



UNIVERSIDADE ESTADUAL DE SANTA CRUZ
PRO-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM COMPUTACIONAL
EM CIÊNCIA E TECNOLOGIA

TADEU NOGUEIRA COSTA DE ANDRADE

RECONHECIMENTO E CLASSIFICAÇÃO DE PADRÕES EM AÇÕES COLETIVAS NO
ESPORTE

ILHÉUS-BA
2016

TADEU NOGUEIRA COSTA DE ANDRADE

**RECONHECIMENTO E CLASSIFICAÇÃO DE PADRÕES EM
AÇÕES COLETIVAS NO ESPORTE**

Dissertação apresentada a Universidade Estadual de Santa Cruz como parte das exigências para obtenção do Título de Mestre em Modelagem Computacional em Ciência e Tecnologia.

Orientador: Prof. Dr. Paulo Eduardo Ambrósio

ILHÉUS-BA
2016

A553

Andrade, Tadeu Nogueira Costa de.

Reconhecimento e classificação de padrões em ações coletivas no esporte / Tadeu Nogueira Costa de Andrade. – Ilhéus, BA: UESC, 2016.

59 f. : il.; anexos.

Orientador: Paulo Eduardo Ambrósio.

Dissertação (mestrado) – Universidade Estadual de Santa Cruz. Programa de Pós-graduação em Modelagem Computacional em Ciência e Tecnologia.

Inclui referências.

1. Redes neurais (Computação). 2. Sistemas de reconhecimento de padrões. 3. Esportes. I. Título.

CDD 006.32

TADEU NOGUEIRA COSTA DE ANDRADE

**RECONHECIMENTO E CLASSIFICAÇÃO DE PADRÕES EM
AÇÕES COLETIVAS NO ESPORTE**

Ilhéus-BA, 03/03/2016



Prof. Dr. Paulo Eduardo Ambrósio
UESC
(Orientador)



Prof. Dr. Francisco Bruno Souza Oliveira
UESC



Prof. Dr. Paulo Roberto Pereira Santiago
EEFERP/USP

*Dedico aos meus pais
Ivan e Taciara*

Agradecimentos

- Ao professor Paulo Ambrósio pela orientação e apoio em todas as etapas da pesquisa.
- Ao professor Marcos Menuchi e ao grupo GEPECOM pela coleta, análise e filtragem dos dados.
- Aos meus colegas que contribuíram para a boa convivência e compartilharam os conhecimentos adquiridos em sala de aula.
- A todos os professores do PPGMC que contribuíram direta ou indiretamente no andamento do trabalho.
- Aos meus pais, Ivan e Taciara, que sempre me apoiaram em todas as decisões.
- A minha esposa Verena e minha filha Alice pelo companheirismo, apoio e amor.
- A todas as pessoas que de alguma forma contribuíram em qualquer etapa da pesquisa.

"Não sabendo que era impossível, foi lá e fez"

Jean Cocteau

Resumo

O avanço tecnológico ocorrido nas últimas décadas tem influenciado diretamente na evolução em diversas áreas e campos de estudos. Na pedagogia do esporte não é diferente, e apresenta-se como uma das áreas com grande investimento computacional nos últimos anos, principalmente quando se deseja melhorar a organização e a aplicação no processo de ensino aprendizagem dos esportistas. A utilização de técnicas computacionais nessa área tem como prioridade estudar o comportamento dos atletas em atividades cotidianas ou analisar detalhadamente o contexto em situações particulares. Com os resultados oferecidos, torna-se muito mais simples modificar ou priorizar determinadas atividades que favoreçam a exploração de possíveis parâmetros auxiliares no desenvolvimento dos esportistas. As técnicas computacionais que oferecem a possibilidade de transformar as informações de entrada que se encontram em um nível não-significativo, como números e sinais, para um nível significativo ou de conhecimento destacam-se na Ciência do Esporte. Tais técnicas são denominadas de Reconhecimento e Classificação de Padrões, e tem como um de seus grandes expoentes, as Redes Neurais Artificiais. Na presente pesquisa é levantada a importância dessas técnicas para reconhecimento de padrões em diversas áreas, mas especificando principalmente sua utilização na Ciência do Esporte e sua importância no desenvolvimento e aprimoramento dos esportistas. O objetivo principal desse trabalho é analisar e modelar a dinâmica interpessoal dos atletas em situações presentes no cotidiano dos esportes coletivos, mais especificamente do futebol, verificando assim, parâmetros espaciais relevantes para uma análise comportamental detalhada. Para isso foram utilizados dados coletados de participantes de uma escola de formação de futebol do Vasco da Gama, localizada na cidade de Ilhéus Bahia, divididos em subcategorias de acordo com a faixa etária. O sistema computacional desenvolvido é baseado em Redes Neurais Artificiais do tipo Perceptron Multicamadas, utilizando o algoritmo de retro-propagação de sinais para treinamento e aprendizagem. Os resultados obtidos no mapeamento do contexto competitivo comprovam que não existe uma relação classificatória entre as diferentes subcategorias e existe uma relação classificatória entre as manipulações espaciais de espaçamento livre ou restrito. Analisando os pesos das variáveis discretas calculados pelo sistema computacional, é verificável que das seis variáveis utilizadas, duas não possuem influência direta no resultado, possibilitando suas retiradas em futuros mapeamentos.

Palavras-chave: Ciência do esporte. Reconhecimento e Classificação de Padrões. Redes Neurais Artificiais.

Abstract

The Technological advances in recent decades has directly influenced in the evolution progress in different areas and fields of study. In pedagogy of sport is not any different, and represent as an area with big investment in the computing last few years, especially when you want to improve the organization and the application in the process of teaching learning sports. The use of computer techniques in this area has as a priority to study the behavior of athletes in daily activities or detailed analysis of the context in particular situations. With the offered results, it becomes much simpler to modify or prioritize certain activities that encourage exploration of possible auxiliary parameters in development of sports. The computer techniques that offer the possibility of transforming the input information found in a non-significant level, as numbers and signs, to a significant degree or knowledge stand out in Sports Science. These techniques seek to associate this set studied an established pre class or finding a partition of the sample elements in groups with common characteristics. These are called techniques Recognition and Pattern Classification, and has as one of its greatest exponents, the Artificial Neural Networks. In this research are raised the importance of technical computing for pattern recognition in several areas, but specifying especially their use in Sports Science and its importance in development and improvement in sports. The main purpose of this study is to analyze and modeling the interpersonal dynamics of athletes in present situations in everyday team sports, specifically football, thereby determining spatial parameters relevant to a detailed behavioral analysis. For this, they used data collected of participants from a school of Vasco da Gama football training, located in Ilhéus Bahia, divided into subcategories agreement to age. The computer system is developed based on Neural Networks Artificial Multilayer Perceptron type, using the back-propagation algorithm signals for training and learning. The results of the mapping the competitive environment show that there is no relation between qualifying the different subcategories, and there is a qualifying relationship between the manipulations space free or restricted spacing. Analyzing the weight of discrete variables calculated by the computer system, it is verifiable that the six variables used, two have no direct influence on the result, enabling their exclusions in future mappings.

Keywords: Artificial Neural Networks. Sports Science. Recognition and Pattern Classification.

Lista de figuras

Figura 1 – Relações formadas com a movimentação dos jogadores no bobinho, MENUCHI (et al 2014)	7
Figura 2 – Representação gráfica PCA. PC1 e PC2 - Componente Principal 1 e 2 respectivamente (PANERO <i>et al.</i> , 2009)	12
Figura 3 – Representação treinamento e classificação (Adaptado JAIN A.K, <i>et al.</i> , 2000)	13
Figura 4 – Neurônio humano e sua composição.	20
Figura 5 – Neurônio matemático e sua composição (HAYKIN, 1998).	21
Figura 6 – RNA com uma camada de entrada e uma camada de saída (HAYKIN, 1998).	23
Figura 7 – RNA com uma camada oculta (HAYKIN, 1998).	23
Figura 8 – RNA com realimentação (HAYKIN, 1998).	24
Figura 9 – Rede Perceptron de camada única (HAYKIN, 1998).	25
Figura 10 – Modelo clássico MLP com sinais de entrada x_n e sinais de saída y_n . (HAYKIN, 1998)	26
Figura 11 – Organização inicial do bobinho. Quatro passadores dispostos ao redor de um marcador	33
Figura 12 – Ilustração do ângulo de passe (MENUCHI, 2014).	35
Figura 13 – Ilustração da técnica de <i>Cross Validation</i>	41

Lista de tabelas

Tabela 1 – Matriz de confusão de 3 classes G1, G2 e G3 com 100% de acerto . . .	41
Tabela 2 – Matriz de confusão de 3 classes G1, G2 e G3 com erros de classificação	41
Tabela 3 – Relação entre modelos, variáveis e resultados	42
Tabela 4 – Matriz de confusão para classificação por espaçamento	42
Tabela 5 – Relação dos pesos sinápticos	43

Lista de quadros

Quadro 1 – Algoritmo de Retro-propagação.	30
---	----

Lista de abreviaturas e siglas

UESC	Universidade Estadual de Santa Cruz
DCET	Departamento de Ciências Exatas e Tecnológicas
PPGMC	Programa de Pós-Graduação em Modelagem Computacional em Ciência e Tecnologia
RNA	Redes Neurais Artificiais
PCA	Análise de Componentes Principais
KPCA	Kernel Análise de Componentes Principais
SOM	Mapas Auto Organizáveis
XOR	Ou exclusivo ou disjunção exclusiva
MLP	Perceptrons Multicamadas
TALE	Termo de Assentimento Livre e Esclarecido
TCLE	Termo de Consentimento Livre e Esclarecido
GEPECOM	Grupo de Estudos e Pesquisa em Comportamento Motor da Universidade Estadual de Santa Cruz
DI	Distância Interpessoal
ELPL	Espaçamento Livre Passe Livre
ELPR	Espaçamento Livre Passe Restrito
ERPL	Espaçamento Restrito Passe Livre
ERPR	Espaçamento Restrito Passe Restrito
DIPM	Distância Interpessoal entre o Passador e o Marcador
DIPR	Distância Interpessoal entre o Passador e o Receptor
DIMR	Distância Interpessoal entre o Marcador e o Receptor
AP	Ângulo de Passe
VMP	Velocidade Média do Passe

Sumário

Lista de quadros	xi
1 – INTRODUÇÃO	1
2 – A CIÊNCIA DO ESPORTE	5
2.1 O Estudo da coordenação interpessoal do esporte	6
2.2 O protocolo experimental do bobinho	7
2.3 Variáveis de análise	8
3 – RECONHECIMENTO E CLASSIFICAÇÃO DE PADRÕES	9
3.1 A extração e seleção das características	10
3.2 Técnicas para o reconhecimento de padrões	12
3.2.1 Técnica do casamento de modelo	12
3.2.2 Técnica estatística para reconhecimento de padrões	13
3.2.3 Redes Neurais Artificiais	14
3.3 Os tipos de classificadores de padrões	15
3.3.1 Classificadores Probabilísticos	16
3.3.2 Classificadores baseado em exemplos	16
3.3.3 Classificadores de Kernel	16
3.3.4 Classificadores de árvore de decisão	17
4 – REDES NEURAIS	18
4.1 Introdução as Redes Neurais	18
4.2 Os princípios da neuro computação	20
4.3 As Redes Neurais Artificiais	22
4.4 As Redes Perceptrons	25
4.5 Aprendizagem - O algoritmo de retro-propagação	27
5 – PROCEDIMENTO E MÉTODOS	31
5.1 Procedimento de coleta de dados	32
5.2 Procedimento fotogramétricos	33
5.3 Variáveis analisadas	34
5.3.1 A Rotina para o cálculo das variáveis	35
5.4 O Sistema computacional	36
5.4.1 Configuração	37
5.4.2 Implementação	38
5.4.3 Treinamento e aprendizagem	40

5.4.3.1	<i>Cross Validation</i> - Validação Cruzada	40
5.4.3.2	Matriz de Confusão	40
5.4.4	Resultados obtidos	41
6	Conclusão	45
6.1	Trabalhos Futuros	45
7	REFERÊNCIAS	47
	 Anexos	 51
	ANEXO A – Rotina para cálculo das variáveis	52

1 INTRODUÇÃO

O processo de ensino-aprendizagem-treinamento dos esportes vem sendo alvo de muitas investigações na Ciência do Esporte. Com relação ao campo pedagógico, a Pedagogia do Esporte apresenta-se como uma das disciplinas que vem investindo em estudos referentes à organização, sistematização, aplicação e avaliação do processo de ensino e aprendizagem do esporte (GALATTI *et al.*, 2008).

Pesquisadores adeptos da abordagem dos sistemas dinâmicos e ecológicos vêm fortalecendo alguns princípios importantes para a compreensão, fundamentação e organização do ambiente de aprendizagem dos esportes, com implicações práticas para a intervenção profissional (MENUCHI *et al.*, 2014). A ideia principal nestas abordagens é que o esporte deve ser analisado com base em suas estruturas funcionais, ou seja, as ações do esportista não podem ser pensadas separadamente do contexto em que são realizadas (ARAÚJO *et al.*, 2006). Segundo os autores, o aprendiz estabelece relações funcionais com o meio que podem ser descritas e analisadas como um sistema dinâmico complexo e auto-organizado. Assim, diferentemente das abordagens cognitivistas tradicionais, entender os fatores que influenciam e delineiam o desempenho esportivo torna-se fundamental.

Para entender estes fatores, a análise dos comportamentos exibidos em uma atividade ou jogo só fazem sentido se o contexto em que este comportamento está sendo observado fizer parte da análise (MENUCHI, 2014). Segundo Davids *et al* (2013), o aprendiz exhibe comportamentos para atender a objetivos específicos no contexto de ação, estabelecendo uma relação funcional balizada pelas propriedades do envolvimento, isto é, na relação entre as capacidades do executante e as restrições no contexto de ação.

Segundo Menuchi (2014), não adianta o aprendiz ficar praticando habilidades técnicas de maneira isolada e fragmentada, pois o comportamento que ele exhibirá durante o jogo não dependerá apenas da capacidade de execução, mas sim do contexto de ação, como um adversário em marcação, oportunidades para o passe ou chute, velocidade de deslocamento, proximidade com os companheiros de equipe, etc. Assim, diante das diversas possibilidades de relações e variáveis envolvidas, o atleta busca por oportunidades para agir em função de seus objetivos. Ou seja, o atleta toma decisões em função das oportunidades que percebe no contexto de ação (PASSOS *et al.*, 2006).

Os Estudos realizados por Duarte (*et al.*, 2010) e Menuchi (2014) vêm explorando a distância interpessoal (distância entre dois ou mais jogadores) associada à velocidade relativa (velocidade de cada jogador). Os autores têm defendido que estes parâmetros são subjacentes à tomada de decisão dos jogadores em subfases do jogo de futebol.

Contudo, estes estudos analisaram díades atacante-defensor durante o drible. Em Duarte *et al.* (2010), as situações de drible foram simuladas e os participantes foram orientados ao confronto direto. Em situações reais de jogo, pode-se verificar nas literaturas consultadas situações de confronto (drible) em jogo recreativo de estudantes universitários (MENUCHI, 2014) bem como jogos de competições profissionais da primeira divisão da liga Portuguesa.

A utilização de técnicas computacionais para estudos comportamentais de atletas em atividades cotidianas do esporte é de grande valia quando se deseja analisar detalhadamente o contexto de uma determinada situação. Nos dias atuais, muito clubes e federações do esporte utilizam de sistemas que buscam de alguma forma modelar o contexto esportivo para se obter o máximo de seus atletas.

Dentre as diversas técnicas disponíveis, as que oferecem a possibilidade de transformar as informações de entrada que se encontram em um nível não-significativo, como números e sinais, para um nível significativo ou de conhecimento, são as que mais se destacam. Tal ciência compreende todo o estudo envolvido na tentativa de associar os elementos de um determinado conjunto em um outro conjunto de rótulos, fornecendo uma identidade classificatória. Esta atividade consiste em associar tal conjunto estudado a uma classe preestabelecida ou em encontrar uma partição dessa amostra em grupos de elementos com características comuns, na qual denomina-se respectivamente em classificação e clusterização de padrões. Segundo Wangenheim *et al* (2006), um sistema de reconhecimento de padrões deve ser capaz de aprender as características e a distribuição dos padrões no espaço vetorial e associar um determinado padrão a classe no qual ele pertence.

O interesse na área de reconhecimento de padrões vem de grande crescimento nas últimas décadas devido a quantidade de aplicações que, além de serem cada vez mais desafiadoras, são computacionalmente mais exigentes que as demais. Com o avanço e a diversidade de recursos computacionais para tal, existe diversas abordagens para realizar um reconhecimento e classificação. Basicamente todo o processo é resumido em quatro etapas: A aquisição dos dados, a análise das informações, o processamento das características e a classificação.

O objetivo principal deste trabalho foi analisar e modelar a dinâmica interpessoal dos atletas em situações presentes no cotidiano dos esportes coletivos, mais especificamente o futebol, verificando assim, parâmetros espaciais relevantes para uma análise comportamental detalhada. Para isso utilizou-se as Redes Neurais Artificiais. Tal técnica tem sido recentemente representada na Ciência do Esporte, abrindo novos caminhos para o entendimento da dinâmica das interações interpessoais que contribuem para a organização e função de diferentes modalidades esportivas.

O desenvolvimento do trabalho foi iniciado com o estudo do experimento

utilizado para simular o contexto esportivo, os parâmetros que deveriam ser considerados e quais variáveis seriam utilizadas para a modelagem computacional de acordo com estudos realizados anteriormente.

O desenvolvimento prático do trabalho dividiu-se em duas etapas: a) a criação de uma rotina capaz de calcular as variáveis de entrada de acordo com os dados utilizados; b) a implementação de um sistema capaz de modelar computacionalmente o contexto esportivo.

