Evaluation of Artificial Neural Networks

Kayamkhani Taher Abbasbhai,
CH. Nova Vijay,
Jhanwar Sourabh.

CMR Technical Campus, Hyderabad, India.

1. Abstract

Artificial Neural Networks are composed of interconnecting artificial neurons (programming constructs that mimic the properties of biological neurons). Artificial neural networks may either be used to gain an understanding of biological neural networks, or for solving artificial intelligence problems without necessarily creating a model of a real biological system. Artificial Neural Network has been shown to be an efficient tool for non-parametric modeling of data in a variety of different context where the output is non-linear function of prediction, medicines, pattern recognition and image processing. ***1.*** *In this paper we present neural network architecture for learning of robotic grasping tasks. Systematic computer simulations have been carried out in order to test learning and generalization capabilities of the system. The proposed model can be used as a high level controller for a robotic dexterous hand during learning and execution of grasping tasks.* ***2.*** *Diagnosis of diseases is well known problem in the medical field. Past research shows that medical database of disease can be train by using various neural network models. Many medical problems face the problem of curse of dimensionality due to the excessively large number of input attributes. Breast cancer is one such problem. Breast cancer is the second leading cause of cancer deaths worldwide and occurs in one out of eight women. In this paper we develop a system for diagnosis, prognosis and prediction of breast cancer using Artificial Neural Network (ANN) models.*

2. Introduction

Artificial neural networks are a computational tool, based on the properties of biological neural systems. Neural networks excel in a number of problem areas where conventional von Neumann computer systems have traditionally been slow and inefficient. This book is going to discuss the creation and use of artificial neural networks.Over the last few years, a number of studies were reported concerning machine learning, and how it has been applied to help robots to improve their operational capabilities. Robots must be able to understand the structure of this environment. To reach their targets without collisions, the robots must be endowed with perception, data processing, recognition, learning, reasoning, interpreting, and decision-making and action capacities. The ability to acquire these faculties to treat and transmit knowledge constitutes the key of a certain kind of artificial intelligence. Reproduce this kind of intelligence is, up to now, a human ambition in the construction and development of intelligent machines, and particularly autonomous mobile robots.

An **Artificial Neural Network (ANN)**, usually called neural network (NN**)**, is a mathematical model or computational model that is inspired by the structure and/or functional aspects of biological neural networks. A neural network consists of an interconnected group of artificial neurons, and it processes information using a connectionist approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. Modern neural networks are non-linear statistical data modeling tools. They are usually used to model complex relationships between inputs and outputs or to find patterns in data.

3. Models

Neural network models in artificial intelligence are usually referred to as artificial neural networks (ANNs); these are essentially simple mathematical models defining a function $f : X \rightarrow Y$ or a distribution over X or both X and Y, but sometimes models are also intimately associated with a particular learning algorithm or learning rule. A common use of the phrase ANN model really means the definition of a *class* of such functions (where members of the class are obtained by varying parameters, connection weights, or specifics of the architecture such as the number of neurons or their connectivity).

3.1. Network function

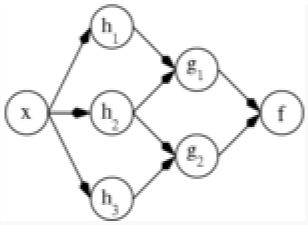
3.1.1. Graphical models

The word *network* in the term 'artificial neural network' refers to the inter–connections between the neurons in the different layers of each system. An example system has three layers. The first layer has input neurons, which send data via synapses to the second layer of neurons, and then via more synapses to the third layer of output neurons. More complex systems will have more layers of neurons with some having increased layers of input neurons and output neurons. The synapses store parameters called "weights" that manipulate the data in the calculations.

An ANN is typically defined by three types of parameters:

1. The interconnection pattern between different layers of neurons
2. The learning process for updating the weights of the interconnections
3. The activation function that converts a neuron's weighted input to its output activation.

