

COMPARAÇÃO ENTRE DOIS MODELOS DE REDES NEURAIS ARTIFICIAIS E MÁQUINAS DE VETORES SUPORTE, APLICADOS À PREDIÇÃO DA POSIÇÃO DA BOLA DA CATEGORIA F180 DA ROBOCUP

ANDRÉ LUIZ CONSTANTINO BOTTA* JOÃO ALBERTO FABRO* CELSO ANTONIO ALVES KAESTNER*

*UTFPR- Universidade Tecnológica Federal do Paraná - Campus Curitiba
 Av. Sete de Setembro, 3165 - Rebouças - CEP 80230-901
 Curitiba - PR - Brasil

Email: alcbotta@gmail.com, fabro@utfpr.edu.br, celsokaestner@utfpr.edu.br

Abstract— This paper presents the evaluation of two models of neural networks, *Back-Propagation* on Multi-Layer Perceptrons, and *Cascade-Correlation* constructive neural network, and Support Vector Machines, applied to the prediction of the ball positioning during robotic soccer games of the F180 category of Robocup. The data for the ball positioning is obtained experimentally using a real-world field, by a top-mounted camera. The data is extracted from the images using the software Tracker. The two models of neural networks and the SVM are compared in the regression problem of predicting the future position of the ball in the field. Experiments show that *Back-Propagation* on Multi-Layer Perceptrons, and Support Vector Machines present better generalization when faced with data not used in the training, while *Cascade-Correlation*, although having less parameters to adjust, tends to overfit.

Keywords— Temporal Series Prediction, Artificial Neural Networks; Support Vector Machines; Robot Soccer.

Resumo— Este trabalho apresenta a avaliação e utilização de dois modelos de redes neurais (*Back-Propagation* em Perceptrons Multi-Camada, e rede neural construtiva *Cascade-Correlation*) e Máquinas de Vetores Suporte (SVMs) para a predição do posicionamento da bola durante jogos de futebol robótico da categoria F180 da Robocup. Os dados utilizados para o treinamento das redes neurais utilizadas para prever o posicionamento bola são obtidos experimentalmente usando um campo real, visualizado por uma câmera posicionada sobre o campo. Os dados são extraídos a partir das imagens, utilizando o software de rastreamento Tracker. Os dois modelos de redes neurais são comparados em termos de precisão do treinamento e facilidade de parametrização. Experimentos mostram que tanto as redes treinadas com o algoritmo *BackPropagation* quanto as SVMs apresentam melhor generalização, quando confrontado com dados que não são utilizados no treinamento, enquanto a rede *Cascade-Correlation*, apesar de possuir menos parâmetros a ajustar, tende ao sobre-treinamento.

Keywords— Previsão de Séries Temporais; Redes Neurais Artificiais; Máquinas de Vetores Suporte; Futebol de Robôs.

1 Introdução

O futebol de robôs tem por objetivo promover pesquisas relacionadas à robótica e à inteligência artificial (Kitano et al., 1997). Atualmente, as categorias que compõem a Robocup Soccer são: simulação 2D e 3D, robôs pequenos (*Small Size* ou F180), robôs médios (*Middle Size*), plataforma padrão (*Standard Platform*) e humanóides. A categoria F180 engloba situações de cooperação multi-agente inteligente em um ambiente altamente dinâmico, sendo que cada time é composto de 6 robôs móveis cilíndricos com 180 mm de diâmetro e 150 mm de altura. O sistema padrão de visão, SSL-Vision, é responsável pela captura dos objetos em campo e posterior envio de informações sobre o posicionamento destes objetos para um computador de cada das equipes, que utilizam estas informações para o devido processamento, possibilitando, assim, a coordenação e controle dos robôs, como mostra a Figura 1. O objetivo deste projeto foi integrar um modelo computacional do comportamento da bola no código de controle dos robôs, afim de possibilitar a realização de comportamentos preditivos pelos robôs, tendo em vista a posição futura da bola

no campo, evitando o comportamento de "contornar a bola" por parte do robô (como apresentado na seção 2). O modelo do comportamento dinâmico da bola foi criado utilizando independentemente duas arquiteturas de redes neurais artificiais, sendo uma o Perceptron de Múltiplas Camadas com o algoritmo *Back-Propagation* e outra uma rede construtiva com o algoritmo *Cascade-Correlation*. Foi também realizado o treinamento utilizando Máquinas de Vetores Suporte, para fins de comparação tanto da facilidade de construção dos modelos, quanto em termos de dificuldade de parametrização e qualidade do modelo obtido em termos de previsão a partir de dados não utilizados na construção dos modelos, em uma aplicação real.

