Interpretability of Neural Networks

Kian Katanforoosh
Neural networks are deployed:

• In our phones: to recommend content,
• In banks: to manage investments,
• In hospitals: to help doctors diagnose disease symptoms,
• In insurance agencies: to evaluate risk and underwrite documents,
• In cars: to help avoid accidents.

How can one who will be held accountable for a decision trust a neural network’s recommendation, and justify its use?

I. Interpreting Neural Networks’ outputs
   A. with saliency maps
   B. with occlusion sensitivity
   C. with class activation maps (Global Average Pooling)

II. Visualizing Neural Networks from the inside
   A. with gradient ascent (class model visualization)
   B. with dataset search
   C. the deconvolution and its applications

III. (Optional: Deep Dream: going deeper in NNs)
I. A. Interpreting and visualizing Neural Networks with saliency maps

**Context:** You have built an animal classifier for a zoo. They are a little reluctant to use your model without human supervision, because they don’t understand the decision process of the model.

**Question:** What can you do to alleviate their concerns?

Can be used for segmentation?

Can be used for segmentation?

![Saliency maps](image)

I. B. Interpreting and visualizing Neural Networks with occlusion sensitivity

**Context:** You have built an animal classifier for a zoo. They are a little reluctant to use your model without human supervision, because they don’t understand the decision process of the model.

**Question:** What can you do to alleviate their concerns?

Probability map of the true class for different positions of the grey square

- Indicates low confidence on the true class for the corresponding position of the grey square
- Indicates high confidence on the true class for the corresponding position of the grey square
I. B. Interpreting and visualizing Neural Networks with occlusion sensitivity

Probability map of the true class for different positions of the grey square

- Indicates low confidence on the true class for the corresponding position of the grey square
- Indicates high confidence on the true class for the corresponding position of the grey square

Occlusion sensitivity

[Matthew D. Zeiler and Rob Fergus (2013): Visualizing and Understanding Convolutional Networks]
III. C. Interpreting NNs using class activation maps

**Context:** Along with the classification output, the zoo now wants real-time visualization of the model’s decision process. You have one day. What do you do?

Using a classification network for localization:

Converted to:

Why this?

---

III. C. Interpreting NNs using class activation maps

![Diagram showing class activation maps](image)

III. C. Interpreting NNs using class activation maps

Source video: Kyle McDonald
I. Interpreting Neural Networks’ outputs
   A. with saliency maps
   B. with occlusion sensitivity
   C. with class activation maps (Global Average Pooling)

II. Visualizing Neural Networks from the inside
   A. with gradient ascent (class model visualization)
   B. with dataset search
   C. The deconvolution and its applications

III. (Optional: Deep Dream: going deeper in NNs)
II. A. Visualizing NNs from the inside using gradient ascent (class model visualization)

**Context:** The zoo trusts that your model correctly locates animals. They get a little scared and they ask you whether the model understands what a dog is.

Given this trained ConvNet, generate an image which is representative of the class “dog” according to the ConvNet

\[ L = s_{dog}(x) - \lambda \| x \|^2 \]

Gradient ascent:

\[ x = x + \alpha \frac{\partial L}{\partial x} \]

“\( x \) should look natural”

Repeat this process:
1. Forward propagate image \( x \)
2. Compute the objective \( L \)
3. Backpropagate to get \( dL/dx \)
4. Update \( x \)'s pixels with gradient ascent

II. A. Visualizing NNs from the inside using gradient ascent (class model visualization)

We can do this for all classes:

Looks better with additional regularization methods.

L2 Regularization

Class model visualization

[Jason Yosinski et al. (2015): Understanding Neural Networks Through Deep Visualization]
II. A. Visualizing NNs from the inside using gradient ascent (class model visualization)

1. DNN model $p(\omega|\mathbf{x})$

2. Expert $p(\mathbf{x})$
   (a) none or $\ell_2$
   (b) underfitted
   (c) true
   (d) overfitted

3. Resulting prototype $\mathbf{x}^*$:
   (a)  (b)  (c)  (d)

**Fig. 2.** Cartoon illustrating how the choice of expert $p(\mathbf{x})$ affects the prototype $\mathbf{x}^*$ found by AM. The horizontal axis represents the input domain.

This method can be applied to any activation in the network in order to interpret what a neuron is detecting.

On the class score:

\[ L = S_{dog}(x) - R(x) \]

On any activation:

\[ L = a_j^{[l]}(x) - R(x) \]

[Jason Yosinski et al. (2015): Understanding Neural Networks Through Deep Visualization]
Given a filter, what examples in the dataset lead to a strongly activated feature map?

