Learning Without Labels

Unsupervised and Weakly Supervised Learning of Deep Models

Presented by Dr. Shazia Akbar
shazia@altislabs.com

AISC Premium Workshop 2019

Outline

- Self training
- Multiple instance learning
- Psuedolabeling
- Other learning paradigms

Cases when *y* is difficult to gather...

Discovering new biological changes/characteristics to treat diseases

- Knowledge is currently unknown
- Medical expertise is expensive and subjective
- Want to gather this information before death

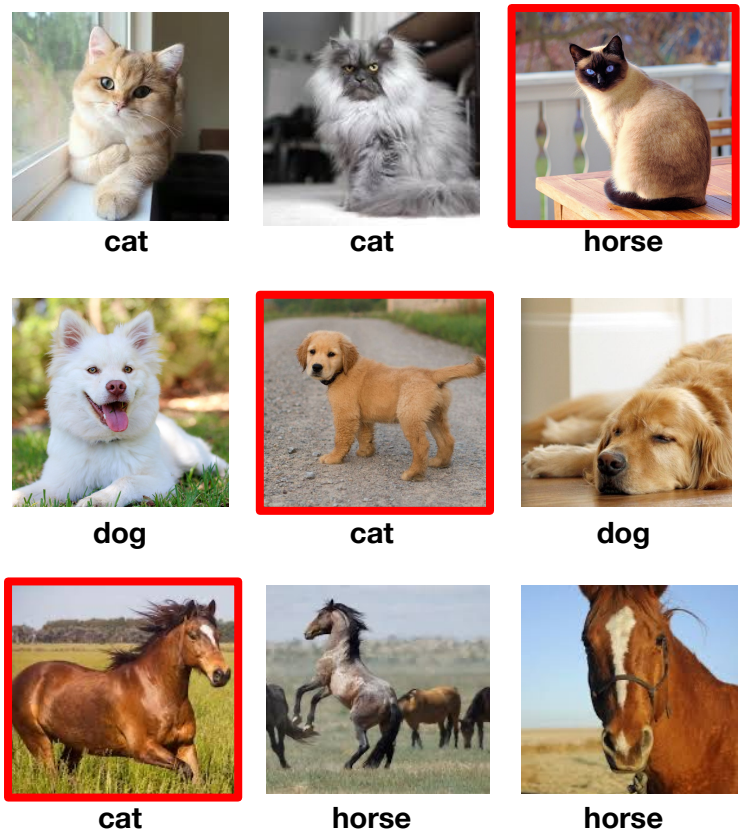
Anomaly detection

- Definition of abnormal is anything “not normal”

Is a Jaffa Cake a biscuit or a cake?

What are weak labels?

Noisy labels where some are correct and others aren’t



What are weak labels?

Noisy labels where some are correct and others aren’t

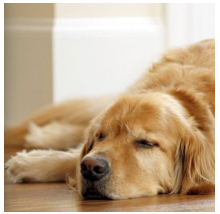
Mixture of labeled and unlabeled data

cat

cat

dog

dog



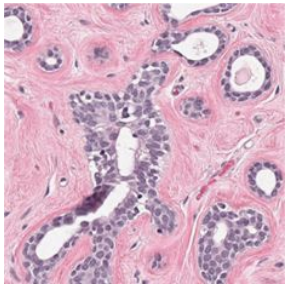
horse

What are weak labels?

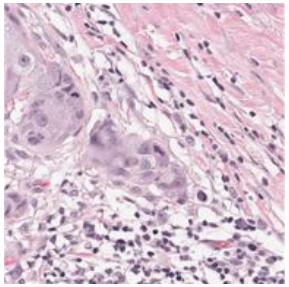
Noisy labels where some are correct and others aren’t

Mixture of labeled and unlabeled data

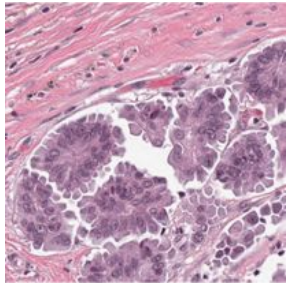
Coarsely labeled data (roughly estimated, eyeballing)



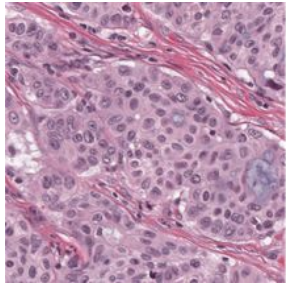
0%



25%



60%



99%

What are weak labels?

