipo de patologia. Diversas perturbações ocorrem na produção da voz, baseadas na frequência fundamental, amplitude e na forma do sinal. Muitas vezes, por serem pequenos distúrbios ou mudanças temporais não alteram o padrão de forma visual e nem definitivo do sinal.

Esta seção tem como objetivo detalhar os parâmetros mais utilizados, extraídos do sinal de voz utilizando o software *Praat* para a extração desses parâmetros.

O *Praat* (Praat08) é um software livre, que permite aplicação em diferentes plataformas, utilizado para análise do sinal de voz desenvolvido por *Paul Boerma e David Weenink – Institute of Phonetic Sciences – Amsterdã*. Seu foco principal é análise sonora através de parâmetros como frequência, comprimento de onda, decibéis, entre outros. As análises são feitas ao longo da forma de onda captada por um microfone. O programa *Praat* será detalhado no Apêndice A.

As medidas acústicas têm sido utilizadas na tentativa de mostrar a evolução do mecanismo vocal ao longo da vida. Entre as mais conhecidas pode-se citar: Frequência Fundamental (F0), *Jitter*, *Shimmer* e *HNR* (*Harmonic-to-Noise Ratio*).

- **Frequência Fundamental (F0):** é a frequência de abertura e fechamento da glote. A F0 tem uma relação direta com a massa, elasticidade e comprimento das pregas vocais. Por meio da análise da F0, podemos determinar a idade e sexo do locutor.
- ***Jitter*:** é uma variação ou perturbação na frequência dos pulsos gerados pelas cordas vocais. Ocorrendo uma assimetria na abertura e fechamento da glote, gera-se uma variação na frequência fundamental.
- ***Shimmer*:** é uma variação ou perturbação na amplitude dos pulsos gerados pelas cordas vocais,

- **HNR:** é à medida que quantifica a presença de ruído no sinal glotal, determinando a qualidade da voz.

Esses parâmetros vão constituir a base de dados de voz. A seguir será detalhada a forma de extração de parâmetros da voz utilizando o software *Aparat*. Este software extrai o sinal gerado na glote, utilizando os sinais captados por um microfone, cancelando todas as influências ocorridas no trato vocal.

5.2

Parâmetros extraídos do sinal glotal

Várias técnicas foram desenvolvidas para obter informações sobre a formação e a modelagem do sinal glotal (Henrich02; Henrich03; Henrich04). Dentre elas, a técnica da filtragem inversa, a partir do sinal de voz, cancela os efeitos dos formantes na voz sendo possível obter o sinal produzido pela glote.

O *Aparat* pode ser definido como um conjunto de ferramentas e interface gráfica utilizada para análise de sinais de voz. Esse software foi desenvolvido na Universidade de Tecnologia de Helsinki (Aparat05). O programa *Aparat* será detalhado no Apêndice A.

A partir do sinal glotal, alguns parâmetros podem ser extraídos conforme mostrado na figura 5.2. Eles serão discutidos a seguir.

Os parâmetros extraídos do sinal glotal são apresentados a seguir com uma breve introdução, conforme ilustrado na figura 5.2 (Pulaka05; Juliano08).

- **Ko (fase de fechamento da glote):** é a duração da fase de fechamento da glote. Tal duração é indicada na figura 5.2(b) corresponde ao intervalo de tempo que as cordas vocais levam da fase totalmente aberta até a fase de fechamento.
- **Ka (fase de abertura):** é a duração da fase de abertura da glote. Tal duração é indicada na figura 5.2(b). Identifica o tempo em que as cordas vocais levam da fase totalmente fechada até a fase de abertura.
- **pp (distância entre os instantes de máxima abertura glotal):** esse parâmetro é definido como a distância entre os valores máximos de pico do sinal glotal (Pulaka05).

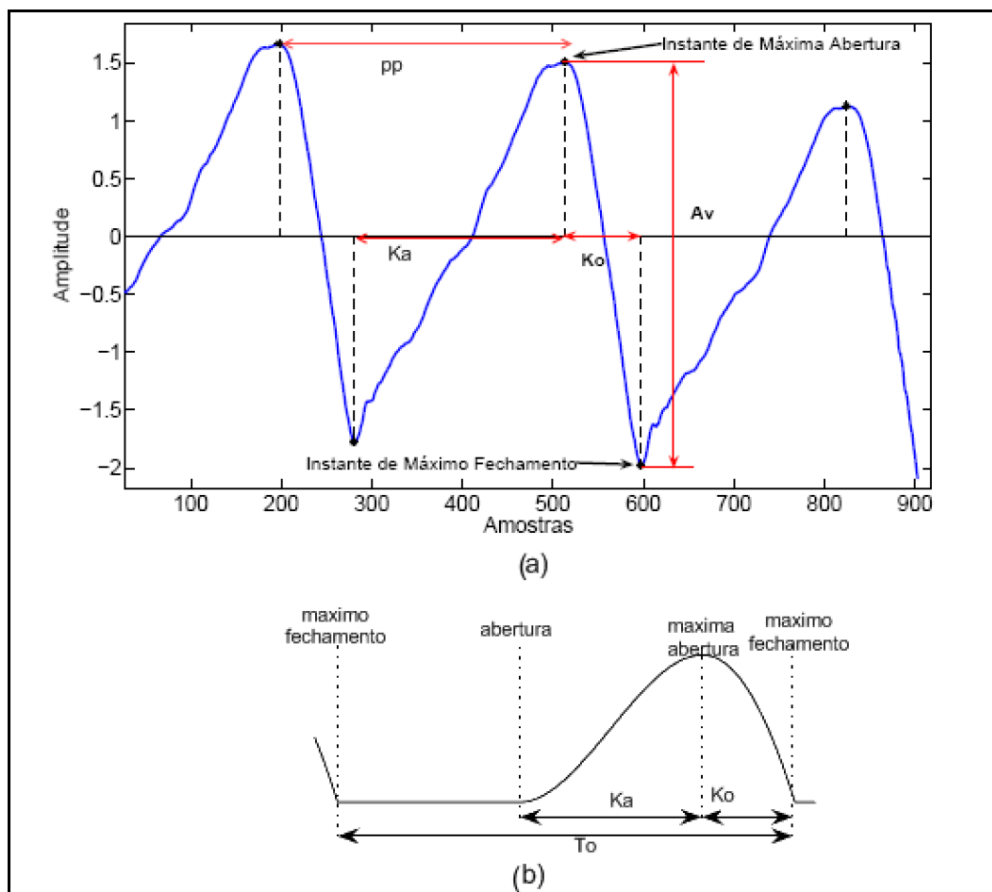


Figura 5.2: (a) Parâmetros do sinal glotal; (b) Sinal glotal. Fonte (Silva10).

- **OQ (quociente de abertura):** é a razão entre o tempo total de abertura da glote e o tempo de um ciclo do sinal glotal. É inversamente proporcional à intensidade da voz, de forma que, quanto menor é o seu valor, maior será a intensidade da voz (Pulaka05).
- **CIQ (quociente de fechamento):** é a razão entre a fase de fechamento (K_o) e o tempo total de um ciclo do sinal glotal. É inversamente proporcional à intensidade da voz, de forma que, quanto menor é seu valor, maior será a intensidade da voz (Pulaka05).
- **SQ (quociente de velocidade):** é definido como a razão entre a duração da fase de abertura e a fase de fechamento do sinal glotal (Pulaka05).

- **AQ (quociente de amplitude):** é a razão entre a amplitude do sinal glotal (máxima abertura) e o valor mínimo do sinal glotal. Esta relacionada com a fonação do locutor (Juliano08).
- **QoQ (quociente de abertura durante o pulso glotal):** é a relação entre o tempo da abertura do sinal glotal no exato momento da oscilação e o tempo de fechamento. Alguns trabalhos utilizam esse parâmetro para determinar emoções (Juliano08).
- **OQa (quociente de abertura definido pelo modelo Liljencrants-Fant):** é calculada utilizando o modelo Liljencrants-Fant para filtragem inversa.
- **NAQ (quociente de amplitude normalizada):** é calculada pela razão entre o quociente de amplitude (AQ) e o tempo total da duração de um pulso glotal (Juliano08).
- **DH12 (diferença entre harmônicos):** é a diferença entre os valores do primeiro e segundo harmônicos do sinal glotal (Titze94). Este parâmetro tem sido utilizado para medir a qualidade vocal.
- **HRF (Fator de relação harmônica):** expressa o fator de riqueza harmônica do sinal. Relaciona o primeiro harmônico com o somatório da energia dos harmônicos restantes. Também é utilizado para medir a qualidade vocal.

Os parâmetros foram separados de acordo com o grupo ao qual pertencem. O OQ foi dividido em OQ1 e OQ2 e são calculados pelas primeira e segunda abertura do fluxo glotal. OQ1 é calculado pelo fechamento do fluxo glotal até o fechamento do próximo fluxo glotal, e o OQ2 é calculado pela abertura até o fechamento do fluxo glotal. O parâmetro SQ, também foi dividido em dois, sendo responsável pela assimetria da glote durante a fonação. A figura 5.3 ilustra o processo feito para extrair os parâmetros e analisar o envelhecimento de voz:

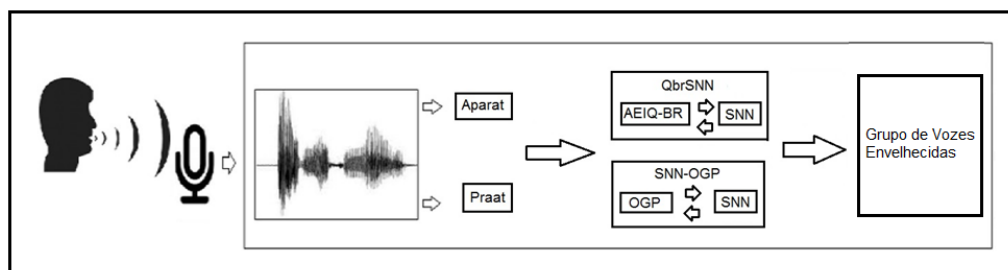


Figura 5.3: Processo para caracterização de uma voz envelhecida.

