Visualising convolutional networks

Hakan Bilen

Machine Learning Practical - MLP Lecture 12
16 January 2019

http://www.inf.ed.ac.uk/teaching/courses/mlp/
Lectures in second semester

- Understanding convolutional networks
- Generative adversarial networks
- Domain adaptation and transfer learning
- Convolutional network design and compression (Dr Elliot Crowley)
- Object detection and semantic segmentation
- Language and vision models
- Video analytics
Recap: Convolutional Neural Networks (CNNs)

What is inside the black box (filters and feature maps)?
Why does it matter?

- Interpretability: understand what they learn and why they work
- Monitor training process (evolution of training)
- Gain intuitions to develop better models
- Diagnose potential problems
Today

1. Visualize filters / weights
2. Analyze activations
3. Deconvolutional networks
4. Saliency deconvolutional networks
5. Adversarial noise
1. Visualize filters

**conv1:**
96 filters 11x11x3
1. Visualize filters

**conv1:**
96 filters 11x11x3

**Question:**
What do these filters detect?
1. Visualize filters

- conv1: 96 filters 11x11x3
- Oriented edge filters (similar to Gabor filters)
- Coloured blob detectors
Monitoring filters during training

Good training: learned filters should exhibit structure and are uncorrelated

Slide credits: Ranzatto & Lecun
2. Analyze activations

1. Pick a neuron at a layer
2. Record it for multiple images
3. Show the images with highest activation value
4. See whether the images correspond to a common concept

Girshick et al. Rich feature hierarchies for accurate object detection and semantic segmentation, CVPR’14
2. Analyze activations

Easy to implement
😞 Only qualitative analysis

Girshick et al. Rich feature hierarchies for accurate object detection and semantic segmentation, CVPR’14
2. Analyze activations quantitatively

Q. How can we quantify alignment with visual concepts?

I. Collect images and label all the pixels with various concepts
   - objects, parts, scenes, textures, colours and materials

Bau et al., Network Dissection: Quantifying Interpretability of Deep Visual Representations, CVPR’17
2. Analyze activations quantitatively

II. Gather responses of neurons to known concepts

- Input image $x$ to CNN
- Take an activation map $A_k(x)$ at layer $l$
- Threshold $P(A_k(x) > T)$
- Upscale to image size

Bau et al., Network Dissection: Quantifying Interpretability of Deep Visual Representations, CVPR’17
2. Analyze activations quantitatively

III. Measure overlap with human labelled concepts

Forward pass

Intersection over union

$\text{IoU} = \frac{A \cap B}{A \cup B}$

Bau et al., Network Dissection: Quantifying Interpretability of Deep Visual Representations, CVPR’17
conv5 unit 79  car (object)  IoU=0.13

conv5 unit 107  road (object)  IoU=0.15

Bau et al., Network Dissection: Quantifying Interpretability of Deep Visual Representations, CVPR’17
More complex concepts emerge at the later layers

Some low level concepts at the later layer are still useful for classification
3. Deconvolutional networks

So far, finding correlations between a set of images and activations

What input pattern originally caused a given activation in the feature maps?

Zeiler & Fergus, Visualizing and Understanding Convolutional Networks, ECCV'14
How to project the activations back to the input pixel space?

Zeiler & Fergus, Visualizing and Understanding Convolutional Networks, ECCV'14
Deconvnet aims to project the activations back to the input pixel space.
- Invert convnet by
  - Unpooling
  - (Un)rectification
  - Convolution transpose

Zeiler & Fergus, Visualizing and Understanding Convolutional Networks, ECCV'14
Question

Is max pool operation invertible?

\[ y = \text{maxpool}(x) \]

\[ x? = (\text{maxpool})^{-1}(y) \]
Unpooling

Layer Above Reconstruction

Unpooling

Max Locations “Switches”

Unpooled Maps

Rectified Feature Maps

Pooled Maps

Pooling

Zeiler & Fergus, Visualizing and Understanding Convolutional Networks, ECCV’14
Unpooling

Relation to backprop (see lecture 8)

\[
\begin{array}{c|c|c}
\frac{\partial E}{\partial H^l} & \frac{\partial H^l}{\partial H^{l-1}} & \frac{\partial E}{\partial H^{l-1}} \\
\hline
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
\end{array}
\times
\begin{array}{c|c|c}
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
\end{array}
= 
\begin{array}{c|c|c}
\end{array}
\]

