Transfer Learning-Based Artificial Neural Network for Forward Kinematic Estimation of 6-DOF Robot

Kesaba P., Indira Gandhi Institute of Technology, Sarang, India
Bibhuti Bhusan Choudhury, Indira Gandhi Institute of Technology, Sarang, India

ABSTRACT

Transfer learning (TL) can significantly lower training time and reduce dependency on a large number of target domain datasets. Such an approach is still not exploited for robotic prediction tasks. Currently, a TL-based artificial neural network (ANN) is explored and validated for robotic forward kinematics estimation of a 6-DOF robot. The robotic positions are estimated from the available joint angle information. The 6-R MTAB Aristo-XT robot is selected as a case study to generate the target experimental training and testing data for validation of ML techniques while the PUMA 560 6-DOF robot is used as a source for prior training of the ANN model. Standard performance measures such as learning error, deviation error, and mean square error (MSE) are evaluated and graphical illustrations are presented for fair comparison of the results. Experimental results reveal that, instead of ANN, the TL-ANN is strongly suggested to improve the training time of ANN regressor, and it also reduces the randomness and improves the accuracy as compared to its counterpart.

KEYWORDS

6-DOF Robot, Artificial Neural Network (ANN), Forward Kinematics, Machine Learning (ML), Mean Square Error (MSE), MTAB

INTRODUCTION

Generally, the joints of robots are connected in series and controlled by the actuators. The gripper, generally called the end effector, defines the position of the robotic arm in the three-dimensional coordinate system. Various mathematical functions are formulated to establish the relationship between joint angles and their target positions. The problem of estimation for robotic models can be widely categorized as 1) forward kinematics and 2) inverse kinematics. The kinematic equation used to compute the position of the end-effector with known specified joint angles is called forward kinematics (Saha, 2014). Similarly, calculating the joint angles from the known position of the end-effector is referred to as inverse kinematics.

In order to establish the relation between these angles and position information, transformation matrices are used. The transformation matrices define rotational and translational relations whose generalization cannot be made prior to all types of robots. As each robot has its own joint spaces and
constraints, the derivation of the mathematical model is necessary to establish the relation between the joint angles and the position. Fuzzy-based strategies have been proven efficient for such estimation (Jamwal et al., 2010), but cannot be generalized for all robots. While, such relationships can be learned with the help of Machine Learning (ML) models, which may solve the hindered generalization problem (Craig, 2009). Several such ML approaches have been proposed in the past decade, addressing the generalization of robotic manipulators to predict the position of robots.

Applications of Artificial Neural Network for forward kinematics estimation problems are prominent in the literature. Several such approaches are established for various robotic setups such as HEXA Parallel Robot (Dehghani et al., 2008), parallel manipulators (Parikh et al., 2009), 3D cable robot (Ghasemi et al., 2010), 7-DOF Sawyer Robotic Arm (Theofanidis et al., 2018), Delta parallel robot (Liu et al., 2019) model, and so on. Further, improvement was observed with integration of PSO to NN for a 6-DOF parallel robot (Li et al., 2007) to fine-tune backpropagation learning. Also, an iterative approach to the NN strategy proved to be efficient for parallel manipulators (Parikh and Lam, 2009). Another hybrid approach combining neural networks with interval arithmetic to solve forward kinematics (Schmidt et al., 2014) is being experimented for Cable-Driven Parallel Robots. The improved Newton iterative method is validated on a 3-DOF parallel manipulator (Wu and Xie, 2019) for the forward kinematics estimation problem. Other than machine learning, optimization approaches have also been found helpful in estimating position information for 2-DOF using genetic programming (Arellano and Rivera, 2019). Also, the Artificial Neural Network (ANN) for a 6-DOF robot has been experimented with forward kinematics for different pose and joint space representations (Grassmann and Burgner, 2019). Optimum values of ANN model parameters like input data, sample size, training algorithm etc. for inverse kinematics solution of a 3R robotic manipulator are also investigated (Karkalos et al., 2017). Fuzzy control-based path planning of industrial robots is analyzed, and it has been found that Gaussian membership function gives a better result compared to other membership functions (Sahu and Choudhury, 2018).