5.3

SNN evoluída aplicada nas bases de vozes para identificação de envelhecimento

As bases de vozes foram compostas por vozes gravadas de oradores do sexo masculino e feminino, em boas condições de saúde de diferentes idades. Alguns locutores têm algum grau de parentesco, e foram adquiridos da seguinte forma: um grupo de moradores de casas de repouso, um grupo da casa Retiro dos Artistas, um grupo de voluntários gravados em suas próprias residências e um grupo de locutores gravados no consultório da fonoaudióloga Dra. Simone Aiex.

Os grupos foram divididos por idade cronológica, nomeados: jovens (classe 1), adultos (classe 2) e idosos (classe 3). Cada base de voz é composta de 3 grupos com 60 registros da vogal sustentada /e/ para o português falado na cidade do Rio de Janeiro, com 18 atributos. Cada grupo foi composto da seguinte forma: 15 a 30 anos (jovens – classe 1), 31 a 60 (adultos – classe 2) e 61 a 100 anos (idosos – classe 3). As classes serão utilizadas para calcular a qualidade da formação dos grupos.

Como função de avaliação utilizada foi o método da Silhueta (Si). A Si foi maximizada no processo de evolução dos modelos e, com o melhor indivíduo da população, calculou-se o cR para avaliar a formação dos grupos para as bases de voz.

De forma idêntica à utilizada para as bases benchmark, os resultados da base de voz foram comparados com uma SNN padrão, onde os parâmetros e atributos das bases não foram otimizados, com o algoritmo *k-means* (Kanungo02). Posteriormente, os resultados da SNN padrão foram comparados com um AG clássico e os modelos propostos, QbrSNN e SNN-OGP. Os modelos híbridos foram evoluídos com um total de 3000 avaliações treinando uma SNN padrão para o agrupamento, com 40 ensaios independentes para cada modelo. Os valores dos parâmetros de cada algoritmo evolutivo foram os mesmos já apresentados na tabela 4.3.

Os resultados da Si, como função de avaliação, e o cálculo do cR para o melhor indivíduo, para o *K-means* e SNN padrão referentes às bases de voz, são mostrados na tabela 5.1 com destaque (cinza) para os melhores resultados.

Resultados - Si e o cR para o melhor indivíduo – 3000 avaliações				
	K-means		SNN padrão	
Base de dados	Si	cR	Si	cR
Masculino	0,71	0,65	0,75	0,55
Feminino	0,69	0,67	0,72	0,69

Tabela 5.1: Resultados obtidos para as bases de voz – K-means e SNN padrão.

Para a configuração dos modelos híbridos, os mesmos parâmetros da SNN definem a parte real do cromossomo com 10 genes (nG , Δ , η , τ , $tmax$, b , c , g , β e $nout$), e o tamanho da parte binária do cromossomo contém 18 genes, referente ao total de atributos extraídos do sinal de voz. Assim, o cromossomo utilizado na análise do envelhecimento de voz é apresentado na figura 5.4:

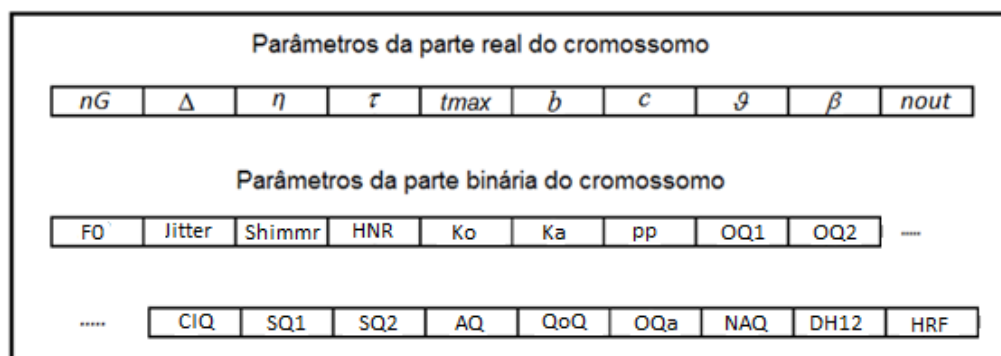


Figura 5.4: Cromossomo com a descrição dos parâmetros para a SNN.

Os resultados das bases de voz, com uma SNN padrão sem atributos e parâmetros otimizados e os modelos híbridos, são apresentados na tabela 5.2. A Si foi maximizada no processo de evolução dos modelos e, com o melhor indivíduo da população, calculou-se cR para avaliar a formação dos grupos.

Resultados - Si e o cR para o melhor indivíduo – 3000 avaliações								
	SNN padrão		AG		QbrSNN		SNN-OGP	
Base de dados	Si	cR	Si	cR	Si	cR	Si	cR
Masculino	0,75	0,55	0,89	0,75	0,96	0,92	0,93	0,89
Feminino	0,72	0,69	0,80	0,55	0,92	0,90	0,91	0,88

Tabela 5.2: Resultados obtidos para as bases de voz para os modelos híbridos.

Os modelos neuro-evolucionários obtiveram resultados do Si e do cR aproximadamente 18% mais altos em termos de média em comparação com os outros algoritmos. Os resultados QbrSNN, em termos de Si e cR, são melhores que os resultados do SNN-OGP com uma pequena diferença.

Na figura 5.5 é apresentado as curvas de evolução dos modelos QbrSNN e SNN-OGP, para 3000 avaliações:

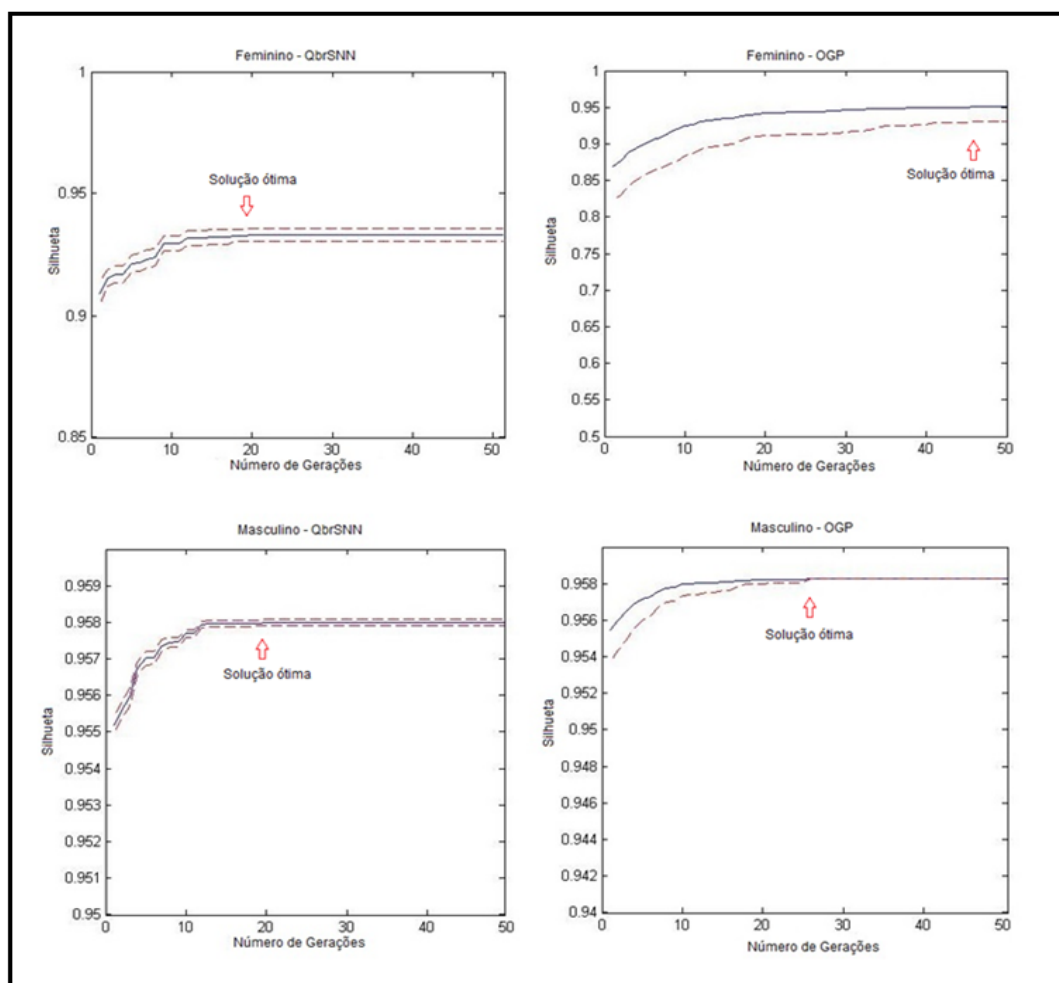


Figura 5.5: Curvas de evolução dos modelos para as bases de voz Masculino e Feminino.

É possível notar que o QbrSNN encontra a solução ótima com um número menor de gerações. Assim, foram feitos testes dos modelos neuro-evolucionários com um número menor de avaliações, acarretando um menor custo operacional dos modelos, o que na prática é um fator muito importante.

