AIMS Big Data
Lectures 4: Deep learning 2
Andrea Vedaldi

For slides and up-to-date information:
http://www.robots.ox.ac.uk/~vedaldi/teach.html

Image representations

An encoder maps the data into a vectorial representation
Facilitate labelling of images, text, sound, videos, …

Modern convolutional nets

Excellent performance in most image understanding tasks
Learn a sequence of general-purpose representations

Understanding visual representations

Visualizing representations
[Mahendran Vedaldi CVPR 2015]

Representations & transformations
[Lenc Vedaldi CVPR 2015]
The representation is not injective

The reconstruction ambiguity provides useful information about the representation

Finding a Pre-Image

A simple yet general and effective method

\[ \min_x \| \Phi(x) - \Phi_0 \|_2^2 \]

Start from random noise
Optimize using stochastic gradient descent

No prior TV-norm \( \beta = 1 \) TV-norm \( \beta = 2 \)
Related Visualizations

Inside CNNs
[Simonyan et al. In ICLR, 2014]
- Do not use a natural image prior.
- Activation maximization, last layer.

DeConvNets
[Zeiler et al. In ECCV, 2014]
- Uses intermediate information at pooling layers.
- Specialized to convnets.

Google Inceptionism
[Blogpost 2015]
- Caricaturization (see later).
- Regularization through jitter.

Deep visualizations
[Yosinksi et al. in ICMLW, 2015]
- Activation maximization.
- Other regularization techniques.

Inverting a Deep CNN

Original Image

AlexNet
[Krizhevsky et al. 2012]
Inverting a Deep CNN

Original Image

Convolutional layers:
- Conv 1
- Conv 2
- Conv 3
- Conv 4
- Conv 5

Fully connected layers:
- FC 6
- FC 7
- FC 8

These layers process the original image through multiple stages of feature extraction and classification to recognize patterns and objects.
Inverting a Deep CNN

Original Image

conv 1 conv 2 conv 3 conv 4 conv 5 fc 6 fc 7 fc 8

Original Image

25

Inverting a Deep CNN

Original Image

conv 1 conv 2 conv 3 conv 4 conv 5 fc 6 fc 7 fc 8

Original Image

26

Inverting a Deep CNN

Original Image

conv 1 conv 2 conv 3 conv 4 conv 5 fc 6 fc 7 fc 8

Original Image

27

Inverting a Deep CNN

Original Image

conv 1 conv 2 conv 3 conv 4 conv 5 fc 6 fc 7 fc 8

Original Image

28
Inverting a Deep CNN

Original Image

Conv 1 Conv 2 Conv 3 Conv 4 Conv 5 FC 6 FC 7 FC 8

CNNs = visual codes?

conv1 conv5 fc8

Inverting a Deep CNN

Original Image

Conv 1 Conv 2 Conv 3 Conv 4 Conv 5 FC 6 FC 7 FC 8

ASIL NADIR RETURNS

Inverting a Deep CNN

Original Image
Look for an image that maximally activates a specific feature component

$$\min_x -\langle e_k, \Phi(x) \rangle + R_T(x) + R_V(x)$$

Activation Maximization

Related Work

**Visualizing higher-layer features of a deep network**
Ethan et al. 2009
[intermediate features]

**Deep inside convolutional networks**
Simonyan et al. 2014
[deepest features, aka “deep dreams”]

**Google “inceptionism”**
Mordvintsev et al. 2015

**Understanding neural networks through deep visualisation**
Yosinksi et al. 2015
[intermediate features]
Our neural network architecture has 60 million parameters. Although the 1000 classes of ILSVRC turn out to be insufficient to learn so many parameters without considerable overfitting. Below, we make each training example impose 10 bits of constraint on the mapping from image to label, this forces us to use much smaller networks. At test time, the network makes a prediction by extracting...
Network comparison

“conv5” features

AlexNet
VGG-M
VGG-VD

Remember: the starting point is white noise

Not an image!
Caricaturization

[Google Inceptionism 2015, Mahendran et al. 2015]

Emphasise patterns that are detected by a certain representation

\[
\min_{x} -\langle \Phi(x_0), \Phi(x) \rangle + R_{TV}(x) + R_{\alpha}(x)
\]

Key differences:

- the starting point is the image \( x_0 \)
- particular configurations of features are emphasized, not individual features
Understanding visual representations

Visualizing representations
[Mahendran Vedaldi CVPR 2015]

Representations & transformations
[Lenc Vedaldi CVPR 2015]

Meaningful representations

Semantic similarity
Vector similarity (distance)

embedding space $\mathbb{R}^d$

congruous pair

incongruous pair

Desiderata:
invariance and distinctiveness

Invariance is task dependent

Are $x$ and $y$ the same category?

$\Phi(x)$

$\Phi(y)$

near

Are $x$ and $y$ the same colour?

$\Phi(x)$

$\Phi(y)$

far

Equivariance

image space

feature space

$g$ is an image transformation

$g$ is the corresponding feature transformation

$\Phi(gx) = M_g \Phi(x)$

Invariance

$x \xrightarrow{g} gx$

$x' \xrightarrow{g} gx'$

$g$ is an image transformation

$\Phi(x) = \Phi(gx)$

$\Phi(x') = \Phi(gx')$

$M_g$ is the identity

the map $M_g$ is the identity

Representations and transformations

Equivariance

There is a (simple) $M_g$

Invariance

$M_g$ is the identity

Neither

There is no (simple) $M_g$

$\Phi(gx) = M_g \Phi(x)$

No equivariance

$\Phi(gx)$

$\Phi(gx')$

$\Phi(x)$

$\Phi(x')$

$g$ is an image transformation

the map $M_g$ does not exists (or is very complex)

What we do know

In some cases (e.g. HOG and image flipping) the transformation is a simple permutation
But what happens with more complex transformations like affine ones?

What happens with more complex representations like CNNs?

Finding equivariance empirically

Regularized linear regression

(learned empirically)
HOG features

Finding equivariance empirically

rotation 45 deg

\[ M_g \Phi \approx M_g \]

\[ \Phi \]

CNN: a sequence of representations

We run the same analysis on a typical CNN architecture

▶ AlexNet [Krizhevsky et al. 12]
▶ 5 convolutional layers + fully-connected layers
▶ Trained on ImageNet ILSVRC

A discriminative goal to learn equivariance

\[ M_g \] is learned empirically

All the other layers (representation and classifier) are frozen
When are two representations the same?

Learning representations means that there is an endless number of them
Variants obtained by learning on different datasets, or different local optima

Equivalence

\[ \Phi_B(x) = E \Phi_A(x) \]

AlexNet, same training data, different parametrization:

Are \( \Phi_A \) and \( \Phi_B \) equivalent?
Equivalence with different random seeds

Baseline

Before training

After training

Equivalence of similar architecture

Train on two different datasets

Places dataset

ILSVRC12 dataset

CNN-IMNET

CNN-PLACES

Equivalence with different training data