Os dados utilizados nesse estudo foram retirados do banco de dados do Grupo de Estudos e Pesquisa em Comportamento Motor (GEPECOM) da Universidade Estadual de Santa Cruz.

As Redes Neurais Artificiais desenvolvida na pesquisa foi do tipo Perceptron Multicamadas (MLP), que consiste de um modelo computacional formado por neurônios artificiais interligados entre si, associados a determinados pesos sinápticos. A utilização desse modelo de rede se deve ao fato das redes MLP serem utilizadas principalmente para classificação de padrões, segundo a própria literatura, com sinais de entrada e saída valorados.

Para o treinamento da rede utilizou-se o algoritmo de retropropagação de sinais (*backpropagation*¹). Tal algoritmo é baseado na análise do sinal de saída, comparando com um determinado erro médio quadrático, onde, caso o sinal de saída não satisfaça, ele é retropropagado para as camadas anteriores, ajustando os pesos sinápticos associados a cada neurônio. Antes de cada sinal ser propagado ele é submetido a uma função de ativação ou transferência, que irá converter o valor da soma de entrada para uma faixa de valores aceitáveis.

Essa dissertação tem como finalidade apresentar os resultados alcançados na pesquisa e todo o procedimento utilizado até a obtenção. Além disso, conceitos referentes a ciência do esporte, reconhecimento e classificação de padrões e redes neurais também são apresentados.

O Capítulo 2 é dedicado a apresentação dos principais conceitos da ciência do esporte e do protocolo experimental que servirão como base teórica para compreender a necessidade e os benefícios fornecidos ao modelar computacionalmente um contexto esportivo. Além disso é apresentado os principais parâmetros estudados atualmente que são capazes de descrever a coordenação global e local do contexto.

No Capítulo 3 são realizadas algumas definições referentes a Classificação e Reconhecimento de Padrões, as diversas áreas de possíveis aplicações, assim como as principais técnicas capazes de reconhecer e classificar padrões de entrada.

O Capítulo 4 é utilizado para apresentar as Redes Neurais Artificiais,

¹Realiza a correção do peso, camada por camada.

principalmente Perceptrons Multicamadas, e o algoritmo de retropropagação de sinais, utilizado para a aprendizagem e treinamento da rede.

No Capítulo 5 é demonstrado o desenvolvimento prático: As variáveis analisadas e o sistema computacional responsável pela modelagem, além da apresentação dos resultados obtidos.

O Capítulo 6 apresenta as conclusões e discussões quanto aos resultados e as propostas de trabalhos futuros.

2 A CIÊNCIA DO ESPORTE

Um fenômeno facilmente observado ao longo da formação de um atleta é sua capacidade de melhoria na produção de movimentos coordenados com pessoas e eventos no contexto competitivo. Tal melhoria está diretamente relacionada à prática na modalidade esportiva, conseguindo favorecer a aquisição e o refinamento das habilidades motoras do atleta. Como resultado, observa-se que as ações motoras dos atletas estão cada vez mais eficientes e harmoniosas no contexto competitivo (MENUCHI, 2014).

Um dos principais comportamentos observados em esportes coletivos é a capacidade dos atletas em interagir coordenadamente com o ambiente e melhorar o seu desempenho. Dessa forma, pesquisadores vem, ao longo do tempo, buscando entender a tomada de decisão, percepção e a ação no esporte como um fenômeno baseado na perspectiva ecológica do contexto apresentado (DAVIDS *et al.*, 1994; ARAÚJO e DAVIDS, 2009). Mantendo essa linha de estudo, a percepção e a tomada de decisão influenciam diretamente no contexto de ação (VILAR *et al.*, 2012), sendo portanto, diretamente dependente. No entanto, cada contexto da ação deve ser tratado individualmente, sendo então único, formado por um conjunto de restrições que relaciona com as características e a potencialidade de cada atleta, do conjunto de regras e da ferramenta do esporte em questão (NEWELL, 1986). Outro fator observado é que as relações estabelecidas no contexto são auto-organizáveis funcionalmente (TURVEY, 2007). Assim, é possível concluir que o atleta aprende a explorar e a adaptar a diferentes contextos apresentados ao longo de sua formação.

Em 1987, Kugler e Turvey utilizaram modelagem computacional para demonstrar que a ação de um atleta em um contexto competitivo é limitada por leis da termodinâmica e dinâmica não linear de sistemas complexos. Dessa forma, o deslocamento espaço-temporal dos atletas relacionados diretamente ao ambiente explorado passou a ser analisado com base na dinâmica da interação interpessoal, buscando identificar as restrições chaves para a percepção, decisão e ação (KUGLER e TURVEY, 1987; DAVIDS *et al.*, 1994; KELSO, 1995; TURVEY e SHAW, 1999; ARAÚJO *et al.*, 2006; McGARRY, 2009).

Alguns estudos têm demonstrado que o drible ou outros padrões interpessoais em subfases do jogo possuem dependência direta a movimentação, posicionamento e velocidade dos companheiros de equipe e dos adversários no contexto apresentado. Ainda, como estes padrões de coordenação interpessoais locais (isto é, observado nas subfases do jogo) podem influenciar o jogo como um todo, mantendo ou quebrando

sua estabilidade (MENUCHI, 2014).

2.1 O Estudo da coordenação interpessoal do esporte

Para McGarry (*et al.*, 2002), o comportamento interpessoal no esporte em equipe deve ser interpretado como um processo das relações espaço-temporais apresentadas pelos jogadores durante uma partida. Assim, tal comportamento deve ser analisado como um sistema dinâmico. Ou seja, as várias relações estabelecidas podem ser formadas pelos jogadores adversários e construídas ou quebradas conforme o contexto apresentado, como a movimentação dos atletas ou da bola.

Baseado nessas definições, diferentes contextos competitivos estão sendo estudados, buscando encontrar os padrões explorados tanto defensivamente como ofensivamente durante toda uma atividade ou partida oficial. Para Davids (2005), o ritmo de um jogo de futebol é caracterizado por diversas interações em diferentes níveis, atingindo seu ápice através das ações dinâmicas interceptivas como passes, chutes e dribles. Seguindo tal conclusão, algumas pesquisas realizadas em subfases do jogo vem apresentando alto grau de potencialidade, demonstrando como os esportes coletivos são representados através de sistemas complexos, direcionados pelas interações entre os subsistemas presentes (GREHAIGNE *et al.*, 1997; DAVIDS *et al.*, 2005; HEADRICK *et al.*, 2012). Para Schmidt (1999), a relação estabelecida entre os atacantes e defensores formam um subsistema com sinergia diática, podendo apresentar propriedades não lineares. Segundo Oullier e Kelso (2006), essas propriedades levam a acreditar que tendências de coordenação parcial, não são nem completamente independente nem totalmente ligados em uma relação mutualmente fixada. Assim, a dinâmica encontrada em um contexto competitivo pode ser analisado tanto em uma escala local, em subfases do jogo, como em uma escala global, verificando o comportamento apresentado por todos os jogadores.

Utilizando como exemplo o futebol, durante um ataque, algumas relações dinâmicas entre os jogadores são apresentadas, podendo essa relação ser dentro da mesma equipe ou entre equipes adversárias. A equipe que está atacando tem como objetivo fazer o “gol”, e para isso é necessário quebrar a simetria apresentada pela equipe que está defendendo. Por outro lado, a equipe que está defendendo deseja manter a simetria, impedindo a progressão da equipe adversária e recuperando a posse da bola. Para que essas decisões possam ser tomadas, diversas informações presentes no contexto competitivo é analisada, como posições dos atletas, velocidade e movimentação. Em síntese, cada situação analisada é única e imprevisível. Assim, pequenas interações no curso de ação podem originar múltiplas variações nas oportunidades de ação que implica na necessidade de atualização permanente da informação que guia a interação

entre jogadores e o envolvimento. (ARAÚJO, *et al.*, 2006).

2.2 O protocolo experimental do bobinho

Bastante conhecido no mundo dos esportes em recreações e treinamentos profissionais, o bobinho é um pequeno jogo de futebol onde os participantes ficam distribuídos em forma de círculo, trocando passes entre si. Um dos participantes, conhecido pelo jogo como “bobinho”, fica localizado no centro do círculo e tem o objetivo de interceptar a troca de passes ou forçar o erro de um dos jogadores. Quando tal fato ocorre, o atual jogador “bobinho” passa a participar da troca de passes, e o jogador responsável pelo erro ou interceptação do adversário torna-se o novo jogador “bobinho”. Essas constantes movimentações de cooperação e competição podem ser visualizadas na Figura 1.

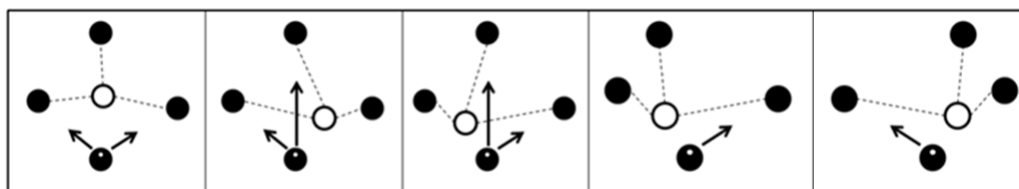


Figura 1 – Relações formadas com a movimentação dos jogadores no bobinho, MENUCHI (*et al* 2014)

É possível verificar na Figura 1 que as diversas possibilidades para o passe dependem da relação estabelecida entre o passador, o receptor e o “bobinho”. Embora as ações do passe possam ser analisadas de maneira periódica, referente à tomada de decisão de cada passador, observações cotidianas da atividade sugerem que as ações são constrangidas pela relação coletiva do posicionamento de todos os jogadores. Assim, uma modelagem adequada do bobinho, deve ser realizada de duas formas: local, contendo apenas os jogadores do contexto da ação, e global, com todos os jogadores presentes na atividade. Segundo Menuchi (*et al.*, 2014) o bobinho consegue representar diferentes estágios do processo de ensino-aprendizagem-treinamento do futebol, contribuindo para o entendimento da coordenação interpessoal na formação de linha de passe do contexto competitivo, com implicações diretas no andamento do jogo. Além disso, é um promissor protocolo experimental representativo de pesquisa para entender a relação funcional e adaptativa da exploração e ocupação de espaços para criar, manter e diluir oportunidades para o passe no contexto competitivo do futebol.

Analisando o sistema representado pelo bobinho, é possível verificar que:

- Os jogadores representam um alto grau de integração no contexto coletivo. As ações são diretamente dependentes do posicionamento e comportamento de todos

os jogadores envolvidos. Ou seja, as ações de um jogador dependem das ações do(s) outro(s) jogador(es).

- As interações interpessoais são capazes de produzir diferentes padrões de comportamento no jogo, tanto local como global.
- O jogo possui uma tendência inerente para a auto-organização, mantendo em função das regras envolvidas. Ou seja, os jogadores se organizam em função da dinâmica do jogo, e não de combinações preestabelecidas.
- Interações interpessoais podem representar respostas similares obtidas por meio de diferentes configurações de comportamento. Por exemplo, pode-se observar diferentes trajetórias de perseguição do marcador, mas mesmas configurações lineares e angulares com o passador no momento do passe.

2.3 Variáveis de análise

Diante do desafio de realizar uma modelagem adequada no experimento do bobinho, surgiu a necessidade de verificar e selecionar quais parâmetros que realmente conseguem descrever a coordenação global e local desse contexto esportivo.

Buscando uma análise da coordenação interpessoal entre o passador, o marcador e o receptor, os parâmetros de distância interpessoal, ângulo de passe e velocidade média do passe foram considerados. Isto justifica porque, de acordo Travassos (*et al.*, 2012), Correa (*et al.*, 2012) e Vilar (*et al.*, 2014), são potenciais parâmetros espaço-temporais que tem envolvimento direto com as tomadas de decisões.

Estudos de Correa (*et al.*, 2012) e Vilar (*et al.*, 2014) demonstram que a velocidade relativa aumenta quando a distância interpessoal assume valores pequenos. Além disso, o ângulo de passe é um potencial parâmetro que influencia consideravelmente a opção escolhida pelo passador. Já de acordo as pesquisas realizadas por Travassos (*et al.*, 2012), o desempenho do passe é constrangido pela convergência nos valores de distância interpessoal entre os jogadores que participam do contexto local. Em outro estudos, Travassos (*et al.*, 2012) consegue mostrar que no momento da inicialização do passe, as distâncias dos defensores à bola restringiram a possibilidade de interceptação.

Baseado nessas conclusões, as variáveis utilizadas para verificar a dinâmica de interação no protocolo experimental do bobinho foram a distância interpessoal entre todos os jogadores que participam da ação do passe, o ângulo de passe e a velocidade média do passe.

3 RECONHECIMENTO E CLASSIFICAÇÃO DE PADRÕES

O termo “Reconhecimento de Padrões” surgiu em meados da década de 60 e inicialmente era conhecido como “detecção de forma simples”. Segundo Duda e Hart (*et al.*, 2000), “O reconhecimento de padrões é um campo interessado no reconhecimento por máquinas de regularidades significativas em ambientes ruidosos ou complexos”. A área de reconhecimento de padrões tem como objetivo a classificação dos objetos em um número de categorias ou classes. Assim, dado um conjunto de n classes, e um padrão desconhecido x , um sistema para reconhecimento de padrões é um sistema que irá extrair e selecionar as características do sistema determinado, sendo auxiliado por sistemas externos de pré-processamento.

O interesse na área de reconhecimento de padrões tem crescido muito nos últimos anos devido as aplicações que estão cada vez mais exigentes computacionalmente. Entre as diversas áreas que se aplica o reconhecimento de padrões, segundo Kanso (1997), podemos listar:

- Comunicação homem-máquinas: Reconhecimento automático de máquinas, reconhecimento automático de fala, reconhecimento de escrita, compreensão de fala, compreensão de imagens e processamento de linguagem natural.
- Defesa: Reconhecimento automático de alvo, orientação e controle;
- Medicina: Diagnóstico médico, análise de imagens e classificação de doenças;
- Veículos: Controladores de automóveis, aviões, trens e barcos;
- Investigações: Detecção criminal a partir de fala, escrita manual, impressão digital e fotografias;
- Estudo e estimativa de recursos naturais: Agricultura, extrativismo, geologia e ambiente;
- Indústria: Testes de montagens de produtos, controle e inspeção de qualidade;
- Sistema doméstico: Utensílios inteligentes;
- Bioinformática: Análise de sequências do genoma;
- Mineração de dados: Busca por padrões significativos em espaços multidimensionais, normalmente obtidos de grande base de dados;
- Reconhecimento biométrico, incluindo faces, íris ou impressões digitais;

Na Ciência do Esporte é possível verificar a utilização de técnicas de

reconhecimento e classificação de padrões em estudos como predição de resultados, seleção de fatores de riscos de lesões, classificação qualitativa de jogadores e estudo da dinâmica interpessoal do esporte.

Para Connel S.D. & Jain A.K. (2000), existe duas maneiras de reconhecer e classificar padrões. O reconhecimento supervisionado, onde o padrão de entrada é identificado como um membro de uma classe pré-definida pelo projetista do sistema e o reconhecimento não supervisionado onde um padrão de entrada é determinada por classes ainda não definidas ou conhecidas. Em outras palavras, um problema de reconhecimento de padrões poderá conter classes que são definidas pelo projetista do sistema ou conter classes que serão aprendidas de acordo com a similaridade dos padrões.

3.1 A extração e seleção das características

A etapa de extração e seleção das características é considerada como a mais importante em todo o processo de reconhecimento de padrões, pois é aqui que as características preestabelecidas do fenômeno são extraídas em busca de um melhor desempenho computacional, sem prejudicar os resultados a serem obtidos. A seleção de características se refere a um processo no qual um espaço de dados é transformado em um espaço de características, de menor dimensão, mas que ainda retenha a maior parte da informação intrínseca dos dados. Em outras palavras, o conjunto de dados sofre uma redução de dimensionalidade. Os métodos de seleção de características tratam exatamente da escolha, dentre todos os atributos da base de dados, daqueles mais relevantes do ponto de vista da informação (MARDIA *et al.*, 1979; DASH *et al.*, 1997).

A escolha da características adequadas para a classificação é feita a partir da observação dos objetos e de suas respectivas classes, ou seja, do conjunto de treinamento. O conjunto de características deve descrever bem os objetos, agrupando os objetos similares e separando os objetos distintos. O sistema precisa ser robustos para tolerar as diferenças dentro de cada classe e ser sensível o bastante para discriminar essas classes. Na prática, raramente uma única característica é suficiente para distinguir duas ou mais classes, sendo geralmente utilizado um conjunto com várias características. Por outro lado a adição de características não significa sempre uma melhora no classificador.

Na prática, um extrator de característica irá determinar e extrair as características mais significativas que contribuam para a descrição do objeto, variando de acordo a classificação. Com a classificação estabelecida, o fenômeno analisado irá se adequar a sua determinada classe, de todas que foram apresentadas. Nessa etapa o classificador irá aprender as diferenças existentes, selecionando as características que mais se adequa a cada classe, construindo uma associação eficiente.

Quanto maior o número de características, mais complexo se torna o sistema, consumindo um alto tempo de execução para treinamento e reduzindo sua capacidade de generalização (THEODORIDIS, KOUTROUMBAS, 2005). Além de que, um grande número de características acaba gerando redundâncias de informações, prejudicando o desempenho do sistema. Infelizmente, para a extração da característica, não existe uma regra geral para saber quais características são relevantes para um problema. A escolha cuidadosa de quais características são importantes é de fundamental importância para evitar o processamento de informações redundantes.