A tabela 5.3 mostra os valores para com os seguintes valores de evolução: 1.000, 2.000 e 3.000 avaliações com 40 ensaios independentes, com destaques (cinza) para os melhores resultados encontrados para cada modelo:

Resultados - Si e cR - 1000 avaliações				
	QbrSNN		SNN-OGP	
Base de dados	Si	cR	Si	cR
Masculino	0,85	0,75	0,81	0,73
Feminino	0,88	0,69	0,73	0,61

Resultados - Si e cR - 2000 avaliações				
	QbrSNN		SNN-OGP	
Base de dados	Si	cR	Si	cR
Masculino	0,96	0,92	0,89	0,85
Feminino	0,91	0,90	0,78	0,78

Resultados - Si e cR - 3000 avaliações				
	QbrSNN		SNN-OGP	
Base de dados	Si	cR	Si	cR
Masculino	0,96	0,92	0,91	0,89
Feminino	0,91	0,90	0,82	0,88

Tabela 5.3: Resultados obtidos para as bases de voz para diferentes valores de avaliações.

É possível constatar na tabela 5.3 que o QbrSNN encontra o valor ótimo com um número menor de avaliações, corroborando com o esperado quando utilizado um algoritmo com inspiração quântica, enquanto que o SNN-OGP alcança o seu melhor resultado com um valor de 3000 avaliações.

A média dos valores finais dos parâmetros, para a parte real do cromossomo, que configura a SNN, é ilustrada na tabela 5.4 para cada base de dados:

Valores dos parâmetros (média aproximada)										
Base de dados	QbrSNN									
	nG	Δ	η	τ	$tmax$	b	c	ϑ	β	$nout$
Masculino	9	15	0,4698	5	7	0,30	5	6	7	3
Feminino	8	12	0,2131	3	7	0,91	5	6	7	3

Valores dos parâmetros (média aproximada)										
Base de dados	SNN-OGP									
	nG	Δ	η	τ	$tmax$	b	c	ϑ	β	$nout$
Masculino	8	20	0,3580	7	7	0,79	4	5	7	3
Feminino	8	15	0,7858	4	7	0,10	5	7	7	3

Tabela 5.4: Valores dos parâmetros para SNN (média).

Em média, os atributos de entrada selecionados para a formação dos grupos foram aproximadamente: 14% para o grupo do sexo feminino e 26% para o grupo masculino. Em ambas as bases de dados, o número de grupos de saída foi concordante com o esperado para cada faixa etária, 3 grupos para cada tipo de locutor (Masculino e Feminino).

Os modelos, QbrSNN e SNN-OGP, obtiveram os mesmos resultados na formação dos grupos. De acordo com as faixas de idade disponibilizada para as classes (15 a 30 anos (jovens – classe 1), 31 a 60 (adultos – classe 2) e 61 a 100 anos (idosos – classe 3)), a base de dados Masculino agrupou corretamente 58 locutores e 2 incorretamente (2 locutores do grupo adulto agrupados no grupo de idosos), e a base Feminino agrupou corretamente 58 locutores e 2 incorretamente (2 locutores do grupo adulto no grupo de idosos). Esses resultados foram idênticos para os dois modelos propostos sendo que o QbrSNN obteve esse resultado com um número menor de avaliações que o OGP-SNN.

A quantidade e os atributos, selecionados para a formação dos grupos entre os modelos, foram diferentes. Todos os atributos, selecionados, pelos modelos neuro-evolutivos, apresentam características que podem se alterar com o avanço da idade.

Os atributos selecionados pelo QbrSNN para a base de dados do sexo Feminino foram: *Shimmer*, NAQ e AQ. Estes atributos são relacionados com a amplitude da voz (*Shimmer*, NAQ e AQ). Na base de dados do sexo Masculino, os atributos selecionados foram: F0, *Jitter*, Ko, SQ1 e SQ2. Estes atributos estão relacionados com a frequência vocal (F0), a perturbação na frequência do sinal (*Jitter*), a fase de fechamento glotal (Ko) e o quociente de velocidade da glote.

Os atributos selecionados pelo SNN-OGP para a base de dados do sexo Feminino foram: *Shimmer* e AQ. Estes atributos são relacionados com a amplitude da voz (*Shimmer*, AQ), o que na fase idosa apresenta uma queda na amplitude do sinal vocal. Para a base de dados do sexo Masculino foram selecionados: F0, *Jitter*, Ko e CIQ. Estes atributos estão relacionados com a frequência vocal (F0), perturbação na frequência do sinal (*Jitter*), e os parâmetros que estão relacionados com a fase de fechamento da glote (Ko e CIQ).

Para a base do sexo Feminino os valores dos atributos de amplitude da voz apresentaram uma queda no grupo de idosos. Este resultado concorda com grande parte da literatura que aponta a tendência desse tipo de alteração na fase idosa (Silva10, Santos05).

Na base do sexo Masculino os parâmetros F0, *Jitter* estão relacionados com a qualidade vocal e condição física das cordas vocais, que, com o avanço da idade, ocorre um relaxamento das cordas vocais podendo ser esse o motivo desses parâmetros terem sido escolhidos (Santos05). Os parâmetros relacionados com a fase de fechamento da glote, Ko e CIQ, apresentaram um crescimento na fase idosa, enquanto que SQ1 e SQ2 apresentaram uma queda. O valor baixo do parâmetro SQ concorda com a literatura, devido ao fato deste ser inversamente proporcional ao Ko, que apresenta um aumento na fase idosa. O CIQ é diretamente proporcional ao parâmetro Ko.

Alguns atributos selecionados para ambos os modelos, QbrSNN e SNN-OGP foram: base de dados Feminino: *Shimmer* e AQ; para a base Masculino: F0, *Jitter* e Ko.

A figura 5.7 mostra o cromossomo final com os parâmetros escolhidos e atributos mais relevantes para o modelo QbrSNN:

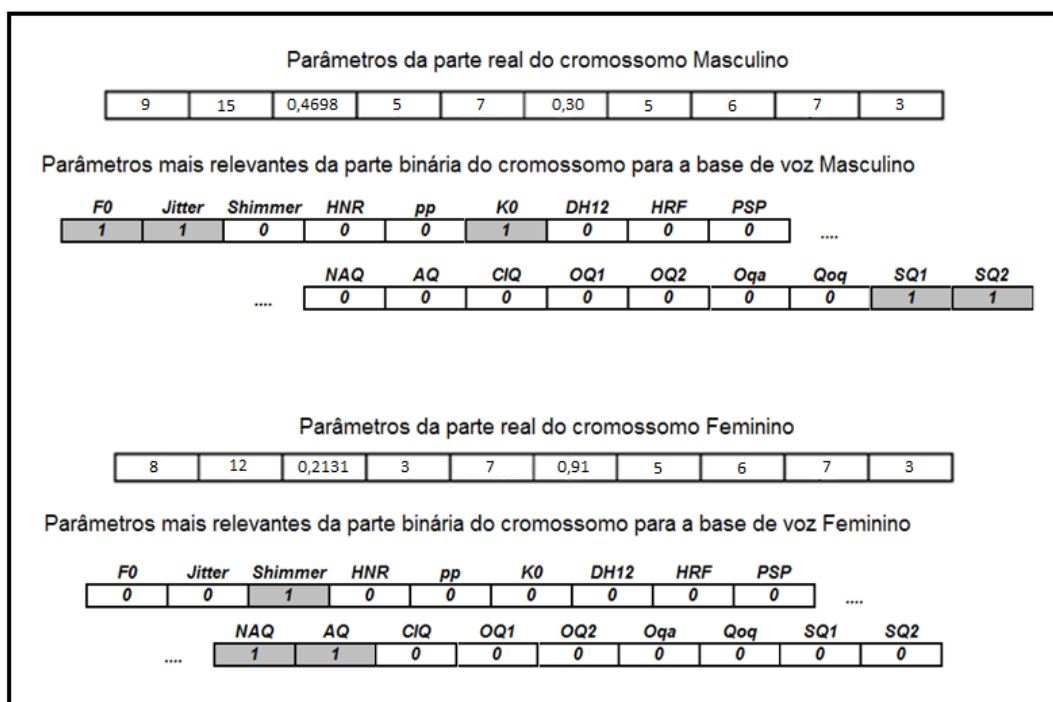


Figura 5.6: Resultado do cromossomo final utilizando o QbrSNN.

A figura 5.8 mostra o cromossomo final com os parâmetros escolhidos e atributos mais relevantes para o modelo SNN-OGP:

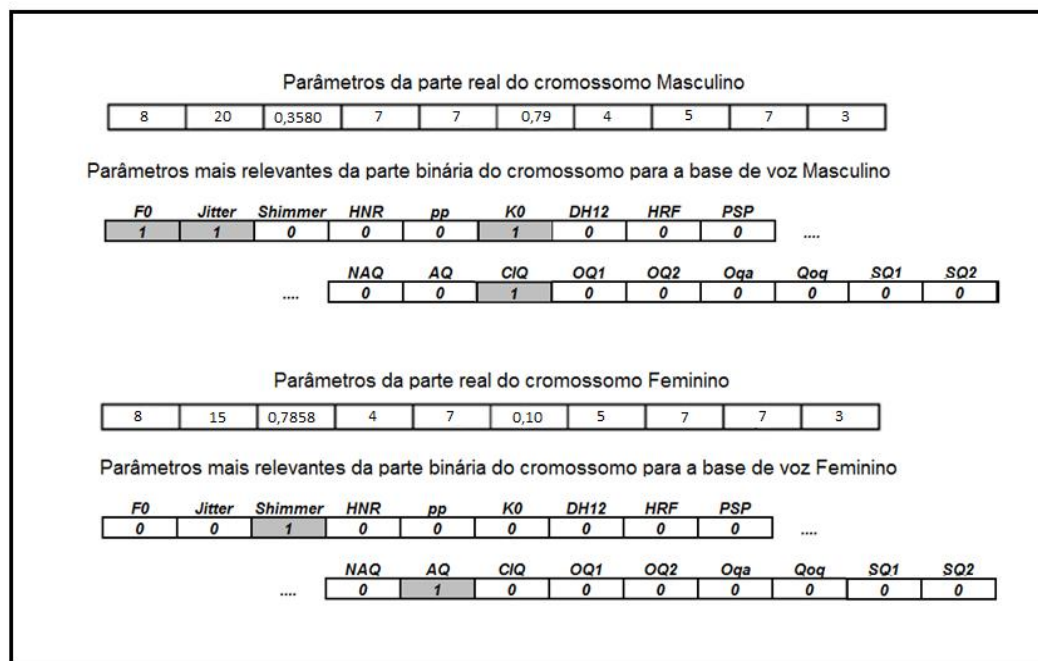


Figura 5.7: Resultado do cromossomo final utilizando o SNN-OGP.