**Top 5 images**
- It seems that the filter has learned to detect shirts
- It seems that the filter has learned to detect edges

**Top 5 images**
II. B. Visualizing NNs from the inside using dataset search

How did we crop the dataset images on the previous slide?

Input image
(64,64,3)

Encoding volume
(13,13,256)

CONV
(5 layers)

Conclusion: the deeper the activation, the more it “sees” from the image
I. Interpreting Neural Networks’ outputs
   A. with saliency maps
   B. with occlusion sensitivity
   C. with class activation maps (Global Average Pooling)

II. Visualizing Neural Networks from the inside
   A. with gradient ascent (class model visualization)
   B. with dataset search

   C. The deconvolution and its applications

III. (Optional: Deep Dream: going deeper in NNs)
II. C. The deconvolution and its applications

Motivation behind deconvolution/upsampling layers

- **Generator “G” (Neural Network)**
- **Discriminator “D” (Neural Network)**
- **Binary classification**

\[
\begin{cases}
y = 0 & \text{if } x = G(z) \\
y = 1 & \text{otherwise}
\end{cases}
\]
II. C. The deconvolution and its applications

Motivation behind deconvolution/upsampling layers

Input image (400, 400, 3) → Convolutions (reduces volume height and width) → Encoding → De-convolutions (increases volume height and width) → Per-Pixel Classification (400, 400, 1)
**III. A. Interpreting NNs using deconvolutions**

**Motivation of DeconvNets for visualization:** Here is a CNN, trained on ImageNet (1.3m images, 1000 classes), we’re trying to interpret by reconstructing the activation’s zone of influence in the input space.

- **Input image** (256x256, 3)

**Diagram:**
- **Input:** Image
- **Zero Pad**
- **Convolution** (CONV)
- **Rectified Linear Unit** (ReLU)
- **Max Pooling** (MAX POOL)
- **Re-Convolution** (DECONV)
- **Unpooling** (UNPOOL)
- **ReLU**
- **De-Convolution** (DECONV)
- **Flatten**
- **Fully Connected (FC)**
- **Softmax**

**Reconstruction:**
- Display only one feature map among 256
- Keep the max activation of a feature map
- Set all other activations of the layer to 0

**Output:**
- Softmax probabilities:
  - 0.02
  - 0.93
  - 0.04
  - ...
  - 0.07
  - 0.11
  - 0.09

[Matthew D. Zeiler and Rob Fergus (2013): Visualizing and Understanding Convolutional Networks]
II. C. The deconvolution and its applications

This allows us to upsample an encoding into an image.

[Hongyang Gao et al.: Pixel Deconvolutional Networks]
[Matthew Zeiler et al.: Deconvolutional Networks]
[Vincent Dumoulin and Francesco Visin: A guide to convolution arithmetic for deep learning]
[Wenzhe Shi, et al.: Is the deconvolution layer the same as a convolutional layer?]
Motivation of DeconvNets for visualization: Here is a CNN, trained on ImageNet (1.3m images, 1000 classes), we’re trying to interpret by reconstructing the activation’s zone of influence in the input space.

III. A. Interpreting NNs using deconvolutions

[Matthew D. Zeiler and Rob Fergus (2013): Visualizing and Understanding Convolutional Networks]
II. C. The deconvolution and its applications
II. C. The deconvolution and its applications

We need to pass the filters and switches from the ConvNet to the DeconvNet.
II. C. The deconvolution and its applications

\[ a^{[l]} = I\{a^{[l+1]} \geq 0\} \cdot a^{[l+1]} \]

“ReLU forward”

“ReLU backward”

“ReLU DeconvNet”

[Sringenberg & Dosovitskiy et al. (2015): Visualizing and Understanding Convolutional Networks]
II. C. The deconvolution and its applications

We need to pass the filters and switches from the ConvNet to the DeconvNet.

- Keep the max activation of a feature map
- Set all other activations of the layer to 0

[Matthew D. Zeiler and Rob Fergus (2013): Visualizing and Understanding Convolutional Networks]
II. C. The deconvolution and its applications

Other CONV layers can be visualized the same way.

[Matthew D. Zeiler and Rob Fergus (2013): Visualizing and Understanding Convolutional Networks]
Results on a validation set of 50,000 images

- Top-9 strongest activations per filter in the 1st layer
- Because we know the position of the activation and all the pooling switches we can crop the part of the image that fired the activation
Results on a validation set of 50,000 images

- Learning a more complex set of patterns than 1st layer edges.
- Covers a much larger space of the image because of the pooling layer before.
- Top-1 strongest activation per feature map in the 2nd layer (256 feature maps.)

