CS230: Lecture 7
Interpretability of Neural Networks

Kian Katanforoosh

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

Questions we will try to answer:

- *What part of the input is responsible for the output?*
- *What is the role of a given neuron/filter/layer?*
- *Can we check what the network focuses on given an input image?*
- *How does a neural network see our world? Does it actually understand it?*

Today's outline

I. Image segmentation

II. The deconvolution

III. Interpreting and visualizing Neural Networks

- A. With deconvolutions
- B. With occlusion sensitivity
- C. With gradient ascent (class model visualization)
- D. With saliency maps
- E. With class activation maps (Global Average Pooling)

IV. Deep Dream: going deeper in NNs

I. Image segmentation

Goal: Segment cells on a picture. In other words, every pixel of the image, indicate if it is part of a cell (1) or not (0).

1. Data?

image

label

2. Input?

3. Output?

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

I. Image segmentation

Architecture?

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)

Loss?

Logistic loss per pixel

$$L^{(i)} = \sum_{pixels} y^{(i)} \log(\hat{y}^{(i)}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})$$

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

I. Image segmentation

If the problem is multi-class

Input image (600, 400, 3)

Convolutions (reduces volume height and width)

Information Encoded

Encoding

De-convolutions (increases volume height and width)

Per-Pixel Classification (600, 400, 1)

Let's understand deconvolutions

Cross entropy per pixel

$$L^{(i)} = \sum_{pixels} y^{(i)} \log(\hat{y}^{(i)})$$

$$\begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad \begin{pmatrix} 0.02 \\ 0.93 \\ 0.04 \\ \vdots \\ 0.07 \\ 0.11 \\ 0.09 \end{pmatrix} \begin{matrix} \text{p(“building”)} \\ \text{p(“grass”)} \\ \text{p(“road”)} \\ \dots \\ \\ \\ \\ \end{matrix}$$

Stanford Drone Dataset: A. Robicquet, A. Sadeghian, A. Alahi, S. Savarese, Learning Social Etiquette: Human Trajectory Prediction In Crowded Scenes in European Conference on Computer Vision (ECCV), 2016

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

II. The deconvolution

Input (4x4)

255	134	202	22
123	94	2	4
11	3	22	192
12	4	23	34

Create sub-pixel image →

Sub-pixel image (9x9)

0	0	0	0	0	0	0	0	0
0	255	0	134	0	202	0	22	0
0	0	0	0	0	0	0	0	0
0	123	0	94	0	2	0	4	0
0	0	0	0	0	0	0	0	0
0	11	0	3	0	22	0	192	0
0	0	0	0	0	0	0	0	0
0	12	0	4	0	23	0	34	0
0	0	0	0	0	0	0	0	0

*

Convolve →

Stride = 1

Output (6x6)

This allows us to upsample an encoding into an image.

[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?]

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

II. The deconvolution

Convolution

A Convolution can be seen as a simple dot product.

$$\begin{pmatrix} w_{11} & w_{12} & w_{13} & 0 & w_{21} & w_{22} & w_{23} & 0 & w_{31} & w_{32} & w_{33} & 0 & 0 & 0 & 0 & 0 \\ 0 & w_{11} & w_{12} & w_{13} & 0 & w_{21} & w_{22} & w_{23} & 0 & w_{31} & w_{32} & w_{33} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_{11} & w_{12} & w_{13} & 0 & w_{21} & w_{22} & w_{23} & 0 & w_{31} & w_{32} & w_{33} & 0 \\ 0 & 0 & 0 & 0 & 0 & w_{11} & w_{12} & w_{13} & 0 & w_{21} & w_{22} & w_{23} & 0 & w_{31} & w_{32} & w_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \\ x_{16} \end{pmatrix}$$

"Deconvolution"

Matrix "faking" the 3x3 window convolved around the 4x4 input.