Foram feitas análises dos atributos selecionados pelos modelos, para confirmar o motivo destes locutores terem sido agrupados diferentemente da sua idade cronológica. As tabelas 5.4 e 5.5 mostram os intervalos dos atributos, das bases de dados Feminino e Masculino, selecionados para os modelos QbrSNN e SNN-OGP, respectivamente:

Intervalos dos atributos relevantes para o agrupamento Feminino - Min e Max			
	<i>Shimmer</i>	<i>NAQ</i>	<i>AQ</i>
Jovens	3,02 - 4,40	0,01 - 0,05	0,41 - 0,70
Adultos	1,10 - 2,85	0,06 - 0,09	0,22 - 0,39
Idosos	8,78 - 10,52	0,10 - 0,29	0,93 - 1,40

Tabela 5.5: Intervalos dos atributos na base Feminina.

Intervalos dos atributos relevantes para o agrupamento Masculino - Min e Max						
	<i>F0</i>	<i>Jitter</i>	<i>Ko</i>	<i>CIQ</i>	<i>SQ1</i>	<i>SQ2</i>
Jovens	100 - 120	0,23 - 0,40	17 - 28	0,04 - 0,10	3,50 - 4,68	3,35 - 4,92
Adultos	125 - 140	0,43 - 0,50	30 - 49	0,12 - 0,18	2,26 - 3,0	2,51 - 2,88
Idosos	120 - 137	0,47 - 0,94	50 - 107	0,20 - 0,70	0,52 - 2,0	0,55 - 1,78

Tabela 5.6: Intervalos dos atributos na base Masculina.

As figuras, 5.9 e 5.10 apresentam as variações dos parâmetros entre as classes (jovens, adultos, idosos) com as médias dos parâmetros para as bases de vozes: masculino e feminino:

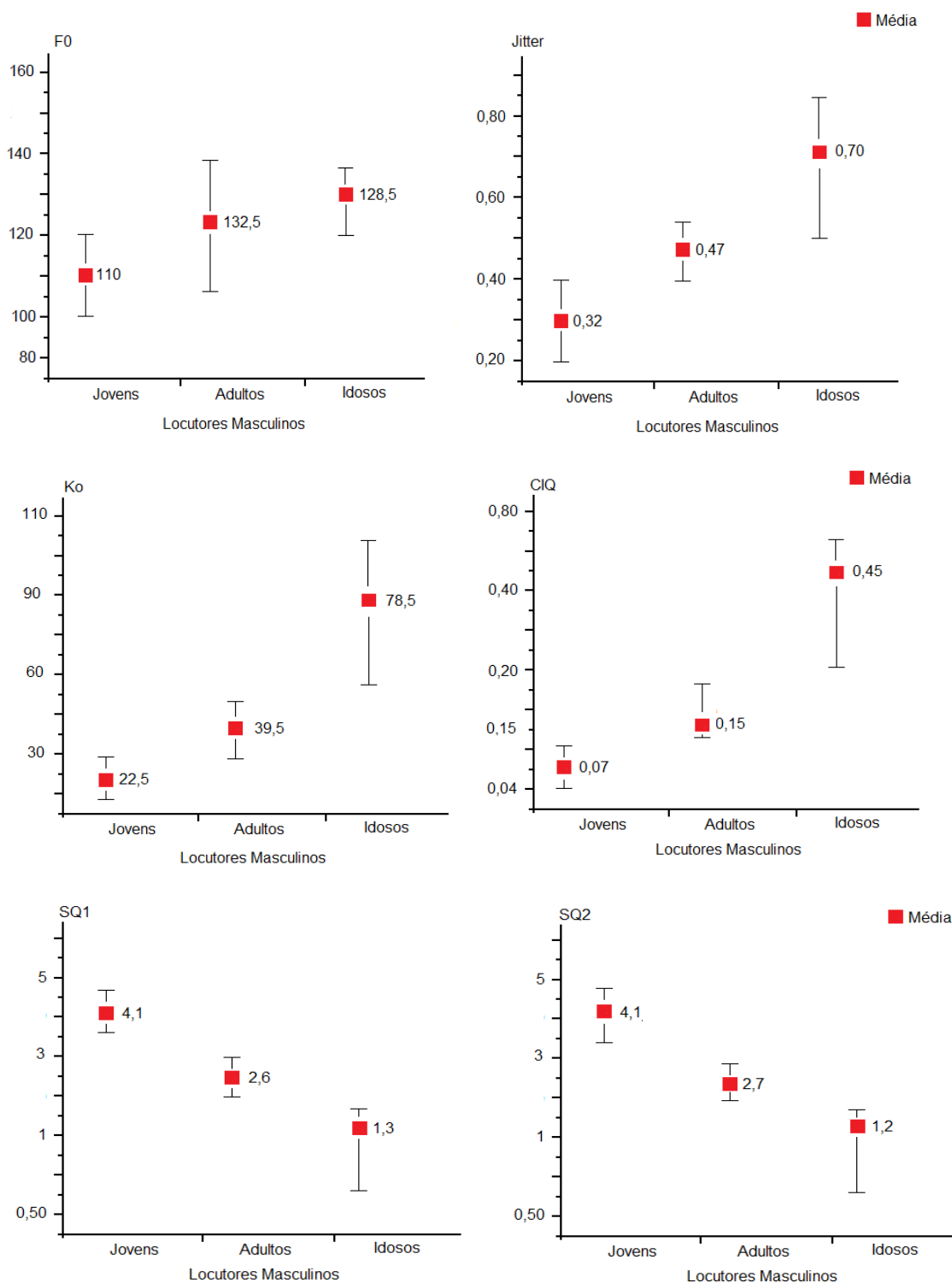


Figura 5.8: Variações dos parâmetros referentes à base do sexo Masculino.

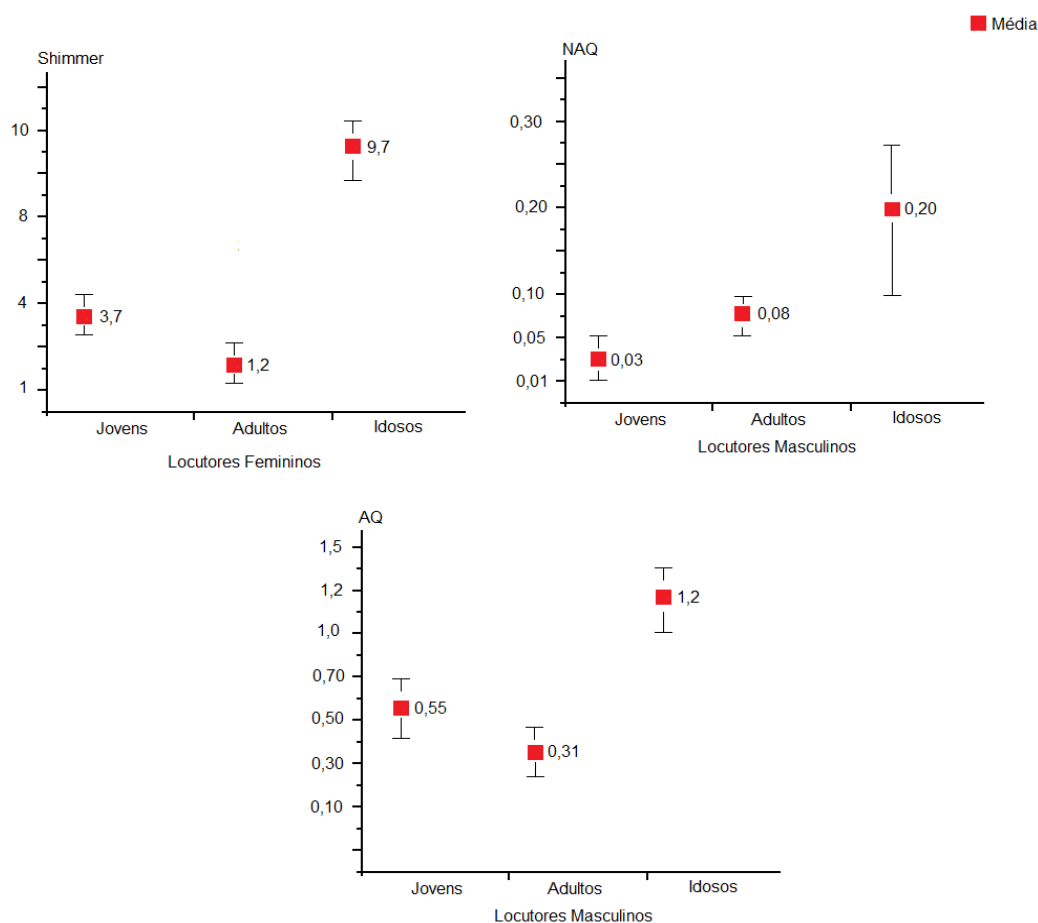


Figura 5.9: Variações dos parâmetros referentes à base do sexo Feminino.

