Verification and Validation of Deep Learning

Xiaowei Huang
University of Liverpool, UK

UDRC Workshop, November, 2021
Outline

Introduction

Formal Verification

Statistical Evaluation

Safety Assurance

Conclusions
Introduction
Deep Learning in Safety-Critical Systems

Figure: Driverless Car [6], Autonomous Underwater Vehicles [7], Drone for inspection [5], Smart Grid [2], Net-zero building [1], etc.
Can We Trust its Decisions?

Question: Can we really trust the decisions made by deep learning models, especially on safety-critical applications?

This question can be broken down into a number of more concrete questions, such as

- How does a deep learning model make a decision?
- Does deep learning always make a correct decision?
- Under what circumstances a deep learning model will make a wrong/correct decision?
- ... ...
Vulnerability

- Robustness – local (input-level) safety
  - wrt input perturbation, weight perturbation, etc
- Generalisation – global (model-level) safety
  - wrt different operational environment
- Security
  - wrt data corruption & poisoning, data privacy, etc
- Explainability
DL model: classifies $\alpha$ and $\alpha'$ **differently**
Human: should remain the **same**
Model Improvement for Robustness

- Decision Boundary by Human Perception
- Adversarial Example
- DL model’s Decision Boundary after Adversarial Training
- Adversarial Example
- \( \alpha \)
- \( \alpha' \)
(Robustness) Verification: verify if a certain input area can exclude misclassification with guarantees
Generalisation Error

How about these unseen datapoints which are far away from known data?
Direction 1: identify the errors
  ▶ adversarial attack, security attack, etc
Direction 2: determine if it is without error
  ▶ verification
Direction 3: reduce errors by improving models
  ▶ adversarial training, adversarial defence, etc
Direction 4: quantify the errors
  ▶ statistical evaluation, software testing, etc
Direction 5: demonstrate the safety for development cycle
  ▶ safety assurance, reliability estimation, etc
Trend of relevant research

Figure: https://nicholas.carlini.com/writing/2019/all-adversarial-example-papers.html

Formal Verification
What is Verification?

Properties, e.g.,
1. robustness,
2. fairness,
3. interpretability,
4. privacy,
etc.

Verification Algorithms

feedback

Counterexamples

fails

holds
Robustness Verification
Robustness Verification for Neural Network

- **Verification Methods**
  - Constraint Solving
  - Over-approximation
  - Global optimisation

- **What is the difficulty?**
  - Scalability
  - Only deal with robustness
Statistical Evaluation
State-of-the-art

- Sampling-based methods
- Software testing methods
- Deep learning theory based methods

May work with both robustness and generalisation.
Sampling-based Methods

- Sampling & fitting distributions
- Guarantee from some theories e.g., minimum adversarial distortion follows extreme value distributions [9].

- Enhanced Monte Carlo sampling [8]
- Guarantee from statistics theory
Software Testing Methods

- Well established in many industrial standard for software used in safety critical systems, such as ISO26262 for automotive systems and DO 178B/C for avionic systems.

- Coverage Metrics
  - structural coverage
  - scenario coverage

- Test Case Generation Methods
  - fuzzing
  - symbolic execution, etc

- to determine if the generated test cases include bugs.
Software Testing Methods

- Industrial standards need to be upgraded

- A few Coverage Metrics
  - Structural Test Coverage Criteria for Deep Neural Networks. EMSOFT 2019

- A few Test Case Generation Methods
  - Concolic Testing for Deep Neural Networks. ASE 2018

- Use a set of generated test cases to either finding bugs or evaluating the performance of a neural network
Deep learning theory based methods

How to statistically predict the generalisation capability of a network, according to some structural information such as weights, architectures, etc?

<table>
<thead>
<tr>
<th>Methods</th>
<th>Generation of input samples?</th>
<th>Utilisation of structural information?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sampling</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Software testing</td>
<td>Yes</td>
<td>Maybe</td>
</tr>
<tr>
<td>DL theory</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table: Comparison between statistical evaluation methods
Deep Learning theory

Which DL theory?

- PAC Bayesian Theory, to upper bound the gap between expected loss on input space and expected loss on training samples, by taking into consideration the change of weights before and after the training.
- Vapnik–Chervonenkis (VC) dimension, to measure of the capacity (complexity, expressive power, richness, or flexibility) of a set of functions that can be learned by a statistical binary classification algorithm.
- etc.
Relax the i.i.d. assumption on the posterior distribution, and define quantities such as weight correlation (WC) based on structural information.