Mathematically, a neuron's network function $f(x)$ is defined as a composition of other functions $g_i(x)$, which can further be defined as a composition of other functions. This can be conveniently represented as a network structure, with arrows depicting the dependencies between variables. A widely used type of composition is the *nonlinear weighted sum*, where $f(x)=K\left(\sum_i w_i g_i(x)\right)$, where K (commonly referred to as the activation function) is some predefined function, such as the hyperbolic tangent. It will be convenient for the following to refer to a collection of functions $g_i$as simply a vector $g=(g_1,g_2,\ldots,g_n)$.



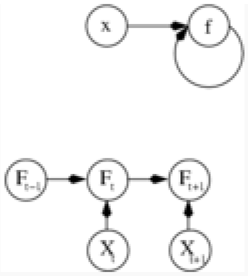
ANN dependency graph

This figure depicts such a decomposition of f, with dependencies between variables indicated by arrows. These can be interpreted in two ways.

The first view is the functional view: the input x is transformed into a 3-dimensional vector h, which is then transformed into a 2-dimensional vector g, which is finally transformed into f. This view is most commonly encountered in the context of optimization.

The second view is the probabilistic view: the random variable $F=f(G)$ depends upon the random variable $G=g(H)$, which depends upon $H=h(X)$, which depends upon the random variable X. This view is most commonly encountered in the context of graphical models.

The two views are largely equivalent. In either case, for this particular network architecture, the components of individual layers are independent of each other (e.g., the components of g are independent of each other given their input h). This naturally enables a degree of parallelism in the implementation.



Two separate depictions of the recurrent ANN dependency graph

Networks such as the previous one are commonly called feed forward, because their graph is a directed acyclic graph. Networks with cycles are commonly called recurrent. Such networks are commonly depicted in the manner shown at the top of the figure, where f is shown as being dependent upon itself. However, an implied temporal dependence is not shown.

4. Application

Real-life applications

The tasks artificial neural networks are applied to tend to fall within the following broad categories:

- Function approximation, or regression analysis, including time series prediction, fitness approximation and modeling.
- Classification, including pattern and sequence recognition, novelty detection and sequential decision making.
- Data processing, including filtering, clustering, blind source separation and compression.
- Robotics, including directing manipulators, Computer numerical control.

Application areas include system identification and control (vehicle control, process control, natural resources management), quantum chemistry, game-playing and decision making (backgammon, chess, poker), pattern recognition (radar systems, face identification, object recognition and more), sequence recognition (gesture, speech, handwritten text recognition), medical diagnosis, financial applications (automated trading systems), data mining (or knowledge discovery in databases, "KDD"), visualization and e-mail spam filtering.

Current Usage

There are many applications of artificial intelligence at present. Some of them have been listed here.

- Banks and other financial institutions rely on intelligent software, which provide accurate analysis of the data and helps make predictions based upon that data.
- Stocks and commodities are being traded without any human interference - all thanks to the intelligent systems.
- Artificial intelligence is used for weather forecasting.
- It is used by airlines to keep a check on its system.
- Robotics is the greatest success story, in the field of artificial intelligence. Spacecrafts are sent by NASA and other space organizations into space, which are completely manned by robots. Even some manufacturing processes are now being completely undertaken by robots. Robots are being used in industrial processes, that are dangerous to human beings, such as in nuclear power plants.
- Usage of artificial intelligence is quite evident in various speech recognition systems, such as IBM ViaVoice software and Windows Vista.

4.1. Robot Learning Using Neural Networks

The examples we have used so far are useful in understanding how neural networks work, but ultimately we want to create a neural network brain for a robot. That is, how does a robot learn to perform a task? The simplest robot learning tasks are navigational tasks such as learning to move about without bumping into obstacles, or following a wall, or learning to detect a wall.

