Introduction to Artificial Neural Networks

EE-589

Who am I

Associate Prof. Dr. Turgay IBRIKCI
Room # 305
Tuesdays 13:30-16:00
(322) 338 6868 / 139
eembdersler.wordpress.com

What you learn from the course

- How to approach a neural network learning classification or clustering
- Basic knowledge of the common linear machine learning algorithms
- Basic knowledge of Neural Networks learning algorithms
- A good understanding of neural network algorithms

Course Outline

The course is divided in two parts: theory and practice.
1. Theory covers basic topics in neural networks theory and application to supervised, unsupervised and reinforcement learning.
2. Practice deals with basics of MATLAB and implementation of NN learning algorithms. We assume that you know MATLAB or you will learn yourself

Course Grading

- Grading the Class:
  - Project 40% (Proposal 5% (31/03/2015), Project paper 35% (CD will be in paper publication format), Presentation (Week 14: 20 mins)
  - Final Exam 20% (Week 15 - We decide together)
  - Homeworks 40% (At least 4 homeworks)
  - Full attending the class 10% (Bonus)

Project

- Choose a topic that is related to your research interest and pertains to the course material. Discuss with your supervisor.
- The proposal should include the following sessions (Due: 31/03)
  - the project goal,
  - the problems to be studied,
  - overview of current methods,
  - proposed methods, expected results, and references (about 4 pages: single space, font size = 12, references are not counted). References should be cited in the proposal.
- A written final paper in the style of a journal article in the following slice is also required.
- Final paper is due by 21/12/2015 - soft&hardcopy.
- Each student will give classroom presentation-ppt about the final paper.
Final Paper

- The paper should not exceed 8 pages (excluding appendix) in the format provided at IEEE Manuscript Templates for Conference Proceedings: http://www.ieee.org/conferences_events/conferences/publishing/templates.html. The paper should address the following topics:
  - **Introduction**: general description of the project topic/research paper.
  - **The problem statement**: why is an artificial neural network required? What are the issues?
  - **Objective**: what are you proposing to do/show?
  - **Approach**: describe the method you have adopted.
  - **Implementation results**: include your graphs, tables, etc. here.
  - **Conclusions**: what have you learned? How can the algorithm be improved?
  - **Appendix**: listing of the source code (exclude these pages from the total page count).

WARNING !!!!!!!!

- **IMPORTANT NOTE**: We don’t accept late materials at all
- We will take your signature every class during the course that you attend.

Academic Integrity

- All programming is to be done alone!
- Do not share code with anyone else in the course (looking at the code counts as sharing)!
- Comparison of homeworks will be to catch cheaters!
- Minimum penalty is 2-letter-grade-drop for the course for everyone involved.

Where to go for help

- You can discuss your programs with anyone!
- Feel free to send your code to turgayibrikci@hotmail.com and ask me for help.
- Please write your name and coursenumber to the subject line
- Bring your code to the class

Introduction to Artificial Neural Networks

Fundamental Concepts of Intelligent

Contents

- Fundamental Concepts of Intelligent
- Fundamental Concepts of ANNs.
- Basic Models and Learning Rules
  - Neuron Models
  - ANN structures
  - Learning
- Distributed Representations
- Conclusions
Philosophers have been trying for over 2000 years to understand and resolve two Big Questions of the Universe:

- How does a human mind work?
- Can non-humans have minds?

These questions are still unanswered. Then we need to ask what Intelligence is.

We can define intelligence as the ability to learn and understand, to solve problems and to make decisions.

The goal of artificial intelligence (AI) as a science is to make machines do things that would require intelligence if done by humans. Therefore, the answer to the question Can Machines Think? was vitally important to the discipline.

The answer is not a simple "Yes" or "No".

One of the most significant papers on machine intelligence, "Computing Machinery and Intelligence," was written by the British mathematician Alan Turing over fifty five years ago. However, it still stands up well under the test of time, and the Turing's approach remains universal.

Turing did not provide definitions of machines and thinking, he just avoided semantic arguments by inventing a game, the Turing Imitation Game.