A seguir serão apresentados, nas tabelas 5.6 e 5.7, as idades cronológicas dos locutores e as médias dos valores dos atributos que foram selecionados:

Feminino				
	Idade	Shimmer	NAQ	AQ
Locutor 1	45 anos	9,98	0,18	1,10
Locutor 2	50 anos	8,22	0,18	1,12
Locutor 3	58 anos	9,09	0,22	1,23

Tabela 5.7: Valores dos parâmetros dos locutores na base de dado Feminino.

Masculino							
	Idade	F0	<i>Jitter</i>	Ko	CIQ	SQ1	SQ2
Locutor 1	51 anos	121	0,58	51	0,68	0,55	0,85
Locutor 2	56 anos	130	0,71	53	0,51	0,96	1,77

Tabela 5.8: Valores dos parâmetros dos locutores na base de dado Masculino.

Os valores estão dentro dos intervalos discutidos na literatura, como representativos das idades dos locutores.

6 Conclusões e Trabalhos Futuros

Foram propostos dois modelos neuro-evolucionários denominados: *Quantum-Inspired Evolution of Spiking Neural Networks with Binary-Real representation* (QbrSNN) e *Spiking Neural Network Optimization by Genetic Programming* (SNN-OGP), para agrupamento de vozes envelhecidas utilizando redes neurais *Spiking*. Esses modelos baseiam-se em dois modelos de algoritmos evolutivos: um algoritmo evolutivo com inspiração quântica com representação mista (Binária-Real) muito utilizado em problemas de otimização combinatória, e outro algoritmo de Otimização por Programação Genética onde um indivíduo é representado como um complexo de estruturas em árvores, com base em princípios de programação genética. Os modelos propostos também foram comparados com uma abordagem clássica de Algoritmo Genético (AG) na evolução da SNN, com a finalidade de comparar o desempenho dos modelos híbridos.

Nesta tese, também foram estudados os fundamentos sobre o funcionamento das SNNs buscando o melhor ajuste dos parâmetros da rede, e selecionando os atributos de entrada que são relevantes para obter bons resultados.

Os testes utilizando bases *benchmark* foram fundamentais para o entendimento do funcionamento das redes utilizando neurônios *spiking*. Os resultados dos modelos híbridos para essas bases, que se apresentam como bases estáticas, demonstraram que os modelos são viáveis, superando significativamente uma SNN padrão sem evolução. Ao comparar as duas abordagens, QbrSNN rendeu um agrupamento ligeiramente de maior qualidade em algumas bases de dados, enquanto que o SNN-OGP foi melhor na maioria dos casos com um esforço computacional, em média, 40% maior. O usuário deve considerar este *trade-off* ao aplicar QbrSNN para seleção de atributos e otimização de parâmetros.

Com relação à seleção de atributos relevantes, os algoritmos encontraram um número diferente de atributos para a maioria das bases, o que determina um melhor rendimento dos modelos.

Para avaliar a capacidade dos modelos, foi utilizado um problema com grande dimensionalidade. Estudos utilizando características da voz foram utilizados em diversos trabalhos, sendo interessante demonstrar como os parâmetros extraídos dos sinais de voz podem contribuir com a identificação de vozes envelhecidas. As utilizações desses parâmetros, que são utilizados para identificar patologias diversas, mostraram-se adequados para a formação dos grupos.

Para as bases de voz, foram utilizadas duas abordagens sem evolução e comparadas entre si, sendo uma clássica de agrupamento (*K-means*) e uma SNN padrão sem evolução. Nessas abordagens é necessário estabelecer previamente a quantidade de grupos desejados, não sendo possível a adaptação estrutural da rede e nem a seleção de atributos.

Os resultados obtidos pelos modelos híbridos mostraram que os modelos propostos são adequados e superam outros modelos de agrupamento. A seleção de atributos foi tal que, as vozes foram corretamente agrupadas com uma elevada taxa de classificação, utilizando um valor menor de atributos.

Os atributos mais significantes na identificação de vozes envelhecidas para o sexo masculino nos modelos híbridos foram: F0, *Jitter*, Ko, SQ1, SQ2 e CIQ. Esses parâmetros estão relacionados com a qualidade e intensidade da voz.

Uma análise de F0 pode contribuir para a identificação da idade, sexo e faixa etária do locutor (Silva10). A justificativa do aumento da F0 na fase idosa se deve a mudanças anatômicas e fisiológicas da laringe, e irregularidades na vibração das cordas vocais, ambos decorrentes do processo de envelhecimento.

O *Jitter* está relacionado ao aspecto perceptual de aspereza na voz que, no caso de vozes envelhecidas, é um atributo significativo. O resultado concorda com a literatura em relação ao fato que, locutores na fase idosa apresentam valores elevados de *Jitter* quando comparados com as outras faixas etárias (Santos05).

O atributo Ko cresce significativamente na fase idosa, o que confirma a escolha dos outros parâmetros (CIQ, SQ1 e SQ2) como atributos relevantes. O CIQ é diretamente proporcional ao valor de Ko ($CIQ = Ko/T$) onde T é o período do sinal glotal, e o SQ é inversamente proporcional ao valor de Ko ($SQ = Ka/Ko$) (Silva10).

Os atributos mais significantes na identificação de vozes envelhecidas para o sexo feminino nos modelos híbridos foram: *Shimmer*, NAQ e AQ. *Shimmer* está relacionado ao aspecto perceptual de rouquidão. O alto valor do shimmer encontrados na fase idosa é discordante com os achados em Santos (Santos05),

que obteve um valor menor na fase idosa. As mudanças que ocorrem na laringe contribuem para a instabilidade da vibração das cordas vocais, devido à ossificação das cartilagens e a degeneração do músculo vocal, podendo causar o aumento da aspereza e rouquidão na voz dos locutores idosos. Por esse motivo, altos valores de *Shimmer* são justificados.

AQ e NAQ têm correlação com mudanças de intensidade da voz, e apresentaram um aumento na fase idosa. Com a idade, o sinal de voz apresenta um aumento de ruído na amplitude (*Shimmer*) do sinal (Silva10).

O QbrSNN comprovou uma maior velocidade de convergência quando comparado com o OGP-SNN em relação ao número de avaliações, e obteve o melhor resultado de agrupamento para as duas bases de dados de voz. Um menor número de avaliações se deve ao fato de, no caso do QbrSNN, cada indivíduo da população quântica representa regiões promissoras do espaço de busca, ao invés de um mapeamento pontual, o que acelera a sua convergência.

No que diz respeito à precisão, o potencial dos modelos com a otimização automática de parâmetros e seleção de atributos mostraram ser consistentes, produzindo bons resultados com menos atributos de entrada. Estas abordagens se mostraram extremamente útil para a análise e modelagem de dados complexos.

6.1

Trabalhos Futuros

Estudar a possibilidade de implementar este trabalho como um aplicativo para a detecção de problemas de envelhecimento vocal, sendo possível captar um sinal de voz para avaliar a idade vocal do orador.

Sobre especificamente os parâmetros extraídos do sinal vocal, testar outros métodos de extração de parâmetros, adicionando a identificação de sexo dos locutores na formação dos grupos, com todas as vogais sustentadas e concatenadas.

Fazer uma análise de sensibilidade dos parâmetros que configuram a SNN, de forma a detectar quais parâmetros são relevantes para serem otimizados. Um grande número de parâmetros otimizados obtém um custo operacional maior.

Acredito como trabalho interessante, analisar os resultados variando os valores dos parâmetros dos algoritmos evolutivos. Os parâmetros foram usados igualmente para todas as bases de dados, sendo: o AG clássico valores *default*,

e o AEIQ-BR e OGP valores com base nas recomendações de (Pinho09) e (Koshiyama13) respectivamente. É possível que se tenha valores diferentes de configuração dos algoritmos para diferentes bases de dados.

Outra sugestão é utilizar esse tipo de redes para aprendizado *online*, onde a tarefa que se deseja aprender é não estacionária, ou seja, os dados de entrada não são estáticos, e podem variar ao longo do tempo. Segundo Cruz (Cruz07), usar algoritmos evolutivos para aprendizado online é usar as melhores soluções, encontradas até o momento anterior a mudança de configuração do problema, para alimentar a população inicial que será usada durante o aprendizado da nova configuração do problema. Classificadores treinados com dados passados podem se tornar obsoletos, necessitando um ajuste nos parâmetros dos modelos tomando como base amostras passadas e novas.

SNNs apresentam-se capazes de fornecer novas soluções computacionais para o processamento eficiente para aspectos temporais de dados. Características temporais são presentes em quase todos os dados do mundo real e processamento humano ou animal. Pode-se também incluir nos próximos passos trabalhos direcionados para:

- Percepção visual de objetos em movimento.
- Percepção auditiva de fonemas e sílabas.
- Percepção olfativa de odores.

Na percepção visual de objetos em movimento, os modelos podem ser modificados para classificação de padrões de imagem, de forma a executar diferentes ações dependendo da imagem capturada.

7

Referências Bibliográficas

[Adibi05] Adibi, P.; Meybodi, M.R.; Safabakhsh, R.; **Unsupervised learning of synaptic delays based on learning automata in a RBF-like network of spiking neurons for data clustering**. Neurocomputing - Elsevier, Vol.64, pp.335-357, 2005.

[Aparat05] <http://www.acoustics.hut.fi/software/>.