As we have seen from our earlier examples, in order to train a neural network, the input data is to be presented at the input layer and then, through a process of feed forward propagation, the neural network produces an output at the output layer. In the case of robot learning, typically in navigational behaviors, the inputs into the neural network are the robot's sensor values, and the outputs are commands for the robot's motors. Depending on the environment you are using you may do the learning experiments on real robots or on simulated ones. Additionally, you may do the learning **offline** or **online**.

4.1.1 Offline Learning/Training

In the case of offline learning, you first collect a training data set by running a robot using a program that exhibits the desired behavior. That is, while the robot is actually carrying out the desired task, you record its sensor and motor values. These will then become the input and target output values during training. Since all the data is collected prior to training, we call this offline training. Once the network has been trained, you can **deploy** the network on the robot to examine and test its behavior. Thus, for offline training, one uses the following steps:

1. Write a behavior (most likely using direct control) that accomplishes the training task
2. Extract a data set from the robot carrying out the task
3. Scale and sample the data to create a training data set
4. Train a network with the data
5. Deploy and test the network on the robot
6. Analyze the results

4.1.2 Online Learning

In the case of online learning, you train the robot in real time, on-the-fly. That is, you get the robot's sensor readings, you feed these to the neural network, propagate and get output, you compare the output with the desired output (i.e. what would the correct response should have been), you compute the error and do a step of neural network learning, you move the robot, and repeat the process until the total error is within an acceptable range. Once the network has learned, you can turn off the learning, and then use the network to control the robot. The following steps are preformed for online learning:

1. Write a network control program with a built-in *trainer* to compute the correct outputs
2. Repeat the following until the error reduces to an acceptable level:
    i. Get sensor readings
    ii. Scale them
    iii. Present them to the input
    iv. Propagate and compute network's output
    v. Use the *trainer* to compute correct output
    vi. Compute the error and use it for a learning step
3. Turn learning off
4. Deploy and test the network on the robot
5. Analyze the results

Example: Detecting an obstacle using offline learning

Let us take a very simple navigational task: The robot goes forward and stops when it is about to run into an obstacle.

4.1.3 Robotics approach in Industrial Applications

In order for a robotic manipulator to perform useful work, it must be programmed to accomplish the desired task or motion cycle. Nowadays industrial robots generally require a tremendous amount of programming to make them useful. Their controllers are very sophisticated, the commercial robot programming environments are typically closed systems and the programming languages varies from manufacturer to manufacturer. Despite the great evolution of the industrial robots controllers, in the majority of the industrial applications, the robot programming is made, using one of the following ways:

- Manual on-line programming;
- Off-line programming;

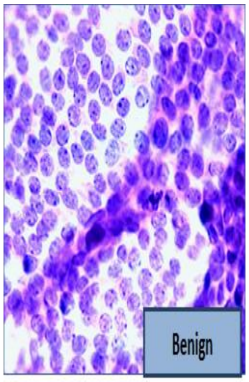
Manual on-line programming refers to physically teaching a robot the required trajectory, through interaction with teach pendant or other similar device (Lee & ElMaraghy, 1990).This programming kind presents the following disadvantages: very slow, it needs that the robot is available, difficulty in the handling of equipment's, need some practice in the language used by the robot, and technical knowledge to understand the operation of the equipment. These disadvantages are very expensive in the industry because the productive process needs to stop for a long time. One simple approach to solve some disadvantages described above is the Off-line programming environments. These environments are based in graphical simulation platforms, in which the programming and execution process are shown using models of the real objects. Consequently, the robot programmer has to learn only the simulation language and not any of the robot programming languages. Other benefits in off-line programming environments include libraries of pre-defined high-level commands for certain types of applications, such as painting or welding, and the possibility to assess the kinematics feasibility of a move, thus enabling the user to plan collision-free paths. The simulation may also be used to determine the cycle time for a sequence of movements. These environments usually provide a set of primitives commonly used by various robots, and produce a such sequence of robot manipulator language primitivesas"move" or "open gripper" that are then downloaded in the respective robot controllers.

