AIMS Big Data Course
Universal, unsupervised and understandable representations

Dr Andrea Vedaldi
Lecture 2
Favignana, August 2018

How can AI impact people?

Horizontal problems

- Autonomous driving
- Internet search
- Social networks
- Medical imaging (?)

Vertical problems

- Measuring plants
- Matching manuscripts
- Matching galaxies
- Counting penguins
- Recognizing flowers
- Measuring condensation
- Tracking crystals
- Measuring artefacts
- Searching Greek vases
- Comparing 19th century paintings
- Matching Mesopotamian clay rolls
- ...
Vertical AI problems

Biology, zoology, areas of medical imaging, material science, etc.

Counting crystals in thousands of videos

The need for (too) big data

One of the most significant bottlenecks of deep learning

ImageNet ILSVRC
1.2M images

ADE 20K
20K images
Dense annotations

Visual Genome
100K images
4M annotations

Open Images
9M images
28M annotations

Will he label 1M images?

Three key challenges of deep learning

Universal Representations
Fewer models to train

Unsupervised Representations
Less effort to train new models

Understandable Representations
Trust, safety, and usability
Three key challenges of deep learning

- Universal Representations
  - Fewer models to train

- Unsupervised Representations
  - Less effort to train new models

- Understandable Representations
  - Trust, safety, and usability

Narrow representations

- Object recognition
- Medical diagnosis
- Text spotting

Universal representation

- Learning multiple visual domains with residual adapters, Rebuffi, Bilen, Vedaldi, NIPS 2017

Universal representation family
Universal representation families

- Object recognition
- Medical diagnosis
- Text spotting

**Preview**
- > 90% of parameters are **shared**
- Same or *better performance* than narrow models
- *No forgetting*

**Applications**
- Better than standard pre-training, especially for small dataset
- Efficient model storage, transmission, updating


Residual adapters

**An efficient parametrization**

- **Adapters**
  - Tweak a fixed neural network block
  - Interleaved with standard convolutions
  - 1 x 1 filter bank
  - Only ~10% of the parameters

- **Residual Adapters**
  - Shrinkage controls the amount of adaptation
  - Better generalization when adapting to small target datasets


Visual Domain Decathlon Challenge

**Goal**: learn a single model that performs as well as possible on 10 very different visual domains

- Aircrafts
- CIFAR-100
- Daimler Pedestrians
- Describable Textures
- German Street Signs
- ImageNet
- VGG Flowers
- Omniglot
- SVHN
- UCF101
- Dyn. Images

Efficient parametrization of multi-domain deep neural networks, Rebuffi, Bilen, Vedaldi, CVPR 2018
Complementary problems

UberNet: more tasks
Universal family: more domains

Detection, segmentation, boundaries, normals, parts, …
Aircraft, people, flowers, text, glyphs, texture, Internet, …

UberNet: more tasks
Universal family: more domains

Detection, segmentation, boundaries, normals, parts, …
Aircraft, people, flowers, text, glyphs, texture, Internet, …


Three key challenges of deep learning

Universal Representations
Fewer models to train

Unsupervised Representations
Less effort to train new models

Understandable Representations
Trust, safety, and usability

Self-supervised learning
Can we drop annotations?
Images are cheap and abundant
However, manual annotations are extremely expensive

Self-supervised features

Self-supervised structure

Deep image prior

Self-supervised features

Self-supervised structure

Deep image prior

Self-supervised features: transfer learning

Pretext task

Cheap supervised data

Encoder $\Phi$

Decoder $\Psi$

Loss

Target task

ImageNet

Encoder $\Phi$

Decoder $\Psi$

Loss

Find a pretext task to pre-train a model $\Phi$

The pretext tasks comes with cheap supervision

Fine-tune the model for a target task

Far less annotations are now required

Where to get supervision from

Self-supervised features

Perturbations

Motion

Modalities

Cheap supervised data

Target task

Pretext task

Encoder $\Phi$

Decoder $\Psi$

Loss

Fine-tune the model for a target task

Far less annotations are now required
Goal: learn a model $\Phi$ that reproduces a human annotator

Standard supervision

Nope: the “auto annotator” is just as complex as the model $\Phi$

Automatic annotator?