For large-scale and wide-class range datasets, the surprising performances of deep learning models to image processing are a hot cake for the research community. Such a hybrid deep learning approach is implemented for 6-DOF serial manipulators (Mohamed et al., 2020). Due to the larger-sized dataset, the learning process, however, takes a long time to optimize. Such problems have also been addressed in the past decade by inheriting knowledge from one domain to another. This is commonly called Transfer Learning (TL). The psychologist C.H. Judd proposed the generalization theory of transfer learning as a consequence of experience. When the experience is generalized, a person can transfer his or her learning experience to learning another. This theory suggests that a connection is needed between two learning activities to speed up the learning process. Such an example is if a person has learned to ride a bicycle, can be able to learn the motorcycle driving faster than others who do not have bicycle riding experience, as they may share some common knowledge (Zhuang et al., 2020). Mostly, the TL approaches were adopted for deep learning architectures. While such approaches can also be applicable to ANN models. The TL approaches are mostly implemented for image processing domains. Several such TL-based image processing applications have been developed for face recognition (Yin et al., 2019), cancer classification (Singh et al., 2020), super resolution (Yuan et al., 2019), land cover classification (Zhang et al., 2019) etc. Other than image processing applications, also TL was found effective for weather prediction (Hu et al., 2016), text classification (Zhuang et al., 2009), sentiment analysis (Pan et al., 2010), sensor localization (Zheng et al., 2008) etc. However, the application of TL to forward kinematics estimation is limited in the literature. Hence, in the present study, a TL-ANN approach is aimed and implemented for robotic estimation problems.

FORwARD KINEMATICS MODELING

The homogeneous transformation matrices of each joint are used to calculate the position and orientation of the end-effector. In other words, the forward kinematics problem can be well-formulated
with information on joint position and rotational constraints. Figure 1 shows the experimental setup of the MTAB Aristo-XT robot. An Aristo robot is a six-axis articulated robot with all rotary joints.

The transformation matrix between end-effector and base frame, for a 6-R robot can be represented as follows:

\[ T_6 = A_1^* A_2^* A_3^* A_4^* A_5^* A_6^* \]  \hspace{1cm} (1)

where, the transformation matrix \( T_6 \) can be described as a product of six homogeneous transformation matrices represented as \( A_i \) (each \( i = 1,2,3...6 \) represents a joint). And each 4 × 4 transformation matrix \( A_i \) can be represented as:

\[
A_i = \begin{bmatrix}
\cos(\theta_i) & -\cos(\alpha_i)\sin(\theta_i) & \sin(\alpha_i)\sin(\theta_i) & a_i \cos(\theta_i) \\
\sin(\theta_i) & \cos(\alpha_i)\cos(\theta_i) & -\sin(\alpha_i)\cos(\theta_i) & a_i \sin(\theta_i) \\
0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\
0 & 0 & 0 & 1
\end{bmatrix}  \hspace{1cm} (2)
\]

\[
T = \begin{bmatrix}
{n_x} & {o_x} & {a_x} & {p_x} \\
{n_y} & {o_y} & {a_y} & {p_y} \\
{n_z} & {o_z} & {a_z} & {p_z} \\
0 & 0 & 0 & 0
\end{bmatrix} \hspace{1cm} (3)
\]

where, with reference to the base frame, the position of the end-effector is represented as vector, and the orientations are represented as \( n, o \) and \( a \) vectors. The \( a \) represents the approach vector, \( o \) represents the orientation vector, and \( n \) is normal to both the \( a \) and \( o \) vectors.

Figure 1. Aristo XT 6-DOF robot in home position
TRANSFER LEARNING-BASED ARTIFICIAL NEURAL NETWORK

Artificial Neural Network

Human brain replication with the help of mathematical models leading to the computational strategy for information processing is widely known as Artificial Neural Network (ANN) (Dora et al., 2020). The ANN consists of multiple layers with interconnected neurons (refer Figure 3), where each neuron can be represented as in Figure 4. The input and output layers are generally characterized by independent variables and dependent variables, respectively. The configuration of ANN for regression can be: 1) Multiple Inputs - Single Output or 2) Multiple Inputs - Multiple Output systems. On the basis of trained data, the relationship between the input and output hyper-dimensional spaces can be learned by ANN. Accordingly, learning performance can be validated with further testing of trained models on real-time data. An ANN learning process which updates the weight and bias of the neurons to minimize error or loss function is normally employed with the backpropagation process. Various architectures of ANN are explored and validated for different applications like classification (Panchal et al., 2016), regression (Eskandarian et al., 2017), pattern recognition (Abiodun et al., 2019) etc. The informative and redundant data, which can be an efficient way to predict the desired outcome, can be extracted instead of storing all the in-hand data (in exponential scale). Many researchers have proved this application fruitful, leading to the application of ANN for numerous prediction tasks. In the past decade, several such predictions have been proposed for stock prediction (Gao et al., 2017), electricity forecasting (Chen., 2017), labor productivity (Golnaranghi et al., 2019) etc.