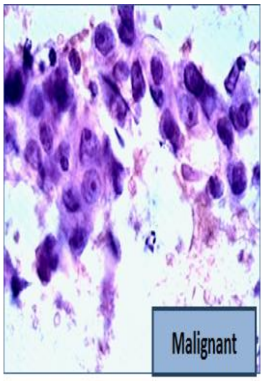
2 Industrial Robotic s - Programming, Simulation and Applications However, the off-line programming tools based in graphically 3D representation presents several problems in many industry applications, particularly, when the robot task or the robot trajectory needs frequent changes, for example: in welding applications where the configuration of the pieces to weld change frequently (the size, the shape, etc.); the robot painting and gluing applications can have similar problems. Nowadays, the CAD tools are often used in the industry to develop and to document the products and its manufacture. There are a lot of commercial CAD tools, like, AutoCAD, Solid Works, Ideas and Cimarron, having each tool its own file format. However, it is possible to export the information of these pieces, in a neutral file format, namely: STL, IGES, STEP and SET formats. This work presents one solution for programming different robots based in the relevant information extracted from neutral files. The solution implemented was tested in the industrial robots Mitsubishi (Mitsubishi Move Master Industrial Robot) and ABB.(model IRB 140 with IRC5 controller). This chapter is organized as follows: section 2 presents an overview about the format of neutral files (STL, IGS, STEP and SET); in the section 3, the algorithms for extraction of the relevant information from the neutral files are described; in

the section 4, the developed tool for code generation for different industrial robots is presented; section 5 and 6 present the results and conclusions; section 7 presents future work.

4.2 Cancer diagnosis using Neural Networks

Artificial neural networks have also been used to diagnose several cancers. An ANN based hybrid lung cancer detection system named HLND improves the accuracy of diagnosis and the speed of lung cancer radiology. These networks have also been used to diagnose prostate cancer. The diagnoses can be used to make specific models taken from a large group of patients compared to information of one given patient. The models do not depend on assumptions about correlations of different variables. Colorectal cancer has also been predicted using the neural networks. Neural networks could predict the outcome for a patient with colorectal cancer with a lot more accuracy than the current clinical methods. After training, the networks could predict multiple patient outcomes from unrelated institutions.





An 'artificial brain' has been developed recently that can diagnose breast cancer. Using various computer-programming languages, a cloud-based artificial neural network is being developed that can assess tissue samples for signs of breast cancer. A breast cancer-diagnosing app based on an artificial neural network has been developed, basically a computer program whose structure is inspired by the way brain cells connect with one another. The artificial neural network along with a procedure called Fine Need Aspirate may help to ease the process of having lumps examined. The artificial neural network has data collected from 7.6 million trials, and according to recent survey, the Java-based artificial brain is 99.1 percent sensitive to malignancy. What's even more amazing is that network is 4.97 percent more sensitive to malignancy than three other commercially available networks. This technology is aimed to deploy the network in hospitals and extend her dataset to include other cancers.

5. Advantages

- A neural network can perform tasks that a linear program can not.
- When an element of the neural network fails, it can continue without any problem by their parallel nature.
- A neural network learns and does not need to be reprogrammed.
- It can be implemented in any application.
- It can be implemented without any problem.

6. Disadvantages

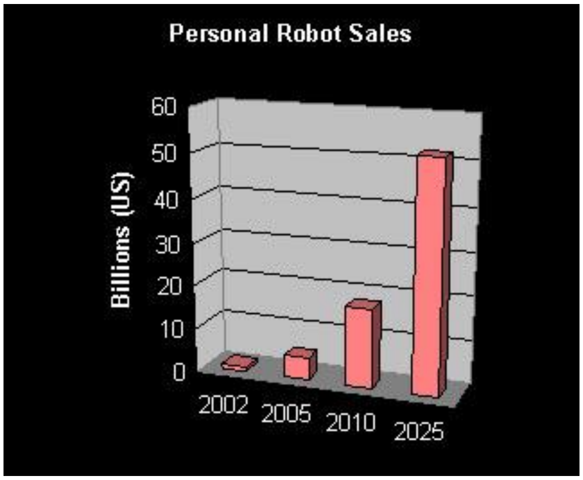
- The neural network needs training to operate.
- The architecture of a neural network is different from the architecture of microprocessors therefore needs to be emulated.
- Requires high processing time for large neural networks.