Flattened 4x4 input

http://deeplearning.net/software/theano_versions/dev/tutorial/conv_arithmetic.html#no-zero-padding-unit-strides-transposed

[Vincent Dumoulin and Francesco Visin : A guide to convolution arithmetic for deep learning]

II. The deconvolution

100-d
random code

$$\begin{pmatrix} 0.47 \\ \vdots \\ 0.19 \end{pmatrix}$$

z

Generator "G"
DECONV
(Neural Network)

(64,64,3)
generated image

x

Discriminator "D"
(Neural Network)

Binary classification

$$\begin{cases} y = 0 \quad if \quad x = G(z) \\ y = 1 \quad otherwise \end{cases}$$

Real images
(database)

These deconvolutions will help us better understand and interpret neural networks.

[Ian Goodfellow (2014): NIPS Tutorial: GANs]

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

Today's outline

I. Image segmentation

II. The deconvolution

III. Interpreting and visualizing Neural Networks

- A. With deconvolutions
- B. With occlusion sensitivity
- C. With gradient ascent (class model visualization)
- D. With saliency maps
- E. With class activation maps (Global Average Pooling)

IV. Deep Dream: going deeper in NNs

Questions we will try to answer:

- *What part of the input is responsible for the output?*
- *What is the role of a given neuron/filter/layer?*
- *Can we check what the network focuses on given an input image?*
- *How does a neural network see our world?*

III. Interpretability of Neural Networks

First problem: Fully-connected layers' neurons lack spatial information

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

III. Interpretability of Neural Networks

III. Interpretability of Neural Networks

III. Interpretability of Neural Networks

III. Interpretability of Neural Networks

III. Interpretability of Neural Networks

Consider a CNN trained on ImageNet (classification), we look at the output of the 5th CONV layer

Input image
(64,64,3)

CONV
(5 layers)

Encoding volume
(13,13,256)

Conclusion: the deeper the activation, the more it "sees" from the image

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

Questions we will try to answer:

- *What part of the input is responsible for the output?*
- *What is the role of a given neuron/filter/layer?*
- *Can we check what the network focuses on given an input image?*
- *How does a neural network see our world?*

III. Interpretability of Neural Networks

Here is our CNN, trained on ImageNet (1.3m images, 1000 classes), we're trying to interpret it.

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

Top 5 images

It seems that the filter has learned to detect edges

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

Questions we will try to answer:

- *What part of the input is responsible for the output?*

- *What is the role of a given neuron/filter/layer?*
    - *Search dataset images maximizing the activation*

- *Can we check what the network focuses on given an input image?*

- *How does a neural network see our world?*

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

III. A. Interpreting NNs using deconvolutions

Here is our CNN, trained on ImageNet (1.3m images, 1000 classes), we're trying to interpret it.

[Matthew D. Zeiler and Rob Fergus (2013): Visualizing and Understanding Convolutional Networks]

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

III. A. Interpreting NNs using deconvolutions

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

III. A. Interpreting NNs using deconvolutions

Here is our CNN, trained on ImageNet (1.3m images, 1000 classes)

[Matthew D. Zeiler and Rob Fergus (2013): Visualizing and Understanding Convolutional Networks]

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

III. A. Interpreting NNs using deconvolutions

Other CONV layers can be visualized the same way

[Matthew D. Zeiler and Rob Fergus (2013): Visualizing and Understanding Convolutional Networks]

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

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]

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

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)

Layer 2 reconstructions

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

[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 ≈80x80 part of a 256x256 image
- Learning objects, faces etc..

[Matthew D. Zeiler and Rob Fergus (2013): Visualizing and Understanding Convolutional Networks]

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

Results on a validation set of 50,000 images

- 3rd layer: increased complexity
- An activated neuron is seeing ≈80x80 part of a 256x256 image
- Learning objects, faces etc..
- Patches: Semantic grouping, not structural

Faces

Clouds

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

[Matthew D. Zeiler and Rob Fergus (2013): Visualizing and Understanding Convolutional Networks]

III. C. Interpreting NNs using gradient ascent (class model visualization)

Here is our CNN, trained on ImageNet (1.3m images, 1000 classes), we're trying to interpret it.