Self-supervision from perturbations

Use the image as label, their perturbations as input

Example perturbation = delete half of the image
Concrete learning scheme

Intuition: completing an image may **require the network to learn about objects**

Perturbations useful for self-supervision

Disadvantage: the network learns to operate on perturbed data

Look for image pairs that are correlated form the outset
Self-supervision using video sequence

Images related through time


Self-supervision using multiple modalities

Data sharing the same root cause


Examples:

- Reconstruction of sound from images:
  - Task: find out which picture is upright
  - The supervisory signal is in the photographer bias: people take pictures with a specific orientation

- Accidental supervision
How to setup the learning problem

Self-supervised features

Perturbations

Motion

Modalities

Probability

Constraining

Marginalising

Correlating

Addressing ambiguous pretext tasks

Many pretext tasks are ill-posed

colorization

future frame prediction

image to sound


Approach 1: use a probability distribution

Explicitly model ambiguity by predicting a probability distribution

Disadvantage: probabilistic modeling can be challenging or too simplistic
Approach 2: constrain the prediction task

Addressing ambiguous pretext tasks

E.g.: reorder frames causally instead of generating them
Reordering is much easier than generation

Approach 3: reduce the amount of predicted information

Predict a “character” of the full data
This is captured by a “marginal loss”
- Via projection
  \[ \| \hat{y} - \pi(y) \| \]
- Via marginalization
  \[ \min \| y - f(\hat{y}, \eta) \| \]

Disadvantage: ad-hoc design

Example: only predict what could move in an image
**Common fate principle**
Group together what moves together

**Pairwise flow similarities**

- Optical flow
- Flow similarity (kernel)

**Pairwise pixel similarities**

- Encoder
- Decoder
- Loss

**Approach 4: learning correspondences**
Addressing ambiguous pretext tasks

Randomly extract $x$ and $y$ from either the same or different data
Learn to tell which one is the case
Reduces to predicting a binary switch

**Audio-visual correspondences**

- Visual network $\phi_1$
- Audio network $\phi_2$
- Switch
- predictor

---

**Matching modalities**

**Addressing ambiguous pretext tasks**

Randomly extract $x$ and $y$ from either the same or different data
Learn to tell which one is the case
Reduces to predicting a binary switch
Co-occurrences may be better captured by mutual information.

Learning objective:

\[
\max_{\Phi_1, \Phi_2} I(\Phi_1(f_1(z)), \Phi_2(f_2(z)))
\]

Image clustering and segmentation

Learn maximally mutually informative classes

Take image pairs related by proximity and/or geometric transformation.

Classes are likely to be the same, or at least correlated.

Learn a classification function to maximize the mutual information between classes.

Clustering results

ImageNet fruits clustering

Satellite image segmentation

Self-supervised features

Self-supervised structure

Deep image prior
Can we learn the structure of visual objects explicitly?

From features to structure

Implicit features

encoder

VS

Explicit concept

Object model

Take I: A 3D surface

$S \subset \mathbb{R}^3$

Object model

Take II: an equivalence class of deformable surfaces

$x, x' \in \mathbb{R}^3$

homeomorphism

induces correspondences through viewpoint changes, deformations

Can put in correspondence different instances, categories

$x, x' \in \mathbb{R}^3$

$x, x' \in \mathbb{R}^3$

$x, x' \in \mathbb{R}^3$
Homeomorphic to a sphere

$S \subset \mathbb{R}^3 \quad \mathbb{Z} = S^2 \quad S' \subset \mathbb{R}^3$

Object model

Take III: Sphere

$\mathbb{Z} = S^2$

Object model

Mathematical dogs

The Hairy-Ball Theorem

These are a few of the concepts and objects studied by topology: now we’ll look at a theorem.