O restante do artigo está organizado como segue. Na seção 2 são apresentadas as características do problema a ser modelado (previsão do comportamento da bola em um campo real). Na seção seguinte (3) são apresentados os modelos a serem comparados neste trabalho, com suas características principais. Na seção 4 são apresentadas a metodologia utilizada, e os experimentos realizados, e seus resultados. E na seção 5 são discutidos os resultados obtidos.

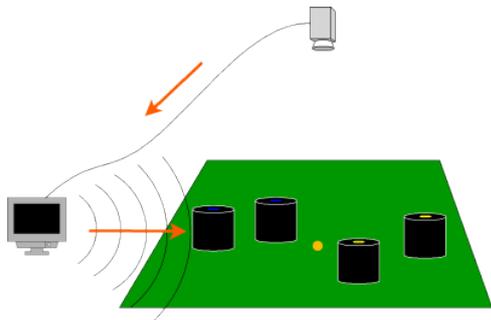


Figura 1: Funcionamento do sistema de visão SSL-Vision.

2 Descrição do Problema

O problema predominante na utilização de uma lógica de ação e reação durante uma partida, reside no fato de que a ação de cada robô é dependente da posição atual da bola, contudo, com a dinâmica do jogo, a bola encontra-se em movimento quase que na totalidade de tempo do jogo, assim sendo, o comportamento efetuado por cada robô torna-se uma trajetória de perseguição à uma bola que não mais se encontra na posição para a qual o robô está deslocando-se. A Figura 2 exemplifica esta proposição utilizando-se de uma escala de tempo imprecisa com o intuito apenas de facilitar o entendimento. A partir do instante de tempo $t = 0$ (s), o robô movimenta-se em direção à bola no mesmo instante de tempo, no instante $t = 1$ (s), a bola está em outra posição, assim sendo, o robô muda sua trajetória em direção a esta, no instante $t = 2$ (s), o robô está em condições de chute, seja em direção ao gol ou de passe para outro integrante de sua equipe, mas para a bola no instante $t = 3$ (s). Consequentemente, do instante $t = 4$ (s) em diante, o robô deslocar-se-á sobre uma trajetória de "seguir" a bola. Esta conduta diminui consideravelmente a possibilidade de ações ofensivas visando o ganho da partida.

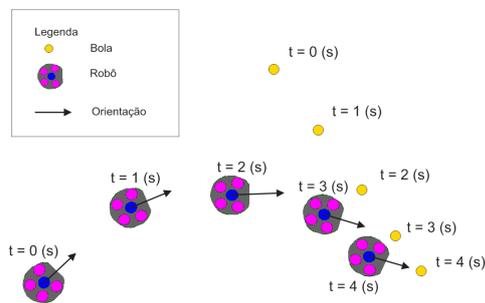


Figura 2: O Problema.

O deslocamento realizado pela bola não pode ser considerado linear. Existe a força de atrito entre esta e o piso, imperfeições do campo que alteram a movimentação da bola, entre outros fatores que inferem a não utilização de cálculos line-

ares mesmo que para uma aproximação do deslocamento da bola.

3 Modelos de Redes Neurais e Máquina de Vetores Suporte utilizados

As redes neurais artificiais simulam o comportamento da estrutura neural de organismos inteligentes com o propósito de adquirir conhecimento através da experiência, podendo ser utilizadas para o controle de processos complexos e não-lineares. Para que a rede neural desempenhe sua função adequadamente, é necessário executar uma etapa de treinamento, concedendo a esta os padrões de aprendizado para o correto refinamento dos pesos das conexões.