Figure 3 shows the general architecture of an ANN consisting of multiple layers. The neurons in input layers are exactly the same as the number of independent variables (feature space) in the dataset, and each neuron in the output layer refers to the prediction outcomes. Also, there may be multiple hidden layers and the size generally depends on the information in the input data. Hidden layers are decided empirically with multiple trials. Each hidden layer can consist of multiple neurons and the exact solution to the volume is still hindered among researchers.

The ANN can be described as a functional representation from the input (observation) space \( X \) to the output (decision) space \( Y \) (Eq. 4):

\[
f : X \rightarrow Y
\]  

(4)

For input feature space \( X \) with \( n \) features as \( \left(x_1, x_2, \ldots, x_n\right) \), the linear relationship to output \( Y' \) of the neuron can be described as in Eq. 5:

\[
Y' = \sum_{i=1}^{n} x_i w_i + b
\]

(5)

where, \( b \) represents a bias term which is added to the sum of multiplicative weights \( W \left(w_1, w_2, \ldots, w_n\right) \) with the input feature space.

The highly non-linear systems of real-time applications can be modeled with the help of ANN by introducing the transfer function. Such transfer function as shown in Figure 4 and Eq. 6 is characterized by the activation function:

\[
Y = \varphi(Y') = \varphi\left(\sum_{i=1}^{n} x_i w_i + b\right)
\]

(6)
where activation function $\varphi(.)$ is a nonlinear function (Wu et al., 2019) such as; $\log \text{sig}(.)$, $\tan \text{sig}(.)$, $\text{purelin}(.)$ etc. The transfer functions are empirically selected based on a trial and error strategy to best fit the specific regression or classification task.

Generally, backpropagation approach is used to train the ANN. The purpose of the training is to reduce error performance of the network by updating the weight and bias associated with each neuron. Commonly, learning errors can be minimized by updating the random initialized weight and bias with the help of a stochastic gradient descent algorithm.

**TRANSFER LEARNING ARCHITECTURE**

Transfer learning is a research problem in machine learning that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem (Zhuang et al., 2020). The adopted learning process is presented in Figure 2.

The definition of transfer learning is given in terms of domains and tasks. A domain $D$ consists of a feature space $X$ and a marginal probability distribution $P(X)$, where $X = \{x_1, x_2, \ldots, x_n\} \in X$.

Given, the domain $D=\{X,P(X)\}$, a task consists of two components: a label space $Y$ and an objective predictive function $f : X \rightarrow Y$. The function $f$ is used to predict the corresponding label $f(x)$ of new instances $x$. This task, denoted by $T=\{Y,f(x)\}$, is learned from training data consisting of pairs $x_i,y_i$, where $x_i \in X$ and $y_i \in Y$.

Given a source domain $D_S$ and learning task $T_S$, a target domain $D_T$ and learning task $T_T$, where $D_S \neq D_T$ or $T_S \neq T_T$, transfer learning aims to help improve the learning of the target predictive function $f_T(x)$ in $T_T$ using the knowledge in $D_S$ and $T_S$.

Presently, the source domain $D_S$ is related to one robotic environment and the target domain $D_T$ refers to another robotic environment. Initially, the ANN model is trained using the $D_S$ domain data set, and further, the learned weights of the neurons are used as initial memory for the target domain $D_T$. In general, instead of initializing the neurons randomly in the ANN architecture, the neurons are adopted from another learning environment and refined further using the target training dataset. This may require fewer training samples and a faster learning experience.

**Figure 2. Proposed model of Transfer Learning**
EXPERIMENTAL SETUP

For the purpose of Transfer Learning, data from two 6-DOF robots are required. One for training the ANN and the other to fine tune the further learning process. In order to validate our assumptions on the proposed TL approach, PUMA 560 and ARISTO XT MTAB robot were selected. Both the robots are industrial robots with six axis joints, hence used for the proposed TL approach.

The nominal operating load of the end-effector of PUMA 560 robot is 2.5kg with 0.1 positional repeatability and the maximum velocity of the end-effector is 1 m/s. The PUMA 560 manipulator occupies a workspace of 0.92m between the wrist center and the center axis. DC servo motors are used for actuators of all six joints. The joint-link parameters have been presented in Table-1, and the specification of this robot with joint limits is visually presented in Figure 5. The PUMA 560 robots are compact in design and used for handling of small objects or components in industries. The PUMA 560 robots are used for assembly of the most complicated, intricate parts like automotive panels, small electric motors, circuit board printings, appliances etc.
The data for initial training of the ANN model was collected from this PUMA 560 robot. The experimented angles act as source domain $D_s$ data, and their respective position information are represented as the learning task as $T_s$.