A redução da dimensionalidade do espaço de características constitui um importante processo no reconhecimento de padrões. A seleção de característica irá diminuir a dimensão de características, de forma que se tenha um melhor desempenho na classificação. No entanto quando se tem um grande número de característica a redução se torna inviável. Neste caso deve-se buscar o desempenho através de conjuntos de características. Os conjuntos selecionados são conhecidos como conjuntos de características subótimos.

A busca para o conjunto subótimo a partir das características que sozinhas apresentam as melhores taxas de acertos não é uma garantia de bom resultados. Diversos métodos de extração e seleção das características são encontrados na literatura (KUDO *et al.*, 2000). A grande dificuldade na implementação de um projeto de reconhecimento de padrões está justamente na escolha da técnica adequada para a extração das características, para que com isso a próxima fase de classificação ocorra de forma satisfatória. Segundo Jain, A.K. *et al.*, (2000), um método de extração de característica determina um subespaço apropriado de dimensionalidade m , de uma maneira linear ou não linear no espaço de características original de dimensionalidade d , sendo $m \leq d$.

Dentre os métodos conhecidos para a extração de características lineares podemos destacar o método de Análise de Componentes Principais (PCA). O estudo das técnicas de PCA foi inicializado por Karl Pearson em 1901, mas foi apenas na década de 30 que tomou maior destaque, através dos trabalhos de Hotteling. Sua finalidade é analisar os dados, buscando uma redução através de combinações lineares das variáveis originais. É considerado um dos principais métodos para reconhecimento e classificação de padrões não supervisionado, utilizado principalmente para análise de similaridades e diferenças entre classes.

Com a redução dos dados por PCA, é possível descartar informações relevantes, ou seja, somente os dados que realmente são úteis para a classificação e construção do sistema serão mantidos. Como o PCA utiliza características mais expressivas, ele efetivamente aproxima os dados para um subespaço linear usando o critério do erro quadrático médio (PANERO *et al.*, 2009). A Figura 2 representa graficamente a análise por PCA em duas e três variáveis.

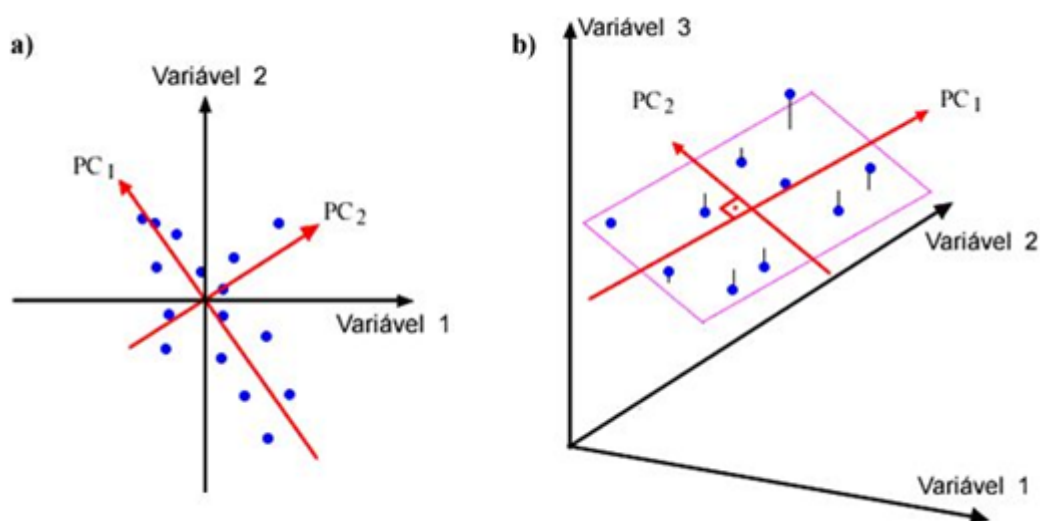


Figura 2 – Representação gráfica PCA. PC1 e PC2 - Componente Principal 1 e 2 respectivamente (PANERO *et al.*, 2009)

Para a extração de características não lineares existe um método que muito se assemelha ao PCA, conhecido como Kernel PCA. O kernel PCA é uma versão não linear do PCA. Esse método é calculado implicitamente através de uma função kernel para um espaço de maior dimensionalidade. O KPCA é capaz de extrair características não lineares em um conjunto de dados apresentados (SCHOLKOPF, KNIRCH, *et al.*, 1998). A principal característica desse método é o mapeamento dos dados de entrada dentro de um novo espaço através de funções não lineares. Uma vez mapeado os dados, o método PCA poderá então ser utilizado.

3.2 Técnicas para o reconhecimento de padrões

As técnicas para reconhecimento de padrões representam as ferramentas que são utilizadas para a extração das características ou de medidas das entidades representadas afim de se obter uma determinada classificação. As técnicas de reconhecimento de padrões se dividem basicamente em dois grandes grupos, as técnicas supervisionadas e as não supervisionadas. Nessa sessão serão apresentadas algumas técnicas utilizadas para o reconhecimento de padrões.

3.2.1 Técnica do casamento de modelo

Considerada uma das principais técnicas para o reconhecimento de padrões, a técnica do casamento de modelo tem como objetivo executar uma operação genérica em busca de uma similaridade entre duas entidades do mesmo tipo. O padrão a ser

reconhecido é comparado com os modelos armazenados, observando todas as variações possíveis em termo de translação, rotação e mudanças de escalas. Muitas vezes o modelo por si mesmo é aprendido a partir do conjunto de treinamento. Este método é computacionalmente exigente, mas a atual disponibilidade de recursos computacionais permite com que essa abordagem seja possível (JAIN, A.K. *et al.*, 2000).

Atualmente essa técnica é bastante utilizada entre os pesquisadores em problemas como:

- Determinar a presença de imagens ou objetos dentro de uma cena (CHOI, M.S & KIN, W.Y, 2000).
- Reconhecimento de caracteres (CONNEL, S.D & JAIN, A.K, 2000)

3.2.2 Técnica estatística para reconhecimento de padrões

Na abordagem estatística, cada padrão é representado em termos de características, medidas ou atributos e é visto como um ponto em um espaço d-dimensional. O objetivo é escolher determinadas características que permitam a vetores padrão, pertencentes a diferentes categorias, ocupar regiões compactas e disjuntas em um espaço de características d-dimensional (MARIN, 2001).

Dado um conjunto de padrões de treinamento, o objetivo é estabelecer decisões de fronteiras no espaço de características, aos quais os padrões separados pertenceriam a diferentes classes. Na abordagem teórica de decisão estatística, a decisão de fronteiras é determinada por distribuições de probabilidade de padrões pertencentes a cada classe, a qual deve ser específica ou aprendida. O sistema de reconhecimento é operado em dois modos: treinamento e classificação (JAIN, A.K. *et al.*, 2000).

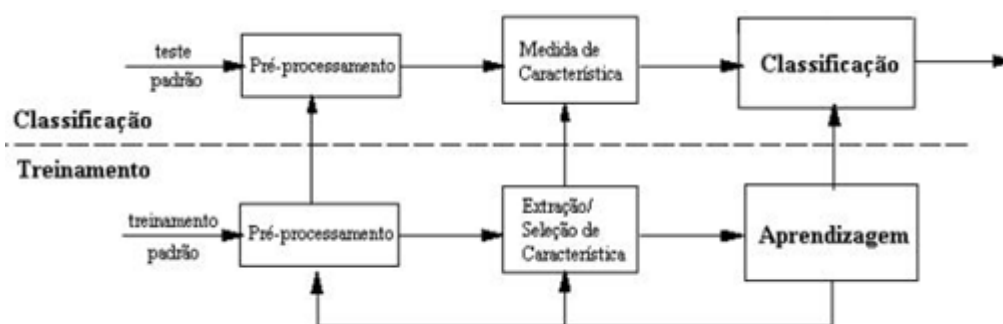


Figura 3 – Representação treinamento e classificação (Adaptado JAIN A.K, *et al.*, 2000)

A função do módulo de pré-processamento (Figura 3) é retirar o padrão de interesse de uma entidade, remover ruídos presentes e normalizar o padrão definido. No modo treinamento, o módulo de extração/seleção de características procura características apropriadas para a representação de padrões de entrada e o classificador

é treinado para particionar o espaço de características. O caminho de volta permite a um projetista otimizar o pré-processamento e estratégias de extração/seleção de características. No modo classificação, o classificador treinado determina o padrão de entrada para uma das classes de padrões sob considerações baseadas nas medidas das características.

Um dos conhecidos modelos estatísticos para reconhecimento de padrões é o modelo Bayesiano. O classificador Bayesiano ingênuo, chamado assim por assumir que as características são independentes entre si, apresenta resultados bastante competitivos em relação aos outros classificadores (FRIEDMAN *et al.*, 1997)

Considerando $p(\omega_i|x)$ como sendo a probabilidade de um dado padrão de características x pertencer à classe $\omega_i, i = 1, 2, \dots, c$, que pode ser definida pelo teorema de Bayes:

$$P(\omega_i|x) = \frac{\rho(x|\omega_i)P(\omega_i)}{\rho(x)} \quad (1)$$

A fórmula de Bayes indica que, analisando a probabilidade de $P(\omega_i)$ e após observar os valores de x , é possível converter a probabilidade a priori $P(\omega_i)$ para a probabilidade a posteriori $P(\omega_i|x)$. A probabilidade a posteriori $P(\omega_i|x)$ faz referência a probabilidade de uma amostra pertencer a classe $P(\omega_i)$, a partir de um vetor de característica x . O $\rho(x|\omega_i)$ é a probabilidade de $P(\omega_i)$ com respeito ao vetor x .

O produto da probabilidade e probabilidade a priori é o mais importante para se determinar a probabilidade a posteriori, uma vez que o fator $\rho(x)$ pode ser visto como apenas um fator para garantir que a soma das probabilidades a posteriori seja igual a 1 (DUDA *et al.*, 2000).

3.2.3 Redes Neurais Artificiais

Nessa sessão será feito uma introdução das Redes Neurais Artificiais (RNAs). Uma descrição mais detalhada será realizada no Capítulo 4. O estudo sobre as RNAs baseiam-se no reconhecimento que o cérebro humano realiza. No entanto, quando essa tarefa é feita computacionalmente, é realizada em um processo totalmente diferente. Segundo Haykin (1998), uma rede neural é um processador distribuído fortemente paralelizado, formado por unidades de processamento que armazena o conhecimento experimental e o torna disponível para uso. Além disso, a rede neural é semelhante ao cérebro em dois aspectos:

- O reconhecimento é adquirido pela rede por meio de um processo de aprendizagem.

- Os pesos de conexões entre neurônios, conhecido como pesos sinápticos, são utilizados para armazenar o conhecimento adquirido.

A rede perceptron foi a primeira rede neural descrita de forma algorítmica, inventada por Rosenblatt (ROSENBLATT, 1958). Em geral, existe três diferentes tipos de redes neurais perceptron (HAYKIN, 1998). Com uma camada sem retro-propagação, onde o sinal é programado em uma camada de neurônio de entrada até a camada de neurônio de saída. Multicamada sem retro-propagação, que difere da primeira por existir uma ou mais camada de propagação de sinal, conhecida como camada intermediária. E as recorrentes, mais utilizadas atualmente, que difere das duas anteriores por possuir um ou mais processo de retro-propagação do sinal de saída.

O procedimento de aprendizagem de uma rede neural perceptron do tipo recorrente consiste na alteração dos seus pesos sinápticos com o intuito de atingir um determinado valor considerado satisfatório como resultado da rede. Um dos algoritmos mais conhecidos no processo de aprendizagem é o algoritmo de *backpropagation*.

O aprendizado por algoritmo *backpropagation* consiste de duas fases entre as diferentes camadas da rede: a fase *forward* e a fase *backward*. Na fase *forward*, um vetor de entrada é aplicado aos neurônios de entrada da rede, e seu efeito é propagado pela rede, camada a camada, até o conjunto de saída ser gerado. Na fase *forward*, todos os pesos sinápticos dos neurônios da rede são fixados. Na fase *backward*, os pesos sinápticos são ajustados de acordo com a regra de correção de erros. A resposta da rede é subtraída da resposta esperada para se obter o sinal de erro, sendo que este é propagado da frente para trás da rede contra a direção das conexões sinápticas. Sendo assim, os pesos sinápticos são ajustados com o intuito de fazer com que a resposta obtida chegue o mais próximo possível da resposta desejada (HAYKIN, 1998).

Outro modelo de rede bastante conhecido é a tipo SOM (Self Organizing Maps - Mapas Auto Organizáveis). As redes SOM representam uma classe especial de redes neurais artificiais. Tais mapas são baseados em um aprendizado competitivo, no qual neurônios de saída da rede competem entre eles para serem ativados, sendo que apenas um neurônio, entre todos ou entre um grupo, é ativado em um determinado tempo. Este modelo é motivado pela característica do cérebro de mapear entradas sensoriais em diferentes áreas do cortex cerebral de uma maneira topologicamente ordenada (HAYKIN, 1998). Em um SOM, os neurônios são dispostos geralmente em uma ou duas dimensões. É possível também utilizar mapas de maior dimensão.

3.3 Os tipos de classificadores de padrões

Um classificador de padrões é uma ferramenta que irá classificar as entradas de acordo com suas categorias. Em uma rede neural, por exemplo, as entradas serão

uma coleção de padrões, e a saída será as possíveis categorias desse padrão de entrada. Segundo Nievola (1996), existe uma grande diferença entre agrupamento e classificação. O agrupamento é o processo de agrupar amostras de entradas que são vizinhos espaciais, enquanto a classificação consiste na rotulação de amostras de entradas através de algum critério externo. Para que um processo utilizando classificação de padrões tenha sucesso, é necessário que exista um mapeamento dos padrões em um espaço de classificação. Com o mapeamento definido, cada padrão definido será configurado como uma determinada classe.

3.3.1 Classificadores Probabilísticos

Este tipo de classificador se baseia na probabilidade com a condição de que a distribuição de dados no espaço do problema seja bem conhecida. Os parâmetros da distribuição utiliza o treinamento supervisionado onde supõe que todos os dados estejam disponíveis simultaneamente. É um classificador que se baseia na probabilidade a priori das classes dos padrões de treinamento em medidas de probabilidade condicionada a posteriori de um dado padrão pertencer a uma classe de treinamento (TODESCO, 1995).

3.3.2 Classificadores baseado em exemplos

Também chamado de classificador preguiçoso, os classificadores exemplares utilizam amostras rotuladas para classificar novas amostras ao invés de construir um classificador a frente do tempo, ou seja, é um tipo de classificador que se baseia nos exemplos treinados previamente. A ideia é atribuir os padrões de entrada para os exemplares mais próximos, utilizando, principalmente, a distância euclidiana ponderada entre as entradas e os exemplares. Normalmente o treinamento desse classificador é uma combinação supervisionada e não supervisionada. Um exemplo é o classificador K-vizinho, que, dado um modelo, tenta encontrar k elementos que são similares ao modelo oferecido.

3.3.3 Classificadores de Kernel

Os classificadores de Kernel tem como objetivo transformar os espaços de entradas para um espaço consideravelmente maior criando regiões de decisão baseado pelas funções kernel. O treinamento dos classificadores de kernel são relativamente rápidos, principalmente quando se utiliza treinamento não supervisionado para a determinação dos centros.

3.3.4 Classificadores de árvore de decisão

Os classificadores de árvore de decisão se baseia em uma estrutura em árvore com seus nós representando os termos, suas bordas representando os testes no termo que os pesos devem possuir e os nós-folhas representando as categorias. Dessa forma o classificador percorre o nó-folha até encontrar a categoria da entidade informada. É um tipo de classificador que tenta superar a dificuldade de interpretação dos classificadores probabilísticos através de suas simbologia.

4 REDES NEURAIS

4.1 Introdução as Redes Neurais

Com os avanços da tecnologia e da busca da automatização dos procedimentos, várias pesquisas estiveram presentes, onde buscava-se reduzir problemas de controle de sistemas que apresentavam características complexas ou não lineares. Entre os estudos realizados, os computadores de Von Neumann deram ao mundo novas perspectivas. Mas foi o desenvolvimento das técnicas que imitam o comportamento da inteligência humana que fez muito desses objetivos serem atingidos, se ampliando bastante nos últimos tempos. Segundo Haykin (1998), o cérebro é o computador altamente complexo, não linear e paralelo. Ele tem a capacidade de organizar seus constituintes estruturais, conhecidos por neurônios, de forma a realizar certos processamentos muito mais rapidamente que o mais rápido computador digital conhecido. Entre as principais técnicas que tentam imitar o comportamento do cérebro humano, as Redes Neurais são as mais conhecidas e vêm de um grande campo de aplicação, principalmente nos últimos anos.

As Redes Neurais Artificiais constituem-se na técnica de inteligência artificial mais antiga que existe. Foi desenvolvida na década de 40, por Walter Pitts e McCulloch, onde tinha como ideia inicial realizar uma analogia entre os neurônios biológicos e os circuitos eletrônicos, que obtinham a conexão sináptica por meio de resistores. Com o surgimento da máquina de Von Neumann também na década de 40, a busca pelo entendimento e estudo das redes neurais foram diminuindo e deixado em segundo plano, já que até o momento as redes ainda não ofereciam resultados satisfatórios. Entretanto o seu estudo, ou a analogia entre o computador e os neurônios, nunca foram deixados de lado.

Em 1956, a Inteligência Artificial passou por uma grande mudança, e seus estudos foram divididos em duas vertentes: A simbólica e a conexionista. A simbólica tentava simular o comportamento do neurônio humano através de uma estrutura linguística, desconsiderando os mecanismos responsáveis pelo fenômeno da inteligência. A conexionista afirmava que, se conseguisse simular as conexões do neurônio humano, era possível apresentar, de certa forma, o processo da inteligência.