If you look at the way the hairs lie on a dog, you will find that they have a “parting” along the dog’s back, and another along the stomach. Now topologically a dog is a sphere (assuming it keeps its mouth shut and neglecting internal organs) because all we have to do is shrink its legs and flatten it up a bit (Figure 60).

Figure 60

One might wonder whether it is possible to comb the hairs in such a way that all partings were eliminated. This would give a smooth

Labelling function $\Phi$

$\Phi$ maps pixels $u$ of image $x$ to object frame coordinates $z$

$\Phi(x; u)$

$z$

$u$

$x$
How can we learn this function without manual supervision?

Geometric consistency of the labelling function

Invariance

∀u : \( \Phi(x; u) = \Phi(gx; gu) \)

Degeneracy: invariance can be trivially satisfied

∀u, u': \( \Phi(x; u) = \Phi(gx; u') \iff u' = gu \)

Addressing the degeneracy

Invariance & distinctiveness

∀u, u': \( \Phi(x; u) = \Phi(gx; u') \implies u' = gu \)
A probabilistic formulation

Induces invariance and distinctiveness

Map pixels to 3D vectors $\Phi(x;u) \in \mathbb{R}^3$

- Vector length codes certainty

Probabilistic vector matching

- Conditional heat map:
  \[ S(u' \mid u) = \langle \Phi(x;u), \Phi(gx;u') \rangle \]

- Conditional correspondence probability:
  \[ p(u' \mid u) = \frac{e^{S(u' \mid u)}}{\int e^{S(v' \mid u)} dv'} \]

**Loss** encourages both precision and accuracy:

\[ \mathcal{L} = \int \| u' - gu \|^2 p(u' \mid u) \, du \, du' \]

Unsupervised learning of a dense equivariant labeller

Data: $(x_i, g)$ where $x_i$ is a random image and $g$ the estimated flow or a random warp

Learning an object from motion

Training data
- Pairs of video frames $x_t$ and $x_{t+\Delta}$
- $g$ is the optical flow

Learned embedding
- Map object points to a fixed reference coordinates
- The absolute reference is learned automatically

Examples & results
### 3D deformable object

<table>
<thead>
<tr>
<th>Input Frames</th>
<th>Labelled Pixels</th>
<th>Projected to sphere</th>
<th>Vector Magnitude</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Input Frames" /></td>
<td><img src="image2.png" alt="Labelled Pixels" /></td>
<td><img src="image3.png" alt="Projected to sphere" /></td>
<td><img src="image4.png" alt="Vector Magnitude" /></td>
</tr>
</tbody>
</table>

### Faces

**Training data**

Pairs of warped images $x$ and $x'$ from a dataset of ~200K celebrity images

$g$ are a synthetic warps

$$ \begin{pmatrix} x_t \\ x_{t+1} \end{pmatrix}, g $$

### Faces

**Learned embedding**

Different face instances are automatically aligned

### Cat faces
Equivariance revisited

Key ideas
1. Equivariance learns a data representation from a relationship \((x, x', g)\) between data pairs
2. It works by trying to map the data into a simpler representation \(y\) where the relationship still holds true

\[ \Phi(g x) = g \Phi(x) \]

Auto-equivariance, factorization, and normalization

A new assumption
The representation space \(Y\) coincides with the transformation group \(G\)

Consequences
1. We can rewrite the constraint as a factorization
2. The factors provide an absolute reference frame for the data, normalizing it

Learning co-variant feature detectors

Aligning 3D object categories

Learning covariant feature detectors, Lenc Vedaldi, ECCV 2016
**Viewpoint factorization network**

![Diagram of Viewpoint factorization network]

- Absolute viewpoint from CNN
- Relative viewpoint from SFM

**Learned absolute viewpoint**

![Image of learned absolute viewpoint]

**Learning to count**

- Train a network to count the number of "objects" within an image.