The imitation game originally included two phases.
In the first phase, the interrogator, a man and a woman are each placed in separate rooms.
The interrogator's objective is to work out who is the man and who is the woman by questioning them.
In the second phase of the game, the man is replaced by a computer programmed to deceive the interrogator as the man did. It would even be programmed to make mistakes and provide fuzzy answers in the way a human would. If the computer can fool the interrogator as often as the man did, we may say this computer has passed the intelligent behaviour test.

Turing believed that by the end of the 20th century it would be possible to program a digital computer to play the imitation game. Although modern computers still cannot pass the Turing test, it provides a basis for the verification and validation of knowledge-based systems.

A program intelligence, in some narrow area of expertise, is evaluated by comparing its performance with the performance of a human expert. To build an intelligent computer system, we have to capture, organize and use human expert knowledge in some narrow area of expertise.

Artificial neural networks are composed of interconnecting artificial neurons (programming constructs that mimic the properties of biological neurons).

What is Neural Networking??

Human Intelligence VS Artificial Intelligence
Human Intelligence VS Artificial Intelligence

**Human Intelligence**
- Intuition, Common sense, Judgment, Creativity, Beliefs etc
- The ability to demonstrate their intelligence by communicating effectively
- Plausible Reasoning and Critical thinking

**Artificial Intelligence**
- Ability to simulate human behavior and cognitive processes
- Capture and preserve human expertise
- Fast Response. The ability to comprehend large amounts of data quickly.

**Humans are fallible**
- They have limited knowledge bases
- Information processing of serial nature proceed very slowly in the brain as compared to computers
- Humans are unable to retain large amounts of data in memory.

**We achieve more than we know. We know more than we understand.**
We understand more than we can explain (Claude Bernard, 19th C French scientific philosopher)

**Artificial Intelligence VS Conventional Computing**

**Artificial Intelligence**
- AI software uses the techniques of search and pattern matching
- Programmers design AI software to give the computer only the problem, not the steps necessary to solve it

**Conventional Computing**
- Conventional computer software follow a logical series of steps to reach a conclusion
- Computer programmers originally designed software that accomplished tasks by completing algorithms

**Psychology And Artificial intelligence**

The functionalist approach of AI views the mind as a representational system and psychology as the study of the various computational processes whereby mental representations are constructed, organized, and interpreted.

(Margaret Boden's essays written between 1982 and 1988)

**Artificial intelligence & Our society**

- To supplement natural intelligence for e.g we are building intelligence in an object so that it can do what we want it to do, as for example-- robots, thus reducing human labour and reducing human mistakes
Introduction to Artificial Neural Networks

What is ANN? Why ANN?
- ANN – Artificial Neural Networks
  - To simulate human brain behavior
  - A new generation of information processing system.

Applications
- Pattern Matching
- Pattern Recognition
- Associate Memory (Content Addressable Memory)
- Function Approximation
- Learning
- Optimization
- Vector Quantization
- Data Clustering
- ...

Traditional Computers are inefficient at these tasks although their computation speed is faster.

The Configuration of ANNs
- An ANN consists of a large number of interconnected processing elements called neurons.
  - A human brain consists of ~10^{11} neurons of many different types.
- How ANN works?
  - Collective behavior.

The Biologic Neuron

Symple
Nucleus
Axon
Cell body
Dendrites
The Biologic Neuron

Connection point

chemical tail

The Biologic Neuron

Excitatory or Inhibitory

The Artificial Neuron

Proposed by McCulloch and Pitts [1943]

M-P neurons

The Artificial Neuron

- positive – excitatory
- negative – inhibitory
- zero – no connection

The Artificial Neuron - Perceptron

\[ y_i(t+1) = \alpha(f) \]

\[ f(\mu_i) = \sum_{j=1}^{m} w_{ij} x_j - \theta_i \]

\[ \alpha(f) = \begin{cases} 1 & f \geq 0 \\ 0 & \text{otherwise} \end{cases} \]
What can be done by M-P neurons?

- A hard limiter.
- A binary threshold unit.
- Hyperspace separation.

What ANNs will be?