Em 1957 surgiram as redes Perceptrons. Criada por Frank Rosenblatt, tinha como objetivo inicial simular o comportamento apresentado pelas retinas, demonstrando como o sistema nervoso visual era capaz de reconhecer os padrões. A partir daí deu início a diversas pesquisas, despertando o interesse de diversos pesquisadores. Porém, Marvin Minsky, responsável pela criação do primeiro neuro computador em 1951, e

Seymour Papert afirmaram, utilizando suas pesquisas, que redes neurais de apenas uma camada, como proposta por Rosenblatt, não eram capazes de resolver diversos problemas conhecidos da inteligência artificial, como por exemplo, simples operações lógicas como o XOR (disjunção exclusiva). Estas pesquisas fizeram com que os estudos envolvendo redes neurais fossem interrompidos. Em 1987, Jonh Hopfield apresentou uma rede neural que simulava o então comportamento do sistema nervoso das lesmas, renascendo então o interesse por tais algoritmos.

Basicamente, uma rede neural é uma estrutura paralela com habilidades de generalização, ou seja, oferece saídas adequadas para entradas que não estavam presentes durante o treinamento. Essa é a grande vantagem das redes em comparação aos demais algoritmos. Segundo Haykin (1998), a rede neural é semelhante ao cérebro em dois aspectos:

1. O conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de aprendizagem.
2. Forças de conexão entre neurônios, conhecidas como pesos sinápticos, são utilizados para armazenar o conhecimento adquirido.

Em problemas complexos normalmente as redes são utilizadas em conjunto com outras tecnologias, já que sozinha, para tais problemas, não consegue oferecer uma solução adequada. Com a decomposição de problemas complexos em outros relativamente simples, as redes neurais poderiam então trabalhar com um subconjunto desses problemas, oferecendo soluções satisfatórias.

Um dos grandes benefícios das redes neurais é a capacidade de representar um universo não estático, onde as estatísticas mudam consideravelmente com o tempo, pois é capaz de alterar seus pesos sinápticos em tempo real, se adaptando a situação encontrada (**Adaptabilidade**). Outra vantagem das redes neurais é a capacidade de se recuperar, por exemplo, as possíveis falhas que possam acontecer em seus neurônios, caso tal rede seja implementada em *hardware* (**Tolerância às falhas**). Ou seja, apesar da perda de qualidade da rede neural com a falha de um dos seus neurônios, provavelmente o resultado final não sofreria consequências que impossibilitasse de obter resultados satisfatórios.

Mas utilizar redes neurais também existem desvantagens, e uma delas é a dificuldade em trabalhar com problemas complexos, além de ser considerada uma verdadeira **caixa preta**, ou seja, é praticamente impossível saber exatamente como uma rede neural chegou a um determinado resultado, ou qual a relevância de um determinado peso sináptico a uma saída apresentada. É apenas possível descobrir que a rede chegou a um resultado satisfatório ou não analisando o erro médio quadrático no processo de validação.

4.2 Os princípios da neuro computação

Para compreender os procedimentos utilizados nas redes neurais, é necessário conhecer os processos referentes ao funcionamento do cérebro humano, principalmente dos neurônios. A formação das conexões entre as células e como se concebe teoricamente a aquisição do conhecimento é de grande importância para compreender as bases da neuro computação.

Para Haykin (1998), o sistema nervoso humano pode ser visto como um sistema de três estágios: os estímulos com seus receptores, a rede neural nervosa e as respostas com seus atuadores. O centro do sistema é o cérebro que recebe continuamente informação e toma decisões apropriadas.

O cérebro humano possui cerca de 10 bilhões de neurônios. Um neurônio é capaz de criar até 10.000 sinapses com outros neurônios. Seus principais componentes são:

- *Os dentritos* – Recebem os estímulos transmitidos por outros neurônios.
- *O corpo do neurônio* – Responsável por coletar informações vindas de outros neurônios.
- *Os axônios* – Responsável por transmitir os estímulos para as outras células.



Figura 4 – Neurônio humano e sua composição.

Cada região do cérebro é especializada por uma determinada função, onde cada processamento estabelecido é obtido através das redes particulares interligadas entre si e realizando um processamento paralelo. Cada região do cérebro possui uma arquitetura diferente, variando de acordo aos números de neurônios, de sinapses e de pesos. Os valores dos pesos são estabelecidos durante a vida útil dos neurônios. Em termos gerais esse é o procedimento de memorização.

Similarmente ao neurônio natural, o neurônio matemático recebe um ou mais sinais de entradas, devolvendo um único sinal de saída. As entradas chegam simultaneamente a rede, por isso as redes são conhecidas também como ferramenta de processamento paralelo. Os dentritos são representados matematicamente pela sinapse, e a intensidade da ligação é representada pelos pesos sinápticos W . Quando as entradas

x são apresentadas aos neurônios, as mesmas são multiplicadas pelos pesos sinápticos gerando o seguinte equação:

$$W_1x_1 + W_2x_2 + \dots + W_nx_n \quad (2)$$

A equação 2 é a tão conhecida função de ativação. O resultado é então apresentado a uma função de transferência que tem como objetivo evitar o acréscimo progressivo dos valores de saída ao longo das camadas da rede. Um exemplo do modelo matemático de rede é apresentado na Figura 5.

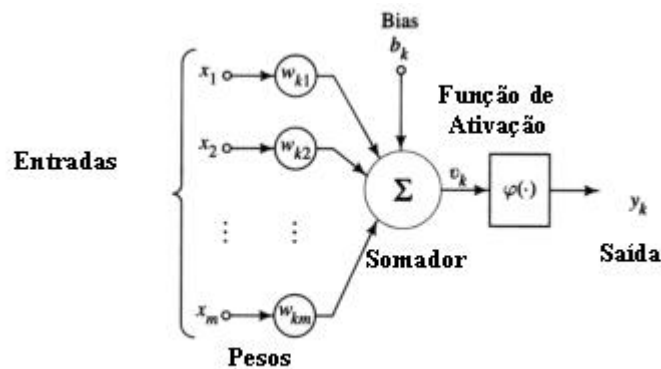


Figura 5 – Neurônio matemático e sua composição (HAYKIN, 1998).

O modelo neural matemático também pode incluir uma polarização, conhecido popularmente como *bias* de entrada, normalmente representado pela letra b . O *bias* tem o efeito de aumentar ou diminuir a entrada líquida da função de ativação, dependendo se ele é positivo ou negativo, respectivamente (HAYKIN, 1998). Além disso, o *bias* possibilita que o neurônio apresente uma saída não nula mesmo que todas as entrada sejam nulas. Isso ajuda com que a rede não tenha problemas para os cálculos realizados em sua função de transferência, evitando resultados insatisfatórios em suas saídas.

A combinação dos neurônios matemáticos em uma ou mais camadas, interligados através das sinapses pode-se formar então as redes neurais, que em geral possui as seguintes características:

- Uma camada de entrada, que tem como única função distribuir os dados de entrada para cada neurônio da camada seguinte.
- Camadas ocultas constituída de neurônios ocultos que irão capacitar a rede a extrair estatísticas. Essa camada não é obrigatória em todas as redes, podendo ou não existir, dependendo de sua arquitetura.
- Uma camada de saída que contém o número de neurônio igual ao número de variáveis de saídas.

As conexões entre as camadas podem partir de várias arquiteturas diferentes. Quando uma rede possui todas as suas saídas conectadas com todos os neurônios das camadas seguintes essa rede é denominada de rede **amplamente conectada**. Quando o sinal de saída de um neurônio servir como sinal de entrada em um neurônio da mesma camada ou da camada anterior a rede possui a característica de **realimentação** (HAIKIN, 1998). Com isso podemos afirmar que a arquitetura de uma rede neural é livre, podendo ser modificada de acordo com a necessidade oferecida pelo problema. Isso significa que não existe padrão para o número de neurônios ou camadas de cada rede. Essa característica irá se adaptar de acordo com o problema.

4.3 As Redes Neurais Artificiais

A arquitetura de uma rede neural depende do problema e da escolha realizada pelo projetista. Segundo Haykin(1998), a maneira pelo qual os neurônios de uma rede estrutural estão estruturados está intimamente relacionado com o algoritmo de aprendizagem utilizado para treinar a rede. No geral, um projeto de redes neurais é dividido basicamente em quatro fases definidas: A coleta de dados, a configuração da rede, o treinamento e o teste.

A coleta de dados tem como finalidade minimizar a ambiguidade dos dados a fim de evitar erros na projeção da rede. Uma análise incorreta dos dados poderá fazer com que a rede não obtenha os resultados desejados. Por outro lado, quando os dados estão bem minimizados e cobrindo amplamente o domínio do problema, a rede oferece resultados consideravelmente satisfatórios. Portanto, é uma tarefa de fundamental importância para o projeto da rede. Além disso, os dados analisados deverão ser separados em duas categorias: Os dados de treinamento, utilizados para treinar a rede, e os dados de validação, utilizados para validar os resultados apresentados pela rede. Outro detalhe é que a quantidade de variáveis de entrada irá definir a quantidade de sinapses da rede.

Na fase de configuração será definido o modelo de rede que melhor se adequa ao problema, a quantidade de neurônios, camadas e nós, o algoritmo de treinamento e a função de transferência utilizada. De uma forma geral, a escolha de cada modelo depende do problema a ser solucionado e da experiência do projetista em relação a tal problema. Somente a prática poderá definir qual será melhor modelo para uso.

A Figura 6 representa a forma mais simples de uma rede em camadas, com apenas uma camada de entrada e uma camada de saída. É uma rede acíclica, pois é a alimentação dos neurônios das camadas posteriores são realizadas apenas pelos neurônios das camadas anteriores.

Uma rede acíclica de múltiplas camadas é apresentado na Figura 7. Tal rede

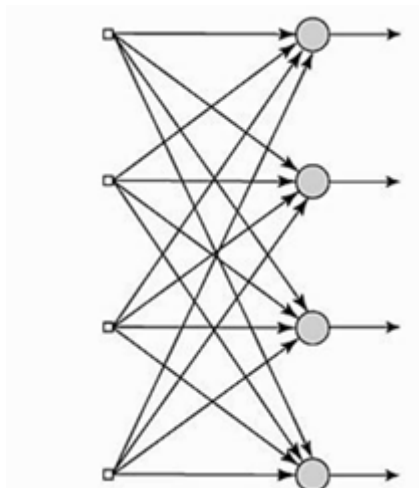


Figura 6 – RNA com uma camada de entrada e uma camada de saída (HAYKIN, 1998).

recebe este nome porque apresenta uma ou mais camadas ocultas ou intermediárias. A função dos neurônios ocultos é intervir entre a entrada externa e a saída da rede de uma maneira útil. Com a adição de uma ou mais camadas ocultas, a rede é capaz de extrair estatísticas de ordem elevada. Em um sentido bastante livre a rede adquire uma perspectiva global apesar de sua conectividade local (HAYKIN, 1998).

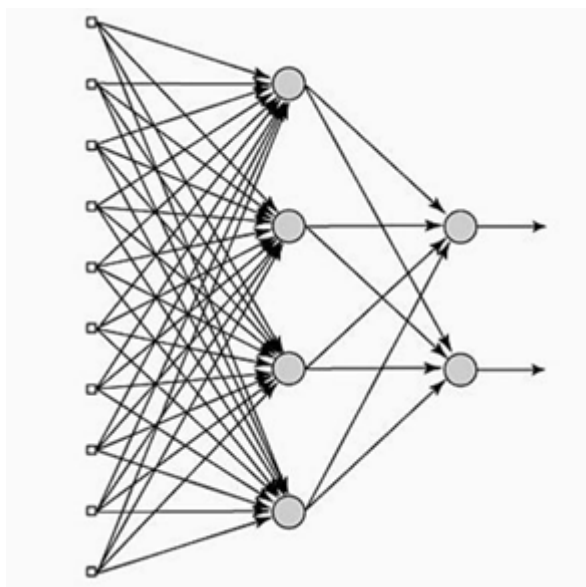


Figura 7 – RNA com uma camada oculta (HAYKIN, 1998).

Outro modelo popularmente conhecido, e já citado, é o modelo de rede recorrente, apresentado na Figura 8. Tal nome é dado justamente porque existe um ou mais neurônios que realizam a realimentação em outros neurônios. Uma rede recorrente pode consistir, por exemplo, de uma única camada de neurônios com cada neurônio alimentando seu sinal de saída de volta para as entradas dos outros neurônios (HAYKIN, 1998).

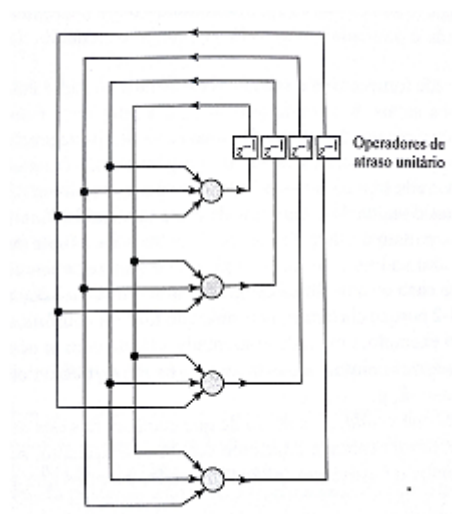


Figura 8 – RNA com realimentação (HAYKIN, 1998).

Na etapa de treinamento da rede os pesos sinápticos são ajustados a fim de se obter um melhor resultado de saída. Normalmente, os valores iniciais dos pesos são valores aleatórios uniformemente distribuídos em um intervalo definido. Em relação ao tempo de execução do treinamento podemos adotar algumas estratégias como quantidade máxima de iterações ou valores mínimos de erros. Obviamente, quanto maior a quantidade de iterações ou menor o erro, maior é o tempo de treinamento. Um problema importante que deve ser evitado é o *overtraining*. Devido à quantidade grande de treinamento, a rede acaba memorizando os dados e perdendo sua capacidade de generalização, que é a característica de reconhecer padrões em entradas ainda não treinadas. O ideal é que o treinamento seja interrompido, assim que a rede conseguir obter bons resultados de generalização.

O próximo passo após o treinamento da rede é o teste. Nessa fase os dados separados para validação são utilizados a fim de verificar o desempenho da rede. Diferentemente da etapa de treinamento, onde são passados os resultados de entrada e de saída, na etapa de teste apenas os resultados de entrada são informados, caso o treinamento realizado seja supervisionado. Com isso é possível realizar a comparação dos resultados apresentados pela rede com os resultados reais de entrada. Uma etapa importante na fase de teste é a análise dos pesos e das bias, pois, neurônios de entrada e bias com valores pequenos poderão ser retirados, já que não afetam diretamente no resultado da rede.

4.4 As Redes Perceptrons

A Rede Perceptron foi representada em seu modelo mais simples em 1957, por Frank Rosenblatt, em seu estudo originado de uma simulação computacional para a retina, demonstrando o reconhecimento de padrões do sistema nervoso visual. Em seu modelo mais simples, de apenas duas camadas, apresenta um conjunto de neurônios na camada de entrada interligados a um conjunto de neurônio na camada posterior de saída. Embora seja uma rede de modelo simples, existem alguns problemas específicos, impedindo sua utilização em aplicações mais complexas. Normalmente é um modelo utilizado para classificações de padrões linearmente separáveis, ou seja, padrões que se encontram de lado opostos a um hiperplano.

Na Rede Perceptron de camada única, como ilustrada na Figura 9, os elementos da camada de entrada, juntamente com o *bias*, fazem a distribuição de cada sinal de entrada para todos os neurônios da camada de saída, formada por um número de neurônios iguais à quantidade de variáveis de saída. Tais neurônios são constituídos por uma função de ativação e uma determinada função de transferência.

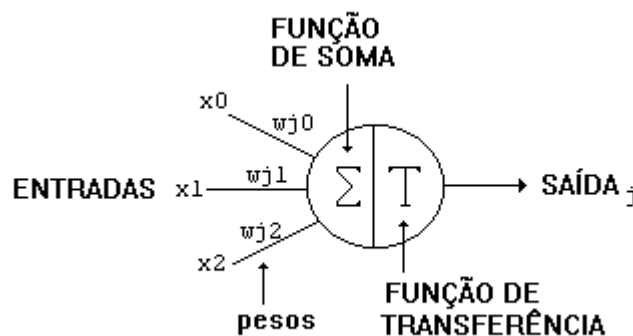


Figura 9 – Rede Perceptron de camada única (HAYKIN, 1998).

Os valores w representam os pesos sinápticos das ligações entre os neurônios da camada de entrada e da camada de saída. As funções de ativação mais utilizadas são:

1- Função Limiar – Nesta equação, a saída do neurônio assume o valor 1, caso o campo local induzido do neurônio seja maior ou igual a 0, e 0 caso contrário.

$$\begin{cases} 1, & \text{se } v_k \geq 0, \\ 0, & \text{se } v_k < 0. \end{cases} \quad (3)$$

2- Função Sigmóide – Função mais comum no desenvolvimento das redes, definida como uma função estritamente crescente, exibindo um balanceamento

adequado entre funções lineares e não-lineares.

$$\varphi(v) = \frac{1}{1 + \exp(-av)} \quad (4)$$

Com o aperfeiçoamento das redes neurais, surgiram então os modelos que conseguiam abranger um maior número de problemas, incapazes de serem resolvidos utilizando as Redes Perceptron de camada simples. A necessidade de imitar o cérebro humano e sua capacidade de reconhecer padrões levou então ao surgimento dessa nova estratégia, que era introduzir uma camada de neurônios ocultos, entre as camadas de entrada e a camada de saída. Tal estratégia foi capaz de alavancar novamente os estudos referente às redes neurais.