Top 5 images

The filter might be detecting wrinkles, not shirts!

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

Questions we will try to answer:

- *What part of the input is responsible for the output?*

- *What is the role of a given neuron/filter/layer?*
    - *Search dataset images maximizing the activation*
    - *Deconvolutions can help visualize the role of a neuron*
- *Can we check what the network focuses on given an input image?*

- *How does a neural network see our world?*

III. B. Interpreting NNs using occlusion sensitivity

In classification, is the model identifying the location of an object in the image, or just the surrounding context?

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

[Matthew D. Zeiler and Rob Fergus (2013): Visualizing and Understanding Convolutional Networks]

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

Questions we will try to answer:

- *What part of the input is responsible for the output?*
    - *Occlusion sensitivity*

- *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*

- *Can we check what the network focuses on given an input image?*
    - *Occlusion sensitivity*

- *How does a neural network see our world?*

III. C. Interpreting NNs using gradient ascent (class model visualization)

Given this trained ConvNet and a class of interest (dog),
generate an image which is representative of the class according to the ConvNet

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

$$L = S_{dog}(x) - \lambda \|x\|_2^2$$

"x should look natural"

Gradient ascent:

$$x = x + \alpha \frac{\partial L}{\partial x}$$

<u>Repeat this process:</u>

1. Forward propagate image x
2. Compute the objective L
3. Backpropagate to get dL/dx
4. Update x's pixels with gradient ascent

[Karen Simonyan et al. (2014): Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps]

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

III. C. Interpreting NNs using gradient ascent (class model visualization)

We can do this for all classes:

Regularizer:
L2

Looks better with additional regularization methods.

[Jason Yosinski et al. (2015): Understanding Neural Networks Through Deep Visualization]

[Karen Simonyan et al. (2014): Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps]

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

III. C. Interpreting NNs using gradient ascent (class model 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:

$$L = S_{dog}(x) - R(x)$$

change to $\longrightarrow$

On any activation:

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

[Jason Yosinski et al. (2015): Understanding Neural Networks Through Deep Visualization]

[Link to video: https://www.youtube.com/watch?v=AgkfIQ4IGaM]

[Jason Yosinski et al. (2015): Understanding Neural Networks Through Deep Visualization]

Questions we will try to answer:

- *What part of the input is responsible for the output?*
    - *Occlusion sensitivity*

- *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*

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

III. D. Interpreting NNs using saliency maps

Given:

Rank the pixels of x based on their influence on the score S of the class.

$$\frac{\partial S_{dog}(x)}{\partial x}$$ indicates which pixels need to be changed the least to affect the class score the most.

CNN

$$S_{dog}(x)$$

backpropagation

$$\frac{\partial S_{dog}(x)}{\partial x}$$

Can be used for segmentation?

Yes

[Karen Simonyan et al. (2014): Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps]

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

Questions we will try to answer:

- *What part of the input is responsible for the output?*
    - *Occlusion sensitivity*

- *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)*

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

- *Do these visualization have use cases?*
    - *Segmentation (saliency maps)*

III. E. Interpreting NNs using class activation maps

Using a classification network for localization

Converted to:

[Bolei Zhou et al. (2016): Learning Deep Features for Discriminative Localization]

III. E. Interpreting NNs using class activation maps

[Bolei Zhou et al. (2016): Learning Deep Features for Discriminative Localization]

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

I. B. Deconvolutional layer

[Bolei Zhou et al. (2016): Learning Deep Features for Discriminative Localization]
Source video: Kyle McDonald

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

Questions we will try 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)*

- *Do these visualization have use cases?*
    - *Segmentation (saliency maps)*

III. Deep Dream

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

"Admiral Dog!" "The Pig-Snail" "The Camel-Bird" "The Dog-Fish"