Usualmente as redes neurais artificiais são dispostas em camadas, sendo que estas podem ser classificadas como:

- *Camadas de Entrada* : os padrões de treinamento são inseridos nesta camada.
- *Camadas Intermediárias ou Ocultas* : responsáveis pela extração de características da massa de dados.
- *Camada de Saída* : expõe o resultado final.

Neste projeto foram utilizados dois algoritmos de aprendizado supervisionado para redes neurais: *Back-Propagation* (Rumelhart et al., 1986) (com uma rede Perceptron multi-camadas) e *Cascade-Correlation* (Fahlman and Lebiere, 1990), que são apresentados nas sub-seções a seguir. Também é apresentado o modelo estatístico de treinamento supervisionado denominado "Máquina de Vetores Suporte", que têm sido muito utilizado em substituição a modelos neurais em aplicações de **classificação** com sucesso, havendo inclusive autores que identifiquem sua maior capacidade de generalização em relação às redes neurais artificiais (Byvatov et al., 2003) (Lin, 2004) (Tyagi, 2008). Entretanto, como a tarefa a ser realizada neste artigo é de **regressão** visando predição de comportamento da bola, há poucos trabalhos disponíveis na literatura, e ainda não há consenso sobre se o uso de máquinas de vetores suporte apresentam vantagens sobre redes neurais (Balabin and Lomakina, 2011).

3.1 Back-Propagation

Tendo sido proposto para o uso com redes neurais artificiais por Rumelhart (Rumelhart et al., 1986), este algoritmo é um dos mais utilizados em redes de múltiplas camadas que utilizam o neurônio simulado do tipo Perceptron. A técnica de treinamento para este algoritmo consiste na apresentação de um padrão para a camada de entrada da rede neural, decorre-se o processamento deste padrão através das camadas intermediárias até a

conclusão, resultando em um valor na camada de saída. Este valor é comparado ao valor esperado da saída referente a este padrão, e, caso o erro entre ambos esteja acima do erro máximo estipulado, este erro é retro-propagado até a camada de entrada, como ilustrado na Figura 3, e durante este processo, os pesos das conexões das camadas intermediária e de saída são ajustados.

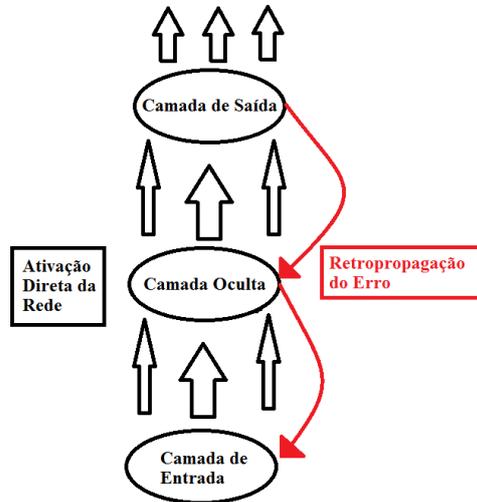


Figura 3: Algoritmo *Back-Propagation*.

3.2 *Cascade-Correlation*

A arquitetura básica desta rede neural artificial fundamenta-se na adição de novas camadas ocultas à rede conforme necessário. Inicialmente a rede neural contém apenas as entradas, saídas e uma unidade de processamento realizando a conexão entre elas. Efetua-se o treinamento com esta rede de camada única. Concluído o treinamento, os valores na camada de saída são avaliados, e caso estejam abaixo do erro máximo estabelecido, o algoritmo cessa, caso contrário, adiciona-se uma nova camada (oculta), e repete-se o processo. A Figura 4 representa o início do algoritmo, caso este não convirja, adiciona-se uma nova camada oculta que recebe todas as entradas e a saída de qualquer outra camada oculta já criada, como mostrado na 5.

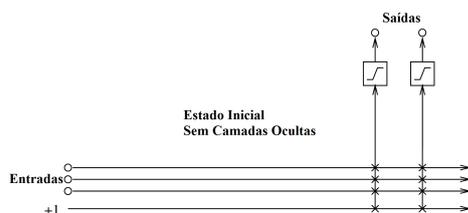


Figura 4: Configuração inicial do algoritmo *Cascade-Correlation*.