O modelo é composto por uma camada de entrada, uma ou mais camadas intermediária ou oculta, e uma camada de saída. O sinal de entrada se propaga para frente, até a camada de saída. Assim como a Rede Perceptron de camada única, a Rede Perceptron Multicamadas (MLP) é treinada de forma supervisionada, através da estratégia de minimização do erro. O mecanismo utilizado para aprendizado da rede é o algoritmo de retro-propagação do erro, que será comentado posteriormente. A Figura 10 representa um modelo clássico das redes MLP.

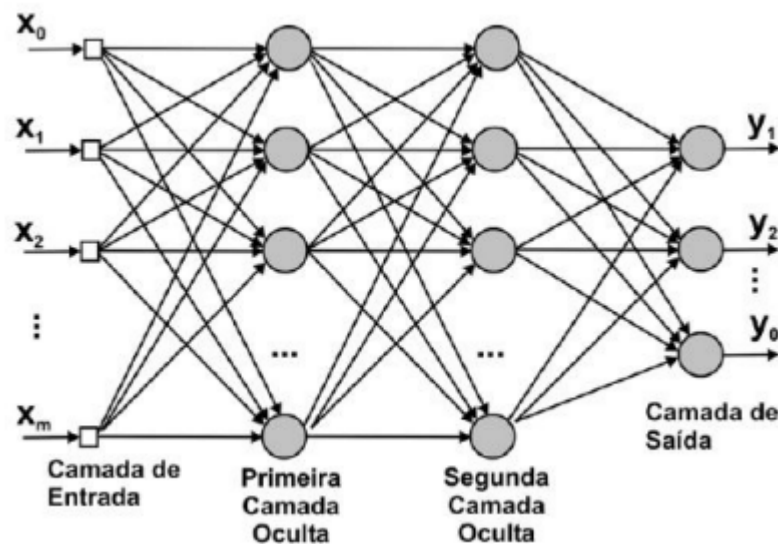


Figura 10 – Modelo clássico MLP com sinais de entrada x_n e sinais de saída y_n . (HAYKIN, 1998)

Segundo Haykin (1998), essa rede apresenta três características que difere da perceptron simples:

1 - O modelo de cada neurônio de rede inclui uma função de ativação não linear, diferenciável em qualquer ponto. Uma forma de representar essa não linearidade é através da função sigmoide.

2 - A rede possui uma ou mais camada de neurônios ocultos que capacitam a rede em aprender tarefas complexas extraindo progressivamente as características mais significativas dos padrões de entrada.

3 - A rede exibe um alto grau de conectividade, representado pela sua sinapse. Uma mudança na conectividade da rede requer uma mudança nas conexões sinápticas e de seus pesos.

4.5 Aprendizagem - O algoritmo de retro-propagação

O procedimento utilizado no processo de aprendizagem das MLPs é conhecido como algoritmo de aprendizagem. Sua principal função é modificar os pesos sinápticos buscando a minimização dos erros, para assim, poder alcançar o resultado de forma desejada.

Para Haykin (1998), a aprendizagem é um processo pelo qual os parâmetros livres de uma rede neural são adaptados através de um processo de estimulação pelo ambiente no qual a rede é inserida. O tipo de aprendizagem é determinado pela maneira pelo qual a modificação dos parâmetros ocorre. A aprendizagem pode acontecer de duas formas: De modo **supervisionado** ou de modo **não supervisionado**.

Conhecido também como aprendizagem que utiliza o auxílio de um "professor", o aprendizado supervisionado considera o conhecimento sobre o problema, representado por um exemplo de entrada-saída. Ao utilizar o processo de treinamento pelo conhecimento supervisionado, a rede já contém a resposta desejada sobre o atual vetor de entrada utilizado no treinamento. Com isso, é possível ajustar o sinal do erro. Este é o modelo de aprendizagem que será utilizado para solucionar o problema proposto por esse estudo.

No modelo de aprendizagem não supervisionada, ou sem o auxílio de um "professor", não existe um valor prévio de saída para utilizar no treinamento. Ou seja, o aprendizado da rede é realizado através da comparação entre os padrões de saída. Como se trata de um modelo não utilizado no estudo proposto, sua descrição é apenas para fins de comparação.

A modificação dos pesos sinápticos é um método tradicional em um projeto de redes neurais. Esta abordagem é bastante próxima dos filtros adaptativos lineares, que já está bem estabelecida e foi aplicada com sucesso em diversas áreas (WIDROW e STEARNS, 1985; HAYKIN, 1998). Entretanto é possível também para uma rede neural modificar sua própria topologia, o que é motivado pelo fato de os neurônios no cérebro humano poderem morrer e que novas conexões sinápticas possam crescer HAYKIN(1998).

Nos modelos MLP's o processo de aprendizado é realizado através do algoritmo de retro-propagação de erro. Obviamente, para realizar o processo de minimização do erro, é necessário o conhecimento do mesmo. Dessa forma, o procedimento de aprendizagem no modelo MLP é realizada de forma supervisionada. O sinal do erro de um determinado neurônio i é representado através da equação abaixo.

$$e_i(n) = d_i(n) - y_i(n) \quad (5)$$

Onde:

$y_i(n)$ - É a resposta oferecida pelo neurônio

$d_i(n)$ - É a resposta desejada pelo neurônio

Dessa forma, o sinal global instantâneo do erro do neurônio, na interação n , é definido por:

$$E(n) = \frac{1}{2} \sum_{j=1}^J e_j^2(n) \quad (6)$$

Onde J inclui todos os neurônios da camada de saída da rede. O erro de cada neurônio é elevado ao quadrado para evitar que um determinado valor de erro negativo compense outro valor positivo.

O erro quadrático global médio da rede para um determinado treinamento é representado pela media aritmética dos erros globais instantâneos, conforme equação abaixo:

$$E = \frac{1}{N} \sum_{n=1}^N E(n) \quad (7)$$

Quando o erro quadrático global médio não é satisfatório, deverá então realizar a retro-propagação do sinal para o ajuste dos pesos nas próximas iterações. Tal processo começa pela última camada em direção à primeira, pois o gradiente local δ_i , calculado em uma camada, será conveniente utilizada para o cálculo dos gradientes de erro da camada imediatamente anterior.

Basicamente, consegue-se resumir o algoritmo de retro-propagação de erros em três etapas:

1. **Inicialização:** Valores aleatórios são atribuídos uniformemente aos pesos sinápticos e aos *bias*.

2. **Apresentação dos treinamentos:** É realizado o processo de iterações, onde em cada iteração ocorre a propagação do sinal e a retro-propagação dos erros, corrigindo os pesos e os *bias*.

- a) *Propagação do sinal:* O vetor de sinal de entrada é aplicado a primeira camada da rede, calculando a saída em todos os neurônios, obtendo-se então o vetor de sinais de entrada para a próxima camada. Em seguida o sinal do erro é calculado para cada neurônio de saída, calculando-se então o erro instantâneo e o erro global daquela iteração.
- b) *Retro-propagação do sinal do erro:* Os gradientes locais $\delta_i(n)$ são calculados em todos os neurônios da camada de saída, através da função:

$$\delta_i(n) = -e(n)\varphi'_j(v_j(n)) \quad (8)$$

Onde φ'_j representa a derivada em relação ao argumento. Em seguida, os pesos e os *bias* de cada camada são ajustados da seguinte forma:

$$\delta W_{ij} = -n\delta_j(n)y_i(n) \quad (9)$$

E por fim o ajuste na camada oculta. Primeiramente o cálculo do gradiente local para cada neurônio:

$$\delta_i(n) = \varphi'_i(v_i(n)) \sum_{j=1}^J (\delta_j(n)W_{ij}(n)) \quad (10)$$

E finalmente os pesos sinápticos e os *bias* dessa camada:

$$\Delta W_{ki} = -n\delta_j(n)y_k(n) \quad (11)$$

3. **Iteração** – Novas iterações são apresentadas para o treinamento da rede de forma aleatória, até que seja satisfeito o critério de parada. Para Haykin (1998), considera-se que o algoritmo de retro-propagação tenha convergido quando a taxa absoluta de variação de erro médio quadrado por época for suficientemente pequena. Na prática, como já mencionado, podemos limitar quantidade de iteração do algoritmo da seguinte forma; ou pelo cálculo do erro ou limitando a quantidade de iterações.

Quadro 1 – Algoritmo de Retro-propagação.

Inicialização - Distribuição aleatória dos pesos sinápticos e bias.

Apresentação dos treinamentos - Propagação dos sinais e a retro-propagação dos erros.

1. *Propagação do sinal*: Vetor de sinais propagado inicialmente na camada de entrada até a camada de saída, calculando o valor do erro médio mínimo.
2. *Retro-propagação do sinal do erro*: Cálculos do gradientes locais em todos os neurônios, ajustando os pesos sinápticos.

Iteração - Novas iterações na rede até obter o erro médio quadrado desejado ou o número máximo de épocas.

5 PROCEDIMENTO E MÉTODOS

As informações relevantes para o desempenho e aprendizagem são adquiridas por meio da contínua interação do aprendiz e seu contexto de ação (CHOW *et al.*, 2011). Dessa forma, torna-se de fundamental importância identificar e manipular as restrições que possam oferecer diferentes demandas no processo da ensino-aprendizagem-treinamento do esporte (ARAÚJO *et al.*, 2004; DAVIDS *et al.*, 2008; RENSHAW *et al.*, 2010). Para Chow (*et al.*, 2011), essas ideias reforçam as necessidades de que os educadores e treinadores devem agir como facilitadores no processo, propondo atividades exploratórias para que os aprendizes possam encontrar soluções estáveis no contexto da ação.

Para Menuchi (2014), alterando os parâmetros específicos, professores e treinadores podem efetivamente guiar os aprendizes a explorarem a funcionalidade de diferentes estados organizacionais para adaptarem às diferentes demandas de restrições das tarefas. Como hipótese, a manipulação da restrição espaço-temporal pode resultar em mudanças na aprendizagem e desenvolvimento de habilidades motoras, bem como a efetividade as adaptações às imprevisíveis demandas no contexto da ação (ARAÚJO *et al.*, 2004; DAVIDS *et al.*, 2003). Além disso, a manipulação das restrições do contexto da ação afetam tanto em termos imediatos quanto em termos longitudinais (CHOW *et al.*, 2011). Assim, a manipulação das restrições é de fundamental importância no processo da formação esportiva.

Dessa forma, o objetivo chave do estudo foi realizar uma modelagem computacional, de acordo com as variáveis selecionadas como potencial parâmetro pela literatura, em diferentes categorias do futebol, utilizando como atividade experimental o bobinho. Com isso, buscou-se observar o desenvolvimento otimizado das habilidades motoras dos atletas de acordo com suas mudanças comportamentais verificando quando o comportamento era mais randômico, onde nenhum padrão é claro devido a alta variabilidade e aumento da complexidade, ou periódico, quando possibilita observar as modificações ocorridas no desenvolvimento de cada categoria.

Assim, nesse capítulo será apresentado o desenvolvimento prático realizado durante o estudo, buscando uma modelagem adequada da condição experimental de acordo com a descrição teórica. Serão representados todos os procedimentos cálculo de variáveis e implementação do sistema computacional utilizado.

5.1 Procedimento de coleta de dados

Os dados utilizados nesse estudo foram retirados do banco de dados do Grupo de Estudos e Pesquisa em Comportamento Motor (GEPECOM) da Universidade Estadual de Santa Cruz.

Para a coleta e análise dos dados, duas manipulações que professores e treinadores utilizam frequentemente foram utilizadas: a restrição espacial e a restrição temporal. A restrição espacial foi obtida manipulando o espaço do jogo, oferecendo limites espaciais para o deslocamento dos jogadores. A restrição temporal foi obtida na manipulação de toques de bola, interferindo diretamente no tempo em que cada jogador pode passar com a bola.

Participaram da coleta de dados praticantes de futebol, distribuídos em 4 grupos diferentes:

1. Grupo 1 - Indivíduos que atuam na categoria sub-13 (entre 10 e 13 anos de idade)
2. Grupo 2 - Indivíduos que atuam na categoria sub-15 (entre 14 e 15 anos de idade)
3. Grupo 3 - Indivíduos que atuam na categoria sub-17 (entre 16 e 17 anos de idade)
4. Grupo 4 - Indivíduos que atuam na categoria sub-20 (entre 18 e 20 anos de idade)

Os participantes referentes aos grupos foram convidados de uma escola de formação do Vasco da Gama - RJ, em Ilhéus - BA. A coleta de dados foi realizada dentro da própria escola de formação. Anteriormente as atividades, os atletas realizaram um aquecimento contendo corridas leves, passes e alongamentos de membros superiores e inferiores. Após o aquecimento os participantes foram organizados e agrupados de 5 em 5, onde quatro participantes denominados passadores foram distribuídos ao redor de um participante denominado marcador. A Figura 11 ilustra o a organização inicial do experimento.

O marcador teve como orientação recuperar a posse de bola, inicialmente na posse de um jogador passador. Por outro lado os jogadores passadores deveriam impedir que a bola fosse recuperada pelo marcador. Caso o marcador recuperasse a bola ou influenciasse a um erro de passe, o jogo era interrompido e o jogador responsável pelo erro passava a posição de marcador, e o antigo marcador passava a posição de passador, recomeçando a atividade. Cada interceptação ou erro de passe representava um **rali**.

Baseado nestas manipulações, quatro combinações foram analisadas:

1. ELPL - Posse de bola livre e dimensão espacial livre;
2. ERPL - Posse de bola livre e dimensão espacial restrita;



Figura 11 – Organização inicial do bobinho. Quatro passadores dispostos ao redor de um marcador

3. ELPR - Posse de bola restrita(um toque na bola) e dimensão espacial livre;
4. ERPR - Posse de bola restrita(um toque na bola) e dimensão espacial restrita;

Na condição de posse de bola livre, os participantes teriam liberdade para efetuar quantos toques na bola acharem necessário, bem como a manutenção da bola nos pés. Na condição de posse de bola restrita os participantes somente poderiam efetuar um toque na bola. Na condição de dimensão espacial livre os participantes poderiam atuar a uma área de 9 metros por 9 metros. Na condição de dimensão espacial restrita os participantes poderiam atuar em uma dimensão espacial de 6 metros por 6 metros.

Cada combinação teve 5 minutos de duração, totalizando 20 minutos no final da atividade. O experimento foi capturado utilizando uma filmadora digital da marca SONY HDD Hybrid - modelo DCR-SR45, configurada para gravação de 25 quadros por segundo, ou seja, 25Hz. A filmadora foi posicionada lateralmente ao campo com uma altura de 3 metros e distância de 14 metros da posição inicial do marcador. Segundo Menuchi (2014) e Duarte (*et al.*, 2010), essa é uma posição satisfatória para a captura do movimento dos participantes em uma modalidade como o bobinho.

5.2 Procedimento fotogramétricos

O procedimento fotogramétrico foi realizado utilizando o *software Digital Video for Windows* (Laboratório de Biomecânica e Instituto de Computação - Unicamp, 1998, versão 5.10). Tais processos foram divididos em captura, digitalização, calibração e reconstrução bidimensional.

A digitalização é o processo onde o *software* irá rastrear o deslocamento dos jogadores ao longo do tempo (espaço-temporal). Após o rastreamento dos pontos dos jogadores ao longo do vídeo, o comportamento espaço-temporal é registrado em pixel. Para transformar as medidas de pixels em metros, coordenadas espaciais referenciais

do espaço de jogo foram usadas para o processo de calibração. Ao final da calibração, um arquivo é gerado, contendo as coordenadas bidimensionais x e y dos pontos médios quadro a quadro.

5.3 Variáveis analisadas

Com a digitalização realizada, foi possível definir as variáveis que serviriam como entrada para modelagem computacional do experimento. O procedimento para os cálculos foi realizado utilizando o *software* Matlab R2013a (versão 8.1). Para a retirada de possíveis ruídos, o filtro de passa-baixa Butterworth de quarta ordem com frequência de 5hz foi utilizado, conforme descrito por Winter (2005). Tal filtro busca uma resposta em frequência o mais plana possível, com o mínimo de oscilações tanto na banda passante como na banda de corte. Todas as coordenadas X e Y dos jogadores e da bola passaram pelo filtro Butterworth.

Após a aplicação do filtro, o próximo passo foi verificar quais as variáveis que participariam do experimento. Inicialmente as variáveis calculadas foram as distâncias interpessoais entre todos os jogadores e a bola, a velocidade média do passe e o ângulo formado pelo passe.

1. **Distâncias Interpessoais:** Distância euclidiana calculada entre o posicionamento de todos os jogadores e da bola em um determinado quadro.

$$di = \sqrt{(j1_x - j2_x)^2 + (j1_y - j2_y)^2} \quad (12)$$

Onde $j1_x$ e $j2_x$ são os valores das coordenadas x dos jogadores 1 e 2 e $j1_y$ e $j2_y$ são os valores das coordenadas y dos jogadores 1 e 2.

2. **Ângulo de passe:** ângulo formado entre os atletas passador, receptor e marcador. O cálculo do ângulo seguiu os princípios básicos do círculo trigonométrico.

$$\cos \theta = \frac{a^2 - (b^2 + c^2)}{-2bc} \quad (13)$$

Onde a , b e c referem-se as distâncias entre os pontos que formam o ângulo de interesse. A Figura 12, ilustra como é formado ângulo entre os participantes.

