Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 133 (2018) 417–423

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Robotics and Smart Manufacturing (RoSMa2018)

Comparative study of neural networks in path planning for catering robots

H.Bharadwaj[a], Vinodh Kumar E[a] *

[a]*School of Electrical Engineering, Vellore Institute of Technology, Vellore-632014, India*

Abstract

Neural Networks (NN) have been the forefront of growth in recent years due to their variety, the opportunities they provide and most importantly their dynamic nature. A control system for catering robots for path planning is proposed with the help of neural networks as a comparative study. Various parameters such as training time, performance of the network, forecasted distance are considered after iterating to obtain the optimal dataset using Probabilistic Roadmap (PRM) algorithm. Approximately 36% improvement in forecasted distance was obtained using neural networks when compared to the traditional PRM algorithm.

Keywords: Neural network; Path planning; catering, robotics

1. Introduction

The world is moving to the age of robotics and automation. With increasing focus on reducing human involvement in mundane tasks, improved algorithms for controlling robots are vital for a wide array of applications. A recently growing trend is the use of robots in catering as waiters as shown by Chen et al. [1]. These usually use programmable logic controllers used by Lama [2] which are not optimized for a dynamic environment such as a restaurant.

Neural networks find application in various fields such as disease detection [3], e-mail and website classification[4], welding saturation control [5], load forecasting [6] etc. These utilize the model-free behavior of

* H.Bharadwaj. Tel.: +91-807-212-9778;
E-mail address:h.bharadwaj2015@vit.ac.in

10.1016/j.procs.2018.07.051

neural networks on situations with extensive data to arrive at optimum results. Neural network based path planning for mobile robotics is a very viable field as seen in Hong [7].

Robots for catering require various features such as stability, communication interface, obstacle avoidance and path planning. With extensive research existing for the other features, path planning offers a new avenue for development. Various path planning algorithms have been compared for realistic indoor application in the past as seen from Pol [8], Galceran[9] and neural networks have been shown to be a viable alternative to traditional methods by Ren et al [10], Li [11].

In this paper the use of heuristic neural network algorithms for robot catering application was compared. Belgith et al showcased the use of various neural network algorithms with existing path planning algorithms such as Probabilistic Roadmap(PRM)[12] using a MATLAB simulation. Results showed the efficiency of Levenberg Marquardt algorithm for this application based on the parameters considered. A wider assortment of neural networks is considered to validate this for a dynamic application considering the performance of networks.

2. Path Planning

2.1. Waypoint generation

Waypoint generation was required to generate the paths which can be used to train the neural network. Initially a binary map had to constructed which marked each location in the map as passable or obstruction. A fixed starting point was chosen for all the iterations but random end points were generated for which path was decided based on Probabilistic Roadmap (PRM) algorithm. PRM requires the user to specify the number of nodes and the connection distance between them as it works by randomly placing nodes into the free space in the map and connecting them to create a path. With high number of nodes and connection distance the PRM solution will be very accurate but will require impractical computational power. The minimum number of nodes and connection distance required to complete the path was considered by increasing the number for particular instance if necessary. The final output was a two dimensional matrix consisting of the path which was vectorized and stored to create the dataset.