The Aristo XT MTAB robot is selected as a case study to perform forward kinematics analysis. The joint-link parameters for the 6R MTAB Aristo-XT robot are presented in Table 2. The experimental Aristo-XT robot has six DOF with all rotary type joints. The first three DOF are located in the arm.

Table 1. Joint-link parameters for the PUMA 560 6R robot

<table>
<thead>
<tr>
<th>Joint $i$</th>
<th>$\theta_i$ (deg)</th>
<th>$\alpha_i$ (deg)</th>
<th>$a_i$ (mm)</th>
<th>$d_i$ (mm)</th>
<th>$\theta_{i\min}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>90</td>
<td>0</td>
<td>0</td>
<td>-160</td>
<td>+160</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-225</td>
<td>+45</td>
</tr>
<tr>
<td>3</td>
<td>-90</td>
<td>4318</td>
<td>1244</td>
<td>-45</td>
<td>+225</td>
</tr>
<tr>
<td>4</td>
<td>90</td>
<td>203</td>
<td>4318</td>
<td>-110</td>
<td>+110</td>
</tr>
<tr>
<td>5</td>
<td>-90</td>
<td>0</td>
<td>0</td>
<td>-100</td>
<td>+100</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-266</td>
<td>+266</td>
</tr>
</tbody>
</table>
which allows determining the robot’s position, and the following three DOF are located in the end effectors to provide orientation. The input variables are six joint angles ($A_1$ to $A_6$) within the given specified limits, and the outputs are the position of the end-effector (world coordinates $X$, $Y$, and $Z$). The software used for the simulation of the robot is Aristo Version 1.4, presented in figure 6. The Aristo-XT robot can be used for various applications like palletizing, loading/unloading, gluing, spray painting, polishing, segregation of objects using vision systems etc. The data for fine-tuning the training process and to test the ANN model is collected from this ARISTO robot. The experimented angles act as source domain $D_s$ data, and their respective position information are represented as the learning task as $T_T$.

**Figure 6. Coordinates of robots using Aristo simulation**

<table>
<thead>
<tr>
<th>Joint $i$</th>
<th>$\theta_i$ (deg)</th>
<th>$\alpha_i$ (deg)</th>
<th>$a_i$ (mm)</th>
<th>$d_i$ (mm)</th>
<th>$\theta_i^{\min}$</th>
<th>$\theta_i^{\max}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>90</td>
<td>0</td>
<td>158</td>
<td>-150</td>
<td>+150</td>
</tr>
<tr>
<td>2</td>
<td>90</td>
<td>0</td>
<td>300</td>
<td>0</td>
<td>+60</td>
<td>+120</td>
</tr>
<tr>
<td>3</td>
<td>160</td>
<td>90</td>
<td>0</td>
<td>0</td>
<td>+130</td>
<td>+190</td>
</tr>
<tr>
<td>4</td>
<td>-180</td>
<td>90</td>
<td>0</td>
<td>-378.5</td>
<td>-210</td>
<td>-150</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>90</td>
<td>0</td>
<td>0</td>
<td>-90</td>
<td>+90</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>64</td>
<td>-165</td>
<td>+165</td>
</tr>
</tbody>
</table>


The experimental variations of the six angles are presented in Figure 7. The collected dataset consists of 800 samples. For training 20% of the data are selected and the rest 80% data are used for testing. In order to make a fair comparison between Artificial Neural Network and proposed Transfer Learning-based ANN architectures. The performances are evaluated in terms of MSE using tables and graphical representations, for each \( X, Y \) and \( Z \) location; and also using three-dimensional representations.

Simulation for the proposed regression is carried out using MATLAB18-b software, installed on a PC with Intel i-5, 12 GB RAM.

**RESULTS AND DISCUSSION**

The learning errors for both the ANN and TL-ANN are reported in Figure 8. From the figure, it is clear that the error is minimized for the proposed TL-ANN approach. The average percentage reduction in error for the proposed TL-ANN approach compared to the ANN method is approximately 70%. While, the convergence of our proposed method is faster as compared to the state-of-art method. The proposed TL-ANN has converged around 200 iterations. However, we can observe that by 2000 iterations the ANN converges equivalently to the TL-ANN approach. The faster convergence of proposed TL-ANN before 1800 iterations compared to ANN signifies the importance of transfer learning, which can be a better alternative to the ANN approach for robotic estimation problems.