**Key idea**: the count of the full image must be the sum of the counts of the subparts.

- Three-ways factorization

**Self-supervised features**

**Self-supervised structure**

**Deep image prior**

A priori information is contained in the structure of the CNN. No supervision! The network provides a parametrization of images:

$$w \text{ is not learned but used as a free image parameter}$$

$$x = \Psi(w; z_0)$$

A parameterization that offers high-impedance to noise:

$$\min_w \|x - \Phi(w)\|^2$$

The convergence speed is proportional to how "natural" the image looks.

For inpainting we only reconstruct the visible pixels, implicitly infer the others:

$$\min_w \|m \odot (x - \Phi(w))\|^2$$
UNet-style network

Deep image prior completion

Three key challenges of deep learning

Explaining deep neural networks

Peeking inside the black box

- Fewer models to train
- Less effort to train new models
- Trust, safety, and usability

- What does a net do?
  - What concepts can it recognise?
  - Spurious correlations?
  - Limitations?

- How does it do it?
  - Template matching?
  - Compositionality?
  - Spatial reasoning?

- How does it learn it?
  - Generalization?
  - Optimisation?

Gold Finch
Peeking inside the black box
Explaining deep neural networks

What does a net do?
- What concepts can it recognise?
- Spurious correlations?
- Limitations?

Deep networks as encoders
Each subnetwork $\Phi$ maps an image $x$ to a code $y$

Images
$x \in \mathbb{R}^m$

Codes
$y \in \mathbb{R}^n$

Generating iconic examples
Attribution
Semantic identification
Generating iconic examples

Attribution

Semantic identification

Find out “how much” of the image \( x \) can be reconstructed from the code \( y \)

How much information about \( x \) does \( y \) contain?

Images
\( \mathcal{X} = \mathbb{R}^m \)

Codes
\( \mathcal{Y} = \mathbb{R}^n \)

Reconstructions are not unique; rather, they form an equivalence class of images that are the same for the network

How much information about \( x \) does \( y \) contain?

Images
\( \mathcal{X} = \mathbb{R}^m \)

Codes
\( \mathcal{Y} = \mathbb{R}^n \)

Starting from random noise, “match” the code via direct optimization

Finding a pre-image in practice

Images
\( \mathcal{X} = \mathbb{R}^m \)

Codes
\( \mathcal{Y} = \mathbb{R}^n \)

\[ \min_x \| \Phi(x) - \Phi(x_0) \|_2^2 \]
Neural nets are “meaningless” outside their training domain. Hence, reconstructions should be constrained to be natural images.

\[
\mathcal{X} = \mathbb{R}^m
\]

\[
\mathcal{X}_n = \text{natural}
\]

\[
\mathcal{X}_{pn} = \text{pseudo-natural}
\]

\[
\mathcal{Y} = \mathbb{R}^n
\]

\[
\Phi^{-1}
\]

Several possible implementations

Inverting codes via the deep image prior

Inverting a Deep CNN

The inverter is only given
- The code \( y_0 \)
- The network to invert \( \Phi \)
- The structure (not the parameters) of the generator \( \psi \)
Inverting a Deep CNN

Original image

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

Original image

Inverting a Deep CNN

Original image

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

Original image

Inverting a Deep CNN

Original image

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

Original image

Inverting a Deep CNN

Original image

conv 1  conv 2  conv 3  conv 4  conv 5  fc 6  fc 7  fc 8
Inverting a Deep CNN

Original image

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

Inverting a Deep CNN

Original image

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

Inverting a Deep CNN

Original image

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

Inverting a Deep CNN

Original image

conv 1 conv 2 conv 3 conv 4 conv 5 fc 6 fc 7 fc 8
Inverting a Deep CNN

<table>
<thead>
<tr>
<th>conv 1</th>
<th>conv 2</th>
<th>conv 3</th>
<th>conv 4</th>
<th>conv 5</th>
<th>fc 6</th>
<th>fc 7</th>
<th>fc 8</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Original image" /></td>
<td><img src="image2.png" alt="Original image" /></td>
<td><img src="image3.png" alt="Original image" /></td>
<td><img src="image4.png" alt="Original image" /></td>
<td><img src="image5.png" alt="Original image" /></td>
<td><img src="image6.png" alt="Original image" /></td>
<td><img src="image7.png" alt="Original image" /></td>
<td><img src="image8.png" alt="Original image" /></td>
</tr>
</tbody>
</table>