- ANN – A neurally inspired mathematical model.
- Consists a large number of highly interconnected PEs.
- Its connections (weights) holds knowledge.
- The response of PE(Process Element) depends only on local information.
- Its collective behavior demonstrates the computation power.
- With learning, recalling and, generalization capability.

Three Basic Entities of ANN Models

- Models of Neurons or PEs.
- Models of synaptic interconnections and structures.
- Training or learning rules.

Introduction to Artificial Neural Networks

Basic Models and Learning Rules

- Neuron Models
- ANN structures
- Learning

Processing Elements

Extensions of M-P neurons

Integration Functions

- M-P neuron: \[ f_i = \text{net}_i = \sum w_i x_i - \theta \]
- Quadratic Function: \[ f_i = \sum w_i x_i^2 - \theta \]
- Spherical Function: \[ f_i = \sum (x_i - w_i)^2 - \theta \]
- Polynomial Function: \[ f_i = \sum \sum (w_i x_j x_k + x_i^m + x_k^n) - \theta \]
Activation Functions

M-P neuron (Step function)

\[ a(f) = \begin{cases} 1 & f \geq 0 \\ 0 & \text{otherwise} \end{cases} \]

Hard Limiter (Threshold function)

\[ a(f) = \text{sgn}(f) = \begin{cases} 1 & f \geq 0 \\ -1 & f < 0 \end{cases} \]

Ramp function:

\[ a(f) = \begin{cases} 1 & f > 1 \\ f & 0 \leq f \leq 1 \\ 0 & f < 0 \end{cases} \]

Unipolar sigmoid function:

\[ a(f) = \frac{1}{1 + e^{-2f}} \]

Bipolar sigmoid function:

\[ a(f) = \frac{2}{1 + e^{-2f}} - 1 \]

Example: Activation Surfaces
Example: Activation Surfaces

M-P neuron: (Step function)

\[ f(f) = \begin{cases} 1, & f \geq 0 \\ 0, & \text{otherwise} \end{cases} \]

Unipolar sigmoid function:

\[ f(x) = \frac{1}{1 + e^{-x}} \]
Introduction to Artificial Neural Networks

Basic Models and Learning Rules
- Neuron Models
- ANN structures
- Learning

ANN Structure (Connections)

Single-Layer Feedforward Networks

Multilayer Feedforward Networks

Pattern Recognition

Multilayer Feedforward Networks

Single Node with Feedback to Itself
Introduction to Artificial Neural Networks

Basic Models and Learning Rules
- Neuron Models
- ANN structures
- Learning

Single-Layer Recurrent Networks

Multilayer Recurrent Networks

Learning
- Consider an ANN with \( n \) neurons and each with \( m \) adaptive weights.
- Weight matrix:

\[
W = \begin{bmatrix}
  w_{11} & w_{12} & \cdots & w_{1m} \\
  w_{21} & w_{22} & \cdots & w_{2m} \\
  \vdots & \vdots & \ddots & \vdots \\
  w_{n1} & w_{n2} & \cdots & w_{nm}
\end{bmatrix}
\]

Learning How?
- To "Learn" the weight matrix.
- Consider an ANN with \( n \) neurons and each with \( m \) adaptive weights.
- Weight matrix:

\[
W = \begin{bmatrix}
w_1^T \\
w_2^T \\
\vdots \\
w_n^T
\end{bmatrix}
= \begin{bmatrix}
w_{11} & w_{12} & \cdots & w_{1m} \\
w_{21} & w_{22} & \cdots & w_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
w_{n1} & w_{n2} & \cdots & w_{nm}
\end{bmatrix}
\]

Learning Rules
- Supervised learning
- Unsupervised learning
- Reinforcement learning
Supervised Learning

- Learning with a teacher
- Learning by examples
  - Training set
  \[ T = \{(x^{(1)}, d^{(1)}), (x^{(2)}, d^{(2)}), \ldots, (x^{(k)}, d^{(k)})\} \]

Unsupervised Learning

- Self-organizing
- Clustering
  - Form proper clusters by discovering the similarities and dissimilarities among objects.