The three-dimensional representation of the location estimation is presented in Figure 9. In this figure, the blue circled line stands for the actual trajectory, whereas the red dotted line and black starred line stand for the TL-ANN trajectory and ANN trajectory of the robot manipulator, respectively. From figure 9, it is evident that by using the proposed TL-ANN method, the generated trajectory output is completely tracking the actual trajectory. However, in the case of the ANN approach, the trajectory does not follow the actual one and deviates from the path at many locations due to error.
Figure 8. Mean Square error comparison of ANN and proposed TL-ANN approaches

Figure 9. Obtained 3-D position for ANN and proposed TL-ANN approaches
Tracking accuracy is highly required for a robot manipulator, especially when it is used in fields with high real-time requirements.

Also, for each $X, Y$ and $Z$ location, the error estimation for both ANN and proposed TL-ANN is presented in Figure 10, Figure 11 and Figure 12, respectively. In these figures, large random variation in error is being observed for the ANN approach, whereas the TL-ANN approach exhibits minimal error. Hence, TL-ANN takes less computation time to train the data as compared to ANN. Also, from Figure 10, Figure 11 and Figure 12, we can observe that errors are highly nonlinear for samples greater than 650. The reason for the random error is in relation to the multiple angle variations in the experimental setup (refer Figure 7), where almost all joints have variations in their angles.

Table 3 presents the MSE and Mean-MSE for ANN and the proposed TL-ANN approach. Both the experiments (ANN and TL-ANN) were carried out using the same number of samples. From the table it is observed that the MSE for each axis is as $X, Y$ and $Z$, and that the mean MSE of the proposed TL-ANN approach is less than the ANN approach. The significant decrease in mean MSE from 0.0138 to 0.0043 represents the traditional ANN and the proposed TL-ANN approaches, respectively. Similar interpretations can also be observed from the bar graph presented in Figure 13, showing the comparative error analysis of the experimented models.

From the results in Table 3, it can be observed that the learning time and error both are minimized for the proposed TL-ANN model as compared to the ANN. So, validation of the proposed TL-based model can be strongly argued for the robotic estimation problem.

CONCLUSION

Due to extensive use of robotic automation in critical and highly precision work, estimation of position from their joint angles is necessary and the relevant approaches are widely available in the literature.
Figure 11. Y-location Mean Square Error comparison of ANN and proposed TL-ANN approaches

Figure 12. Z-location Mean Square Error comparison of ANN and proposed TL-ANN approaches

Table 3. Mean Square Error for $X, Y$ and $Z$ location with mean of MSE comparison for ANN and TL-ANN model

<table>
<thead>
<tr>
<th>$Joint_i$</th>
<th>X-MSE</th>
<th>Y-MSE</th>
<th>Z-MSE</th>
<th>Mean MSE</th>
<th>Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANN</td>
<td>0.0164</td>
<td>0.0127</td>
<td>0.0124</td>
<td>0.0138</td>
<td>640</td>
</tr>
<tr>
<td>TL-ANN</td>
<td>0.0052</td>
<td>0.0049</td>
<td>0.0029</td>
<td>0.0043</td>
<td>640</td>
</tr>
</tbody>
</table>
Several such literature are fuzzy-based and ML-based. Other than robotic estimation, the TL-based approaches are widely available for image processing, signal processing, and data mining domains. Also, due to high volume of data, these approaches are applied to deep learning approaches to reduce training time. However, such approaches are limited to robotic estimation problems. Hence, such a TL-based ANN approach is implemented, and primarily results are reported which are in favor of the proposed TL-ANN approach as compared to the traditional ANN model. The convergence curve of the TL-ANN approach is lower than the traditional ANN model, which indicates that learning time is minimized, and accuracy is maximized by adopting the TL-ANN approach. The obtained results strongly suggest the incorporation of Transfer Learning for ANN models to estimate the robotic forward kinematics problem. From Table 3, it is clearly observed that the mean MSE of the proposed TL-ANN approach is 0.0043, which is much lower than the MSE of 0.0138 of the state-of-art ANN approach. However, further experiments can be carried out on other robotic configurations for global validation of the proposed approach.

**FUNDING AGENCY**

The publisher has waived the Open Access Processing fee for this article.
REFERENCES


---

**P. Kesaba is currently a PhD candidate at the Mechanical Engineering Department, IGIT, Sarang, Odisha, India. He holds B.E. in mechanical engineering from UCE Burla and M.E. in mechanical engineering from Indian Institute of Engineering Science and Technology, Shibpur.**

**Bibhuti Bhusan Choudhury completed his PhD degree in 2009 from the National Institute of Technology, Rourkela. He is currently Professor and the Head of Production Engineering Department at IGIT, Sarang, Odisha, India. The author’s research area is CAD/CAM/CIM and robotics. He has 20 years of experience in teaching and research and is currently working as the head of the Department in Production Engineering at IGIT Sarang, Odisha, India.**