Inverting a Deep CNN

<table>
<thead>
<tr>
<th>conv 1</th>
<th>conv 2</th>
<th>conv 3</th>
<th>conv 4</th>
<th>conv 5</th>
<th>fc 6</th>
<th>fc 7</th>
<th>fc 8</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Original image" /></td>
<td><img src="image2.png" alt="Original image" /></td>
<td><img src="image3.png" alt="Original image" /></td>
<td><img src="image4.png" alt="Original image" /></td>
<td><img src="image5.png" alt="Original image" /></td>
<td><img src="image6.png" alt="Original image" /></td>
<td><img src="image7.png" alt="Original image" /></td>
<td><img src="image8.png" alt="Original image" /></td>
</tr>
</tbody>
</table>
Inverting a Deep CNN

Activation maximization

Look for an image that maximally activates a specific neuron activation

Is the code semantic or visual?

Decoding AlexNet trained on ImageNet

fc8 is a 1000-dimensional class score vector... or is it?

Deep Quiz

https://goo.gl/jURsCP
**Visualization via direct optimization**

**Reading list**

- Visualizing higher-layer features of a deep network. Erhan, Bengio, Courville, U Montreal, 2009
- Visualizing and understanding convolutional networks
  Zeiler Fergus. ECCV, 2014.
- Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps
- Understanding deep image representations by inverting them
  Mahendran Vedaldi, CVPR, 2015
- Google “Inceptionism”
  Mordvintsev et al. 2015
- Understanding neural networks through deep visualisation
  Yosinski et al. ICMLW, 2015
- Plug & play generative networks: Conditional iterative generation of images in latent space
  Nguyen, Yosinksi, Bengio, Dosovitskiy, Clune, CVPR, 2017
- Deep image prior
  Ulyanov Vedaldi Lempistky, CVPR, 2018

**The choice of inversion prior has a strong effect**

**AlexNet Visualizations**

- Deep Image Prior
- TV-Norm Prior
The inverter is only given:
- The code $y_0$
- The network to invert $\Phi$
- The structure (not the parameters) of the generator $\psi$

Using the deep image prior is just one option...

... another is to learn an inverse generator network

The inverter is now given:
- The code $y_0$
- A large training set of images to learn a generator from

Visualization via learning a generator

Train a strong prior from examples

Our goal: diagnose a given discriminator network $\Phi$
But inversions also reflect the chosen "natural image" prior $p(x)$

Diagnostic vs aesthetic value

Inverting convolutional networks with convolutional networks
Dosovitskiy Brox, CVPR, 2016

Generating images with perceptual similarity metrics based on deep networks
Dosovitskiy Brox, NIPS, 2016

Synthesizing the preferred inputs for neurons in neural networks via deep generator networks
Nguyen, Dosovitskiy, Yosinski, Brox, Clune, NIPS, 2016

Plug & play generative networks: Conditional iterative generation of images in latent space
Nguyen, Yosinksi, Bengio, Dosovitskiy, Clune, CVPR, 2017

Deep Image Prior
Empirical prior

Illustrates the model $\Phi$
Illustrates the prior $p(x)$

ImageNet validation set (empirical distribution)
If you want to dig further

**Reviews and interfaces**

The building blocks of interpretability
Olah, Satyanarayan, Johnson, Carter, Schubert, Ye, Mordvintsev

Understanding neural networks through deep visualisation
Yosinksi et al. ICMLW, 2015

.. another is to learn an inverse generator network

The inverter is now given:

\[
\min_{\Psi} \frac{1}{N} \sum_{i=1}^{N} \| \Psi(\Phi(x_i)) - x_i \|^2
\]