Noisy labels where some are correct and others aren’t

Mixture of labeled and unlabeled data

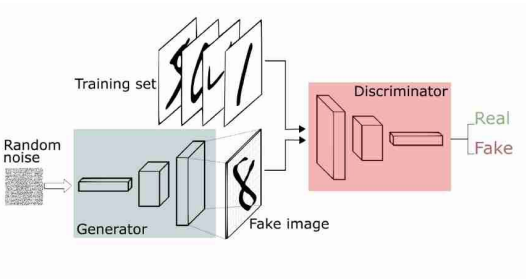
Coarsely labeled data (roughly estimated, eyeballing)

Unstructured labels (grouped, many-to-one, one-to-many)

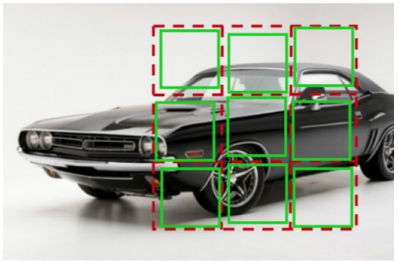
Street lamp
Car
Person
Trees
Road
Zebra Crossing

Trees
Car
Road

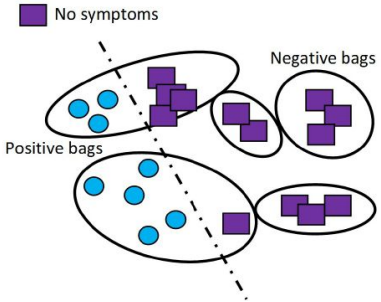
Weak supervision techniques



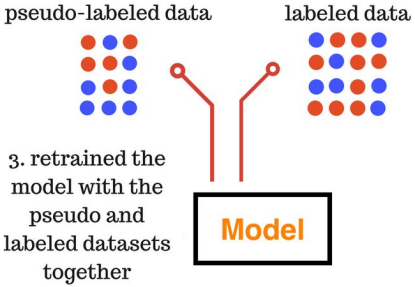
Generative Models
(data is modeled)



Self Training
(data provides the supervision)

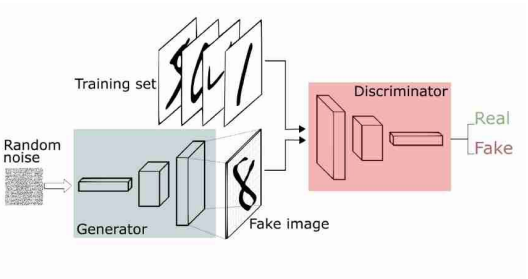


Multi-Instance Learning
(labels are coarse/noisy)

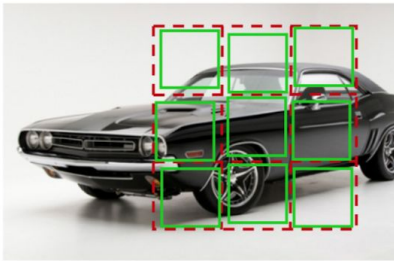


Psuedo-labeling
(labels are generated)

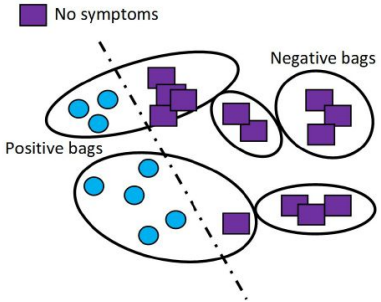
Weak supervision techniques



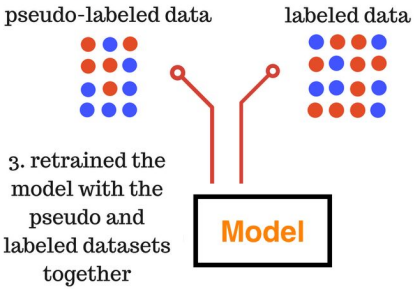
Generative Models
(data is modeled)