**Figure:** (FCN) The WC of any two neurons is the cosine similarity of the associated weight vectors. (CNN) The WC of any two filters is the cosine similarity of the reshaped filter matrices.
Table 1: Complexity Measures (Measured Quantities)

<table>
<thead>
<tr>
<th>Measure</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generalisation Error (GE)</td>
<td>$\mathcal{L}<em>D(f</em>{\theta^F}) - \mathcal{L}<em>S(f</em>{\theta^F})$</td>
</tr>
<tr>
<td>Product of Frobenius Norms (PFN)</td>
<td>$\prod_\ell |\theta_\ell^F|_{Fr}$</td>
</tr>
<tr>
<td>Product of Spectral Norms (PSN)</td>
<td>$\prod_\ell |\theta_\ell^F|_2$</td>
</tr>
<tr>
<td>Number of Parameters (NoP)</td>
<td>Total number of parameters in the network</td>
</tr>
<tr>
<td>Sum of Spectral Norms (SoSP)</td>
<td>Total number of parameters $\times \sum_\ell |\theta_\ell^0 - \theta_\ell^F|_2$</td>
</tr>
<tr>
<td>Weight Correlation (WC)</td>
<td>$\frac{1}{T} \sum_\ell \rho(w_\ell)$</td>
</tr>
<tr>
<td>PAC Bayes (PB)</td>
<td>$\sum_\ell |\theta_\ell^0 - \theta_\ell^F|<em>{Fr}^2/2\sigma</em>\ell^2$</td>
</tr>
<tr>
<td>PAC Bayes &amp; Correlation (PBC)</td>
<td>$\sum_\ell (|\theta_\ell^0 - \theta_\ell^F|<em>{Fr}^2/2\sigma</em>\ell^2 + g(w_\ell))$</td>
</tr>
</tbody>
</table>

Table 2: Complexity measures for CIFAR-10

<table>
<thead>
<tr>
<th>Network</th>
<th>PFN</th>
<th>PSN</th>
<th>NoP</th>
<th>SoSP</th>
<th>PB</th>
<th>PBC</th>
<th>WC</th>
<th>GE</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCN1</td>
<td>8.1e7</td>
<td>1.4e4</td>
<td>3.7e7</td>
<td>1.6e9</td>
<td>1.1e4</td>
<td>1.14e5</td>
<td>0.297</td>
<td>2.056</td>
</tr>
<tr>
<td>FCN2</td>
<td>3.3e7</td>
<td>8.5e3</td>
<td>4.2e7</td>
<td>1.61e9</td>
<td>8.8e3</td>
<td>1.24e5</td>
<td>0.296</td>
<td>2.354</td>
</tr>
<tr>
<td>VGG11</td>
<td>8.5e10</td>
<td>1.4e5</td>
<td>9.7e6</td>
<td>2.4e8</td>
<td>2.0e3</td>
<td>3.41e4</td>
<td>0.273</td>
<td>0.929</td>
</tr>
<tr>
<td>VGG16</td>
<td>5.1e15</td>
<td>1.3e7</td>
<td>1.5e7</td>
<td>5.2e8</td>
<td>2.6e3</td>
<td>3.73e4</td>
<td>0.275</td>
<td>0.553</td>
</tr>
<tr>
<td>VGG19</td>
<td>1.1e19</td>
<td>2.9e8</td>
<td>2.1e7</td>
<td>8.1e8</td>
<td>3.3e3</td>
<td>4.26e4</td>
<td>0.274</td>
<td>0.678</td>
</tr>
<tr>
<td>ResNet18</td>
<td>2.5e22</td>
<td>1.1e12</td>
<td>1.1e7</td>
<td>8.4e8</td>
<td>4.7e3</td>
<td>1.34e5</td>
<td>0.732</td>
<td>2.681</td>
</tr>
<tr>
<td>ResNet34</td>
<td>9.9e34</td>
<td>4.9e16</td>
<td>2.1e7</td>
<td>3.1e9</td>
<td>1.0e4</td>
<td>1.30e5</td>
<td>0.733</td>
<td>2.552</td>
</tr>
<tr>
<td>ResNet50</td>
<td>1.4e76</td>
<td>7.5e46</td>
<td>2.3e7</td>
<td>6.1e9</td>
<td>1.6e7</td>
<td>1.62e7</td>
<td>0.278</td>
<td>2.807</td>
</tr>
<tr>
<td>DenseNet121</td>
<td>5.9e176</td>
<td>1.4e151</td>
<td>6.8e6</td>
<td>1.5e10</td>
<td>1.0e9</td>
<td>1.04e9</td>
<td>0.357</td>
<td>1.437</td>
</tr>
<tr>
<td>Concordant Pairs</td>
<td>21</td>
<td>21</td>
<td>22</td>
<td>26</td>
<td>24</td>
<td>29</td>
<td>24</td>
<td>-</td>
</tr>
<tr>
<td>Discordant Pairs</td>
<td>15</td>
<td>15</td>
<td>14</td>
<td>10</td>
<td>12</td>
<td>7</td>
<td>12</td>
<td>-</td>
</tr>
<tr>
<td>Kendall’s $\tau$</td>
<td>0.16</td>
<td>0.16</td>
<td>0.22</td>
<td>0.44</td>
<td>0.33</td>
<td>0.61</td>
<td>0.33</td>
<td>-</td>
</tr>
</tbody>
</table>

New measure
Safety Assurance
Play video demo:
https://www.youtube.com/watch?v=akY8f5sSFpY&t=1s
Conclusions
Conclusions

▶ Most efforts are taken on finding errors on pre-trained models;
▶ Adversarial examples are inherent to deep learning, i.e., errors cannot be eliminated;

What shall we do?
▶ focus on acceptable level of safety;
▶ consider how the deep learning models are used;
▶ consider safety assurance on not only the pre-trained models but also the development cycle.
Getting serious about net zero buildings.

Smart grid.


RBR. Neurala, avisight team up for ai-powered drone inspections, 2020.