- \(E\) is loss function
- \[\frac{\partial E}{\partial H_{l-1}} = \frac{\partial E}{\partial H_l} \cdot \frac{\partial H_l}{\partial H_{l-1}}\]
- Unpooling corresponds to backprop of maxpooling
Unrectification (UnReLU)

\[ \text{ReLU} \]

\[ H_{l-1} \rightarrow \text{ReLU} \rightarrow H_l \]

\[ H_l = \max(H_{l-1}, 0) \]

\[ \text{UnReLU} \]

\[ R_{l-1} \rightarrow \text{UnReLU} \rightarrow R_l \]

\[ R_{l-1} = \max(R_l, 0) \]

Relation to backpropagation

\[ \frac{\partial E}{\partial H_{l-1}} = \frac{\partial E}{\partial H_l} \cdot 1(H_l > 0) \]

UnReLU does not utilise \[ R_l \cdot 1(R_l > 0) \] but \[ \max(R_l, 0) \]
Transpose convolution (deconvolution?)

Convolution
\[ H_l = \text{conv}(H_{l-1}, W_l) \]

Transpose convolution
\[ R_{l-1} = \text{conv}(R_l, W_l^T) \]

- It is not inverse convolution!
- Usually \( \text{conv}(H_{l-1}, W_l) \neq \text{conv}(H_{l-1}, W_l^T) \)

Relation to backprop (see lecture 8)
Layer 1-2: Top-9 Patches

Top 9 activations are projected down to pixel space using deconvolutional net

Patches from validation images that give maximal activation of a given feature map

Zeiler & Fergus, Visualizing and Understanding Convolutional Networks, ECCV‘14
Layer 3: Top-9 Patches

Top 9 activations are projected down to pixel space using deconvolutional net

Patches from validation images that give maximal activation of a given feature map

Zeiler & Fergus, Visualizing and Understanding Convolutional Networks, ECCV’14
Layer 4-5: Top-9 Patches

Zeiler & Fergus, Visualizing and Understanding Convolutional Networks, ECCV'14
4. Image specific saliency

\[ S_c(I) \]

Which pixels matter most for the prediction?

Simonyan, Vedaldi and Zisserman, Deep Inside Convolutional Networks, NIPS’14
4. Image specific saliency

Question
Can we calculate influence of each pixel on the class probability?

Simonyan, Vedaldi and Zisserman, Deep Inside Convolutional Networks, NIPS’14
4. Image specific saliency

\[
\delta S_c(I) \quad \frac{\delta S_c}{\delta I}
\]

Simonyan, Vedaldi and Zisserman, Deep Inside Convolutional Networks, NIPS'14
4. Image specific saliency

Simonyan, Vedaldi and Zisserman, Deep Inside Convolutional Networks, NIPS’14
Deconv net vs Saliency net

\[ R_{l-1} = \max(R_l, 0) \]

\[ R_{l-1} = M \cdot R_l \]

\[ M = 1(R_l > 0) \]
4. Generic class saliency maps

Can we generate an image that outputs high score for dog?

Simonyan, Vedaldi and Zisserman, Deep Inside Convolutional Networks, NIPS'14
4. Generic class saliency maps

\[ S_c(I) \]

\[ \delta S_c / \delta I \]

- \( \text{argmax}_I S_c(I) - \lambda \| I \|_2^2 \)
- Maximize “dogness” by modifying pixel values

Simonyan, Vedaldi and Zisserman, Deep Inside Convolutional Networks, NIPS’14
4. Generic class saliency maps

Simonyan, Vedaldi and Zisserman, Deep Inside Convolutional Networks, NIPS’14
4. Image and generic class saliency (Deep dream)

Mordvintsev et al, Inceptionism: Going Deeper into Neural Networks
4. Image and generic class saliency maps

\[ S_c(I) \]

\[ \delta S_c / \delta I \]

\[ argmax_I S_c(I) - \lambda \| I \|_2^2 \]

What happens if we add saliency for another class?

Simonyan, Vedaldi and Zisserman, Deep Inside Convolutional Networks, NIPS'14
Adversarial Examples

Problem common to any discriminative method!

Szegedy et al., Intriguing properties of neural networks, ICLR’14
Summary

Visualize CNNs
- Filters
- Highest activations
- Deconv network
- Saliency network
- Generating adversarial samples
Reading material

Recommended
- Zeiler & Fergus, Visualizing and Understanding Convolutional Networks, ECCV’14

Extra
- Simonyan, Vedaldi, Zisserman, Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, ICLR’14
- Szegedy et al. Intriguing properties of neural networks, ICLR’14
- Nice summary of adversarial techniques by Karpathy
- Try to generate adversarial examples or interesting pictures!