7. Conclusion

Recognizing objects, physically maneuvering the environment, performing tasks. These are the functions that will first make AI accessible to most of the people, which will increase total revenue available to the AI industry which will in turn further promote the development of artificial neural networks. The consumer robotics marketplace is just emerging, with a gross of US$600 million in 2002, comprised mainly of programmable robots which mow lawns, clean floors and amuse children. But momentum is growing and with expertise concentrated in one spot (Japan) and reducing manufacturing costs the genuinely useful, autonomous household robot is almost upon us.

According to projections from UNEC the personal robotics industry will have grown to US$5.4 billion by 2005 and to US$17.1 billion by 2010 spectacular growth indeed, but humble by comparison to the following decade. A quarter century from now, the robotics industry is expected to rival the automobile and computer industries in both dollars and jobs. This next generation of robots will be capable of interacting with humans on a sophisticated level, making decisions, helping, caring and giving companionship.

(Figure 5.1)PROJECTED GLOBAL PERSONAL ROBOTICS REVENUES 2002 -2025



If robots are to become truly useful for humanity, they must possess the ability to perceive and distinguish between objects, places and persons. A requirement that presents a set of unique challenges for robotics programmers and engineers to grapple with. If such engineering challenges can be overcome, some believe that we could see robots become as ubiquitous as cell phones; "In five or ten years robots will routinely be functioning in human environments."

With continuous advances in robotic methods like behavior-based control, future space missions might be able to function without relying heavily on human commands. On the home front, similar technology is already used in many practical applications such as digital cameras, computer programs, dishwashers, washing machines and some car engines. The post office even uses neural networks to read handwriting and sort mail.

"Do these mean robots in the near future will think like humans? No, But by mimicking human techniques, they could become easier to communicate with, more independent, and ultimately more efficient".

8. References

Ajith Abraham, 129-Artificial Neural Network journal, Oklahoma State University, Stillwater, OK, USA

Vitor Bottazzi & Jaime Fonseca
Department of Industrial Electronics - University of Minho
Portugal

Tiemin Hu, Computational Intelligence in Robotics and Automation, 2001. Proceedings 2001 IEEE International Symposium on, Sch. of Eng., Guelph Univ., Ont.
Yang, S.X.

Omidvar, O., Van der Smagt, P., (Eds.). (1997) Neural Systems for Robotics. New York: Academic Press

Kai-Hui Chi, Consumer Electronics, Communications and Networks (CECNet), 2011 International Conference on, Grad. Inst. of Autom. & Control, Nat. Taiwan of Univ. Sci. of Technol., Taipei, Taiwan
Lee, M.R.

Mansur Ghaffari Moghaddam
Faculty of Science
University Putra Malaysia
43400 UPM Serdang, Selangor, Malaysia

Faujan Bin H. Ahmad
Faculty of Science
University Putra Malaysia
43400 UPM Serdang, Selangor, Malaysia

Mahiran Basri
Faculty of Science

University Putra Malaysia
43400 UPM Serdang, Selangor, Malaysia

Mohd Basyaruddin Abdul Rahman
Faculty of Science
University Putra Malaysia
43400 UPM Serdang, Selangor, Malaysia

Wikipedia, Artificial Neural Networks.

Books:

Artificial Neural Networks, B. Yegnanarayana

Neural Networks
***by** Ivan F Wilde - King's College London, 2009*

A Brief Introduction to Neural Networks
***by** David Kriesel - dkriesel.com , 2011*

A Brief Introduction to Neural Networks
***by** David Kriesel - dkriesel.com, 2011*