Towards Better Understanding Of Deep Learning With Visualization

Presenter: Haipeng Zeng
Supervised by: Huamin Qu
Department of Computer Science & Engineering
2016.11.10
• Deep Learning

Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction.

(LeCun et al., 2015)
Introduction-Background

- Remarkable Progress

Deep Learning

... Image Recognition

Speech Recognition

Natural language Processing
Introduction-Background

• Black Box
  – No clear understanding of the inner working mechanism
  – Compared with other machine learning models
  – A substantial amount of trial-and-error procedures

Deep Learning → Visualization
• A simple analogy

Visualization → Deep Learning

Microscope → cells

Introduction - Motivation
Introduction-Motivation

• Visualization on Deep Learning
  – Help understand deep learning models intuitively
  – Help train a better model efficiently
  – Make decisions more interpretable
Taxonomy

• A taxonomy based on:
  – the **challenges** that **deep learning** faces
  – the **purposes** that **visualization techniques** serve

Challenges on Deep Learning:

- How a deep learning model **works**?
- How to **improve** a deep learning model?

Purposes of Visualization:

- Visualize the **features** learned by a deep learning model
- Visualize the **relationships** in a deep learning model
- Visualize the whole **process** of a deep learning model
Taxonomy

- A taxonomy based on:
  - the **challenges** that deep learning **faces**
  - the **purposes** that visualization techniques **serve**
## Taxonomy

- **The related papers in this survey**

<table>
<thead>
<tr>
<th>Feature Visualization</th>
<th>Representation Depiction Methods</th>
<th>Related Paper</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Karpathy, 2014a), (Yosinski et al., 2015), (Karpathy et al., 2015), (Strobelt et al., 2016), etc.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Zeiler &amp; Fergus, 2014), (Zhou et al., 2014), (Girshick et al., 2014), etc.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Contribution Computation Methods</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Zeiler &amp; Fergus, 2014), (Simonyan et al., 2013), (Springenberg et al., 2014), (Bach et al., 2015), (Zhou et al., 2015), (Bahdanau et al., 2014), (Socher et al., 2013), (Hermann et al., 2015), (Goyal et al., 2016), etc.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Input Reconstruction Methods</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(Long et al., 2014), (Erhan et al., 2009), (Simonyan et al., 2013), (Alexander et al., 2015), (Mahendran &amp; Vedaldi, 2015), (Yosinski et al., 2015), (Mahendran &amp; Vedaldi, 2016), (Dosovitskiy &amp; Brox, 2015), etc</td>
<td></td>
</tr>
<tr>
<td>Relationship Visualization</td>
<td>Relationships Between Representations</td>
<td>Related Paper</td>
</tr>
<tr>
<td></td>
<td>(Maaten &amp; Hinton, 2008), (Cho et al., 2014), (Karpathy, 2014b), (Rauber et al., 2016), etc.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Relationships Between Neurons</td>
<td>(Liu et al., 2016), (Rauber et al., 2016), etc.</td>
</tr>
<tr>
<td>Process Visualization</td>
<td>Neural Network Structure (Model)</td>
<td>Related Paper</td>
</tr>
<tr>
<td></td>
<td>(Karpathy, 2014a), (Yosinski et al., 2015), (Smilkov et al., 2015), (Harley, 2015), (Chung et al., 2016), (Liu et al., 2016), (Bruckner, 2014), (Google, 2015), etc.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Training Information (Data)</td>
<td>(Google, 2015), (Bruckner, 2014), (Smilkov et al., 2015), (Skymind, 2013), (Chung et al., 2016), etc.</td>
</tr>
</tbody>
</table>
Outline

- Deep Learning Models
- Feature Visualization
- Relationship Visualization
- Process Visualization
- Conclusion and Future Work
Outline

• Deep Learning Models
  – Deep Neural Network (DNN)
  – Convolutional Neural Network (CNN)
  – Recurrent Neural Network (RNN)

• Feature Visualization

• Relationship Visualization

• Process Visualization

• Conclusion and Future Work
Deep Learning Models - DNN

- Input layer
- Output layer
- Hidden layers
- Neurons (units)

- Activation
- Representation
- Backpropagation
Deep Learning Models - CNN

• Architecture
  – Convolution layer
  – Pooling layer
  – Fully connected layer

• Convolution
  – Filter (kernel)
  – Feature map (Activation Map)

• Max-pooling
  – Translation invariance
Deep Learning Models - RNN

- Input
- Output
- Hidden state ("memory")

- LSTMs
  - Input gate
  - Forget gate
  - Output gate

Recurrent Neural Network (RNN)

Long Short Term Memory networks (LSTM)
Outline

- Deep Learning Models
- Feature Visualization
  - Representation Depiction Methods

Representation Depiction Methods refer to those methods that visually depict representation directly.