Train using lots of empirical examples

Find what parts of an image are salient for a deep network

Generating iconic examples

Attribution

Semantic identification
Saliency: Backpropagation

Sensitivity analysis of target neuron w.r.t. input pixels

\[ J = \frac{\partial \Phi(x)}{\partial x} \]

The "salient" pixels usually light up


Three popular methods

- **Deconvolution**
  - Visualizing and understanding convolutional networks
  - Zeiler Fergus, ECCV, 2014

- **Gradient (backpropagation)**
  - Deep inside convolutional networks: Visualising image classification models and saliency maps
  - Simonyan, Vedaldi, Zisserman, ICLR, 2014

- **Guided backpropagation**
  - Striving for simplicity: The all convolutional net
  - Springenberg, Dosovitskiy, Brox, Riedmiller, ICLR, 2015

The only difference is in how ReLU is reversed!

Salient deconvolutional networks, Mahendran Vedaldi, ECCV, 2016
The saliency of neurons at the same location is about the same.

Lack of channel specificity:
- Saliency for: maximally activated neuron
- Saliency for: random neuron
- Saliency for: minimally activated neuron

Define some other rules to back-propagate the “relevance” of activations.

Saliency: Relevance and excitation backpropagation

A simple example: gradients modulated by the forward activations.

Actuation
\[ \begin{pmatrix} a_{1,1} \ldots a_{1,N} \\ \vdots \ldots \vdots \\ a_{D,1} \ldots a_{D,N} \end{pmatrix} \]
\[ \begin{pmatrix} w_{11} \ldots w_{1D} \\ \vdots \ldots \vdots \\ w_{C1} \ldots w_{CD} \end{pmatrix} \]
\[ \begin{pmatrix} z_{1a} \ldots z_{1b} \\ \vdots \ldots \vdots \\ z_{Da} \ldots z_{Db} \end{pmatrix} \]

ReLU
\[ \max(0, z) \]

Activation
\[ \begin{pmatrix} r_{1,1} \ldots r_{1,N} \\ \vdots \ldots \vdots \\ r_{D,1} \ldots r_{D,N} \end{pmatrix} \]
\[ \begin{pmatrix} z_{12} \ldots z_{1b} \\ \vdots \ldots \vdots \\ z_{D2} \ldots z_{Db} \end{pmatrix} \]

Relevance backpropagation formula

On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation
Bach, Binder, Montavon, Klauschen, Müller. PLOS one, 2015

Top-down neural attention by excitation backprop
Zhang, Lin, Brandt, Shen, Sclaroff, ECCV, 2016

CAM and Grad-CAM
Better channel specificity can be achieved by backpropagating only a few layers.

Saliency for: cat class neuron
Saliency for: dog class neuron

Learning deep features for discriminative localization
Zhou, Khosla, Lapedriza, Oliva, Torralba. CVPR, 2016

Grad-CAM: Visual explanations from deep networks via gradient-based localization
Selvaraju, Cogswell, Das, Vedantam, Parikh, Batra. ICCV, 2017

Due to chaining and cancelations we get that at any level m relevance is the modulated gradient.

\[ r_m = \frac{d}{dx_m} (\text{diag}(x_m)) \]

See extensive tutorial this morning!
Methods have been defined by specifying a “backpropagation formula.”

But what does the result of this computation actually mean?

It “looks good”

Deconvolution
- Sharp
- Poor spatial selectivity

Gradient
- Blurry
- Good spatial selectivity

Guided Backprop
- Sharp
- Good spatial sensitivity

Reminder: they all still have poor channel selectivity

The meaning of saliency

It generates “semantic” heat maps

A good correlation means that:
1) the diagnosed model “understand” the location of objects and
2) the saliency method can diagnose this fact

Drawbacks:
1) failure of localization confound limitations of the model and the saliency method
2) difficult to say which is which since the saliency formulas are largely heuristics

Gradients prove a local approximation of the model

The gradient can be directly interpreted as a local linear approximation of the model.
However, all other saliency propagation rules do not have simple interpretations such as this.
Towards a formal approach to explanations

Meaningful perturbation analysis

Study how $\Phi(x)$ changes up to perturbations $\pi(x)$ of the input $x$

**Perturbation** should be meaningful (interpretable). E.g:
- Injecting noise
- Rotating or translating the image
- Erasing parts of the image

The representation may
- Be invariant (stay the same)
- Be equivariant (respond predictably)

The analysis may be
- Local around $x$ and $\pi$
- For a distribution $p(x)$ and a fixed $\pi$
- For a distribution $p(\pi)$ and a fixed $x$
- ...