É importante atentar para o fato de que o número máximo de camadas ocultas adicionadas

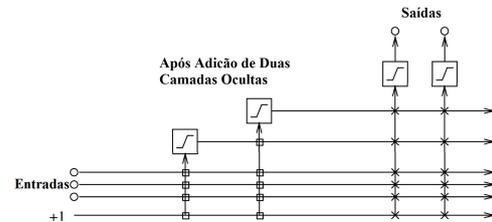


Figura 5: Configuração do algoritmo *Cascade-Correlation* após a inserção de dois neurônios.

também deve ser definido para evitar-se um processo em que as saídas nunca estão abaixo do erro definido, evitando a consequência do algoritmo continuar a criar novas camadas sucessivamente, mesmo sem a ocorrência da convergência das saídas. Desta forma, os únicos parâmetros para o treinamento são o erro máximo e o número máximo de unidades a serem inseridas na rede.

3.3 *Máquina de Vetores Suporte ou Support Vector Machine-SVM*

SVM é um algoritmo de aprendizado supervisionado, que pode ser utilizado tanto para problemas de classificação quanto de regressão. Apesar de poder ser utilizado nas mesmas tarefas que as redes neurais artificiais com treinamento supervisionado, as bases teóricas são estatísticas, sendo que o treinamento é realizado através da otimização de um problema quadrático. Desta forma, uma das propriedades das SVMs é a capacidade de simultaneamente minimizar o erro de classificação, maximizando a distância entre o hiperplano separador das classes. Para que o algoritmo de SVM funcione em problemas não lineares é preciso transformar o espaço inicial dos dados para um espaço de maior dimensão, onde os dados iniciais agora são separáveis linearmente. A transformação de espaço é feita por uma função de *kernel* (ou núcleo) que, aplicada aos dados no espaço original, retorna-os num espaço com mais dimensões. Existem vários tipos de função de kernel, sendo a função polinomial a mais usada (Varpnik, 1999).

A extensão de SVMs para o tratamento de problemas de regressão é feita através do uso de uma função de perda (Smola and Schölkopf, 2004). De forma contrária ao que ocorre no uso de SVMs para classificação (onde se busca forçar as amostras a se posicionarem o mais distante possível do hiperplano ótimo, no espaço de características) o propósito passa a ser de forçar as amostras a se posicionarem o mais próximo possível do hiperplano ótimo. Embora os propósitos da otimização sejam opostos, o objetivo de ambas as formulações é o mesmo: maximizar a capacidade de generalização do modelo para dados não utilizados no treinamento.

4 Desenvolvimento

Com uma câmera de vídeo, ajustada para capturar a área desejada, fez-se testes com uma bola de características oficiais da categoria F180 da Robocup, armazenando cada teste para posterior seleção. Com o auxílio do software *Tracker*¹, analisou-se os vídeos para a obtenção de dados referentes à velocidade, posição e tempo, decorrentes do deslocamento da bola.

4.1 Exame e filtragem dos dados.

Concluída a etapa de filmagem e análise dos vídeos, obteve-se uma tabela com dados brutos relacionando cada instante de tempo a uma posição e a uma velocidade. As Figuras 6 e 7 exibem os gráficos da magnitude da posição em função do tempo e magnitude da velocidade em função do tempo, respectivamente, plotados pelo *Tracker*.

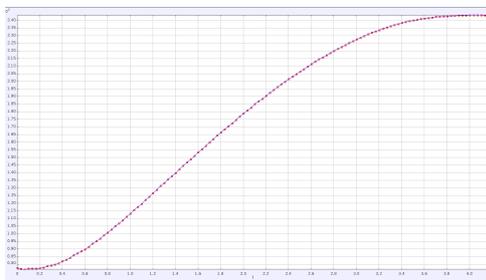


Figura 6: Magnitude da posição em função do tempo.



Figura 7: Magnitude da velocidade em função do tempo.