Self Training
(data provides the supervision)

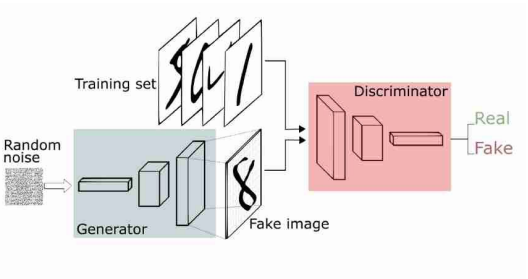


Multi-Instance Learning
(labels are coarse/noisy)

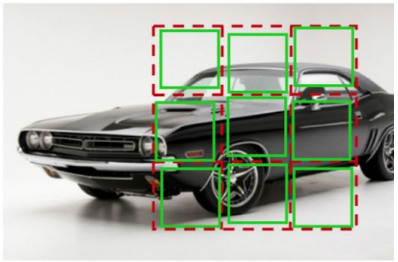


Psuedo-labeling
(labels are generated)

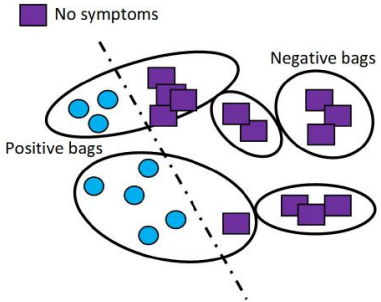
Weak supervision techniques



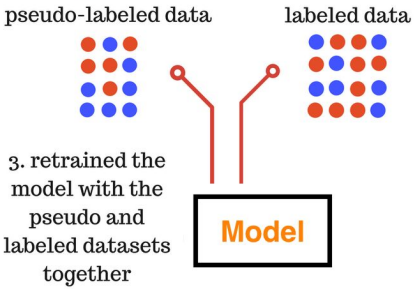
Generative Models
(data is modeled)



Self Training
(data provides the supervision)



Multi-Instance Learning
(labels are coarse/noisy)



Psuedo-labeling
(labels are generated)

Self Training (Image-Based)

An unsupervised learning problem that is framed as a supervised learning problem in order to apply supervised learning algorithms to solve it.

- Ground-truth is available for free

Examples:

- Making images grayscale
- Having a model predict a color representation (colorization)
- Removing blocks of the image and have a model predict the missing parts (inpainting)

Colorization

Train network to predict pixel colour from a grayscale input

The model outputs colors in the the CIE Lab* color space (approximates human vision)

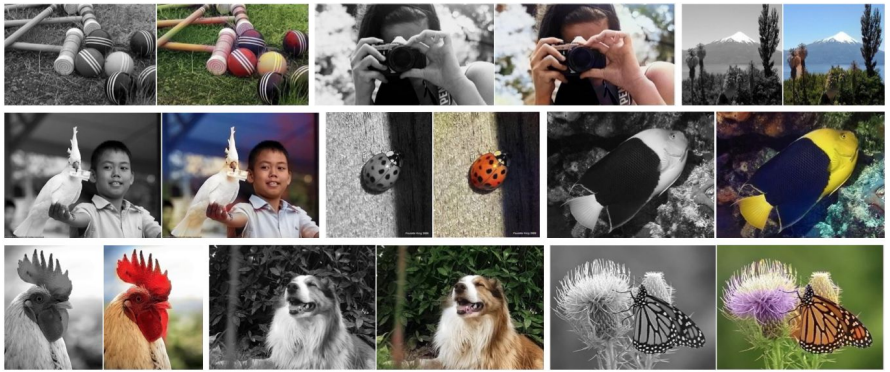


Fig. 1. Example input grayscale photos and output colorizations from our algorithm. These examples are cases where our model works especially well. Please visit `http://richzhang.github.io/colorization/` to see the full range of results and to try our model and code. Best viewed in color (obviously).