Reinforcement Learning

- Learning with a critic
- Learning by comments
The General Weight Learning Rule

Input: $net_i = \sum_{j=1}^{m} w_{ij} x_j - \theta_i$

Output: $y_i = a(net_i)$

We want to learn the weights & bias.

Let $x_m = -1$ and $w_{im} = \theta_i$.

$net_i = \sum_{j=1}^{m} w_{ij} x_j$

$\Delta w_i(t) = ?$

We want to learn the weights & bias.

The General Weight Learning Rule

Input: $net_i = \sum_{j=1}^{m} w_{ij} x_j - \theta_i$

Output: $y_i = a(net_i)$

The General Weight Learning Rule

Input: $net_i = \sum_{j=1}^{m} w_{ij} x_j - \theta_i$

Output: $y_i = a(net_i)$

We want to learn $w = (w_{i1}, w_{i2}, \ldots, w_{im})^T$.

The General Weight Learning Rule

Input: $net_i = \sum_{j=1}^{m} w_{ij} x_j$

Output: $y_i = a(net_i)$

The General Weight Learning Rule

Input: $net_i = \sum_{j=1}^{m} w_{ij} x_j - \theta_i$

Output: $y_i = a(net_i)$

Learning Signal Generator

$w_i = (w_{i1}, w_{i2}, \ldots, w_{im})^T$

Learning Signal Generator

$w_i$
The General Weight Learning Rule

\[ \Delta w_i(t) \propto r x(t) \]

We want to learn \( w = (w_{i1}, w_{i2}, \ldots, w_{im})^T \)

Discrete-Time Weight Modification Rule:

\[ w_{i(t+1)} = w_{i(t)} + \eta f_i(w_{i(t)}, x(t), d_i(t)) x(t) \]

Continuous-Time Weight Modification Rule:

\[ \frac{dw_i(t)}{dt} = \eta r x(t) \]

Hebb's Learning Law

- Hebb [1949] hypothesis that when an axonal input from A to B causes neuron B to immediately emit a pulse (fire) and this situation happens repeatedly or persistently.

- Then, the efficacy of that axonal input, in terms of ability to help neuron B to fire in future, is somehow increased.

- Hebb's learning rule is a unsupervised learning rule.

\[ r = f_r(w_i, x, d_i) = a(w_i^T x) = y_i \]

\[ \Delta w_i(t) = \eta y_i x \]

\[ \Delta w_{ij} = \eta y_i x_j \]
Introduction to Artificial Neural Networks

Distributed Representations

- Make induction easy.
- Each entity is represented by one PE.

Advantages

- Act as a content addressable memory.
- Act as a content addressable memory.
- Make induction easy.

Example

<table>
<thead>
<tr>
<th>P_0</th>
<th>P_1</th>
<th>P_2</th>
<th>P_3</th>
<th>P_4</th>
<th>P_5</th>
<th>P_6</th>
<th>P_7</th>
<th>P_8</th>
<th>P_9</th>
<th>P_10</th>
<th>P_11</th>
<th>P_12</th>
<th>P_13</th>
<th>P_14</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dog</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Cat</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Bread</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
</tbody>
</table>

Advantages

- Act as a content addressable memory.
- Act as a content addressable memory.
- Make the creation of new entities or concept easy (without allocation of new hardware).

What is this?

Add doughnut by changing weights.
Advantages

- Act as a content addressable memory.
- Make induction easy.
- Make the creation of new entities or concept easy (without allocation of new hardware).
- Fault Tolerance.

<table>
<thead>
<tr>
<th>P_0</th>
<th>P_1</th>
<th>P_2</th>
<th>P_3</th>
<th>P_4</th>
<th>P_5</th>
<th>P_6</th>
<th>P_7</th>
<th>P_8</th>
<th>P_9</th>
<th>P_{10}</th>
<th>P_{11}</th>
<th>P_{12}</th>
<th>P_{13}</th>
<th>P_{14}</th>
<th>P_{15}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dog</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Cat</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Bread</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

Some PEs break down don’t cause problem.

Disadvantages

- How to understand?
- How to modify?

Learning procedures are required.

REFERENCES

- All materials were collected from different web pages.
- Thanks for the authors of these slides.