3. **Velocidade do passe:** Velocidade da bola no momento em que sai do passador até chegar ao receptor. Passes interceptados ou não concluídos não foram levados em consideração. O cálculo é realizado utilizando a fórmula básica de velocidade

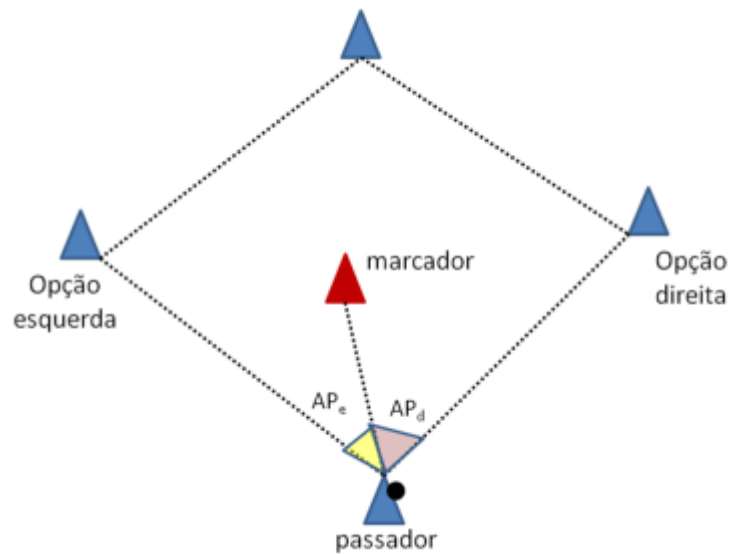


Figura 12 – Ilustração do ângulo de passe (MENUCHI, 2014).

relativa: A fração entre a distância interpessoal dos jogadores (ΔS) e o tempo do passe (ΔT).

$$V = \frac{\Delta S}{\Delta T} \quad (14)$$

5.3.1 A Rotina para o cálculo das variáveis

Após a definição das variáveis, o desenvolvimento da rotina foi inicializada. Primeiramente, os arquivos gerados pelo *software Dvideo* foram recebidos para a leitura. Como já informado, os arquivos continham a posição de todos os jogadores e da bola nas coordenadas (x, y) . Cada linha do arquivo representava as posições em um frame. A totalidade dos arquivos dependiam da quantidade de ralis encontrados em cada grupo, em cada categoria. Cada rali representava um determinado arquivo. No total foram gerados 64 arquivos para a categoria sub-13, 71 arquivos para a categoria sub 15, 82 arquivos para a categoria sub-17 e 66 arquivos para a categoria sub-20. Totalizando 283 ralis.

Além dos ralis, algumas outras informações foram necessárias e utilizadas como entrada na rotina para o cálculo das variáveis:

- O frame em que o passe foi realizado.
- O frame que ocorreu recepção.
- O frame com o fim da posse de bola do receptor.
- Qual o jogador que está realizando o passe.

- Qual jogador que recebeu o passe.

Com os *inputs* informados, a rotina foi capaz de realizar o cálculo de todas as variáveis que seriam utilizadas como entrada na modelagem computacional, em todos os ralis. Para cada execução com o os arquivos e os *inputs* informados, a rotina gerava um arquivo contendo as variáveis:

1. DIPM - Distância interpessoal entre o passador e o marcador.
2. DIPR - Distância interpessoal entre o passador e o receptor.
3. DIMR - Distância interpessoal entre o marcador e o receptor.
4. AP - Ângulo de passe formado entre o marcador, o receptor e o passador.
5. VMP - Velocidade média do passe realizado.
6. A categoria que foi executada (sub-13, sub-15, sub-17 ou sub-20).
7. Qual modelo de experimento utilizado:
 - ELPL* - Espaçamento livre e passe livre.
 - ERPL* - Espaçamento restrito e passe livre.
 - ELPR* - Espaçamento livre e passe restrito .
 - ERPR* - Espaçamento restrito e passe restrito.

5.4 O Sistema computacional

A utilização de um sistema computacional para modelar um experimento como o bobinho consegue auxiliar na compreensão do desenvolvimento e da adaptabilidade da coordenação interpessoal em um contexto competitivo. A modelagem computacional oferece a possibilidade de mapear a configuração espaço-temporal, trazendo informações que auxilie no entendimento dos detalhes que fazem parte na tomada de decisão, bem como os fatores que são considerados influenciadores. Resumidamente, podemos afirmar que a utilização de um sistema computacional, nesse contexto esportivo, oferece a possibilidade de:

1. Mapear a dinâmica da coordenação interpessoal.
2. Reconhecer e classificar padrões presentes no contexto.
3. Analisar as alterações na dinâmica da coordenação interpessoal em função das categorias de informação.

4. Identificar as adaptações da dinâmica da coordenação interpessoal, em função de manipulações de restrições da tarefa nas diferentes categorias de formação do futebol.

Diante disso, buscou-se desenvolver um modelo computacional que possibilite auxiliar no entendimento do contexto do bobinho e em suas respectivas tomadas de decisões ocorridas no andamento da atividade. Para melhor compreender a dinâmica apresentada, quatro estruturas-modelos para criação das redes neurais artificiais foram configuradas. As diferenças entre os modelos estavam relacionadas as variáveis que funcionariam como sinais de entrada e saída da rede. Os modelos estruturados foram:

1. **Modelo 1** - Teste de reconhecimento e classificação do experimento.

Entradas - DIPM, DIPR, DIMR, AP, VMP e categoria.

Saída - Um dos quatro experimento proposto.

2. **Modelo 2** - Teste de reconhecimento e classificação da categoria.

Entradas - DIPM, DIPR, DIMR, AP, VMP e experimento.

Saída - Uma das quatro categorias estabelecidas.

3. **Modelo 3** - Teste de reconhecimento e classificação do passe livre/restrito.

Entradas - DIPM, DIPR, DIMR, AP, VMP e categoria.

Saídas - Se o passe executado no contexto está relacionado aos experimentos de passe restrito ou livre.

4. **Modelo 4** - Teste de reconhecimento e classificação do espaçamento livre/restrito.

Entradas - DIPM, DIPR, DIMR, AP, VMP e categoria.

Saída - Se o espaçamento presente no contexto está relacionado aos experimentos de espaçamento livre ou restrito.

5.4.1 Configuração

Com a preparação e identificação das possibilidades de modelagem do problema proposto, o próximo passo foi configurar o sistema computacional para o mapeamento adequado da dinâmica do bobinho. Para isto, foi utilizado redes neurais artificiais do tipo Perceptron Multicamadas (MLP), que consiste de um modelo computacional formado por neurônios artificiais interligados entre si, associados a determinados pesos. A utilização desse modelo de rede se deve ao fato das redes MLP serem utilizadas principalmente para reconhecimento e classificação de padrões, segundo a própria literatura, com sinais de entrada e saída valorados, e este é justamente o tipo de

reconhecimento e classificação que a pesquisa propõe. Para o treinamento da rede utilizou-se o algoritmo de retro-propagação de sinais denominado *backpropagation*. Tal algoritmo é baseado na análise do sinal de saída, comparando com um determinado erro médio quadrático, onde, caso o sinal de saída não satisfaça, ele é retro-propagado para as camadas anteriores, ajustando os pesos associados a cada neurônio. Antes de cada sinal ser propagado ele é submetido a uma função de ativação ou transferência, que irá converter o valor da soma de entrada para uma faixa de valores aceitáveis. A função de transferência utilizada nos modelos propostos em cada neurônio foi a função sigmoidal logística, conforme abaixo.

$$f(x) = \frac{1}{1 + e^{-ax}} \quad (15)$$

As variáveis de entrada e saídas presentes em cada modelo definido representaram os sinais de entrada da primeira camada e o sinal de saída da última camada da rede neural respectivamente.

5.4.2 Implementação

Ao implementar uma rede neural, principalmente para classificação de padrões, busca-se desenvolver um modelo que não seja muito rígido, não modelando fielmente o problema, e que não seja muito flexível, permitindo também a modelagem de ruídos presentes nos dados. O processo de implementação e preparação dos parâmetros em uma RNA é considerado em algumas literaturas como um procedimento oculto, justamente por ser um processo pouco compreendido. Ainda não existe fórmula para uma configuração ideal das redes neurais, mas existe sugestões de desenvolvedores experientes que poderão ser seguidos para se obter uma rede com desempenho satisfatório. Diante disso, algumas sugestões foram seguidas para a implementação e definição dos parâmetros da rede:

- **Número de camadas ocultas** - Não é recomendado um grande número de camadas ocultas. Isso ocorre devido a falta de precisão do erro médio, que irá atualizar os pesos sinápticos, a medida que o número de camadas aumentam. Diversos testes realizados em redes neurais MLP não demonstram vantagens quando se aumentam o número de camadas ocultas. Além disso, quanto maior a quantidade de camadas, mais complexa se torna a rede. Para problemas de classificação de padrões, uma camada oculta é suficiente.
- **Número de neurônios ocultos** - Um alto número de neurônios na camada oculta pode fazer com que a rede memorize os dados (*overfitting*), perdendo a capacidade de generalização. Em contrapartida o baixo número faz com que a rede demore

para encontrar um resultado satisfatório. Diante disso, a maioria dos autores recomendam que a quantidade de neurônios pode ser definido como a média geométrica ou a aritmética das camadas de entrada e saída. Para casos que a saída é definida com apenas um neurônio, o número de neurônios da camada oculta, pode ser calculado através do número de neurônios da camada de entrada. Outro critério que também pode ser considerado, e que foi utilizado na pesquisa, é calcular a quantidade de neurônios ocultos através do dobro de neurônios da camada de entrada acrescido de 1.

- **Taxa de aprendizado** - O valor da taxa de aprendizado tem grande influência durante o treinamento. Um valor muito baixo poderá causar um treinamento demorado, diminuindo o desempenho da rede. Um valor alto, poderá causar oscilações no treinamento, impedindo a convergência do resultado. A literatura recomenda que a taxa de aprendizado seja definido por um valor que se encontre no intervalo $[0.1, 1]$.
- **Número de ciclos como critério de parada** - Um número alto de ciclos poderá causar *overfitting*, ou seja, a memorização dos dados. Por outro lado um número pequeno de ciclos poderá fazer com que a rede não chegue em seu melhor desempenho. É recomendado que a quantidade de ciclos fique entre 1000 e 3000.
- **Erro médio quadrático como critério de parada** - É sugerido que o valor do erro médio quadrático inicialmente se encontre entre 0,01. A medida que a rede é treinada, esse valor poderá ser ajustado.

Todos os modelos estruturados contém 6 neurônios na camada de entrada e 1 neurônio na camada de saída. Seguindo as sugestões de desenvolvimento apresentadas, as redes foram, inicialmente, definida com as seguintes configurações:

- Quantidade de camadas oculta: 1;
- Quantidade de neurônios na camada oculta: 13;
- Taxa de aprendizado: 0,3;
- Número máximo de ciclos ou repetições: 3000;
- Valor do erro médio quadrático mínimo: 0,01;

A implementação da rede foi realizada utilizando o Matlab, através de seu toolbox de redes neurais denominado **nn toolbox** (*Neural Network Toolbox*) versão 8.0. O **nn toolbox** permite importar, criar, utilizar e exportar dados das redes neurais implementadas. No final da implementação o toolbox oferece os sinais de entrada

e saída, a figura da rede implementada, os valores dos erros e gráficos de performance e desempenho da rede.

Em determinado momento do projeto foi necessário implementar a rede utilizando Java, linguagem de programação desenvolvida pela Sun Microsystems e mantida pela Oracle. A necessidade de implementação das redes em uma linguagem de programação como o Java veio do desejo de analisar todo o procedimento da entrada das variáveis até o valor de saída, possibilitando o acompanhamento das alterações ocorridas nos pesos sinápticos e do valor do gradiente de cada neurônio. Além disso, uma implementação sem o auxílio de um framework ou um toolbox oferece uma completa compreensão de todo o procedimento existente em uma rede neural MLP.

5.4.3 Treinamento e aprendizagem

O treinamento e aprendizagem da rede foram realizados utilizando o algoritmo de retro-propagação de erros, o *backpropagation*, onde os sinais são retro-propagados, ajustando os devidos pesos de cada neurônio. Os dados utilizados em cada modelo foram preparados e separados para utilização. No total foram 816 registros de entradas e saídas, onde cada registro era adaptado para os 4 modelos estruturados, executados separadamente.

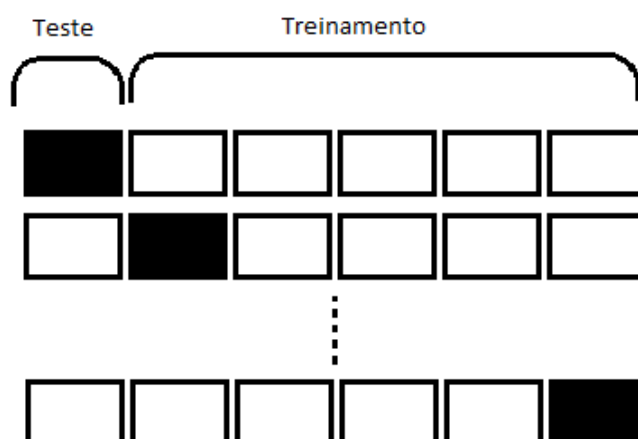
Duas importantes técnicas de análise de resultados foram utilizadas no treinamento e aprendizagem da rede. A *Cross Validation* (validação cruzada) e a matriz de confusão.

5.4.3.1 *Cross Validation* - Validação Cruzada

A técnica da validação cruzada tem como objetivo analisar a capacidade de generalização dos dados utilizados em projetos com redes neurais. Seu conceito principal é a partição dos dados em subconjuntos, utilizando um dos subconjuntos para o teste dos dados e o restante para o treinamento. Após esse procedimento, o subconjunto separado para teste retorna para os subconjuntos de treinamento, e outro subconjunto é separado para análise no teste. No final do processo todos os elementos dos conjuntos participarão das etapas de testes e treinamento. O uso de validação cruzada é atrativo particularmente quando temos que projetar uma rede neural grande cujo objetivo seja uma boa generalização (HAYKIN, 1998). A Figura 13 ilustra o processo de validação cruzada.

5.4.3.2 Matriz de Confusão

A Matriz de Confusão, também conhecida como tabela de contingências, é um modelo de matriz que permite visualizar as principais confusões presentes nos dados,

Figura 13 – Ilustração da técnica de *Cross Validation*

oferecendo uma medida efetiva de classificação, relacionando os resultados esperados x resultados obtidos e ajudando na compreensão e análise de cada saída correlacionadas às demais. É um modelo bastante utilizado em validação de dados em treinamentos supervisionados. Exemplos de matrizes de confusão podem ser visualizados em Tabela 1 e Tabela 2. Para todos os subconjuntos de treinamento utilizado com a validação cruzada, foram gerados suas respectivas matrizes de confusão.

Tabela 1 – Matriz de confusão de 3 classes G1, G2 e G3 com 100% de acerto

	G1	G2	G3
G1	100	0	0
G2	0	100	0
G3	0	0	100

Tabela 2 – Matriz de confusão de 3 classes G1, G2 e G3 com erros de classificação

	G1	G2	G3
G1	80	20	0
G2	10	90	0
G3	0	0	100

5.4.4 Resultados obtidos

Conforme descrito na subseção anterior, os quatros modelos estruturados foram executados separadamente. Para a análise da convergência, foram seguidos os critérios de paradas de acordo ao número de ciclos ou repetições e do valor de erro médio quadrático mínimo. Quando um dos critérios eram atingidos o treinamento da rede era interrompido e seu teste era executado. Após o fim da execução, os resultados eram comparados aos valores reais, podendo então, realizar as avaliações em torno de cada

um dos modelos. A Tabela 3 representa a relação entre os modelos, as variáveis e os resultados obtidos.

Tabela 3 – Relação entre modelos, variáveis e resultados

Modelos	Entrada	Saída	Resultado
Experimento	DIPM, DIPR, DIMR, AP, VMP, categoria	ELPL, ELPR, ERPL, ERPR	Insatisfatório
Categoria	DIPM, DIPR, DIMR, AP, VMP, experimento	Sub13, Sub15, Sub17, Sub20	Insatisfatório
Passe livre/restrito	DIPM, DIPR, DIMR, AP, VMP, categoria	ELPL/ERPL, ELPR/ERPR	Insatisfatório
Espaçamento	DIPM, DIPR, DIMR, AP, VMP, categoria	ELPL/ELPR, ERPL/ERPR	satisfatório

Os três primeiros modelos (classificação por experimento, classificação por categoria e classificação por restrição de passe) não apresentaram resultados satisfatórios, resultando em divergência nas suas saídas, quando comparado com as saídas reais, impossibilitando de realizar qualquer tipo classificação. Como o valor de erro médio quadrático mínimo não foi atingido, o critério de parada para essa situação foi estabelecido pelo número de ciclos. Esta é uma divergência causada principalmente pelo valor do gradiente descendente, que atinge seu ponto mínimo de convergência, fazendo com que seus valores em ciclos posteriores se tornem desfavoráveis para os valores de saída. Assim, podemos afirmar que **não existe separabilidade ou uma relação classificatória entre os experimentos, entre as categorias e entre a restrição dos passes** quando utiliza as variáveis discretas calculadas para entrada da rede.

Já o modelo de classificação por espaçamento apresentou convergência em seus resultados, permitindo uma análise mais detalhada. A matriz de confusão para o espaçamento é apresentada na Tabela 4.

Tabela 4 – Matriz de confusão para classificação por espaçamento

<i>Experimento</i>	ELPL	ELPR	ERPL	ERPR
ELPL	95	91	11	2
ELPR	84	93	21	4
ERPL	2	10	111	78
ERPR	1	8	105	99

Através da matriz de confusão é possível verificar uma separação entre os experimentos, relacionando diretamente ao espaçamento. Ou seja, é possível realizar uma classificação dos dados pelo espaçamento livre (ELPL e ELPR) ou pelo espaçamento restrito (ERPL e ERPR). Dessa forma, podemos afirmar que **existe uma separabilidade ou uma relação classificatória entre os espaçamentos** quando utiliza as variáveis discretas calculadas para entrada da rede.

