Learning Deep Representations by Mutual Information Estimation & Maximization

Karan Grewal, University of Toronto
Collaborators

Devon Hjelm
Microsoft Research, Montréal
Mila

Alex Fedorov
Mind Research Network

Phil Bachman
Microsoft Research, Montréal

Adam Trischler
Microsoft Research, Montréal

Samuel Lavoie-Marchildon
Mila

Yoshua Bengio
Mila
Overview

1. representation learning

2. learning representations by maximizing mutual information

3. evaluating representations
Why learn representations?

raw data is often useless!

- low-dimensional “summary” of data
- can be used for other tasks - e.g. classification
- discover underlying structure in data
- deal with limited data
What is a good representation?

“one that captures the posterior distribution of the underlying explanatory factors for the observed input”

(Bengio et al., 2013)

1. high mutual information with the input
   → capture “useful” information, not low-level noise
2. task-dependent statistical properties
   → separability, independence
High-level overview

1. encode image into feature vector

2. combine feature vector with feature map of image (+)

3. combine feature vector with feature map of some other image (-)

4. train a GAN-like discriminator to tell apart (+) pair from (-) pair

(Goodfellow et al., 2014)
Mutual Information (MI)

measures the amount of information that two variables share

if $X, Y$ are independent, then $P_{XY} = P_X P_Y \iff$ MI is zero!

\[
I(X; Y) = \mathbb{E}_{P_{XY}} \left[ \log \frac{P_{XY}}{P_X P_Y} \right] = D_{KL}(P_{XY} \parallel P_X P_Y)
\]
Objectives of Deep InfoMax

\[ \text{enc}_\psi = h_\psi \circ f_\psi \] is an encoder that maps image \( x \in X \) to a feature map \( f_\psi(x) \) and then to a representation \( y = h_\psi(f_\psi(x)) \in Y \).

the distribution of \( y \)'s is called the “pushforward distribution” induced by pushing samples from \( f_\psi(X) \) through \( h_\psi \), denoted by \( Q_{\psi,X} \)

(1) maximize MI between \( f_\psi(X) \) and \( Y \)

(2) match the pushforward distribution \( Q_{\psi,X} \) to a prior \( Q_{\text{prior}} \), similar to adversarial autoencoders

(Makhzani et al., 2015)
General encoder architecture

\[ x \rightarrow \text{conv layers} \rightarrow f_\psi(x) \rightarrow \text{fc layers} \rightarrow y = h_\psi(f_\psi(x)) \]

- **Input image**: \(x\)
- **M x M feature map**: \(f_\psi(x)\)
- **64-dimensional representation**: \(y = h_\psi(f_\psi(x))\)
Loss functions

Donsker-Varadhan representation of KL divergence:

\[
D_{KL}(P \parallel Q) \geq \sup_{T_\omega : \Omega \to \mathbb{R}} \mathbb{E}_P [T_\omega] - \log \mathbb{E}_Q \left[ e^{T_\omega} \right]
\]

(Donsker & Varadhan, 1983)

Let \( T_\omega : f_\psi(X) \times Y \to \mathbb{R} \) be a discriminator with parameters \( \omega \) that distinguishes samples from the joint distribution \( P_{XY} \) and product of marginals \( P_X P_Y \)

\[
\max_{\omega, \psi} \hat{I}^{(DV)}(f_\psi(X); \text{enc}_\psi(X))
\]

\[\iff\]

\[
\max_{\omega, \psi} \mathbb{E}_{P_{XY}} [T_\omega(f_\psi(x), \text{enc}_\psi(x))] - \log \mathbb{E}_{P_X \times \tilde{P}_X} \left[ e^{T_\omega(f_\psi(\tilde{x}), \text{enc}_\psi(x))} \right]
\]
Global InfoMax
Global InfoMax

1. sample (+) images $x_+^{(1)}, \ldots, x_+^{(n)} \sim P_X$, compute feature maps $f_\psi(x_+^{(i)}) \ \forall \ i$

2. compute $y^{(i)} = h_\psi(f_\psi(x_+^{(i)})) \ \forall \ i$

3. call $\{(f_\psi(x_+^{(i)}), y^{(i)})\}$ the (+) pairs

4. sample (-) images $x_-^{(1)}, \ldots, x_-^{(n)} \sim P_X$, compute feature maps $f_\psi(x_-^{(i)}) \ \forall \ i$

5. call $\{(f_\psi(x_-^{(i)}), y^{(i)})\}$ the (-) pairs

6. maximize $\frac{1}{n} \sum_{i=1}^{n} T_\omega(f_\psi(x_+^{(i)}), y^{(i)}) - \log \frac{1}{n} \sum_{i=1}^{n} e^{T_\omega(f_\psi(x_-^{(i)}), y^{(i)})}$ by iteratively optimizing $\psi$ and $\omega$
Local InfoMax