**Saliency via eliding objects**

We seek the "smallest elision" that maximally changes the neuron activation (more meaningful, ineffective)

Neural networks are fragile to adversarial perturbations

### Adversarial vs meaningful perturbations

Looking beyond neural network artifacts

**Adversarial elision**

- Improbable in nature

**Meaningful elision**

- Likely in nature

**Regularization** can help finding more meaningful perturbations

Examples: simplify the mask, look for the average effect of a pool of similar masks

---

### Meaningful minimal elisions

<table>
<thead>
<tr>
<th>Crisp regions</th>
<th>Similar to gradient, meaning is “obvious” by definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interpretable explanations of black boxes by meaningful perturbation, Fong Vedaldi, CVPR, 2017</td>
<td></td>
</tr>
</tbody>
</table>

---

### Diagnosing networks

What is salient may not be meaningful

<table>
<thead>
<tr>
<th>Mask Overlay</th>
<th>0.610 =&gt; 0.351</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mask Overlay</td>
<td>0.610 =&gt; 0.015</td>
</tr>
</tbody>
</table>

Example: the hot chocolate is recognized via the spoon and the truck vs the license plate

---

### CNN fragility

Easily fooled by adversarial examples

Let \( y = \Phi(x) \) be the label predicted for image \( x \) by the deep net

Empirically, we can find tiny perturbations \( x + \delta \) that change \( y \) arbitrarily!

\[
\delta^* = \text{argmin}_{\|\delta\|<\varepsilon} \|y_{\text{arbitrary}} - \Phi(x + \delta)\|
\]

**Intriguing properties of neural networks**

**Adversarial examples can be successfully “injected” in real life**

**Dangerous adversaries**

Adversarial glasses fooling face recognition

Adversarial stickers fooling sign recognition

**Access to a crime: Real and stealthy attacks on state-of-the-art face recognition** Sharif, Bhagavatula, Bauer, Reiter. CSS, 2016.


**Adversarial defence**

**Method:** recognize genuine vs adversarial images by learning a classifier on top of the saliency maps (Illustrative of properties of saliency, not really a recommended defense strategy!)

**Equivariance**

*How is a representation affected by an image warp?*

Short answer: warping image usually reduces to sparse linear tf in feature space.

Long answer: Understanding image representations by measuring their equivariance and equivalence. Lenc Vedaldi. CVPR 2015 & IJCV 2018

**Equivalence**

*Are different neural networks “the same”?*

Short answer: there generally are corresponding features in different networks (up to 1x1 linear tfs).

Long answer: Understanding image representations by measuring their equivariance and equivalence. Lenc Vedaldi. CVPR 2015 & IJCV 2018
Generating iconic examples

Semantic identification

Neurons and concepts: equivalence classes of images

Each neuron / concept “activates” for a subset of natural images (patches)

Assume that neurons have binary activation and that concepts apply deterministically

concept_i = \{ x \in \mathcal{X} : \text{concept}_i(x) = \text{true} \}

neuron_k = \{ x \in \mathcal{X} : \Phi_k(x) = 1 \}

Questions:
- Do neurons and concepts correspond one-to-one?
- How many neurons are required to express a concept?
- How many concepts are required to express a neuron?