Calculando a acurácia da rede no método estruturado de espaçamento temos:

$$ac = \frac{756}{815} = 0.92 \quad (16)$$

Onde 756 representa o total de acertos e 815 o total de registros. A relação dos pesos sinápticos apresentado pela rede no método estruturado de espaçamento é apresentado na Tabela 5:

Tabela 5 – Relação dos pesos sinápticos

Variável	Peso
DIPR	2.115
DIPM	1.544
DIMR	1.212
AP	1.172
VMP	1.062
Categoria	1.057

É possível verificar que a categoria e a velocidade média do passe, com valores de 1.057 e 1.062 respectivamente, não possuem influência direta no resultado da rede. Podemos dizer então que, mesmo que retirássemos essas duas variáveis de entrada, a classificação seria possível, sem sofrer divergência em suas saídas. Assim, o experimento apresentou os seguintes resultados:

- Não existe uma relação classificatória entre as categorias sub-13, sub-15, sub-17 e sub-20 quando analisado a distância interpessoal, o ângulo de passe, a velocidade média do passe e o tipo de experimento.
- Não existe uma relação classificatória entre os diferentes experimentos propostos quando analisado a distância interpessoal, o ângulo de passe, a velocidade média do passe e a categoria.
- Não existe uma relação classificatória entre as restrições temporais de passe, quando analisado a distância interpessoal, o ângulo de passe e a velocidade média do passe.
- Existe uma relação classificatória entre as manipulações espaciais, relacionados aos espaçamento livre ou restrito, quando analisado como variáveis a distância interpessoal, o ângulo de passe e a velocidade média do passe.
- As variáveis de categoria e velocidade média do passe não apresenta influência direta quando se deseja realizar uma classificação analisando os espaçamentos.
- A Rede Neural possibilita analisar a importância de cada variável de entrada no resultado de saída, de acordo com seus respectivos pesos.

- A Rede Neural considera cada entrada, verificando a relação que estabelece entre os parâmetros analisados.

6 Conclusão

A análise comportamental de atletas realizadas sem auxílio tecnológico por pedagogo ou especialistas esportivos é de alta complexidade, oferecendo grandes oportunidades para falhas ou imprecisões. Além disso, os resultados encontrados apresentam alto grau de subjetividade.

A utilização de ferramentas computacionais no processo de ensino aprendizagem na pedagogia do esporte e na análise comportamental de seus participantes é de grande valia quando se deseja analisar detalhadamente o contexto de uma determinada situação. Atualmente muitos clubes e federações utilizam de sistemas que buscam de alguma forma modelar as atividades realizadas afim de obter o máximo de seus atletas.

Particularmente ao mapeamento apresentado, através das variáveis discretas oferecidas e utilizando RNA, não foi possível estabelecer relações classificatórias quanto as diferentes categorias e experimento. Já em referência as manipulações diretas no espaçamento, a rede obteve resultados satisfatórios, com alta porcentagem de acerto. Observa-se ser um resultado já esperado, pois com o espaçamento livre, teoricamente, as distâncias interpessoais sofreriam alterações. Mas tais resultados demonstram que, quando é oferecida a oportunidade ao atleta, sua estratégia é modificada, explorando as variadas possibilidades. Além disso, a RNA foi capaz de afirmar que duas das seis variáveis discretas não apresentaram influência direta nos resultados, contrariando alguns estudos já realizados.

Uma característica importante na RNA é que ela consegue oferecer, através dos pesos calculados, a importância de cada variável no resultado apresentado, possibilitando ajustes diretos nas entradas, enriquecendo o mapeamento realizado.

6.1 Trabalhos Futuros

Dando continuidade ao estudo, alguns trabalhos são propostos:

1. Aumento do número de dados, experimentos ou variáveis de entrada para obtenção de resultados com maior generalidade.
2. Utilização de outras ferramentas de reconhecimento e classificação de padrões, como por exemplo as SVM (Support Vector Machine), comparando ou trabalhando em conjunto com os resultados alcançados pelas Redes Neurais Artificiais.
3. Realização do experimento em outros esportes como vôlei ou basquetebol, utilizando diferentes variáveis de entrada.

4. Utilizar os dados obtidos do experimento para a implementação de um classificador de padrões em outras situações presente no contexto esportivo do bobinho, aproveitando variáveis contínuas como centroide, que é a média de posicionamento entre os passadores, e índice de dispersão, que é a relação entre o centroide e o posicionamento dos jogadores.

7 REFERÊNCIAS

ARAÚJO, D.; DAVIDS, K.; HRISTOVSKI, R. The ecological dynamics of decision making in sport. *Psychology of Sport and Exercise*, Amsterdam, v. 7, n. 6, p. 653-676, 2005.

ARAÚJO, D.; DAVIDS, K. Embodied cognition and emergent decision-making in dynamical movement systems. *Junctures*, v.2, p.45-57, 2004.

CHOI, M.S.; KIM, W.Y. A novel two stage template matching method for rotation and illumination invariance. *Pattern Recognition*. p.119-129, 2000.

CHOW, J. Y.; DAVIDS, K.; BUTTON, C.; KOH, M. Coordination changes in a discrete multi-articular action as a function of practice. *Acta Psychologica*, v.127, p.163-176, 2008.

CHOW, J. Y.; DAVIDS, K.; BUTTON, C.; SHUTTLEWORTH, R.; RENSHAW, I.; ARAÚJO, D. Nonlinear pedagogy: a constraints-led framework for understanding emergence of game play and movement skills. *Nonlinear Dynamics, Psychology, and Life Sciences*, v.10, n.1, p.71-103, 2006.

CHOW, J. Y.; DAVIDS, K.; HRISTOVSKI, R.; ARAÚJO, D.; PASSOS, P. Nonlinear pedagogy: learning design for self-organizing neurobiological systems. *New Ideas in Psychology*, v.29, p.189-200, 2011.

CONNEL, S.D; JAIN, A.K; Template-based online character recognition. *Pattern Recognition Society*. Published by Elsevier Science, 2000.

CORREIA, V.; ARAÚJO, D.; DUARTE, R.; TRAVASSOS, B.; PASSOS, P.; DAVIDS, K. Changes in practice task constraints shape decision-making behaviours of team games players. *Journal of Science and Medicine in Sport*, v.15, n.3, p.244-249, 2012.

DAVIDS, K.; ARAÚJO, D.; VILAR, L.; RENSHAW, I.; PINDER, R. An ecological dynamics approach to skill acquisition: implications for development of talent in sport. *Talent Development & Excellence*, v.5, n.1, p.21-34, 2013.

DAVIDS, K., GLAZIER, P., ARAÚJO, D.; BARTLETT, R. Movement systems as dynamical systems. *Sports Medicine*, v.33, n.4, p.245-260, 2003

DAVIDS, K.; HANDFORD, C.; WILLIAMS, M. The natural physical alternative to cognitive theories of motor behaviour: an invitation for interdisciplinary research in sports science? *Journal of Sports Sciences*, v.12, n.6, p.495-528, 1994.

DASH M.; H. Lui. Dimensionality Reduction for Unsupervised Data. *Ninth IEEE International Conference on Tools with*, 1997.

DUARTE, R.; ARAÚJO, D.; FERNANDES, O.; FONSECA, C.; CORREIA, V.; GAZIMBA, V.; TRAVASSOS, B.; ESTEVES, P.; VILAR, L.; LOPES, J. Capturing complex human behaviors in representative sports contexts with a single câmera. *Medicina (Kaunas)*, v.46, n.6, p.408-414, 2010.

DUARTE, R.; ARAÚJO, D.; GAZIMBA, V.; FERNANDES, O.; FOLGADO, H.; MARMELEIRA, J.; DAVIDS, K. The Ecological Dynamics of 1v1 Sub-Phases in Association Football. *The Open Sports Sciences Journal*, v.3, p.16-18, 2009.

DUDA, R. O.; HART, P. E.; STORK, D. G. *Pattern Classification (2nd Edition)*. 2. ed. Wiley-Interscience, 2000.

FRIEDMAN, N.; GEIGER, D.; GOLDSZMIDT, M. Bayesian network classifiers. In: *Machine Learning*. [S.l.: s.n.], 1997.

GALATTI, L. R. FERREIRA, H. B. SILVA, Y. P. G. PAES, R. R. *Pedagogia do esporte: procedimentos pedagógicos aplicados aos jogos esportivos coletivos*. *Revista da faculdade de Educação Física da Unicamp, Campinas*, v. 6, ed. especial, p. 397-408, jul. 2008.

GREHAIGNE, J-F.; BOUTHIER, D.; DAVID, B. Dynamic-system analysis of opponent relationships in collective actions in soccer. *Journal of Sports Sciences*, v.15, p.137-149, 1997.

HAYKIN, S. *Redes Neurais: Princípios e práticas*. 2. ed. [S.l.], 1998: trad: Paulo Martins Rangel - Porto Alegre: Bookman, 2001.

HEADRICK, J.; DAVIDS, K.; RENSHAW, I.; ARAÚJO, D.; PASSOS, P.; FERNANDES, O. Proximity-to-goal as a constraint on patterns of behaviour in attacker-defender dyads in team games. *Journal of Sports Sciences*, v.30, n.3, p.247-253, 2012.

KANSO, M.A; *Redes Neurais*, 1997.

KELSO, J. A. S. *Dynamic patterns*. Cambridge, MA: MIT Press, 1995.

KUDO M.; Sklansky J. Comparasion of Algorithms that Select Features for Pattern Classifiers. *Pattern Recognition*, 2000.

KUGLER, P. N.; TURVEY, M. T. *Information, Natural Law, and the SelfAssembly of Rhythmic Movement*. Hillsdale, NJ: Lawrence Erlbaum Associates Inc, 1987.

MARDIA K.V; Kent J.T; Bibby J.M. *Multivariate Analysis*. Academic Press Inc. (London) Ltd. 1979.

MARIN L.O., *Técnicas Estatísticas para Reconhecimento e Detecção de Face*, 2001.

MARQUE, S., *Reconhecimento de Padrões*, Faculdade de Engenharia da

Universidade do Porto, 2000.

MCGARRY, T. Applied and theoretical perspectives of performance analysis in sport: scientific issues and challenges. *International Journal of Performance Analysis of Sport*, v.9, p.128-140, 2009.

MCGARRY, T.; ANDERSON, D. I.; WALLACE, S. A.; HUGHES, M. D.; FRANKS, I. M. Sport competition as a dynamical self-organizing system. *Journal of Sports Sciences*, v.20, p.771-781, 2002.

MENUCHI, M. R. T. P.; GALATTI, L. R.; NASCIMENTO, J. V. Formação em Educação Física e a organização do ambiente de aprendizagem na iniciação esportiva. In: NASCIMENTO, J. V. (Org). *Formação em Educação Física VI*. UFSC, 2014.

MENUCHI, M. Padrões de coordenação interpessoal atacante-defensor no futsal. *Revista Mineira de Educação Física*, n.9, p. 515-521, 2014.

NEWELL, K. M. Constraints on the development of coordination. In: WADE, M.; WHITING, H. T. A. (Orgs.). *Motor development in children: aspects of coordination and control*. Dordrecht: Martinus Nijhoff, 1986, p.341-359.

OULLIER, O.; KELSO, J. A. S. Neuroeconomics and the metastable brain. *Trends in Cognitive Science*, v.10, p.353-354, 2006.

NIEVOLA, Júlio C. *Redes Competitivas de Kohonen*, Pontifícia Universidade Católica do Paraná, 1996.

PANERO, F.S; VIEIRA, M. D.;CRUZ, A. M.; MOURA, M. D. Application of the exploratory analysis od data in the geographical discrimination of okra of Rio Grande do Norte and Pernambuco, 2009.

PASSOS, P.; ARAÚJO, D.; DAVIDS, K.; GOUVEIA, L.; SERPA, S.; MILHO, J.; FONSECA, S. Interpersonal pattern dynamics and adaptive behavior in multiagent neurobiological systems: conceptual model and data. *Journal of Motor Behavior*, v.41, n.5, p.445-459, 2009.

PASSOS, P.; BATALAU, R. GONÇALVES, R. *Comparação entre as abordagens ecológica e cognitivista para o treino da tomada de decisão no ténis e no rugby*. Lisboa: Universidade Lusófona de Humanidades e Tecnologias, 2006.

RENSHAW, I.; CHOW, J-Y.; DAVIDS, K.; HAMMOND, J. A constraints-led perspective to understanding skill acquisition and game play: a basis for integration of motor learning theory and physical education praxis? *Physical Education and Sport Pedagogy*, v.15, n.2, p.117-137, 2010.

ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, v. 65, n. 6, nov 1958.

SCHMIDT, R. C.; O'BRIEN, B.; SYSKO, R. Self-organization of betweenpersons cooperative tasks and possible applications to sport. *International Journal of Sport Psychology*, v.30, p.558-579, 1999.

SCHOLKOPT, B.; KNIRSCH, P., SMOLA, A., & BURGESS, C. Fast approximation of support vector kernel expansions, and an interpretation of clustering as approximation in feature spaces. *Proceedings of the DAGM Symposium Mustererkennung*, 1998.

THEODORIDIS, S.; KOUTROUMBAS, K. *Pattern Recognition*, Third Edition. Orlando, FL, USA: Academic Press, Inc., 2005.

TODESCO, J.L. Reconhecimento de Padrões usando Redes Neurais Artificiais com uma função de Base Radial. Uma aplicação na Classificação de Cromossomos Humanos. Tese (Doutorado - Programa de Pós Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina), 1995.

TURVEY, M. T. Action and perception at the level of synergies. *Human Movement Science*, v.26, p.657-697, 2007.

TURVEY, M. T.; SHAW, R. Ecological foundations of cognition I: symmetry and specificity of animal-environment systems. *Journal of Consciousness Studies*, v.6, n.11-12, p.95-110, 1999.

VILAR, L.; ARAÚJO, D.; DAVIDS, K.; TRAVASSOS, B. Constraints on competitive performance of attacker-defender dyads in team sports. *Journal of Sports Sciences*, v.30, n.5, 2012

VILAR, L.; ARAÚJO, D.; DAVIDS, K.; TRAVASSOS, B.; DUARTE, R.; PARREIRA, J. Interpersonal coordination tendencies supporting the creation/prevention of goal scoring opportunities in futsal. *European Journal of Sport Science*, v.14, n.1, 2014.

WANGENHEIM, ALDO; Reconhecimento de Padrões. Wangenheim (UFSC), 2006. Site: <http://www.inf.ufsc.br/patrec/>, Acesso em junho/2014.

WIDROW B.; STEARNS S.D. *Adaptive Signal Processing*. 1st edition, 1985.

WINTER, D. A. *Biomechanics and Motor Control of Human Movement*. New York: John Wiley & Sons, Inc, 2005.

Anexos

ANEXO A – Rotina para cálculo das variáveis

```
% Marcos Rodrigo T. P. Menuchi
% Tadeu Nogueira Costa de Andrade
% Ralli bobinho filtragem Butterworth
% 06/04/2015

clear all

%% carregando arquivos do Rali
dados = load('LOCAL_CARREGAR_ARQUIVO_GERADO_PELO_DVIDEO', 'txt');
%converte o arquivo em uma matriz x,y
[m,n] = size(dados);

% frequencia para calculo do butterworth
freq=30;
fim_rali = m;

%Nomeando as colunas para cada jogador e declarando a
%matriz de uma linha
X1 = fim_rali:1;
Y1 = fim_rali:1;
X2 = fim_rali:1;
Y2 = fim_rali:1;
X3 = fim_rali:1;
Y3 = fim_rali:1;
X4 = fim_rali:1;
Y4 = fim_rali:1;
X5 = fim_rali:1;
Y5 = fim_rali:1;
X6 = fim_rali:1;
Y6 = fim_rali:1;

%variaveis
DI_P1_P2 = fim_rali:1;
DI_P1_P3 = fim_rali:1;
DI_P1_P4 = fim_rali:1;
DI_P1_P5 = fim_rali:1;
DI_P1_P6 = fim_rali:1;
DI_P1_CENTROIDE = fim_rali:1;
DI_P2_P3 = fim_rali:1;
DI_P2_P4 = fim_rali:1;
DI_P2_P5 = fim_rali:1;
DI_P2_P6 = fim_rali:1;
```

```
DI_P2_CENTROIDE = fim_rali:1;
DI_P3_P4 = fim_rali:1;
DI_P3_P5 = fim_rali:1;
DI_P3_P6 = fim_rali:1;
DI_P3_CENTROIDE = fim_rali:1;
DI_P4_P5 = fim_rali:1;
DI_P4_P6 = fim_rali:1;
DI_P4_CENTROIDE = fim_rali:1;
DI_P5_P6 = fim_rali:1;
DI_P5_CENTROIDE = fim_rali:1;
DI_P6_CENTROIDE = fim_rali:1;
VE_P1_P2 = fim_rali:1;
VE_P1_P3 = fim_rali:1;
VE_P1_P4 = fim_rali:1;
VE_P1_P5 = fim_rali:1;
VE_P1_P6 = fim_rali:1;
AN_P1_P2_P5 = fim_rali:1;
AN_P1_P5_P2 = fim_rali:1;
AN_P2_P1_P5 = fim_rali:1;
AN_P1_P2_P3 = fim_rali:1;
AN_P2_P1_P3 = fim_rali:1;
AN_P1_P3_P2 = fim_rali:1;
AN_P3_P1_P4 = fim_rali:1;
AN_P1_P4_P3 = fim_rali:1;
AN_P1_P3_P4 = fim_rali:1;
AN_P5_P1_P4 = fim_rali:1;
AN_P1_P4_P5 = fim_rali:1;
AN_P1_P5_P4 = fim_rali:1;
AN_P1_P4_P2 = fim_rali:1;
AN_P1_P3_P5 = fim_rali:1;
AN_P1_P2_P4 = fim_rali:1;
AN_P1_P5_P3 = fim_rali:1;
DI_id_centroide = fim_rali:1;

%centroide x e y
cX = fim_rali:1;
cY = fim_rali:1;

%indice de dispersao
idX = fim_rali:1;
idY = fim_rali:1;

for i=1:fim_rali
    % preenchimento dos jogadores com suas colunas
    X1(i,1) = dados(i,2);
    Y1(i,1) = dados(i,3);
    X2(i,1) = dados(i,4);
```

```

Y2(i,1) = dados(i,5);
X3(i,1) = dados(i,6);
Y3(i,1) = dados(i,7);
X4(i,1) = dados(i,8);
Y4(i,1) = dados(i,9);
X5(i,1) = dados(i,10);
Y5(i,1) = dados(i,11);
X6(i,1) = dados(i,12);
Y6(i,1) = dados(i,13);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Filtrando os dados com filtro Butterworth

n=4;                                %filtro de 4a ordem
Wn = 5/(freq/2);                    %freq de corte/(freq coleta/2)
[b,a] = butter(n,Wn);               %filtro butterworth
x1_f = filtfilt(b,a,X1);
y1_f = filtfilt(b,a,Y1);
x2_f = filtfilt(b,a,X2);
y2_f = filtfilt(b,a,Y2);
x3_f = filtfilt(b,a,X3);
y3_f = filtfilt(b,a,Y3);
x4_f = filtfilt(b,a,X4);
y4_f = filtfilt(b,a,Y4);
x5_f = filtfilt(b,a,X5);
y5_f = filtfilt(b,a,Y5);
x6_f = filtfilt(b,a,X6);
y6_f = filtfilt(b,a,Y6);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%plotando o grafico com os jogadores X1,X2,X3,X4,X5 e a bola X6 FILTRADO
plot(x1_f,y1_f, 'red',x2_f,y2_f, 'black',x3_f,y3_f, 'green',x4_f,y4_f,
'blue',x5_f,y5_f, 'yellow', x6_f,y6_f, 'magenta');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculo das variaveis
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for i=1:fim_rali

    % calculo da centroide
    cX(i,1) = (x2_f(i,1) + x3_f(i,1) + x4_f(i,1) + x5_f(i,1) )/4;
    cY(i,1) = (y2_f(i,1) + y3_f(i,1) + y4_f(i,1) + y5_f(i,1) )/4;