**hypothesis** - information that is shared between all patches is more likely to be captured by $\text{enc}_\psi$.

**pro** → global structure of input is encoded

**con** → model has no incentive to focus on relevant information, pixel-level noise is likely to be captured.

Relevant information

Irrelevant
Local InfoMax

let \( f_\psi(X)_j \) be patch \( j \) of the feature map, \( \forall j = 1, \ldots, M \times M \)

maximize MI between \( f_\psi(X)_j \) and \( y \) for all patches \( j \), just like global infomax

\[
\max_{\omega, \psi} \frac{1}{M^2} \sum_{j=1}^{M^2} \hat{I}^{(DV)}(f_\psi(X)_j; \text{enc}_\psi(X))
\]
Local InfoMax

- M x M features
- Local Feature (+)
- Real
  - M x M
- Local Feature (+)
- Fake
  - M x M
- M x M features from another image
- Local Feature (-)
Local InfoMax

1. sample (+) images \( x_+^{(1)}, \ldots, x_+^{(n)} \sim P_X \), compute feature maps \( f_\psi(x_+^{(i)}) \forall i \)

2. compute \( y^{(i)} = h_\psi(f_\psi(x_+^{(i)})) \forall i \)

3. call \( \{(f_\psi(x_+^{(i)}), y^{(i)})\} \) the (+) pairs \( \forall j \)

4. sample (-) images \( x_-^{(1)}, \ldots, x_-^{(n)} \sim P_X \), compute feature maps \( f_\psi(x_-^{(i)}) \forall i \)

5. call \( \{(f_\psi(x_-^{(i)}), y^{(i)})\} \) the (-) pairs \( \forall j \)

6. maximize \( \frac{1}{M^2} \sum_{j=1}^{M^2} \left( \frac{1}{n} \sum_{i=1}^{n} T_\omega(f(x_+^{(i)}), y^{(i)}) - \log \frac{1}{n} \sum_{i=1}^{n} e^{T_\omega(f(x_-^{(i)}), y^{(i)})} \right) \) by iteratively optimizing \( \psi \) and \( \omega \)

\[ \text{patch } j \]

\[ \text{patch } j \text{ of another image} \]
Matching representations to prior

motivation: want the pushforward distribution $Q_{\psi,X}$ to match a specified prior $Q_{\text{prior}}$

→ train a discriminator $\mathcal{D}_\theta : Y \to \mathbb{R}$ to tell apart samples drawn from $Q_{\text{prior}}$ vs $Q_{\psi,X}$

$$\min_\psi \max_\theta \mathcal{D}_{\text{JSD}}(Q_{\text{prior}} \parallel Q_{\psi,X})$$

$$\iff \min_\psi \max_\theta \mathbb{E}_{Q_{\text{prior}}} [\log \mathcal{D}_\theta(u)] + \mathbb{E}_{Q_{\psi,X}} [\log(1 - \mathcal{D}_\theta(y))]$$

(Makhzani et al., 2015)
Putting it all together

DIM objective:

\[
\max_{\omega, \psi} \alpha \hat{I}^{(DV)}(f_\psi(X); \text{enc}_\psi(X)) + \max_{\omega, \psi} \beta \frac{1}{M^2} \sum_{j=1}^{M^2} \hat{I}^{(DV)}(f_\psi(X)_j; \text{enc}_\psi(X)) + \min_{\psi} \max_{\theta} \gamma D_{\text{JSD}}(Q_{\text{prior}} \parallel Q_{\psi,X})
\]

\(\alpha, \beta, \gamma \rightarrow \) tunable hyperparameters

global InfoMax

local InfoMax

prior matching
## Alternate objectives

<table>
<thead>
<tr>
<th>Objective</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Donsker-Varadhan bound</td>
<td>Tightest available bound on KL divergence</td>
<td>Unstable when only few negative samples used</td>
</tr>
<tr>
<td>Jensen-Shannon divergence</td>
<td>Stable</td>
<td>Undesirable statistical properties</td>
</tr>
<tr>
<td>Noise contrastive estimation</td>
<td>Impressive results</td>
<td>Requires many negative samples</td>
</tr>
<tr>
<td>Others - Wasserstein, MMD, etc</td>
<td></td>
<td>Did not try!</td>
</tr>
</tbody>
</table>
**infoNCE** (Oord et al., 2018)

noise contrastive estimation:

“a good model should be able to tell apart data from noise”

$$\max_{\omega,\psi} \hat{\mathcal{I}}^{(\text{infoNCE})}(f_\psi(X); \text{enc}_\psi(X))$$

$$\iff \max_{\omega,\psi} \mathbb{E}_{P_X} \left[ T_\omega(f_\psi(x), \text{enc}_\psi(x)) - \mathbb{E}_{\tilde{P}_X} \left[ \log \sum_{\tilde{x}} e^{T_\omega(f_\psi(\tilde{x}), \text{enc}_\psi(x))} \right] \right]$$
## Results - classification

(1) linear classifiers - train SVM to perform classification on learned representations

<table>
<thead>
<tr>
<th>Method</th>
<th>conv</th>
<th>fc</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>variational autoencoder (VAE)</td>
<td>53.8</td>
<td>42.1</td>
<td>39.6</td>
</tr>
<tr>
<td>adversarial autoencoder</td>
<td>55.2</td>
<td>43.3</td>
<td>37.8</td>
</tr>
<tr>
<td>BiGAN</td>
<td>56.4</td>
<td>38.4</td>
<td>44.9</td>
</tr>
<tr>
<td>NAT</td>
<td>48.6</td>
<td>42.6</td>
<td>39.6</td>
</tr>
<tr>
<td>DIM</td>
<td>57.6</td>
<td>45.6</td>
<td>18.6</td>
</tr>
<tr>
<td>DIM - global only</td>
<td>46.8</td>
<td>28.8</td>
<td>29.1</td>
</tr>
<tr>
<td>DIM - local only</td>
<td>63.3</td>
<td>54.1</td>
<td>49.6</td>
</tr>
</tbody>
</table>

CIFAR-10, SVM

- **conv**: last conv layer
- **fc**: 2nd last fc layer
- **Y**: 64-dimensional representation
(2) non-linear classifiers - train neural network with a single hidden layer to perform classification on learned representations

<table>
<thead>
<tr>
<th></th>
<th>CIFAR-10</th>
<th></th>
<th></th>
<th>CIFAR-100</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>conv</td>
<td>fc</td>
<td>Y</td>
<td>conv</td>
<td>fc</td>
<td>Y</td>
</tr>
<tr>
<td>fully supervised</td>
<td></td>
<td></td>
<td>75.4</td>
<td></td>
<td></td>
<td>42.3</td>
</tr>
<tr>
<td>VAE</td>
<td>60.7</td>
<td>60.5</td>
<td>54.6</td>
<td>37.2</td>
<td>34.1</td>
<td>24.2</td>
</tr>
<tr>
<td>$\beta$-VAE</td>
<td>62.4</td>
<td>57.9</td>
<td>55.4</td>
<td>32.3</td>
<td>26.9</td>
<td>29.0</td>
</tr>
<tr>
<td>adversarial autoencoder</td>
<td>59.4</td>
<td>57.2</td>
<td>52.8</td>
<td>36.2</td>
<td>33.4</td>
<td>23.3</td>
</tr>
<tr>
<td>BiGAN</td>
<td>62.6</td>
<td>62.7</td>
<td>52.9</td>
<td>37.6</td>
<td>33.3</td>
<td>21.5</td>
</tr>
<tr>
<td>DIM - global</td>
<td>52.2</td>
<td>52.8</td>
<td>43.2</td>
<td>27.7</td>
<td>24.4</td>
<td>20.0</td>
</tr>
<tr>
<td>DIM - local (Donsker-Varadhan)</td>
<td>72.7</td>
<td>70.6</td>
<td>64.7</td>
<td>48.5</td>
<td>44.4</td>
<td>39.3</td>
</tr>
<tr>
<td>DIM - local (Jensen-Shannon)</td>
<td>73.3</td>
<td>73.6</td>
<td>67.0</td>
<td>48.1</td>
<td>45.9</td>
<td>39.6</td>
</tr>
<tr>
<td>DIM - local (noise contrastive est.)</td>
<td>75.2</td>
<td>75.6</td>
<td>69.1</td>
<td>49.7</td>
<td>47.7</td>
<td>41.6</td>
</tr>
</tbody>
</table>

conv → last conv layer
fc → 2nd last fc layer
Y → 64-dimensional representation
(2) non-linear classifiers - train neural network with a single hidden layer to perform classification on learned representations