Após um pré-processamento que consistiu de avaliar a variação de posição e a variação de tempo para cada velocidade pertencente à esta primeira massa de dados, produziu-se um segundo conjunto de dados que relaciona uma mesma velocidade a diferentes variações de tempo e suas respectivas variações de posição. O aspecto relevante para o

¹ *Tracker* é um software livre (*Open Source*) para análise e modelagem de imagens e vídeos que possibilita a obtenção de dados como velocidade e aceleração, tanto linear quanto angular, plotagem de gráficos das grandezas aferidas, mudança do referencial, entre outras características. Disponível em: <http://www.cabrillo.edu/~dbrown/tracker>

movimento da bola diz respeito ao seu deslocamento em função da sua velocidade inicial e uma variação de tempo. Com isso, a velocidade e variação de tempo, contidas no segundo conjunto de dados, exerceram a função de entrada e a variação de posição, deste mesmo conjunto, desempenhou a função de saída no âmbito de parâmetros de treinamento das redes neurais. A finalidade de múltiplas bases de tempo deve-se à possibilidade de se arquitetar um comportamento no qual o robô realize uma série de cálculos com o intuito de adequar o parâmetro tempo viabilizando um melhor posicionamento deste mesmo robô. Da Figura 8 observa-se que o robô está avaliando diferentes instantes de tempo e as respectivas posições deste e da bola. No instante $t = 3$ (s), atenta-se que o robô está em condições favoráveis de interceptação da bola e consequente execução da estratégia predeterminada. Este comportamento não foi implementado pela necessidade da criação de um modelo para cada robô, referente à variação de posição do mesmo em função do tempo, onde a entrada consistiria do parâmetro tempo e a saída representaria o deslocamento, em módulo, efetuado pelo robô.

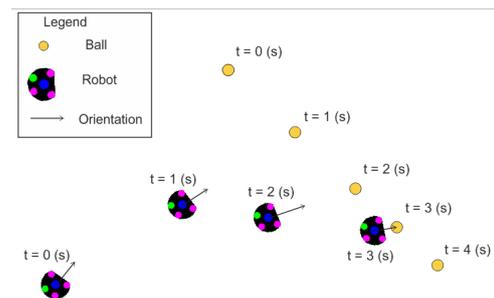


Figura 8: Estimativa realizada pelo robô para diferentes instantes de tempo de sua posição e da posição da bola.

Para serem utilizadas no treinamento da rede neural, as informações presentes no segundo conjunto de dados deveriam estar no intervalo $[-1,1]$, afim de normalizá-los. Assim, utilizando os limitantes correspondentes da Tabela 1, produziu-se a massa de dados final adequada para iniciar a fase de treinamento da rede neural.

Apresentou-se à cada rede neural os padrões de comportamento, entradas e saídas normalizadas, para que esta realizasse o processo de treinamento supervisionado. Ao término, a rede neural passa então a responder, gerando uma saída, com base neste conjunto de dados, mas podendo receber entradas que não foram originalmente fornecidas.

A velocidade máxima oficial da bola durante a competição é de 8 m/s. A dimensões do campo, tanto horizontal quanto vertical, não excedem a 8 metros, ou seja, em menos de 1 segundo a bola, em velocidade máxima, cruza toda a ex-

Tabela 1: Limites de Normalização

Grandeza	Máximo	Mínimo
Variação de Tempo (s)	2	0.5
Velocidade Instantânea (mm/s)	2000	100
Variação de Posição (mm)	1000	90

tensão do campo mesmo que sua posição inicial seja uma das quatro bordas do campo. Consequentemente, a predição da variação da posição da bola na condição de velocidade máxima não apresenta informações de grande valia, sendo mais adequado o uso de valores mais baixos. Desenvolveu-se um software para realizar a análise dos dados, definir seus valores extremos para a normalização, executar o processo de treinamento e exercer um papel de interface, depois de finalizado o treinamento, permitindo assim a passagem de parâmetros, velocidade instantânea e variação de tempo, obtidos em tempo real durante uma partida real ou simulada, para que este retorne a variação de posição estipulada pela rede neural. Este software foi desenvolvido tanto para a rede neural *Back-Propagation* quanto para a *Cascade-Correlation*, alterando-se apenas particularidades do mesmo para o correto relacionamento com o código referente à cada rede neural, mantendo-se sua finalidade inalterada.