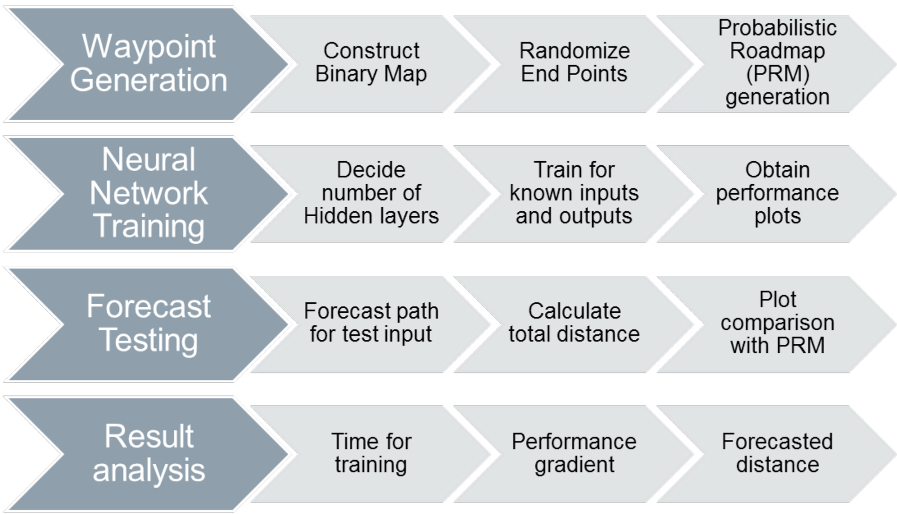

Figure 1: Flowchart

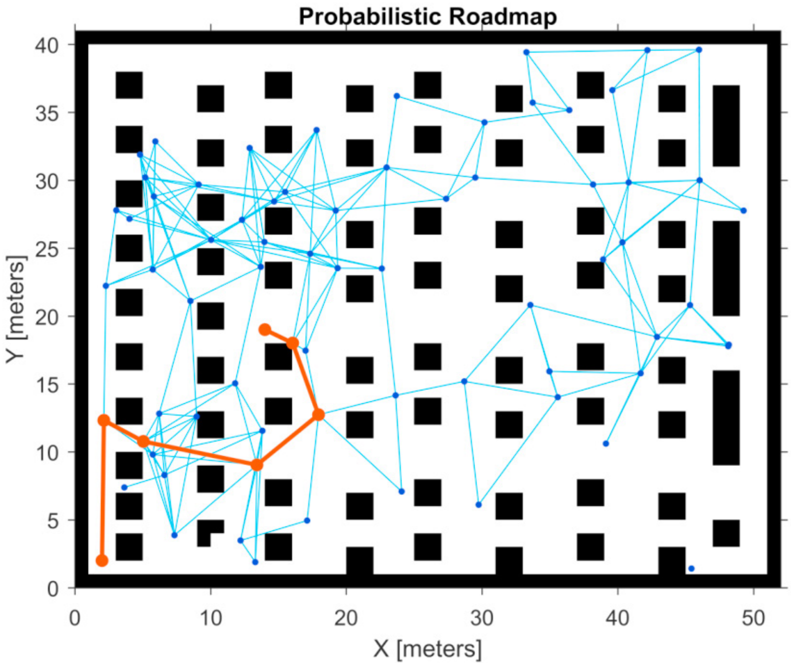

Figure 2: Probabilistic Roadmap

2.2. Neural Network Mathematical Modelling

Neural networks can be broadly classified on functionality for pattern recognition and function approximation and the networks are optimized for it. The weight function update equations which give the coefficients of the neural network models are given for the nine different neural networks which were considered to forecast the path of the catering robot .

2.2.1. Levenberg-Marquardt Algorithm

This algorithm is efficient for small datasets in fitting applications as it avoids the calculation of the Hessian matrix by approximating it through Jacobians. [13][14]

$$\mathrm{X}_{\mathrm{k}+1} = \mathrm{X}_{\mathrm{k}} - [\mathrm{J}^{\mathrm{T}}\mathrm{J} + \mu\mathrm{I}]^{-1}\mathrm{J}^{\mathrm{T}}\mathrm{e} \tag{1}$$

where J =Jacobian Matrix, e = vector of network errors, μ is a scalar

2.2.2. BFGS quasi-Newton backpropagation

The BFGS algorithm requires more computation as it calculates and stores the hessian matrix and is applied to small datasets. It usually converges in lesser iterations.[15]