[Matthew D. Zeiler and Rob Fergus (2013): Visualizing and Understanding Convolutional Networks]
Results on a validation set of 50,000 images

- Learning a more complex set of patterns than 1st layer edges
- Covers a much larger space of the image probably because of the pooling layer before.
- Features are more invariant to small changes. Ex: A dot, spiral, circle all fire the same 2nd layer feature very strongly

[Matthew D. Zeiler and Rob Fergus (2013): Visualizing and Understanding Convolutional Networks]
Results on a validation set of 50,000 images

- 3rd layer: increased complexity
- An activated neuron is seeing \(\approx 80\times80\) part of a \(256\times256\) image
- Learning objects, faces etc..

[Matthew D. Zeiler and Rob Fergus (2013): Visualizing and Understanding Convolutional Networks]
Results on a validation set of 50,000 images

• 3rd layer: increased complexity

• An activated neuron is seeing $\approx 80 \times 80$ part of a $256 \times 256$ image

• Learning objects, faces etc..

• Patches: Semantic grouping, not structural

[Matthew D. Zeiler and Rob Fergus (2013): Visualizing and Understanding Convolutional Networks]
<table>
<thead>
<tr>
<th>conv1</th>
<th>pl</th>
<th>n1</th>
<th>conv2</th>
<th>p2</th>
<th>n2</th>
<th>conv3</th>
<th>conv4</th>
<th>conv5</th>
<th>p5</th>
<th>fc6</th>
<th>fc7</th>
<th>fc8</th>
<th>prob</th>
</tr>
</thead>
</table>

[Link to video: https://www.youtube.com/watch?v=AgkBO4IgAM]

[Jason Yosinski et al. (2015): Understanding Neural Networks Through Deep Visualization]
I. Interpreting Neural Networks’ outputs
   A. with saliency maps
   B. with occlusion sensitivity
   C. with class activation maps (Global Average Pooling)

II. Visualizing Neural Networks from the inside
   A. with gradient ascent (class model visualization)
   B. with dataset search
   C. The deconvolution and its applications

III. (Optional: Deep Dream: going deeper in NNs)
How to boost the activation of a neuron?

Dreaming process (to repeat):
1. Forward propagate image until dreaming layer
2. Set gradients of dreaming layer to be equal to its activations
3. Backpropagate gradients to input image
4. Update Pixels of the image

[Alexander Mordvintsev et al. (2015): Inceptionism: Going Deeper into Neural Networks]
III. Deep Dream: going deeper in NNs

Horizon → Trees → Leaves

Towers & Pagodas → Buildings → Birds & Insects

[Alexander Mordvintsev et al. (2015): Inceptionism: Going Deeper into Neural Networks]
III. Deep Dream: going deeper in NNs

If you dream in lower layers:

Github repo (deepdream): https://github.com/google/deepdream/

Alexander Mordvintsev et al. (2015): Inceptionism: Going Deeper into Neural Networks

Kian Katanforoosh
III. Deep Dream: going deeper in NNs
III. Deep Dream

Did the neural network learned the right features to detect an object?

Dumbbells

*the network failed to understand the essence of a dumbbell*

[Khan, 2015: Inceptionism: Going Deeper into Neural Networks]

[GitHub repo (deepdream): https://github.com/google/deepdream]
Questions we are now able to answer:

- What part of the input is responsible for the output?
  - Occlusion sensitivity
  - Class Activation Maps

- What is the role of a given neuron/filter/layer?
  - Deconvolutions can help visualize the role of a neuron
  - Search dataset images maximizing the activation
  - Gradient ascent (class model visualization)

- Can we check what the network focuses on given an input image?
  - Occlusion sensitivity
  - Saliency maps (one-time gradient ascent)
  - Class Activation Maps

- How does a neural network see our world?
  - Gradient ascent (class model visualization)
  - Deep Dream

- Do these visualization have use cases?
  - Segmentation (saliency maps)
  - Art (Deep Dream)
Duties for next week

**Sunday 05/24 at 11:59 PM PT:**
- Project Meeting #3 (Meet with your assigned TA between 5/09 and 5/24 to discuss your second milestone report.)
- Project Milestone #2 Due

**Tuesday 05/26 at 9am PT:**
- Complete C5M1

**Others:**
- Friday discussion section