4.2 Testes e comparações

O algoritmo *Cascade-Correlation* propicia uma ação de treinamento em que parâmetros como número de neurônios, número de camadas, taxa de aprendizado, entre outros, não precisam ser apresentados para que o treinamento seja efetuado, pois o próprio algoritmo realiza as iterações necessárias para adequar estes parâmetros. Em contrapartida, o algoritmo *Back-Propagation* não possui tal característica, sendo assim, é indispensável o uso de tais parâmetros, o que deve ser feito por meio de testes para se achar uma configuração que obtenha resultados apropriados. Em ambos algoritmos é imprescindível a especificação do erro máximo. A configuração final aplicada às redes neurais está exposta na Tabela 2.

Já no caso das SVMs, utilizou-se a biblioteca *LibSVM* (Chih-Chung and Chih-Jen, 2011) do ambiente Weka (2009). Foram utilizados os parâmetros padronizados desta ferramenta (função *kernel* de base radial, $C=1.0$, função de perda *nu-SVM*). Com os modelos obtidos definiu-se matematicamente a divergência entre os valores gerados por cada modelo quando comparados aos valores esperados. Este processo foi realizado com o conjunto de dados original com o qual as redes neurais foram treinadas e com um segundo conjunto de dados obtidos com o procedimento descrito nos tópicos anteriores desta seção. Estes erros, em seu valor médio, estão expostos na Tabela 3. Observa-

se que o algoritmo *Cascade-Correlation*, obteve melhor desempenho quando testado com o conjunto de dados original, entretanto, a situação se inverte quando os dados fornecidos são referentes a um segundo experimento.

A curva que representa a variação de posição da bola em função do tempo, exposta na Figura 6, exibe característica não-linear. Por meio do software Tracker é possível traçar uma reta com o objetivo de linearizar a curva. Por meio deste processo de linearização, obtém-se uma equação que representa a posição da bola em função do tempo, como mostra a Figura 9. Aplicando-se a equação da reta às diferentes entradas de tempo, e calculando-se o erro entre o valor gerado e o valor esperado, verifica-se um erro médio de 0,20. Este erro é maior do que o erro encontrado nos testes com o algoritmo *Back-Propagation*.

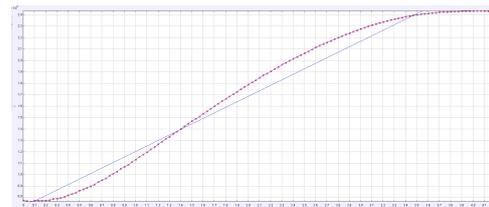


Figura 9: Linearização da curva da magnitude da posição em função do tempo obtida através do Tracker.

5 Discussões e Conclusões

Com a metodologia proposta na Seção 4, desenvolveu-se um modelo referente à variação de posição da bola utilizada na categoria *Small Size* da Robocup. A partir dos parâmetros de entrada, variação de tempo e velocidade, o modelo deve ser capaz de gerar uma saída referente à variação de posição da bola. Está em desenvolvimento um sistema de controle para robôs baseado nestas predições, permitindo uma otimização no comportamento dos robôs se comparado ao sistema de controle que não dispõe do uso de redes neurais. Fazendo-se a comparação entre os resultados gerados pelas redes neurais Perceptron multi-camadas treinada com o algoritmo *Back-Propagation*, a rede construtiva *Cascade-Correlation*, e o modelo SVM, observa-se que tanto a rede *Back-Propagation* quanto o modelo SVM obtiveram resultados semelhantes em termos de generalização do aprendizado obtido nos

Tabela 2: Parâmetros Aplicados.