$$\mathrm{X}_{\mathrm{k}+1} = \mathrm{X}_{\mathrm{k}} - A_k^{-1} g_k \tag{2}$$

where $A_k^{-1} g_k$ is the Newton's direction for weight update

2.2.3. Resilient Backpropagation

Multilayer neural networks utilize sigmoid transfer functions to a compress the inputs into a finite output range. This causes only small changes in weights due to partial derivatives so that the network doesn't converge. Resilient backpropagation eliminates this by determining just the direction of weight update through partial derivative hence it requires lesser memory.[16]

$$dX = deltaX.* \, sign(gX) \tag{3}$$

where $deltaX$ =derivative of performance, gX = gradient, dX = change in weight

2.2.4. Scaled conjugate gradient backpropagation

Basic backpropagation algorithms update weights on the basis of negative of gradient but that doesn't necessarily converge quickly. It implements a combination of model-trust region approach using in Levenberg-Marquardt algorithm with conjugate gradient approach and requires many intermediate updates.[17]

$$X_{k+1} = X_k + \alpha_k p_k \tag{4}$$

where p_k =search direction, α_k = scalar coefficients

2.2.5. Conjugate gradient backpropagation with Powell-Beale restarts

Conjugate gradient algorithms require the direction of the gradient to the set at specified intervals. This method proposes the restart only when there is little orthogonality change in the gradient.[18][19]

2.2.6. Conjugate gradient backpropagation with Fletcher-Reeves updates

Fletcher-Reeves algorithm involves determining the optimal distance such that searches are conjugates of each other by a parameter β.[20]

$$\begin{gathered} X_{k+1} = X_k \alpha_k p_k \\ p_k = -g_k + \beta_k p_{k-1} \\ \beta_k = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}} \end{gathered} \tag{5}$$

where β_k= Fletcher Reeves parameter

2.2.7. Conjugate gradient backpropagation with Polak-Ribiére updates

Polak-Ribiere updates is another conjugate gradient backpropagation algorithm similar to the Fletcher-Reeves with a different parameter β.

$$\beta_k = \frac{\Delta g_k^T g_k}{g_{k-1}^T g_{k-1}} \tag{6}$$

2.2.8. One-step secant backpropagation

It is an extension of the BFGS algorithm to decrease the storage and computation by performing a secant approximation and assuming that the hessian matrix is an identity matrix in each iteration. It implements a combination of conjugate gradient algorithm with quasi-Newton algorithm.[21]

$$dX = -gX + \alpha \, \Delta X_{k-1} + \beta \, \Delta gX_{k-1} \tag{7}$$

where ΔX_{k-1} = change in weights in previous iteration, ΔgX_{k-1}= change in gradient in previous iteration

2.2.9. Gradient descent with momentum and adaptive learning rate backpropagation

This algorithm implements a gradient descent with momentum which prevents the networks from getting stuck in a shallow local minimum.

$$dX = m_c \, \Delta X_{k-1} + lr * m_c * \frac{deltaX}{dX} \quad (8)$$

where m_c= momentum coefficient, lr = learning rate, $deltaX$ = derivative of performance

3. Results and Discussion

The waypoint generation PRM algorithm was performed for multiple number of datasets ranging from 100 to 10000, with number of nodes between 10 to 200 and with inter nodal distance ranging from 5 to 100. The computational requirements which are comparable with neural networks yielded a result at 2000 datasets with initial value of 20 nodes and maximum nodal distance of 10 without overfitting.

