Impact of Deep Learning on our Society

2/14, 2018
Hiroshi Maruyama
Agenda

1. What is Deep Learning
2. Implications to Software Development
3. Implications to us Individuals
Agenda

1. What is Deep Learning
2. Implications to Software Development
3. Implications to us Individuals
What is Deep Learning? – A ( Stateless) Function

\[ Y = f(X) \]

- Very high-dimensional, any combination of continuous and categorial variables
- Low-dimensional for classification, very high-dimensional for generation
Example: Converting Celsius to Fahrenheit

Requirements

Input: C
Output: F
Where F is Fahrenheit equivalent of C in Celsius

Model

\[ F = 1.8 \times C + 32 \]

Implementation

```c
double c2f(double c) {
    return 1.8*c + 32.0;
}
```
Alternative Approach – Data-Driven, Inductive Programming

Find a model that represents this data set
Machine Learning (aka Statistical Modeling) does this!

**Estimated Model**

\[ F = 1.8 \times C + 32 + e \]

\[ e \sim N(0,10) \]
A Catch: the Model must be fixed in advance

What is the function that represents this data?

Too many parameters result in overfitting!

Choosing the right model is difficult
Deep Learning can approximate a function without too much overfitting (in many cases)

Mean-square error is back-propagated

Ramp function as activation function

2-layers of 10-node hidden layers

Total 141 parameters

Good approximation without too much overfitting

Input

Output

Graph showing the ramp function as an activation function and the total 141 parameters.
Deep Learning as a Universal Computing Mechanism

- Very large number of parameters
- Can approximate ANY high-dimensional function*

⇒ Pseudo Turing Complete!

How Deep Learning Works by Yann LeCun

https://code.facebook.com/pages/1902086376686983
Image Segmentation for autonomous driving

https://www.youtube.com/watch?v=IGOjchGdVQs
Image Segmentation for Picking Robot

2nd Place in the Picking Task in Amazon Picking Challenge
Image + Natural Language for Controlling Robot
Auto Coloring Line Drawings
Reinforcement Learning for Autonomous Driving

Consumer Electronics Show (CES) 2016
Development with Reinforcement Learning

https://research.preferred.jp/2015/06/distributed-deep-reinforcement-learning/
Generative Model Captures Statistical Essence of Training Data
Deep Learning Requires lots of Computation

**Image**

- \(10^P\) (Image) \(\sim\) \(10^E\) (Video) Flops
- 100 million images

**Life Science**

- \(100^P\sim 1^E\) Flops
- 10M SNPs per person. 100PF for 1 million, 1EF for 100 million.

**Speech Rec.**

- \(10^P\sim\) Flops
- 5K hours of 10K people of audio data
- 100K hours of synthetic audio data for training [Baidu 2015]

**Autonomous Driving**

- \(1^E\sim 100^E\) Flops
- 1TB/day/autonomous cars
- 100 days of data
- 1~100M cars

**Robotics/Drone**

- \(1^E\sim 100^E\) Flops
- 1TB/car/year
- Data from 1~100M cars

Machine generated data is much bigger than human generated data

These estimation is based on;
To finish training using 1GB within 1 day require 1Tflops
Fundamental Limitation of ML (1)

Statistical Machine Learning works only if the future is similar to the past
Fundamental Limitation of ML (2)

- Powerless on data in unseen regions
Fundamental Limitation of ML (3)

- Always works statistically

No guarantee of “100% correctness”
Agenda

1. What is Deep Learning
2. Implications to Software Development
3. Implications to us Individuals
Typical ML-Based System

Instrumentation

Pipeline

Data Source

ETL

Data prep

Data prep

Training

Serving

Raw Data

Training data set

Prerained Model

Pipeline
ML-Based System Lifecycle

The Process is more exploratory
Conventional Deductive Development (Manufacturing-Inspired)

System with modularized components (e.g., recognition, planning, feedback control, …)
Inductive Development (=Machine Learning)

Requirements = Training Data Sets

Monolithic end-to-end system (deep neural net)

Training Data (input/output pairs)

Input → Machine Learning → Output
Autonomous Vehicle Demo at Toyota’s Booth, Consumer Electronics Show 2016

https://www.youtube.com/watch?v=7A9UwxvgcV0
Trial-and-Error Development (a.k.a. Reinforcement Learning)

Requirements = After-the-fact Feedback
New Paradigm of System Development

- **Manufacturing-Inspired**
  - Huge cost of modification after development
  - Requirement Definition in advance

- **Machine Learning-Inspired**
  - System modification cost is virtually zero
  - After-the-fact Requirement Definition