[Alexandre06] Simões, A. S.; **Aprendizado não supervisionado em redes neurais da base radial**. Tese de Doutorado, Escola Politécnica da Universidade de São Paulo, São Paulo, 2006.

[Antonio04] Timoszczuk, A. P.; **Reconhecimento Automático do Locutor com Redes Neurais Pulsadas**. Tese de Doutorado, Escola Politécnica da Universidade de São Paulo, São Paulo, 2004.

[Bodyanskiy14] Bodyanskiy, Y.; Dolotov, A.; Vynokurova, O.; **Evolving spiking wavelet-neuro-fuzzy self-learning system**. Applied Soft Computing, Vol.14, pp.252-258, 2014.

[Bohte03] Bohte, S.M.; **Spiking Neural Networks**. PhD thesis, Institute for Programming Research and Algorithmic.Center for Mathematics and Computer Science, Amsterdam, 2003.

[Booi05] Booi, O.; Nguyen, H.; **A gradient descent rule for spiking neurons emitting multiple spikes**. Information Processing Letters – Journal, Vol.95, pp.552-558, 2005.

[Chandra12] Chandra, R.; Zhang, M.; **Cooperative coevolution of Elman recurrent neural networks for chaotic time series prediction**. Neurocomputing, Vol.86, pp.116-123, 2012.

[Cluster] **Clustering datasets**. <http://cs.joensuu.fi/sipu/datasets/>.

[Cruz07] Cruz, A. V. Abs.; **Algoritmos Evolutivos com Inspiração Quântica para Problemas com Representação Numérica**. Tese de Doutorado, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2007.

[Derrac11] Derrac, J.; Garcia, S.; Molina, D.; Herrera, F.; **A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms**. Swarm and Evolutionary Computation, Vol.1, pp.3-18, 2011.

[Everitt01] Everitt, B. S.; Landau S.; Leese, M.; **Cluster Analysis**. 4th Edition, Oxford University Press, Inc. New York, 2001.

[Gerstner99] Gerstner, W.; **Pulsed Neural Networks**. pp.3–48, MIT Press, Cambridge, MA, USA, 1999.

[Gerstner96] Gerstner, W.; Kistler, W.; **A Neuronal Learning Rule for Sub-millisecond Temporal Coding**. Nature, N.384, pp.76–78, 1996.

[Gerstner02] Gerstner, W.; Kistler, W.; **Spiking Neuron Models: Single Neurons, Populations, Plasticity**. Cambridge University Press, New York, USA, 2002.

[Han00] Han, K.; Kin, J.; **Quantum Algorithm and its Application to Combinatorial Optimization Problem**. In Proceedings of the 2000 Congress on Evolutionary Computation, pp. 1354–1360, IEEE, 2000.

[Han02] Han, K.; Kin, J.; **Quantum-Inspired Evolutionary Algorithm for a Class of Combinatorial Optimization**. IEEE Transaction on Neural Networks, Vol.6, N.6, pp. 580–593, 2002.

[Hamed09] Hamed, H. N. A.; Kasabov, N.; Shamsuddin, S.M.; **Integrated Feature Selection and Parameter Optimization for Envolving Spiking Neural Networks using Quantum Inspired Particle Swarm Optimization**. IEEE Computer Society, International Conference of Soft Computing and Pattern Recognition, IEEE Press, pp.695-698, 2009.

[Hamed11] Hamed, H. N. A.; Kasabov, N.; Shamsuddin, S.M.; **Quantum-inspired particle swarm optimization for feature selection and parameters optimization in evolving spiking neural networks for classification tasks.** Numerical Analysis and Scientific Computing, pp.133-148, 2011.

[Henrich02] Henrich, N.; d'Alessandro C.; Doval B.; **Glottal flow models: waveforms, spectra and physical measurements.** Forum Acusticum, Sevilla, 2002.

[Henrich03] Henrich, N.; Roubeau B.; Castellengo M.; **On the use of electroglottography for characterization of the laryngeal mechanisms.** Proceedings of the Stockholm Music Acoustic Conference, Stockholm, Sweden, 2003.

[Henrich04] Henrich, N.; d'Alessandro C.; Castellengo M.; Doval B.; **On the use derivative of electroglottography signals for characterization of nonpathological phonation.** Acoustical Society of America, pp.1321-1332, 2004.

[Hélène06] Paugam-Moysy, H.; **Spiking Neuron Networks – A Survey.** IDIAP Research Institute – IDIAP-RR, Switzerland, 2006.

[Hopfield95] Hopfield, J.; **Pattern Recognition Computation using Action Potential Timing for Stimulus Representation.** Nature, Vol.376, pp. 33–36, 1995.

[Hinchliffe96] Hinchliffe M. P.; Willis M. J.; Hiden H.; Tham M. T.; McKay B.; Barton G. W.; **Modeling chemical process systems using a multi-gene genetic programming algorithm.** In Genetic Programming: Proceedings of the First Annual Conference. Massachusetts, MIT Press, pp.56-65, 1996.

[Izhikevich04] Izhikevich, E. M.; **Which model to use for cortical spiking neurons?.** IEEE Transaction on Neural Networks, Vol.15, N.5, 2004.

[Iannella05] Iannella, N.; Kindermann, L.; **Finding iterative roots with a spiking neural network**. Information Processing Letters - Elsevier, Vol.95, pp.545-551, 2005.

[Juliano08] Juliano, S. M.; **Um estudo comparativo entre o sinal electroglotográfico e o sinal de voz**. Dissertação de Mestrado em Engenharia de Telecomunicações, UFF, 2008.

[Kandel00] Kandel, E.R.; Schwartz, J.H.; Jessel T.M.; **Principles of Neural Sciences**. McGraw-Hill, 4th edition, 2000.

[Kanungo02] Kanungo, T.; Mount, D. M.; Ntenyahu, N. S.; Piatko, C. D.; Silverman, R.; Wu, A. Y.; **An Efficient k-Means Clustering Algorithm: Analysis and Implementation**. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.24, No.7, 2002.

[Kasabov10] Kasabov, N.; **To spike or not to spike: A probabilistic spiking neuron model**. Neural Networks – Journal, Vol.23, pp.16-19, 2010.

[Kasabov11] Kasabov, N.; Schliebs, R.; Kojima, H.; **Probabilistic Computational Neurogenetic Modeling: From Cognitive Systems to Alzheimer's Disease**. IEEE Transactions, Vol.3, pp.300-311, 2011.

[Kasabov12] Kasabov, N.; **NeuCube EvoSpike Architecture for Spatio-temporal Modelling and Pattern Recognition of Brain Signals**. ANNPR, Lecture Notes in Artificial Intelligence - Springer, pp.225-243, 2012.

[Kasabov13] Kasabov, N.; Dhoble, K.; Nuntalid, N.; Indiveri, G.; **Dynamic evolving spiking neural networks for on-line spatio and spectro-temporal pattern recognition**. Neural Networks, Vol.41, pp.188-201, 2013.

[Kasabov14] Kasabov, N.; **NeuCube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data**. Neural Networks, Vol.52, pp.62-76, 2014.

[Kaufman90] Kaufman, L.; Rousseeuw, P. J.; **Finding groups in data: An introduction to cluster analysis**. John Wiley&Sons, Inc., New York, 1990.

[Knoblauch05] Knoblauch, A.; **Neural associative memory for brain modeling and information retrieval**. Information Processing Letters - Elsevier, Vol.95, pp.537-544, 2005.

[Koza92] Koza, J. R.; **Genetic Programming: On the Programming of computers by Means of Natural Selection**. Cambridge, MA, USA, MIT Press, 1992.

[Koshiyama13] Koshiyama, A. S.; Dias, D. M.; Abs da Cruz, A. V.; Pacheco, M. A. C.; **Numerical Optimization by Multi-Genetic Programming**. GECCO, pp.145-146, 2013

[Kuncheva04] Kuncheva, L. I.; **Combining Pattern Classifiers: Methods and Algorithms**. Wiley-Interscience, 2004.

[Langdon10] Langdon, W. B.; Gustafson, S. M.; **Genetic Programming and Evolvable Machines**. Genetic programming and evolvable machines: ten years of reviews – journal, Vol.11, No.3, pp.321-338, 2010.

[Maass97] Maass, W.; **Networks of Spiking Neurons: The third generation of neural network models**. Elsevier Science, Vol.10, No.9, pp.1659–1671, 1997.

[Maass99] Maass, W. Bishop, C. M.; **Pulsed Neural Networks**. MIT Press, Cambridge, MA, USA, 1ª edição, 1999.

[Matlab10] **MATLABR2010a**. Mathworks, Natick, 2010.

[Mitchell92] Mitchell, M.; **An Introduction to Genetic Algorithms**. The MIT Press, USA, 1992.

[Mohammed12] Mohammed, A.; Schliebs, S.; Satoshi, M.; Kasabov, N.; **SPAN: Spike Pattern Association Neuron for learning spatio-temporal spike patterns**. International Journal of Neural Systems, Vol.22, 2012.

[Nat98] Natschlager, T.; Ruf, B.; **Spatial and Temporal Pattern Analysis via Spiking Neurons**. Network, Vol.9, N.3, pp.319-332, 1998.

[Nat01] Natschlager, T.; Maass, W.; Zador, A.; **Efficient Temporal Processing with Biologically Realistic Dynamic Synapses**. Network Computing Neural Syst. 12, pp. 75–87, 2001.

[Pinho09] Pinho, A. G.; Vellasco, M.; Abs da Cruz, A. V.; **A New Model for Credit Approval Problems: A Quantum-Inspired Neuro-Evolutionary Algorithm with Binary-Real Representation**. World Congress on Nature & Biologically Inspired Computing – NaBIC, pp.445-450, 2009.