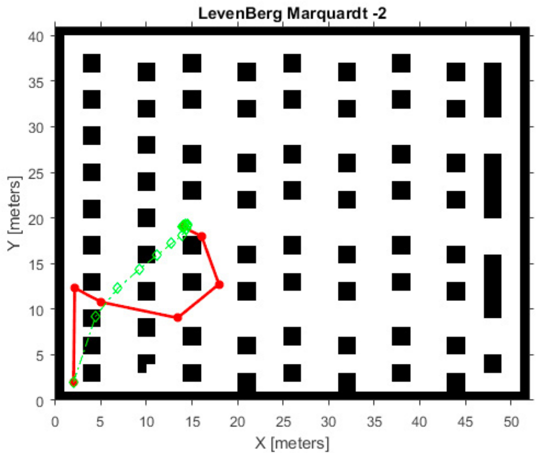

Figure 3: Forecasted path using Levenberg-Marquardt algorithm

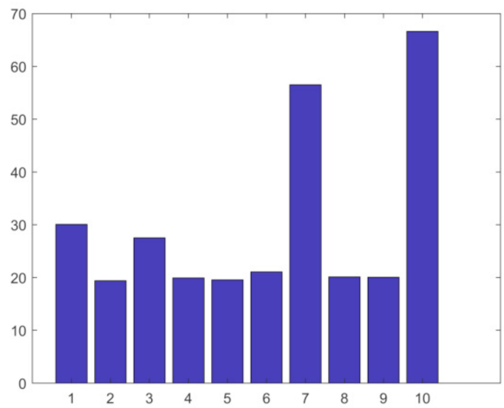

Figure 4: Distance Comparison

The parameters considered for comparison of the neural networks were total distance to be traversed, computational time and performance gradient. These parameters were chosen over other possible methods such as chi-squared statistical analysis or confusion matrix as they were not ideal to compare various neural networks in catering application requiring path planning.

Variable learning rate backpropagation and Fletcher-Powell conjugate gradient approaches were the least suitable for this purpose. This is due to the higher memory requirements of variable learning rate algorithm and it's specific nature to complex situations. These algorithms work best in situations having the highest number of weights which is not required in the given situation.

BFGS-Quasi Newton, Resilient Backpropagation, Scaled Conjugate Gradient, Conjugate Gradient with Powell/Beale restarts and Polar-Ribiere Conjugate Gradient provide nearly identical results with respect to performance and distance. Levenberg-Marquardt and One step secant are the best for a catering robot as they give the best results with smaller number of weights with high performance.

Also on comparison with the PRM benchmark, the neural network results produce a smoother output curve which avoid sharp turns and adapts easily to dynamic data providing a 36% improvement on distance. Thus Levenberg Marquardt algorithm is shown to be the ideal choice for a neural network based path planning.

Table 1: Parameters

Sl no	Algorithm	Training time(s)	Performance	Distance
1	Probabilistic Roadmap	02:00	-	35.8980
2	Levenberg-Marquardt	00:34	13.6518	22.6859
3	BFGS Quasi-newton	00:11	13.6613	23.1070
4	Resilient backpropagation	00:10	12.7808	23.0782
5	Scaled Conjugate Gradient	00:03	13.2856	23.1681
6	Conjugate Gradient with Powell/Beale Restarts	01:31	12.5787	23.3231
7	Fletcher-Powell Conjugate Gradient	00:44	31.8634	180.6451
8	Polar-Ribiere Conjugate Gradient	01:27	12.5758	22.9435
9	One Step Secant	01:20	15.0973	22.5731
10	Variable Learning Rate Backpropagation	00:04	127.7452	155.8027

4. Conclusion

With the increased requirements of robust control algorithms for ever increasing applications of robots, neural networks appear to be the way forward. Out of all the algorithms tested for these randomized datasets with small amount of hidden layers, Levenberg-Marquardt algorithm proved to be the fastest to converge while providing optimum results consistently. Path planning using neural networks includes the possibility of making dynamic datasets so that the system can adapt to new changes in the system. Further modules can be added to the path planning system to achieve object avoidance, and self-balancing to make the complete system functional.