Colorful Image Colorization, Zhang et al. (2016)

Colorization

Train network to predict pixel colour from a grayscale input

The model outputs colors in the the CIE Lab* color space (approximates human vision)

Treated as a classification problem

The loss function is rebalanced with a weighting term that boosts the loss of infrequent color buckets

Colorful Image Colorization, Zhang et al. (2016)

Colorization

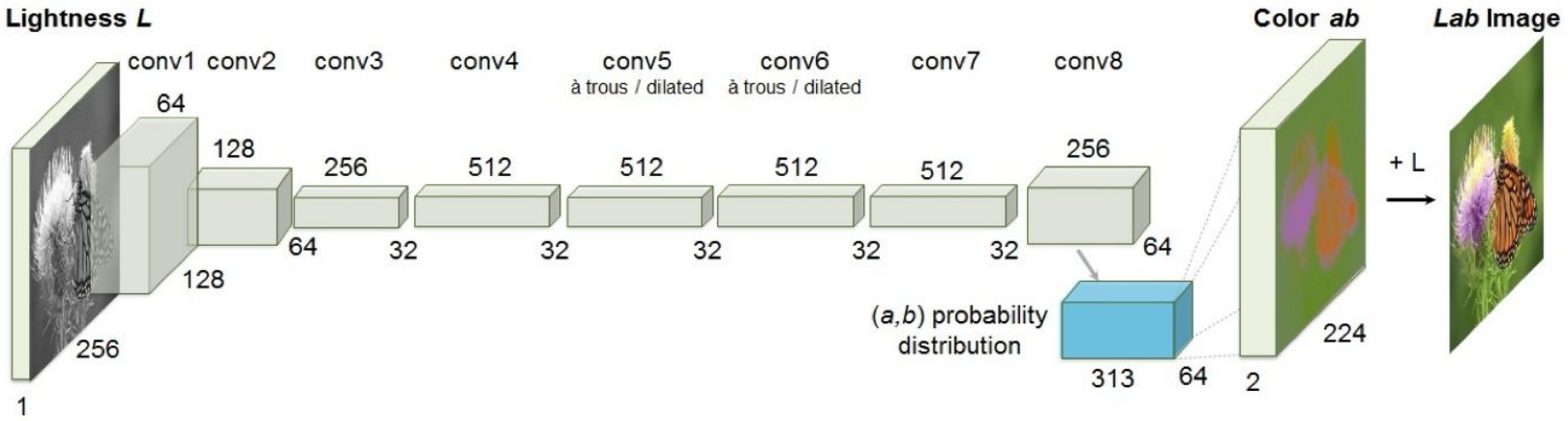


Fig. 2. Our network architecture. Each `conv` layer refers to a block of 2 or 3 repeated `conv` and `ReLU` layers, followed by a `BatchNorm` [30] layer. The net has no `pool` layers. All changes in resolution are achieved through spatial downsampling or upsampling between `conv` blocks.

Colorful Image Colorization, Zhang et al. (2016)