	<i>Back-Propagation</i>	<i>Cascade-Correlation</i>
momentum	0,1	-
learning rate	0,1	-
max error	0,07	0,07
number of neurons by layer	6	-
number of hidden layers	1	-

Tabela 3: Comparação entre as saídas dos modelos treinados e a saída esperada

	<i>Back-Propagation</i>	<i>Cascade-Correlation</i>	<i>SVM</i>
Erro (Conjunto original de treino)	0,0779	0,0147	0,0470
Erro (Conjunto não apresentado)	0,0934	0,5910	0,0492

dados de treinamento para os dados de testes. A rede *Cascade-Correlation*, apesar de obter o melhor ajuste para os dados de treino, acabou por "decorar" os padrões de treinamento, obtendo resultados insatisfatórios com os dados de testes. Em termos de simplicidade de treinamento, o único parâmetro da rede *Cascade-Correlation* foi o erro desejado para o treinamento, enquanto que tanto o algoritmo *BackPropagation* quanto o SVM necessitam de diversos parâmetros (número de neurônios, taxas e fatores diversos, definição de função de *kernel*, entre outros). Mas, mesmo usando parâmetros *default* da ferramenta Weka, ambos obtiveram bons resultados de generalização nos dados de teste, sem necessidade de ajustes posteriores. O problema de *overfitting* das redes *Cascade-Correlation* têm sido alvo de pesquisas, e a simplicidade que este modelo provê na parametrização merece maior estudo, porém o que se conclui é que para o estudo de caso apresentado neste artigo, tanto a rede *BackPropagation* quanto o algoritmo SVM obtiveram resultados semelhantes, mesmo sem ajustes de seus parâmetros de treinamento.

- Trabalhos Futuros

O modelo treinado foi elaborado exclusivamente para a bola, entretanto, utilizando-se da abordagem definida na seção 4, é possível desenvolver um modelo para o movimento realizado pelos robôs, com isso, o robô não apenas calcula a posição da bola mas também a sua própria posição para o mesmo intervalo de tempo, viabilizando uma melhor conduta por parte do robô, em que este se desloca para a posição que lhe permitiria maiores chances de chute, seja para o gol ou para um companheiro de equipe. Esta etapa está em desenvolvimento, visando integrar a predição de comportamentos ao sistema de controle dos robôs numa próxima edição da Robocup.

Referências

Balabin, R. M. and Lomakina, E. I. (2011). Support vector machine for regression (ls-svm):

An alternative to artificial neural networks (anns) for the analysis of quantum chemistry data?, *Analyst* (136): 1703–1712. DOI: [10.1039/c0an00387e](https://doi.org/10.1039/c0an00387e)

Byvatov, E., Fechner, U., Sadowski, J. and G., S. (2003). Comparison of support vector machine and artificial neural network systems for drug/nondrug classification., *J Chem Inf Comput Sci.* **43**(6). DOI: [10.1021/ci0341161](https://doi.org/10.1021/ci0341161)

Chih-Chung, C. and Chih-Jen, L. (2011). Libsvm: A library for support vector machines, *ACM Transactions on Intelligent Systems and Technology (TIST)* **2**(3).

Fahlman, S. and Lebiere, C. (1990). The cascade-correlation learning architecture, *Advances in Neural Information Processing Systems* (2).

Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., Osawa, E. and Matsubara, H. (1997). *RoboCup A challenge problem for AI.*, Vol. 18, AI Magazine.

Lin, W.-C. (2004). *A case study on support vector machines versus artificial neural networks*, Master's thesis, University of Pittsburg, MSEE - Master of Science in Electrical Engineering.

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986). Learning representations by back-propagating errors, *Nature* (323).

Smola, A. J. and Schölkopf, B. (2004). A tutorial on support vector regression, *Statistics and Computing* **14**(3): 199–222. DOI: [10.1023/B:STCO.0000035301.49549.88](https://doi.org/10.1023/B:STCO.0000035301.49549.88)

Tyagi, S. (2008). A comparative study of svm classifiers and artificial neural networks application for rolling element bearing fault diagnosis using wavelet transform preprocessing, *World Academy of Science, Engineering and Technology* **43**: 309–317.

Varpnik, V. (1999). An overview of statistical learning theory., *IEEE Trans. Neural Net-works* (10): 988-999). DOI: [10.1109/72.788640](https://doi.org/10.1109/72.788640)