PS.
A representation is a vector of activations in a hidden layer.
Representation Depiction Methods

- CNNs
  - Activation map (Feature map)
- LSTMs
  - Hidden state

(Karpathy et al., 2015)

(Yosinski et al., 2015)

(Strobelt et al., 2016)

Introduction
Taxonomy
Deep Learning Models
Feature Visualization
Relationship Visualization
Process Visualization
Conclusion and Future Work
**Representation Depiction Methods**

- CNNs
  - Deep visualization system
  - Activation map (Feature map)

**Visualization: bitmap**

**Pros:**
- simple implement and informative
- support webcam input

**Cons:**
- scalability problem
- Hard to explain in some cases

(Yosinski et al., 2015)
• LSTMs
  – Hidden state

Visualization: parallel coordinates + heatmap
Pros:
  + show the hidden state dynamics
  + match similar hidden state patterns
Cons:
  - visual clutter
Outline

• Deep Learning Models
• Feature Visualization
  – Representation Depiction Methods
  – Input Modification Methods

Input Modification Methods refer to those methods where we modify the input and then measure the changes of the output or activations in hidden layers.
Input Modification Methods

- A gray square
  - mono colored
- A randomized patch

Visualization: heatmap+bitmap
Pros:
  + know which parts are important
Cons:
  - affected by the square or patch

(Zeiler & Fergus, 2014)

(Zhou et al., 2014)
Outline

• Deep Learning Models
• Feature Visualization
  – Representation Depiction Methods
  – Input Modification Methods
  – Contribution Computation Methods

Contribution Computation Methods refer to those methods that compute the contributions of input to the result.
Contribution Computation Methods

- Deconvolutional network
- Backpropagation
- Guide backpropagation
- Relevance propagation
- Class activation map

(Zhou et al., 2015)
(Simonyan et al., 2013)
(Springenberg et al., 2014)
(Zeiler & Fergus, 2014)
(Bach et al., 2015)
• Deconvolutional network

Remarkable results:
Low layers detect low features: edge, color, etc.
High layers detect high features: object, etc.

Visualization: bitmap
Pros:
+ know each pixel’s contribution
+ provide a non-parametric view of invariance

Cons:
- cannot visualize the joint activity in a layer
  pose variation, e.g. keyboards, dogs

(Zeiler & Fergus, 2014)
Visualization on text:
- Tree structure
- Matrix
- Heatmap

very negative to very positive
(−−, −, 0, +, ++)

(Socher et al., 2013)

(Bahdanau et al., 2014)

(Hermann et al., 2015)

(Goyal et al., 2016)
Outline

• Deep Learning Models

• Feature Visualization
  – Representation Depiction Methods
  – Input Modification Methods
  – Contribution Computation Methods
  – Input Reconstruction Methods

Contribution Computation Methods refer to the methods that reconstruct the input based on representations in the network.
Input Reconstruction Methods

- **Gradient**
- **Replacement**
- **A generative network**

(Mahendran & Vedaldi, 2015)

(Simonyan et al., 2013)

(Alexander et al., 2015)

(Dosovitskiy & Brox, 2015)

Introduction	Taxonomy	Deep Learning Models	Feature Visualization	Relationship Visualization	Process Visualization	Conclusion and Future Work
Input Reconstruction Methods

• Gradient
  – Activation maximization
  – Code inversion

Visualization: image
Pros:
  + generate a artificial input image
Cons:
  - need natural-image priors
  - hard to get global optimization
  - some repeats

Activation maximization aims to find an image that maximally activates the neuron of interest.

(Simonyan et al., 2013)
Input Reconstruction Methods

• Gradient
  – Activation maximization
  – Code inversion

Deep Dream (inceptionism)

(Alexander et al., 2015)
Input Reconstruction Methods

• Gradient
  – Activation maximization
  – Code inversion

Visualization: image
Pros:
  1. generate artificial input images
Cons:
  1. need natural-image priors
  2. inversion is not unique

Code inversion aims to synthesize an image starting from the encoded image representation.

(Mahendran & Vedaldi, 2015)
# Feature Visualization

<table>
<thead>
<tr>
<th>Category</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Representation Depiction Methods</td>
<td>the most direct way to visualize representations, etc.</td>
<td>scalability problem; visual clutter; may be difficult to be explained in some cases, etc.</td>
</tr>
<tr>
<td>Input Modification Methods</td>
<td>easy to know which parts are important, etc.</td>
<td>just crops of input images; don’t know detail pixels’ contributions, etc.</td>
</tr>
<tr>
<td>Contribution Computation Methods</td>
<td>Intuitively get how inputs contribute to results, etc.</td>
<td>need natural-image priors, not clear how to evaluate the quality of a heatmap, etc.</td>
</tr>
<tr>
<td>Input Reconstruction Methods</td>
<td>generate artificial input images, etc.</td>
<td>need natural-image priors, images not unique, hard to get global optimization, etc.</td>
</tr>
</tbody>
</table>
Outline

• Deep Learning Models
• Feature Visualization
• **Relationship Visualization**  
  – Relationships between representations  
  – Relationships between neurons
• Process Visualization
• Conclusion and Future Work
Scatter-plot visualization
  – T-SNE projection

It projects high dimensional vectors into 2D plane and preserve neighborhoods and clusters.