Reading list

Identifying the meaning of neuron responses

Analyzing the performance of multilayer neural networks for object recognition
Agrawal, Girshick, Malik. ECCV, 2014

Object-centric representation learning from unlabeled videos.
Gao, Jayaraman, Grauman, ACCV, 2016

Places: An image database for deep scene understanding
Zhou, Khosla, Lapedriza, Torralba, Oliva. PAMI, 2016

Network dissection: Quantifying interpretability of deep visual representations
Bau, Zhou, Khosla, Oliva, Torralba. CVPR, 2017

Understanding intermediate layers using linear probes
Alain Bengio. ICLR Workshop, 2017

Revisiting the importance of individual units via ablation
Zhou He Bau Torrab, arXiv 2018

Correlates filters with ImageNet patches
Identifies the semantics of some convolutional filters
BRODEN, fine-grained semantic of individual filters
Understand training task performance
Learn linear predictor for diagnostics
Relation between interpretability and classification performance
Beyond just responding to images

Concepts as a structured abstraction

Concepts form a structured space. For example, WordNet induces an is-a hierarchy:

\[ \text{is-a-wing}(x) \approx \langle w_{\text{wing}}, \Phi(x) \rangle \]

Train a classifier from a network slice \( \Phi \)

\[ \text{is-a-wing}(x) \approx (w_{\text{wing}}, \Phi(x)) \]

Net2Vec associates a linear concept space to a network

Net2vec associates a linear concept space to a network

Net2Vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. Fong Vedaldi. CVPR, 2018
Thousands of images annotated with hundreds of concepts, often densely.

Image-level Annotations
- street (scene)
- flower (object)
- headboard (part)

Pixel-level Annotations
- swirly (texture)
- pink (color)
- metal (material)

Network dissection: Quantifying interpretability of deep visual representations
Bau, Zhou, Khosla, Oliva, Torralba. CVPR, 2017

Vecs: pixel-wise linear predictor of a concept
Net2vec: building concept embeddings

Threshold activations
“Dog” vec embedding
Channel-wise sum
Segmentation mask
IoU = .77

Subset selection follows
[Agrawal et al., 2014]

Singleton vests (only one neuron)
Best filter predictor
~ [Bau et al., 2017]
**Observation:** using more than one channel performs much better for most concepts

Individual neuron do not “isolate” concepts

**Concepts may need to be combined to explain neurons**

Neuron may correspond to concept combinations, or to “unknown” concepts

Sheep (IoU set = .21)

Horse (IoU set = .21)

Cow (IoU set = .20)

AlexNet conv5 66 is highly selective for multiple farm animals

**Exploring the structure of the learned concept space**

Net2vec associates a linear concept space to a network

The representation induces a similarity between concepts

\[ [K_{\Phi}]_{ij} = \langle w_{\text{concept}}, w_{\text{concept}} \rangle \]

We can now compare “conceptualizations”

\[ \text{similarity}(\Phi, \Psi) = \frac{\sum_{ij} [K_{\Phi}]_{ij} [K_{\Psi}]_{ij}}{\sqrt{\sum_{ij} [K_{\Phi}]_{ij}^2} \cdot \sqrt{\sum_{ij} [K_{\Psi}]_{ij}^2}} \]

**Number of feature channels per concept**

<table>
<thead>
<tr>
<th>Single channel</th>
<th>All channels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color</td>
<td>Conv1</td>
</tr>
<tr>
<td>Object</td>
<td>Conv2</td>
</tr>
<tr>
<td>Part</td>
<td>Conv3</td>
</tr>
<tr>
<td>Material</td>
<td>Conv4</td>
</tr>
<tr>
<td>...</td>
<td>Conv5</td>
</tr>
</tbody>
</table>

**Depends strongly on the concept, even at similar level of abstractions**

Number of Top Filters Used (F)

- **airplane**
  - 64 channels

- **person**
  - 8 channels
Comparing “conceptualizations”

AlexNet vs VGG16 vs GoogleNet vs Self-supervised vs Word Embeddings

Summary

Universal Representation
- Compact representation families
- Unsupervised Representation
- Self-supervision for learning features
- Self-supervision for learning structure
- What’s in the prior

Understandable Representations
- Iconic visualizations
- Attribution
- Semantic identification