Colorization

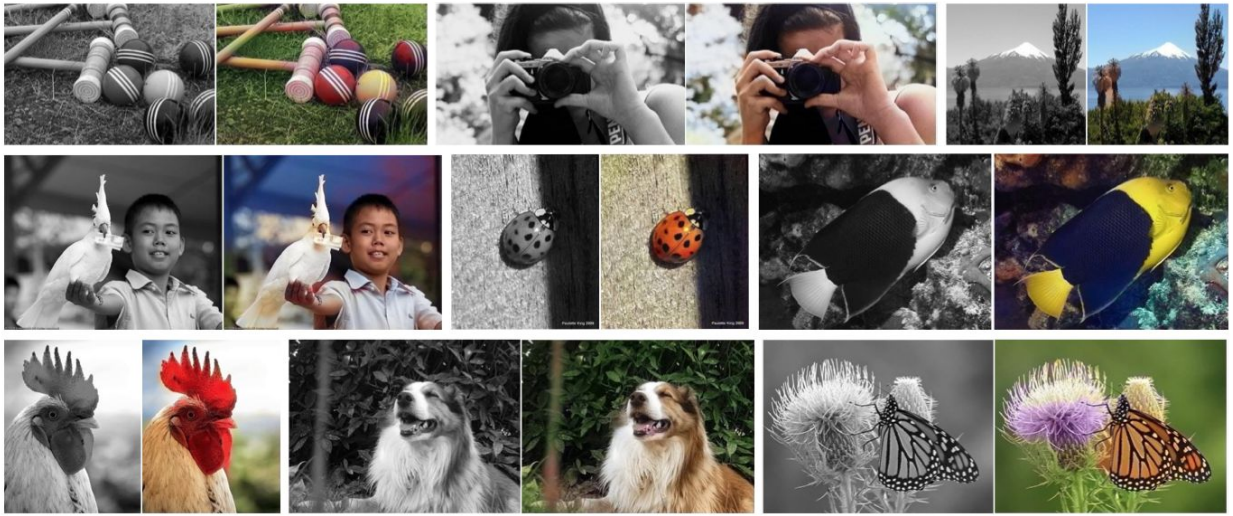
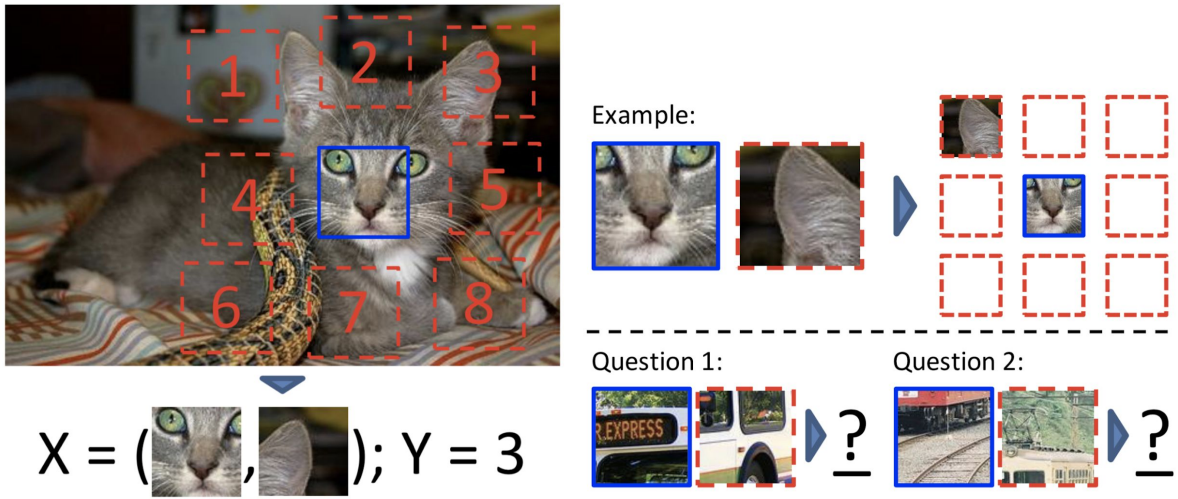


Fig. 1. Example input grayscale photos and output colorizations from our algorithm. These examples are cases where our model works especially well. Please visit `http://richzhang.github.io/colorization/` to see the full range of results and to try our model and code. Best viewed in color (obviously).

Colorful Image Colorization, Zhang et al. (2016)

(Pre) Jigsaw

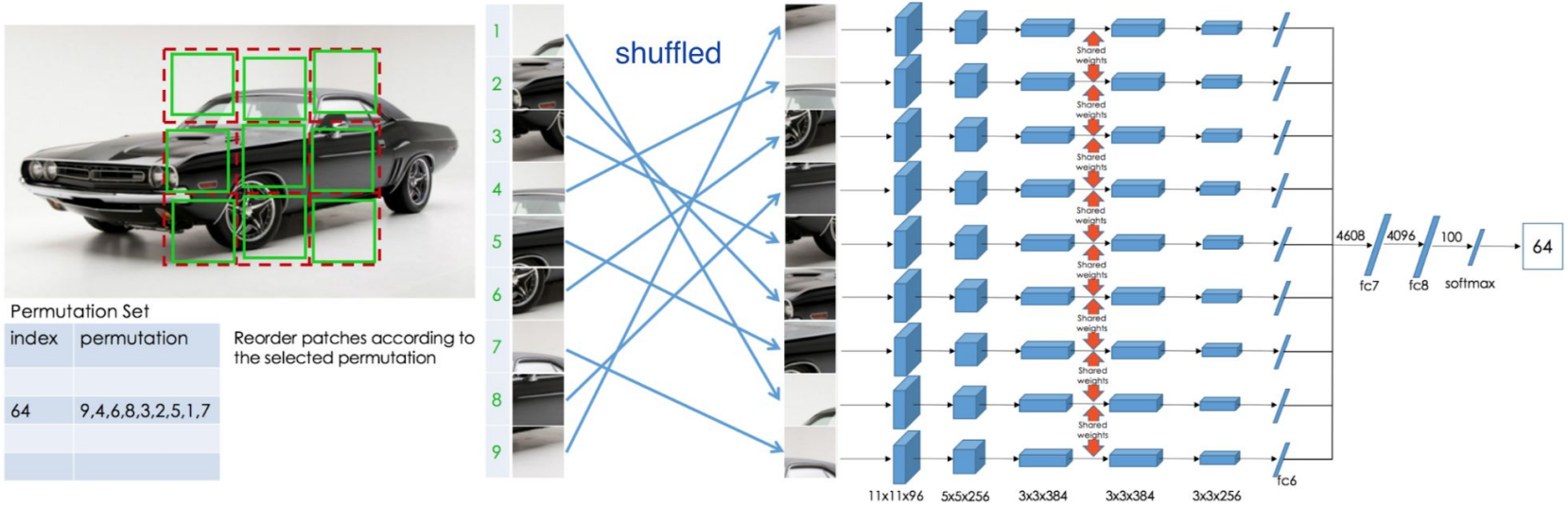
Predicting the *relative position* between two random patches from one image