[Poli08] Poli, R.; Langdon, W.B.; Mcphee, N. F.; **A Field Guide to Genetic Programming**. Rayleigh, 2008.

[Praat08] <http://www.fon.hum.uva.nl/praat/>.

[Pulaka05] Pulaka, H.; **Analysis of Human Voice Production Using Inverse Filtering High-Speed Imaging, and Electroglottography**. Helsinki University of Technology, Dept. of Computer Science and Engineering, Master's Thesis, 2005.

[Ruf98] Ruf, B.; **Computing and Learning with Spiking Neurons – Theory and Simulations**. PhD thesis, Institute for Theoretical Computer Science, Austria, 1998.

[Santos05] Santos, I.R.; **Análise Acústica da Voz de Indivíduos na Terceira Idade**. Dissertação de Mestrado, Universidade de São Carlos (USP), 2005.

[Schliebs09] Schliebs, S.;Platel, M.D.; Worner, S.; Kasabov, N.; **Quantum Inspired Feature and Parameter Optimization of Envolving Spiking Neural Networks with a Case Study from Ecological Modeling**. IEEE Computer Society, Proceedings of International Joint Conference on Neural Networks, USA,2009.

[Schliebs11] Schliebs, S.; Hamed, H. N. A.; Kasabov, N.; **A reservoir-based evolving spiking neural networks for on-line spatio-temporal pattern learning and recognition**. 18th International Conference on Neural Information Processing, No.7063, pp.160-168, Springer, Shanghai, China, 2011.

[Schliebs13] Schliebs, S.; Kasabov, N.; **Evolving spiking neural network – a survey**. Evolving Systems, Vol.4, pp.87-98, Springer, 2013.

[Searson07] Searson D. P.; Willis M. J.; Montague G. A.; **Co-evolution of non-linear PLS model components**. Journal of Chemometrics, Vol.2, pp.592-603, 2007.

[Silva10] Silva, M.A.B.; **Uma contribuição para a caracterização do sinal de voz envelhecida**. Dissertação de Mestrado em Engenharia de Telecomunicações, Universidade Federal Fluminense (UFF), 2010.

[Soltic08] Soltic, S.; Wysoski, S.; Kasabov, N.; **Evolving Spiking Neural Networks for taste recognition**. IEEE World Congress on Computational Intelligent (WCCI), Hon Kong, 2008.

[Stanley02] Stanley, K. O.; Miikkulainen, R.; **Evolving neural networks through augmenting topologies**. Evolutionary Computation, Vol.10, pp.99-127, 2002.

[Thorpe01] Thorpe, S.; Delorme, A.; Rullen, R.; **Spike-based strategies for rapid processing**. Neural Networks, Vol.14, pp. 715–725, 2001.

[Titze94] Titze, I.R.; **Principles of Voice Production**. Prentice-Hall, Englewood Cliffs, New Jersey, 1994.

[UCI] **UCI Machine Learning Repository**. <http://archive.ics.uci.edu/ml>.

[Verstraeten05] Verstraeten, D.; Schrauwen, B.; Stroobandt, D.; **Isolated word recognition using a Liquid State Machine: a case study**. Information Processing Letters - Elsevier, Vol.95, pp.521-528, 2005.

[Wysoski08] Wysoski, S. G.; Benuskova, L. Kasabov, N.; **Fast and adaptive network of spiking neurons for multi-view visual pattern recognition**. Neurocomputing - Elsevier, Vol.71, pp.2563-2575, 2008.

[Wysoski10] Wysoski, S. G.; Benuskova, L. Kasabov, N.; **Evolving spiking neural networks for audiovisual information processing**. Neural Networks, Vol.23, pp.819-835, 2010.

[Zhang09] Zhang, G.; **Quantum-inspired Evolutionary algorithms: a survey and empirical study**. Springer Science Business Media - LLC, 2009.

8

Apêndice A

Modelo Izhikevich

O modelo apresentado por Izhikevich reproduz de forma simples o comportamento de neurônios corticais, utilizando a plausibilidade biológica do modelo Hodgkin-Huxley, mas de fácil implementação (Izhikevich04). Os modelos a seguir utilizam um simples pulso DC como estímulo de entrada, e as respostas dos neurônios podem ser visualizados na figura 2.6.

- **Tonic Spiking:** o neurônio dispara um trem de pulsos enquanto ocorrer o pulso na entrada.
- **Phasic Spiking:** o neurônio dispara somente no momento do pulso.
- **Tonic Bursting:** o neurônio dispara uma rajada de pulsos durante o pulso na entrada.
- **Phasic Bursting:** o neurônio dispara somente uma rajada de pulsos durante o pulso na entrada.
- **Mixed Model:** o neurônio dispara somente uma rajada de pulsos e depois um trem de pulsos durante o pulso na entrada.
- **Spike Frequency Adaptation:** o neurônio dispara um trem de pulsos com uma diminuição na frequência dos pulsos de saída, enquanto ocorrer o pulso na entrada.
- **Class 1 Excitability:** o neurônio dispara um trem de pulsos cuja frequência depende do valor do pulso de entrada.
- **Class 2 Excitability:** o neurônio dispara um trem de pulsos de baixa frequência e quando o pulso de entrada aumenta, a frequência dos pulsos de saída aumenta.
- **Spike Latency:** o neurônio dispara com um atraso que depende do valor do pulso de entrada.
- **Subthreshold Oscillations:** o neurônio apresenta potencial de oscilação.

- ***Frequency Preference and Resonance***: devido ao fenômeno da ressonância, um neurônio com potenciais oscilatórios pode responder seletivamente às entradas com conteúdo de frequência semelhante à frequência das oscilações subliminares. Esses neurônios são chamados ressonadores
- ***Integration and Coincidence Detection***: o neurônio sem os potenciais oscilatórios atua como um integrador. Quanto maior a frequência, maior a probabilidade de disparo.
- ***Rebound Spike***: o neurônio dispara um pós-inibitório quando ele recebe o pulso e, em seguida, é libertada a partir de uma entrada inibitória.
- ***Rebound Burst***: o neurônio dispara uma rajada de pulsos.
- ***Threshold Variability***: o neurônio tem um limiar variável que depende de uma atividade anterior. Só dispara quando uma entrada inibitória é aplicada imediatamente antes do estímulo excitatório.
- ***Bi-stability of Resting and Spiking States***: o neurônio apresenta dois modos estáveis de operação. Os modos são ligados por um pulso excitatório ou inibitório.
- ***Depolarizing After-Potentials***: após o disparo do pulso, o potencial da membrana pode apresentar um potencial prolongado.
- ***Accommodation***: o neurônio perde sua sensibilidade a um estímulo forte, sendo habilitado a disparar com um aumento lento (rampa) disparando com um estímulo fraco.
- ***Inhibition-Induced Spiking***: o neurônio dispara quando ocorre uma entrada inibitória constante, disparando durante o pulso na entrada.
- ***Inhibition-Induced Bursting***: idêntico ao caso anterior, o neurônio dispara rajadas de pulsos a uma entrada constante.

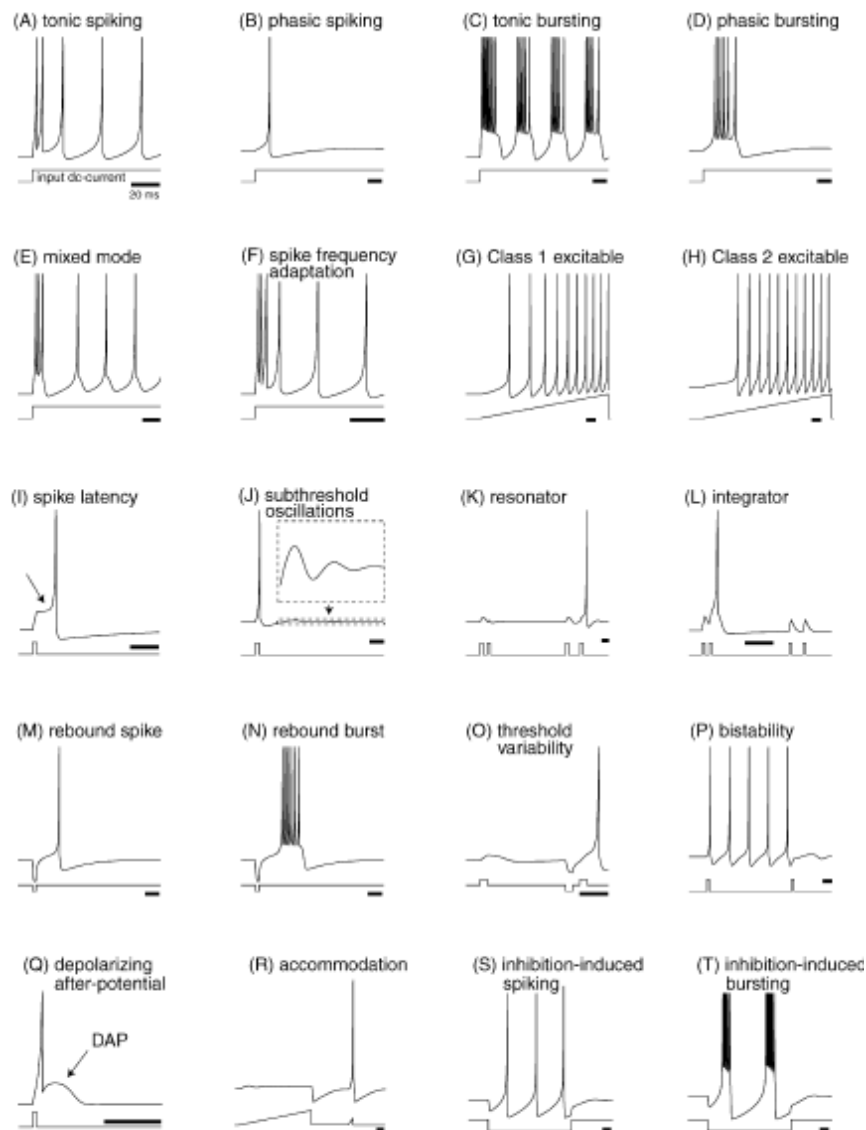


Figura 8.1: Modelo de neurônios. Fonte (Izhikevich04).