<table>
<thead>
<tr>
<th>Method</th>
<th>Tiny ImageNet</th>
<th>STL-10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>conv</td>
<td>fc</td>
</tr>
<tr>
<td>fully supervised</td>
<td></td>
<td>75.4</td>
</tr>
<tr>
<td>VAE</td>
<td>18.6</td>
<td>16.9</td>
</tr>
<tr>
<td>$\beta$-VAE</td>
<td>19.3</td>
<td>16.8</td>
</tr>
<tr>
<td>adversarial autoencoder</td>
<td>18.0</td>
<td>17.3</td>
</tr>
<tr>
<td>BiGAN</td>
<td>24.4</td>
<td>20.2</td>
</tr>
<tr>
<td>DIM - global</td>
<td>11.3</td>
<td>6.3</td>
</tr>
<tr>
<td>DIM - local (Donsker-Varadhan)</td>
<td>30.4</td>
<td>29.5</td>
</tr>
<tr>
<td>DIM - local (Jensen-Shannon)</td>
<td>33.5</td>
<td>36.9</td>
</tr>
<tr>
<td>DIM - local (noise contrastive est.)</td>
<td>34.2</td>
<td>38.1</td>
</tr>
</tbody>
</table>

conv → last conv layer
fc → 2nd last fc layer
Y → 64-dimensional representation
Results - MINE

(3) Mutual information neural estimation (MINE) - train neural network to estimate $\mathcal{I}(f_\psi(X); Y)$

(Belghazi et al., 2018)

<table>
<thead>
<tr>
<th>CIFAR-10</th>
<th>MINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAE</td>
<td>93.0</td>
</tr>
<tr>
<td>adversarial autoencoder</td>
<td>87.5</td>
</tr>
<tr>
<td>BiGAN</td>
<td>37.7</td>
</tr>
<tr>
<td>NAT</td>
<td>6.0</td>
</tr>
<tr>
<td>DIM</td>
<td>101.7</td>
</tr>
<tr>
<td>DIM - global only</td>
<td>49.6</td>
</tr>
<tr>
<td>DIM - local only</td>
<td>45.1</td>
</tr>
</tbody>
</table>
(4) Neural dependency measure (NDM)

- encode features, shuffle them along the batch axis, and train another discriminator to tell apart true representations from shuffled one

- measure of how co-dependent features are

(Brakel & Bengio, 2017)
Results - NDM

- Independent
- Dependent
(4) Neural dependency measure (NDM) (Brakel & Bengio, 2017)

- encode features, shuffle them along the batch axis, and train another discriminator to tell apart true representations from shuffled one

- measure of how co-dependent features are

<table>
<thead>
<tr>
<th>CIFAR-10</th>
<th>NDM</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAE</td>
<td>1.6</td>
</tr>
<tr>
<td>adversarial autoencoder</td>
<td>0.1</td>
</tr>
<tr>
<td>BiGAN</td>
<td>24.5</td>
</tr>
<tr>
<td>NAT</td>
<td>0.1</td>
</tr>
<tr>
<td>DIM</td>
<td>22.9</td>
</tr>
<tr>
<td>DIM - global only</td>
<td>10.0</td>
</tr>
<tr>
<td>DIM - local only</td>
<td>9.2</td>
</tr>
</tbody>
</table>
Results - occlusion

(5) occlusion

- given image $x$, randomly occlude part of the input image and call this $x'$
- maximize MI between $y' = \text{enc}_\psi(x')$ and $f_\psi(x)_j$
Results - occlusion

(5) occlusion

- given image $x$, randomly occlude part of the input image and call this $x'$
- maximize MI between $y' = \text{enc}_\psi(x')$ and $f_\psi(x)_j$

<table>
<thead>
<tr>
<th></th>
<th>CIFAR-10</th>
<th></th>
<th></th>
<th>CIFAR-100</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>conv</td>
<td>fc</td>
<td>Y</td>
<td>conv</td>
<td>fc</td>
<td>Y</td>
</tr>
<tr>
<td>DIM</td>
<td>77.5</td>
<td>73.3</td>
<td>70.7</td>
<td>49.9</td>
<td>48.0</td>
<td>44.3</td>
</tr>
<tr>
<td>DIM - occlusion</td>
<td>76.8</td>
<td>74.5</td>
<td>72.9</td>
<td>48.9</td>
<td>47.7</td>
<td>44.9</td>
</tr>
<tr>
<td>DIM - coordinate prediction</td>
<td>77.3</td>
<td>73.9</td>
<td>71.6</td>
<td>50.3</td>
<td>48.6</td>
<td>45.4</td>
</tr>
<tr>
<td>DIM - both</td>
<td>77.3</td>
<td>75.2</td>
<td>74.0</td>
<td>48.7</td>
<td>48.0</td>
<td>46.0</td>
</tr>
</tbody>
</table>

conv → last conv layer  
fc → 2nd last fc layer  
Y → 64-dimensional representation
Closest L1 neighbors

query

DIM - global

DIM - local
Results - number of negative samples
DIM on graphs

maximize MI between local patch representations and high-level summaries of graphs

(Veličković et al., 2019)
Conclusions

- high MI between input and representation is important!

- maximizing MI tends to capture global information, and to avoid capturing pixel-level noise, need to encode local information into representations

- encoding local information can also be harmful (see below)