Unsupervised Visual Representation Learning by Context Prediction, Doersch et al. (2015)

Jigsaw

Model is trained to place 9 shuffled patches back to the original locations



Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles, Noroozi & Favaro (2017)

Jigsaw

Context free network (CFN) because each patch is analyzed independently until the last few layers

Given 9 tiles, there are 9! = 362,880 possible permutations

Shortcuts: to avoid just learning arbitrary 2D positioning

- Multiple Jigsaw puzzles from the same image
- Random gap between tiles
- Resize whilst retaining original aspect ratio
- Random shifts, jitters

Jigsaw

Table 2: Comparison of classification results on ImageNet 2012 [9]. The numbers are obtained by averaging 10 random crops predictions.

	🔒 conv1	🔒 conv2	🔒 conv3	🔒 conv4	🔒 conv5
CFN	**54.7**	**52.8**	**49.7**	45.3	**34.6**
Doersch *et al.* [10]	53.1	47.6	48.7	**45.6**	30.4
Wang and Gupta [39]	51.8	46.9	42.8	38.8	29.8
Random	48.5	41.0	34.8	27.1	12.0

Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles, Noroozi & Favaro (2017)

Exemplar Networks

Exemplar = seed image

Perturb/distort image patches, e.g. by cropping and affine transformations

Train a CNN to classify these exemplars into their “seed index”

- The output softmax is of length [number of seed images]
- E.g. if you have 8000 seed images, you end up solving a 8000-way classification problem

Exemplar Networks

Transformations include:

- Translation
- Scale
- Rotation
- Contrast
- HSV color

Showed to work well on modest datasets (CIFAR, Caltech)

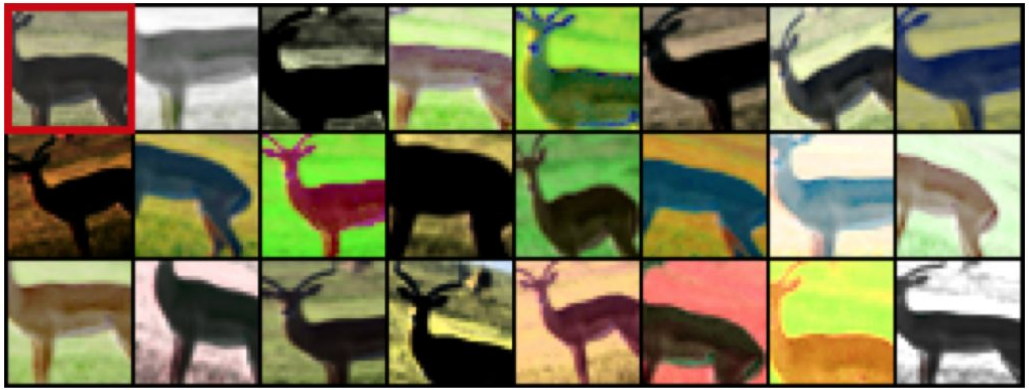
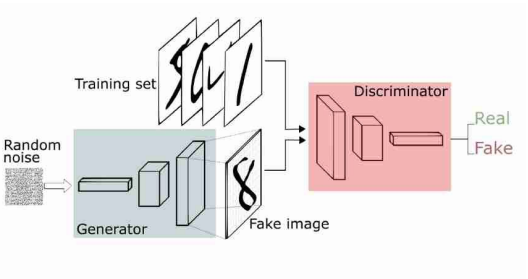