```

```

% calculo do indice de dispersao
idX(i,1) = (1/4) * ( abs(x2_f(i,1) - cX(i,1)) + abs(x3_f(i,1) - cX(i,1)) +
abs(x4_f(i,1) - cX(i,1)) + abs(x5_f(i,1) - cX(i,1)) );
idY(i,1) = (1/4) * ( abs(y2_f(i,1) - cY(i,1)) + abs(y3_f(i,1) - cY(i,1)) +
abs(y4_f(i,1) - cY(i,1)) + abs(y5_f(i,1) - cY(i,1)) );

DI_id_centroide(i,1) = ((idX(i,1) - cX(i,1))^2 + ((idY(i,1) - \
cY(i,1))^2))^(1/2);

% calculo das distancias interpessoais
%%INICIO%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
DI_P1_P2(i,1) = ((x1_f(i,1) - x2_f(i,1))^2 + ((y1_f(i,1) -
y2_f(i,1))^2))^(1/2);
DI_P1_P3(i,1) = ((x1_f(i,1) - x3_f(i,1))^2 + ((y1_f(i,1) -
y3_f(i,1))^2))^(1/2);
DI_P1_P4(i,1) = ((x1_f(i,1) - x4_f(i,1))^2 + ((y1_f(i,1) -
y4_f(i,1))^2))^(1/2);
DI_P1_P5(i,1) = ((x1_f(i,1) - x5_f(i,1))^2 + ((y1_f(i,1) -
y5_f(i,1))^2))^(1/2);
DI_P1_P6(i,1) = ((x1_f(i,1) - x6_f(i,1))^2 + ((y1_f(i,1) -
y6_f(i,1))^2))^(1/2);
DI_P1_CENTROIDE(i,1) = ((x1_f(i,1) - cX(i,1))^2 + ((y1_f(i,1) -
cY(i,1))^2))^(1/2);
DI_P2_P3(i,1) = ((x2_f(i,1) - x3_f(i,1))^2 + ((y2_f(i,1) -
y3_f(i,1))^2))^(1/2);
DI_P2_P4(i,1) = ((x2_f(i,1) - x4_f(i,1))^2 + ((y2_f(i,1) -
y4_f(i,1))^2))^(1/2);
DI_P2_P5(i,1) = ((x2_f(i,1) - x5_f(i,1))^2 + ((y2_f(i,1) -
y5_f(i,1))^2))^(1/2);
DI_P2_P6(i,1) = ((x2_f(i,1) - x6_f(i,1))^2 + ((y2_f(i,1) -
y6_f(i,1))^2))^(1/2);
DI_P2_CENTROIDE(i,1) = ((x2_f(i,1) - cX(i,1))^2 + ((y2_f(i,1) -
cY(i,1))^2))^(1/2);
DI_P3_P4(i,1) = ((x3_f(i,1) - x4_f(i,1))^2 + ((y3_f(i,1) -
y4_f(i,1))^2))^(1/2);
DI_P3_P5(i,1) = ((x3_f(i,1) - x5_f(i,1))^2 + ((y3_f(i,1) -
y5_f(i,1))^2))^(1/2);
DI_P3_P6(i,1) = ((x3_f(i,1) - x6_f(i,1))^2 + ((y3_f(i,1) -
y6_f(i,1))^2))^(1/2);
DI_P3_CENTROIDE(i,1) = ((x1_f(i,1) - x2_f(i,1))^2 + ((y1_f(i,1) -
y2_f(i,1))^2))^(1/2);
DI_P4_P5(i,1) = ((x4_f(i,1) - x5_f(i,1))^2 + ((y4_f(i,1) -
y5_f(i,1))^2))^(1/2);
DI_P4_P6(i,1) = ((x4_f(i,1) - x6_f(i,1))^2 + ((y4_f(i,1) -
y6_f(i,1))^2))^(1/2);
DI_P4_CENTROIDE(i,1) = ((x4_f(i,1) - cX(i,1))^2 + ((y4_f(i,1) -
cY(i,1))^2))^(1/2);

```

```

DI_P5_P6(i,1) = ((x5_f(i,1) - x6_f(i,1))^2 + ((y5_f(i,1) -
y6_f(i,1))^2))^(1/2);
DI_P5_CENTROIDE(i,1) = ((x5_f(i,1) - cX(i,1))^2 + ((y5_f(i,1) -
cY(i,1))^2))^(1/2);
DI_P6_CENTROIDE(i,1) = ((x6_f(i,1) - cX(i,1))^2 + ((y6_f(i,1) -
cY(i,1))^2))^(1/2);
%%FIM%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% calculo angulo de passe
%%INICIO
AN_P1_P2_P5(i,1) = radtodeg(acos(((DI_P1_P5(i,1)^2)-(DI_P2_P5(i,1)^2)-
(DI_P1_P2(i,1)^2))/(-2*DI_P2_P5(i,1)*DI_P1_P2(i,1))));
AN_P1_P5_P2(i,1) = radtodeg(acos(((DI_P1_P2(i,1)^2)-(DI_P2_P5(i,1)^2)-
(DI_P1_P5(i,1)^2))/(-2*DI_P2_P5(i,1)*DI_P1_P5(i,1))));
AN_P2_P1_P5(i,1) = radtodeg(acos(((DI_P2_P5(i,1)^2)-(DI_P1_P5(i,1)^2)-
(DI_P1_P2(i,1)^2))/(-2*DI_P1_P5(i,1)*DI_P1_P2(i,1))));
AN_P1_P2_P3(i,1) = radtodeg(acos(((DI_P1_P3(i,1)^2)-(DI_P2_P3(i,1)^2)-
(DI_P1_P2(i,1)^2))/(-2*DI_P2_P3(i,1)*DI_P1_P2(i,1))));
AN_P2_P1_P3(i,1) = radtodeg(acos(((DI_P2_P3(i,1)^2)-(DI_P1_P3(i,1)^2)-
(DI_P1_P2(i,1)^2))/(-2*DI_P1_P3(i,1)*DI_P1_P2(i,1))));
AN_P1_P3_P2(i,1) = radtodeg(acos(((DI_P1_P2(i,1)^2)-(DI_P2_P3(i,1)^2)-
(DI_P1_P3(i,1)^2))/(-2*DI_P2_P3(i,1)*DI_P1_P3(i,1))));
AN_P3_P1_P4(i,1) = radtodeg(acos(((DI_P3_P4(i,1)^2)-(DI_P1_P4(i,1)^2)-
(DI_P1_P3(i,1)^2))/(-2*DI_P1_P4(i,1)*DI_P1_P3(i,1))));
AN_P1_P4_P3(i,1) = radtodeg(acos(((DI_P1_P3(i,1)^2)-(DI_P3_P4(i,1)^2)-
(DI_P1_P4(i,1)^2))/(-2*DI_P3_P4(i,1)*DI_P1_P4(i,1))));
AN_P1_P3_P4(i,1) = radtodeg(acos(((DI_P1_P4(i,1)^2)-(DI_P3_P4(i,1)^2)-
(DI_P1_P3(i,1)^2))/(-2*DI_P3_P4(i,1)*DI_P1_P3(i,1))));
AN_P5_P1_P4(i,1) = radtodeg(acos(((DI_P4_P5(i,1)^2)-(DI_P1_P4(i,1)^2)-
(DI_P1_P5(i,1)^2))/(-2*DI_P1_P4(i,1)*DI_P1_P5(i,1))));
AN_P1_P4_P5(i,1) = radtodeg(acos(((DI_P1_P5(i,1)^2)-(DI_P4_P5(i,1)^2)-
(DI_P1_P4(i,1)^2))/(-2*DI_P4_P5(i,1)*DI_P1_P4(i,1))));
AN_P1_P5_P4(i,1) = radtodeg(acos(((DI_P1_P4(i,1)^2)-(DI_P4_P5(i,1)^2)-
(DI_P1_P5(i,1)^2))/(-2*DI_P4_P5(i,1)*DI_P1_P5(i,1))));
AN_P1_P4_P2(i,1) = radtodeg(acos(((DI_P1_P2(i,1)^2)-(DI_P2_P4(i,1)^2)-
(DI_P1_P4(i,1)^2))/(-2*DI_P2_P4(i,1)*DI_P1_P4(i,1))));
AN_P1_P3_P5(i,1) = radtodeg(acos(((DI_P1_P5(i,1)^2)-(DI_P3_P5(i,1)^2)-
(DI_P1_P3(i,1)^2))/(-2*DI_P3_P5(i,1)*DI_P1_P3(i,1))));
AN_P1_P2_P4(i,1) = radtodeg(acos(((DI_P1_P4(i,1)^2)-(DI_P2_P4(i,1)^2)-
(DI_P1_P2(i,1)^2))/(-2*DI_P2_P4(i,1)*DI_P1_P2(i,1))));
AN_P1_P5_P3(i,1) = radtodeg(acos(((DI_P1_P3(i,1)^2)-(DI_P3_P5(i,1)^2)-
(DI_P1_P5(i,1)^2))/(-2*DI_P3_P5(i,1)*DI_P1_P5(i,1))));
%%FIM
end

%plot(DI_P5_CENTROIDE,DI_id_centroide,'red');

```

```

for i=1:fim_rali - 1
    % calculo velocidade relativa
    %%INICIO
    VE_P1_P2(i,1) = abs((DI_P1_P2(i+1,1)- DI_P1_P2(i,1))/100)/0.04;
    VE_P1_P3(i,1) = abs((DI_P1_P3(i+1,1)-DI_P1_P3(i,1))/100)/0.04;
    VE_P1_P4(i,1) = abs((DI_P1_P4(i+1,1)-DI_P1_P4(i,1))/100)/0.04;
    VE_P1_P5(i,1) = abs((DI_P1_P5(i+1,1)-DI_P1_P5(i,1))/100)/0.04;
    VE_P1_P6(i,1) = abs((DI_P1_P6(i+1,1)-DI_P1_P6(i,1))/100)/0.04;
    %%FIM
end

%% saida dos dados
nome_arquivo = input('Entre com o nome do arquivo: ', 's');
quant_passe = input('Entre com o numero de passes no rally: ');

% declaracao dos vetores de passe, recepcao e posse de bola
vetor_passe = 1:quant_passe;
vetor_rec = 1:quant_passe;
vetor_posse = 1:quant_passe;
saida = m:n;

for i=1:quant_passe

    % input dos passes de saida da bola
    vetor_passe(1,i) = input(strcat('Entre com o frame do passe _',
    int2str(i), ':'));
    %input da recepcao
    vetor_rec(1,i) = input(strcat('Entre com o frame da recepcao _',
    int2str(i), ':'));

    %input da posse de bola
    vetor_posse(1,i) = input(strcat('Entre com o frame da posse de bola _',
    int2str(i), ':'));

    %jogadores de 2 a 5
    jogador_passando = input('Informe o jogador que esta passando: ');

    jogador_recebendo = input('Informe o jogador que esta recebendo: ');

    %jogador passando 4
    if jogador_passando == 4 && jogador_recebendo == 5
        saida(i,1) = DI_P1_P4(vetor_passe(1,i), 1);
        saida(i,2) = DI_P4_P5(vetor_passe(1,i), 1);
        saida(i,3) = AN_P1_P4_P5(vetor_passe(1,i), 1);
    elseif jogador_passando == 4 && jogador_recebendo == 2
        saida(i,1) = DI_P1_P4(vetor_passe(1,i), 1);
        saida(i,2) = DI_P2_P4(vetor_passe(1,i), 1);
    end
end

```



```
saida(i,3) = AN_P1_P4_P2(vetor_passe(1,i), 1);
elseif jogador_passando == 4 && jogador_recebendo == 3
    saida(i,1) = DI_P1_P4(vetor_passe(1,i), 1);
    saida(i,2) = DI_P3_P4(vetor_passe(1,i), 1);
    saida(i,3) = AN_P1_P4_P3(vetor_passe(1,i), 1);
end

%jogador passando 2
if jogador_passando == 2 && jogador_recebendo == 5
    saida(i,1) = DI_P1_P2(vetor_passe(1,i), 1);
    saida(i,2) = DI_P2_P5(vetor_passe(1,i), 1);
    saida(i,3) = AN_P1_P2_P5(vetor_passe(1,i), 1);
elseif jogador_passando == 2 && jogador_recebendo == 4
    saida(i,1) = DI_P1_P2(vetor_passe(1,i), 1);
    saida(i,2) = DI_P2_P4(vetor_passe(1,i), 1);
    saida(i,3) = AN_P1_P4_P2(vetor_passe(1,i), 1);
elseif jogador_passando == 2 && jogador_recebendo == 3
    saida(i,1) = DI_P1_P2(vetor_passe(1,i), 1);
    saida(i,2) = DI_P2_P3(vetor_passe(1,i), 1);
    saida(i,3) = AN_P1_P3_P2(vetor_passe(1,i), 1);
end

%jogador passando 3
if jogador_passando == 3 && jogador_recebendo == 5
    saida(i,1) = DI_P1_P3(vetor_passe(1,i), 1);
    saida(i,2) = DI_P3_P5(vetor_passe(1,i), 1);
    saida(i,3) = AN_P1_P3_P5(vetor_passe(1,i), 1);
elseif jogador_passando == 3 && jogador_recebendo == 4
    saida(i,1) = DI_P1_P3(vetor_passe(1,i), 1);
    saida(i,2) = DI_P3_P4(vetor_passe(1,i), 1);
    saida(i,3) = AN_P1_P3_P4(vetor_passe(1,i), 1);
elseif jogador_passando == 3 && jogador_recebendo == 2
    saida(i,1) = DI_P1_P3(vetor_passe(1,i), 1);
    saida(i,2) = DI_P2_P3(vetor_passe(1,i), 1);
    saida(i,3) = AN_P1_P3_P2(vetor_passe(1,i), 1);
end

%jogador passando 5
if jogador_passando == 5 && jogador_recebendo == 2
    saida(i,1) = DI_P1_P5(vetor_passe(1,i), 1);
    saida(i,2) = DI_P2_P5(vetor_passe(1,i), 1);
    saida(i,3) = AN_P1_P5_P2(vetor_passe(1,i), 1);
elseif jogador_passando == 5 && jogador_recebendo == 4
    saida(i,1) = DI_P1_P5(vetor_passe(1,i), 1);
    saida(i,2) = DI_P4_P5(vetor_passe(1,i), 1);
    saida(i,3) = AN_P1_P5_P4(vetor_passe(1,i), 1);
elseif jogador_passando == 5 && jogador_recebendo == 3
```

```
saida(i,1) = DI_P1_P5(vetor_passe(1,i), 1);
saida(i,2) = DI_P3_P5(vetor_passe(1,i), 1);
saida(i,3) = AN_P1_P5_P3(vetor_passe(1,i), 1);
end

saida(i,4) = (vetor_posse(1,i) - vetor_rec(1,i)) * 1/30;
di = (vetor_rec(1,i) - vetor_passe(1,i)) * 1/30;

% velocidade
saida(i,5) = (((x6_f(vetor_rec(1,i),1) - x6_f(vetor_passe(1,i),1))^2 +
((y6_f(vetor_rec(1,i),1) - y6_f(vetor_passe(1,i),1))^2))^(1/2))/di;

end

%saida
nome_arquivo = strcat(nome_arquivo, '.xls');
xlswrite(strcat('LOCAL_SALVAR_ARQUIVO_XLS_COM_SAIDAS', nome_arquivo),
saida_pontos_di_centroide);
```