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

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

III. Deep Dream

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

III. Deep Dream

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

[Link to video: https://www.youtube.com/watch?v=DgPaCWJL7XI]
[Github repo (deepdream): https://github.com/google/deepdream/]
[Alexander Mordvintsev et al. (2015): Inceptionism: Going Deeper into Neural Networks]

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

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

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

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

Questions we will try 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

Check out the project example code (cs230-stanford.github.io)

For Tuesday 05/15, 9am:

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**

For this Friday (05/18):

- **Project Milestone due. (we expect running code.)**
- **TA Section this Friday**

Question for me:

- **What visualization methods to use and why?**

My answer: CAMs and Saliency mast are the quickest and most common, DeconvNets is hard to implement, and gradient ascent is used for adversarial examples and art mostly.

- **What are the next steps in interpreting a neural network?**

- Class activation maps vs. Saliency maps?

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

I. B. Deconvolutional layer

Input (4x4)

255	134	202	22
123	94	2	4
11	3	22	192
12	4	23	34

Create sub-pixel image →

Sub-pixel image (9x9)

0	0	0	0	0	0	0	0	0
0	255	0	134	0	202	0	22	0
0	0	0	0	0	0	0	0	0
0	123	0	94	0	2	0	4	0
0	0	0	0	0	0	0	0	0
0	11	0	3	0	22	0	192	0
0	0	0	0	0	0	0	0	0
0	12	0	4	0	23	0	34	0
0	0	0	0	0	0	0	0	0

*

Convolve →

Stride = 1

Output (6x6)

[Wenzhe Shi, et al. : Is the deconvolution layer the same as a convolutional layer?]

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

http://cnnlocalization.csail.mit.edu/

VIDEO to motivate

We have seen convNets, how do you think we can do this visualization in practice?

Start with AlexNet, NetinNet, GoogLeNet.

Add GAP + SOFTMAX FC

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

https://arxiv.org/pdf/1312.6034.pdf

Saliency maps

It should be noted that we used the (unnormalised) class scores Sc, rather than the class posteriors, returned by the soft-max layer: Pc = Pexp Sc c exp Sc . The reason is that the maximisation of the class posterior can be achieved by minimising the scores of other classes. Therefore, we optimise Sc to ensure that the optimisation concentrates only on the class in question c. We also experimented with optimising the posterior Pc, but the results were not visually prominent, thus confirming our intuition.

You can now convert a classification algorithm in a localization also using saliency maps (weak supervision).

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

https://arxiv.org/pdf/1311.2901.pdf

Despite this encouraging progress, there is still little insight into the internal operation and behavior of these complex models, or how they achieve such good performance. From a scientific standpoint, this is deeply unsatisfactory. Without clear understanding of how and why they work, the development of better models is reduced to trial-and-error

reveals the input stimuli that excite individual feature maps at any layer in the model. It also allows us to observe the evolution of features during training and to diagnose potential problems with the model. to project the feature activations back to the input pixel space.

Our approach, by contrast, provides a non-parametric view of invariance, showing which patterns from the training set activate the feature map.

To examine a convnet, a deconvnet is attached to each of its layers, as illustrated in Fig. 1(top), providing a continuous path back to image pixels

Motivation: Earlier in the course we talk about the fact that early layers detect edges while later layers detect more complex patterns. We did not talk about how to visualize these.

The number of activations is often not equal to the number of pixels in the input image. How can we project a small volume back to the input image?

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri

Deep Dream

We can dream at any layer in the convnet

We forwarded the image in the network up to a certain layer,
Now we set the gradients of this to be equal to its activations, then we do backward from that point back to the image.
Then update the image pixels.

You always dream after Relu units so these activations are zero cropped.
Your are amplifying the features that maximally activated the network.

We will find the image update that will boost the strong activations of the network

Kian Katanforoosh, Andrew Ng, Younes Bensouda Mourri