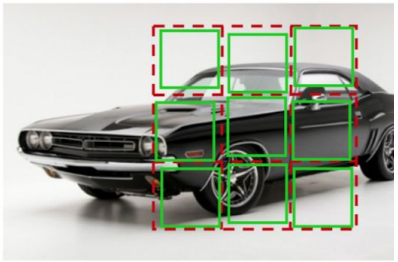
Fig. 2. Several random transformations applied to one of the patches extracted from the STL unlabeled dataset. The original ('seed') patch is in the top left corner.

Discriminative Unsupervised Feature Learning with Exemplar Convolutional Neural Networks, Dosovitskiy et al (2015)

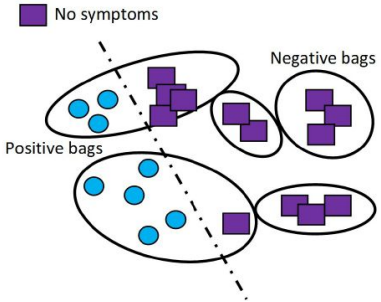
Weak supervision techniques



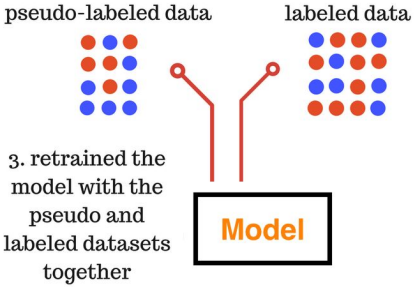
Generative Models
(data is modeled)



Self Training
(data provides the supervision)

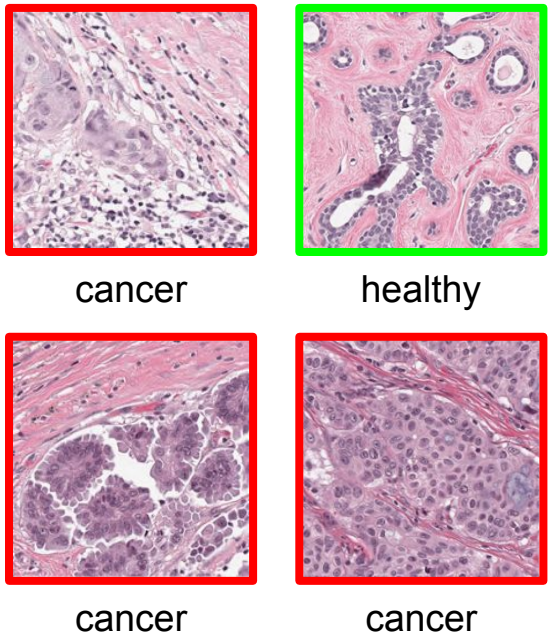


Multi-Instance Learning
(labels are coarse/noisy)



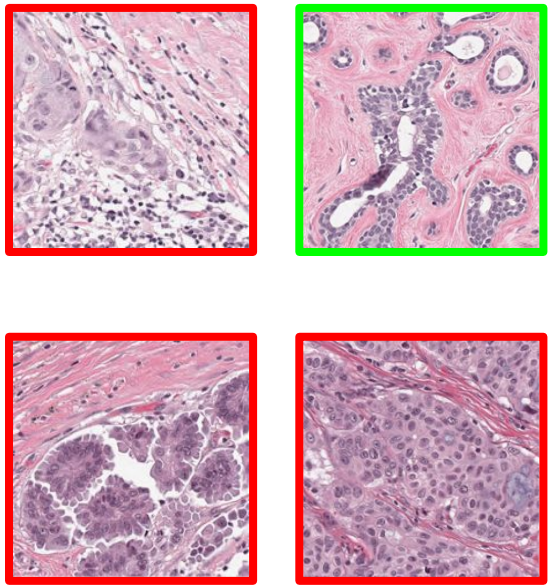
Psuedo-labeling
(labels are generated)