Text

Image

(Cho et al., 2014)

(Karpathy, 2014b)
Relationships Between Neurons

- Directed acyclic graph
- A hybrid visualization
  - rectangle packing
  - matrix visualization
  - a biclustering-based edge bundling

Pros:
- + better understand, diagnose, and refine deep CNNs.

Cons:
- offline
- DAG, cannot support RNNs, etc.

(Liu et al., 2016)
<table>
<thead>
<tr>
<th>Category</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scatter-plot visualization</td>
<td>intuitive; visualize relationships between representation; visualize</td>
<td>visual clutter; Projection result change, etc.</td>
</tr>
<tr>
<td>(t-SNE, MDS, etc.)</td>
<td>relationships between neurons; easy to extend to other models, etc.</td>
<td></td>
</tr>
<tr>
<td>DAG-based visualization</td>
<td>intuitive; capture whole picture; visualize relationships between</td>
<td>visual clutter; offline; can not apply to RNNs, etc.</td>
</tr>
<tr>
<td>(clustering, etc.)</td>
<td>neurons; show relationships between layers, etc.</td>
<td></td>
</tr>
</tbody>
</table>
Outline

- Deep Learning Models
- Feature Visualization
- Relationship Visualization
- Process Visualization
  - Neural Network Structure
  - Training Information
- Conclusion and Future Work
Neural Network Structure

- Grid-Based Diagrams
- **Node-Link Diagrams**
- Block-Link Diagrams

(Karpathy, 2014a)
(Yosinski et al., 2015)
(Smilkov et al., 2015)
(Bruckner, 2014)
(Google, 2015)
(Harley, 2015)
(Chung et al., 2016)
(Liu et al., 2016)
Neural Network Structure

- Node-Link Diagrams

**Pro**:
- give an overview
- simplicity and intuitiveness

**Cons**:
- scalability
- visual clutter

---

**CNNVis**:
Aggregate related layers and cluster neurons; bicluster edge bundling

---

**Visualization**: Node-link diagrams

---

**Tensorflow Playground**
(Smilkov et al., 2015)

---

**CNN-3D**
(Harley, 2015)

---

**ReVACNN**
(Chung et al., 2016)

---

**CNNVis**
(Liu et al., 2016)
Training Information

- Visualization after Training
- Visualization during Training

Visualization:
- line chart, bar chart, histogram, matrix, etc.

Pros:
+ intuitive

Cons:
- trivial design

(Skymind, 2013)

Visualization (Google, 2015)

Accuracy
Activations
Activations
Weights
Bias
Bias

Process Visualization
Conclusion and Future Work
# Process Visualization

<table>
<thead>
<tr>
<th>Data Category</th>
<th>Data types</th>
<th>Visualization</th>
</tr>
</thead>
<tbody>
<tr>
<td>input data</td>
<td>images, text, etc.</td>
<td>bitmap, heatmap, t-SNE, tree structure, matrix, etc.</td>
</tr>
<tr>
<td>hidden layer data</td>
<td>activation maps, filters, hidden state, etc.</td>
<td>bitmap, heatmap, parallel coordinates, t-SNE, etc.</td>
</tr>
<tr>
<td>output data</td>
<td>loss function, accuracy, classification results, etc.</td>
<td>line chart, confusion matrix, etc.</td>
</tr>
<tr>
<td>parameters</td>
<td>weights, biases, etc.</td>
<td>line chart, histogram, bar chart, etc.</td>
</tr>
<tr>
<td>hyper-parameters</td>
<td>the number of layers, the number of neurons in each layer, learning rate, batch size, etc.</td>
<td>line chart, bar chart, etc.</td>
</tr>
</tbody>
</table>
Outline

• Deep Learning Models
• Feature Visualization
• Relationship Visualization
• Process Visualization
• **Conclusion and Future Work**
Conclusion

• Integrate visualization with deep learning
  – Feature Visualization
  – Relationship Visualization
  – Process Visualization

• Advantages
  – Understand what features learned by deep learning models
  – Grasp the inner working mechanism of deep learning models
  – Facilitate people to design and train better deep learning models
  – Make deep learning models more understandable and accessible to people
Future Work

• **New designs** are needed.
  – Solve scalability issues
  – Provide informative insight

• **A visual analysis system** is needed.
  – Instant and iterative feedback
  – A friendly user interface
  – Smooth interaction

• **Extend** visualization techniques to other data and models.
• **Combine** visualization with deep learning models on different applications.
Ph.D. Qualifying Examination

Thank you!

Nov.10th, 2016