CS230: Lecture 7
Interpretability of Neural Networks
Kian Katanforoosh
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. B. Interpreting and visualizing Neural Networks with saliency maps

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

**Question:** How do you prove quickly that the model is actually looking at the cat?

Can be used for segmentation?

Indicates which pixels need to be changed the least to affect the class score the most.


Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri
Context: You have built an animal classifier for a pet shop. They are a little reluctant to use your model without human supervision, because they don’t understand the decision process of the model.

Question: How do you prove that the model is actually looking at the cat?

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

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri
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:** The pet shop now wants to localize the animals. You have two days. What do you do?

Using a classification network for localization:

Converted to:

Why this?

III. C. Interpreting NNs using class activation maps


Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri
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 pet shop trusts that your model is correctly locating animals. They get a little scared and they ask you to explain what the model thinks a dog is.

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

Keep the weights fixed and use gradient ascent on the input image to maximize this loss:

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

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 \( \frac{\partial L}{\partial x} \)
4. Update x's pixels with gradient ascent


Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri]
II. A. Visualizing NNs from the inside using gradient ascent (class model visualization)

We can do this for all classes:

L2 Regularization

Looks better with additional regularization methods.

[Jason Yosinski et al. (2015): Understanding Neural Networks Through Deep Visualization]
This method can be applied to any activation in the network in order to interpret what a neuron is detecting.

On the class score:

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

change to

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

[Jason Yosinski et al. (2015): Understanding Neural Networks Through Deep Visualization]
II. B. Visualizing NNs from the inside using dataset search

**Context:** The pet shop loved the technique, and asks if there are other alternatives.

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

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri
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)

Convergence volume
(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

100-d random code

\[
\begin{bmatrix}
0.47 \\
\vdots \\
0.19
\end{bmatrix}
\]

Z

Generator “G”
(Neural Network)

\[
\text{(64, 64, 3) generated image}
\]

Discriminator “D”
(Neural Network)

\[
\begin{align*}
\text{Binary classification} \\
& 
\begin{cases}
  y = 0 \quad \text{if} \quad x = G(z) \\
  y = 1 \quad \text{otherwise}
\end{cases}
\end{align*}
\]

Real images
(database)
Motivation behind deconvolution/upsampling layers

II. C. The deconvolution and its applications

Input image (400, 400, 3)

Convolutions
(reduces volume height and width)

Information Encoded

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.

```
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.
```

[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.
**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.

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

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)

---

**Input image**

(256,256,3)
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.

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

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

[Sringenberg & Dosovitskiy et al. (2015): Visualizing and Understanding Convolutional Networks]
We need to pass the filters and switches from the ConvNet to the DeconvNet.

[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.
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

[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 because of the pooling layer before.

- Top-1 strongest activation per feature map in the 2nd layer (256 feature maps.)
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 \( \approx 80 \times 80 \) part of a \( 256 \times 256 \) 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]
[Link to video: https://www.youtube.com/watch?v=AgkBo4JGaM]
[Jason Yosinski et al. (2015): Understanding Neural Networks Through Deep Visualization]
Today’s outline

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)
III. 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]

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri
III. Deep Dream: going deeper in NNs

[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]
III. Deep Dream: going deeper in NNs
Did the neural network learned the right features to detect an object?

Dumbbells

the network failed to understand the essence of a dumbbell

III. Deep Dream

[Alexander Mordvintsev et al. (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)
Announcements

For Wednesday 11/14, 11am:

C5M1
- Quiz: Recurrent Neural Networks
- Programming Assignment:
  - Building a Recurrent Neural Network - Step by Step
  - Dinosaur Land -- Character-level Language Modeling
  - Jazz improvisation with LSTM

This Friday (11/09):
- Project Milestone due. (we expect running code.)
- TA Section