Programa Aparat

A plataforma principal de desenvolvimento é o Linux. A instalação em ambiente Matlab® necessita do Matsig, que consiste em uma biblioteca de processamento de sinais que têm que ser instalado separadamente. A partir da versão 0.3 o pacote de instalação do *Aparat* já contém o *Mastig*.

O Aparat pode ser executado fora do ambiente do Matlab®, bastando ter uma pasta com todos os arquivos do pacote de instalação, e criar um executável.

Os diferentes parâmetros na base de tempo são demonstrados graficamente na janela de *signal view*, parâmetros na base de frequência e

modelo LF. Os parâmetros de tempo e frequência serão descritos de forma mais detalhada nas seções *glottalttimeparams* e *glottalfreqparams*. O LF-model simula a forma de onda ideal dentro do espectro selecionado. Esta função tem uma demanda computacional grande e o tempo de execução depende do computador utilizado (Aparat 2005).

Spectrum view - Espectro do sinal

Esta janela mostra o espectro do sinal. É possível escolher qual espectro deseja visualizar, e cada um é designado com uma curva e cor diferente.

Na figura 7.1 o espectro do locutor 1:

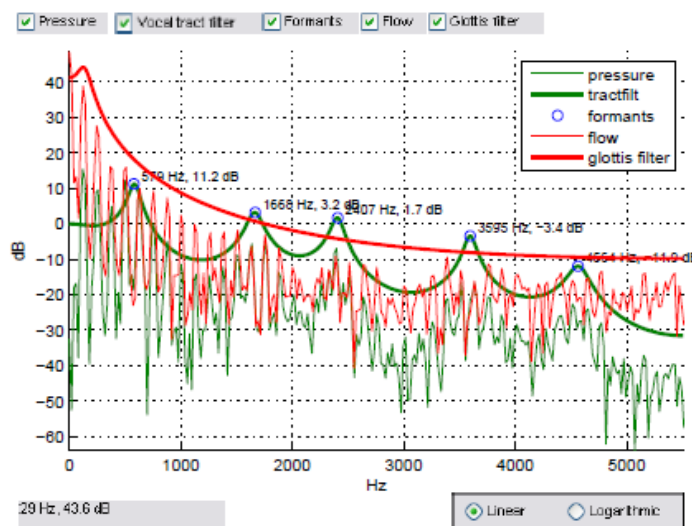


Figura 8.2: Espectro

Podemos ainda visualizar a curva linear ou logarítmica. As linhas finas: verde e vermelha ilustram a pressão do sinal e o fluxo glotal, e as linhas grossas: verde e vermelha o filtro do trato vocal e o filtro da glote.

Os picos do trato vocal também são mostrados como estimativas de formantes. Aquisição de dados ou gravação dos sinais deve ser feitos separadamente. Certos cuidados que se deve ter na captação:

- É importante a utilização de microfone omnidirecional, evitando os microfones dinâmicos e os microfones cardioides que têm características não lineares de fase.
- Todos os filtros no pré-amplificador devem ser desligados.
- Verificar que o dispositivo de gravação tem uma grandeza linear e resposta de frequência de fase ao longo da banda de voz.

- Não utilizar gravadores de MP3 ou minidiscos para a captação das vozes. Se possível gravar em câmaras anecóicas ou ambiente com pouca reverberação. Alta reverberação influencia na qualidade do sinal.
- Calibrar os equipamentos antes da gravação.
- Manter as mesmas condições durante o processo de gravação.

Outra forma de visualização é através de gráficos denominados: Plano-z, Fases do sinal e Trato vocal. A seguir as **Erro! Fonte de referência não encontrada.** 7.2, 7.3 e 7.4:

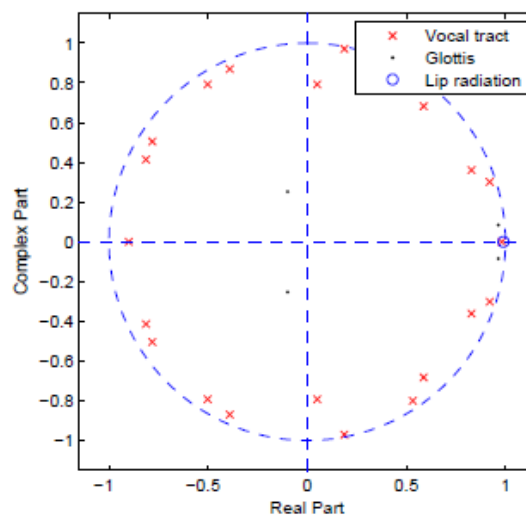


Figura 8.3: Plano z

O Plano-z visualiza os pólos do trato vocal e os filtros da glote. Caso haja pólos no eixo horizontal, estes podem ser retirados através do menu da janela principal (Aparat 2005).

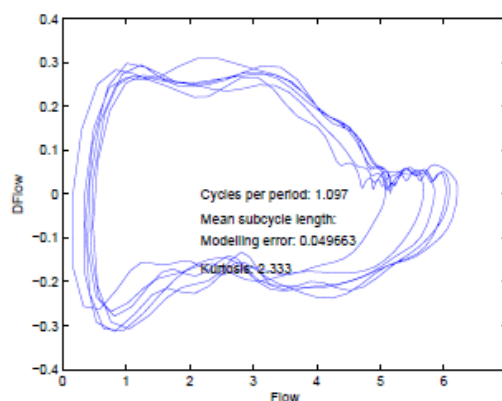


Figura 8.4: Diagrama de fase

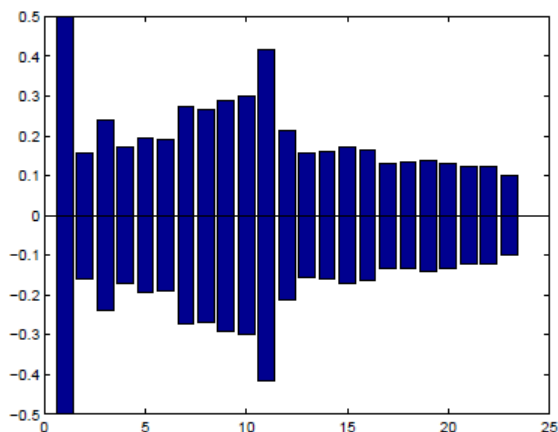


Figura 8.5: Trato vocal

O gráfico do Trato vocal visualiza a seção transversal do tubo do trato vocal. Neste gráfico, obtêm-se informações para avaliar a qualidade da filtragem inversa. Qualquer ressonância no fluxo glotal é mostrada no plano de fase assim com *loops* na trajetória são mostrados à direita da figura.

laif.m

Utilizando o IAIF, extraímos diversos parâmetros relacionados à filtragem inversa do sinal.

Glottalfreqparams.m

Os parâmetros à base de frequências podem ser vistos em um arquivo. A análise consta de múltiplos parâmetros relacionados ao fluxo glotal.

Glottaltimparams.m

Múltiplos parâmetros referentes à base do tempo são vistos neste arquivo, e todos relacionados ao fluxo glotal.

Programa Praat

Ao iniciar o software 2 janelas principais são mostradas e informações importantes são visualizadas:

- Objetos – encontram-se análises de sianis e sons. Todas as funções de gravação, reprodução, audição, desenho e edição também se encontram nessa janela;
- Figura – encontram-se desenhos para edição e a adição de textos. É o único lugar que permite a impressão dos gráficos.

Após a escolha do tipo de áudio a ser gravado, é necessário escolher a taxa de amostragem e efetuar a gravação. O nível do sinal gravado pode ser ajustado e é acompanhado pelo medidor de nível, evitando distorção na gravação.

O Praat oferece uma ferramenta muito flexível para visualizar, reproduzir e extrair informações de um arquivo de áudio. A Figura 7.5 ilustra a janela do programa.

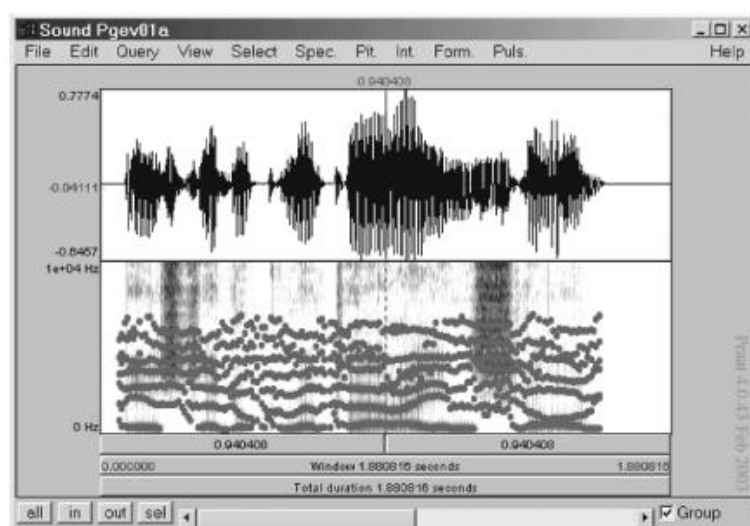


Figura 8.6: Programa *Praat*.