5. References

[1]C. Chen, Q. Gao, Z. Song, O. Liping, and X. Wu, “Catering service robot,” *Proc. World Congr. Intell. Control Autom.*, pp. 599–604, 2010.
[2]P. Technology and B. B. Lama, “Controlling a Robotic Arm Manipulator with a PLC,” 2012.
[3]P. S. Sasongko, H. A. Wibawa, F. Maulana, and N. Bahtiar, “Performance Comparison of Artificial Neural Network Models for Dengue Fever Disease Detection,” pp. 183–188, 2017.
[4]A. Chandra and M. Suaib, “W EB S PAM C LASSIFICATION USING S UPERVISED,” vol. 2, no. 1, pp. 21–30, 2015.
[5]R. Subbanna, “Performance Analysis of ANN Training Algorithms to Detect the Magnetization Level in the

Magnetic Core of a Welding Transformer," vol. 6, no. 6, pp. 17–25, 2013.
[6]G. Mitchell, S. Bahadoorsingh, N. Ramsamooj, and C. Sharma, "A Comparison of Artificial Neural Networks and Support Vector Machines for Short-term Load Forecasting using Various Load Types," pp. 3–6, 1990.
[7]H. Qu, S. X. Yang, S. Member, A. R. Willms, and Z. Yi, "Real-Time Robot Path Planning Based on a Modified Pulse-Coupled Neural Network Model," vol. 20, no. 11, pp. 1724–1739, 2009.
[8]R. S. Pol and M. Murugan, "A review on indoor human aware autonomous mobile robot navigation through a dynamic environment survey of different path planning algorithm and methods," *2015 Int. Conf. Ind. Instrum. Control. ICIC 2015*, no. July, pp. 1339–1344, 2015.
[9]E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Rob. Auton. Syst.*, vol. 61, no. 12, pp. 1258–1276, 2013.
[10]L. Ren, W. Wang, and Z. Du, "Intelligent Obstacle Avoidance Control Strategy for Wheeled Mobile Robot," *ICROS-SICE Int. Jt. Conf. 2009*, pp. 3199–3204, 2009.
[11]C. Li and J. Zhang, "Application of Artificial Neural Network Based on Q-learning for Mobile Robot Path Planning tnfite belteved," *Sci. Technol.*, pp. 978–982, 2006.
[12]K. Belghith, F. Kabanza, L. Hartman, and R. Nkambou, "Anytime dynamic path-planning with flexible probabilistic roadmaps," *Proc. - IEEE Int. Conf. Robot. Autom.*, vol. 2006, pp. 2372–2377, 2006.
[13]*Hagan, M.T., H.B. Demuth, and M.H. Beale, Neural Network Design, Boston, MA: PWS Publishing, 1996. .*
[14]"Marquardt, D., 'An Algorithm for Least-Squares Estimation of Nonlinear Parameters,' SIAM Journal on Applied Mathematics, Vol. 11, No. 2, June 1963, pp. 431–441."
[15]"Dennis, J.E., and R.B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Englewood Cliffs, NJ: Prentice-Hall, 1983."
[16]"Riedmiller, M., and H. Braun, 'A direct adaptive method for faster backpropagation learning: The RPROP algorithm,' Proceedings of the IEEE International Conference on Neural Networks, 1993."
[17]*Moller, M.F., "A scaled conjugate gradient algorithm for fast supervised learning," Neural Networks, Vol. 6, 1993, pp. 525–533. .*
[18]"Powell, M.J.D., 'Restart procedures for the conjugate gradient method,' Mathematical Programming, Vol. 12, 1977, pp. 241–254."
[19]"Beale, E.M.L., 'A derivation of conjugate gradients,' in F.A. Lootsma, Ed., Numerical methods for nonlinear optimization, London: Academic Press, 1972."
[20]"Fletcher, R., and C.M. Reeves, 'Function minimization by conjugate gradients,' Computer Journal, Vol. 7, 1964, pp. 149–154."
[21]"Battiti, R., 'First and second order methods for learning: Between steepest descent and Newton's method,' Neural Computation, Vol. 4, No. 2, 1992, pp. 141–166."