Cf. Waterfall -> Agile -> DevOps -> …
Software Crisis in 1960’s

Dawn of Software Engineering

---

<table>
<thead>
<tr>
<th>Table 1</th>
<th>EXTENDED FLOATING POINT ADD ROUTINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>* Special extended precision addition of FFRs 0 and 4</td>
</tr>
<tr>
<td>3</td>
<td>** DxoXard04 Eceed</td>
</tr>
<tr>
<td>4</td>
<td>** RHsA EQU 0</td>
</tr>
<tr>
<td>5</td>
<td>** RHsB EQU 2</td>
</tr>
<tr>
<td>6</td>
<td>** RHsB EQU 4</td>
</tr>
<tr>
<td>7</td>
<td>** RHsB EQU 6</td>
</tr>
<tr>
<td>8</td>
<td>** USING 'A', 'B'</td>
</tr>
<tr>
<td>9</td>
<td>** START B BEGIN</td>
</tr>
<tr>
<td>10</td>
<td>** Dc AL1(8)</td>
</tr>
<tr>
<td>11</td>
<td>** Before touching GPR’s, save them</td>
</tr>
<tr>
<td>12</td>
<td>** BEGIN ERM 14,12,12(13) Save caller’s registers</td>
</tr>
<tr>
<td>13</td>
<td>** LR 2,15</td>
</tr>
<tr>
<td>14</td>
<td>** ST 2,SAVEX6</td>
</tr>
<tr>
<td>15</td>
<td>** LA 12,SAVER</td>
</tr>
<tr>
<td>16</td>
<td>** ST 12,SA(2)</td>
</tr>
<tr>
<td>17</td>
<td>** LTR RHs,SHoB Check high order part of B</td>
</tr>
<tr>
<td>18</td>
<td>** BZ RETURN B is zero so done</td>
</tr>
<tr>
<td>19</td>
<td>** MNOTO LTR RHs,SHoA Check high order part of A</td>
</tr>
<tr>
<td>20</td>
<td>** BNZ AMOTO RHs,SHoA A is zero so set A to B</td>
</tr>
<tr>
<td>21</td>
<td>** LDH RHs,SHoB</td>
</tr>
<tr>
<td>22</td>
<td>** LDH RHs,SHoB</td>
</tr>
<tr>
<td>23</td>
<td>** RETURN Finished</td>
</tr>
<tr>
<td>24</td>
<td>** Neither A nor B is zero -- save A and B and continue</td>
</tr>
<tr>
<td>25</td>
<td>** AMOTO STD RHs,SHoA Save high order part of A</td>
</tr>
<tr>
<td>26</td>
<td>** STD XLsA,SHoA Save low order part of A</td>
</tr>
<tr>
<td>27</td>
<td>** STD X,SHoB Save high order part of B</td>
</tr>
<tr>
<td>28</td>
<td>** STD XLsB,SHoB Save low order part of B</td>
</tr>
<tr>
<td>29</td>
<td>** Check for result in range so standard add can be done</td>
</tr>
<tr>
<td>30</td>
<td>** L 2,SHoA Load high order A</td>
</tr>
<tr>
<td>31</td>
<td>** X 2,SHoA Exclusive or to check sign match</td>
</tr>
<tr>
<td>32</td>
<td>** L 3,SHoB Load high order B</td>
</tr>
<tr>
<td>33</td>
<td>** X 3,SHoB Exclusive or to check sign match</td>
</tr>
<tr>
<td>34</td>
<td>** LTR 2,2 Result = 0 they watch -- A in range</td>
</tr>
<tr>
<td>35</td>
<td>** BM NOTSD No match-out of range</td>
</tr>
<tr>
<td>36</td>
<td>** LTR 2,3 Result = 0 they watch -- B is range</td>
</tr>
<tr>
<td>37</td>
<td>** BM NOTSD No match-out of range</td>
</tr>
</tbody>
</table>
Similarly, we should start “Machine Learning Engineering” now!

- Requirements engineering
  - How to translate requirements into training data sets?
  - How to communicate pros & cons of ML with stakeholders?
- Testing
  - How to do “Regression Test” in ML?
  - What can we learn from the practice of Test-Driven Development (TDD)?
- Tools
  - What should ML IDE look like?
  - How should we manage ML work products, especially training data sets and pretrained models?
  - How to avoid overfitting caused by overly automated tools?
- Operation
  - How to detect “concept drift” in operations?
Agenda

1. What is Deep Learning
2. Implications to Software Development
3. Implications to us Individuals
Please make clear distinction between generalized AI and specialized AI

“There’s a distinction, which is probably familiar to a lot of your readers, between generalized AI and specialized AI.”

https://www.wired.com/2016/10/president-obama-mit-joji-ito-interview/
(Specialized) A. I. is not a technology per se – It is an activity to extend horizon of computer science

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Symbol Processing (LISP)</td>
<td>• Knowledge Representation</td>
<td>• Statistical Machine Learning</td>
</tr>
<tr>
<td>• Means-End Analysis</td>
<td>• Expert System</td>
<td>• Deep Learning</td>
</tr>
<tr>
<td>• Language Parsing</td>
<td>• Ontology</td>
<td></td>
</tr>
</tbody>
</table>

- Garbage Collection
- Search Algorithms
- Formal Language Theory
- :

- Object-Oriented Language
- Modeling
- Semantic Web
- :

Inductive Programming
What is Intelligence? – Two Types of Thinking

• System 1 Thinking
  • Fast
  • Automatic
  • Intuitive
  • E.g., $2 \times 2 = ?$

• System 2 Thinking
  • Slow
  • Consumes attention
  • Logical
  • E.g., $27 \times 31 = ?$

How AI Research Evolved

1st and 2nd waves of AI
• System 2 Thinking
  • Logical
  • Reason
  • Symbolic
  • Deductive

3rd wave of AI
• System 1 Thinking
  • Perception
  • Intuition
  • Distributed
  • Inductive

Which is more “human-like” intelligence?
I asked a carpenter’s son, “what would you like to be in the future?”

He said, “I want to be a great tofu maker like my father.”
Big Data (and ML) can manipulate you via your cognitive bias

“Echo-chamber effect” – everybody you see has the same opinion with you
Remember, our society is based on the fundamental value assumption that people are rational

- Democracy, capitalism, science, etc. are all strongly influenced by “enlightenment (啓蒙思想)”

John Locke  Jean-Jacques Rousseau  Immanuel Kant

Thank You