Multiple Instance Learning (MIL)



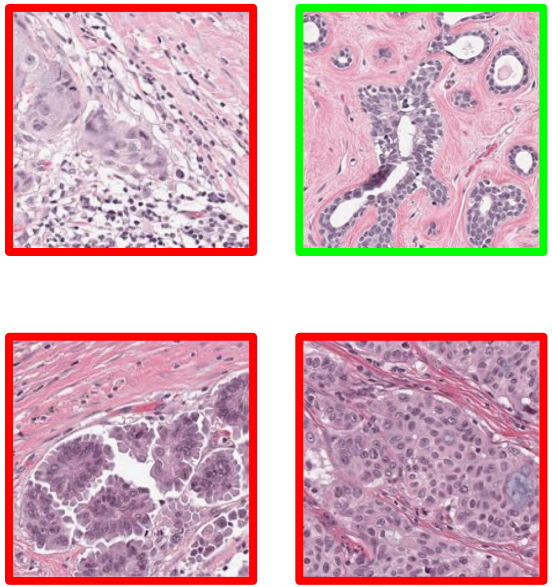
T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. Artificial Intelligence, 89:31–71, 1997.

Multiple Instance Learning (MIL)



T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. Artificial Intelligence, 89:31–71, 1997.

Multiple Instance Learning (MIL)



cancer

T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. Artificial Intelligence, 89:31–71, 1997.

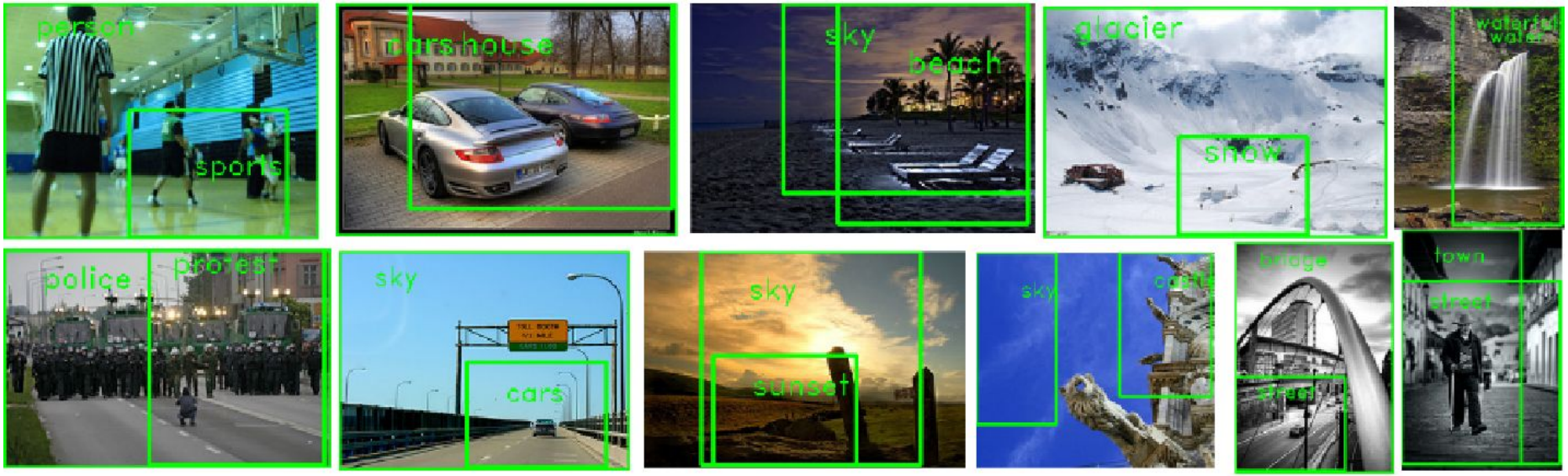
Multiple Instance Learning (MIL)

If bag is +ve, at least one instance **must** be +ve

If bag is -ve, all instances **must** be -ve

T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. Artificial Intelligence, 89:31–71, 1997.

Examples of MIL



** Zero-shot, few-shot learning

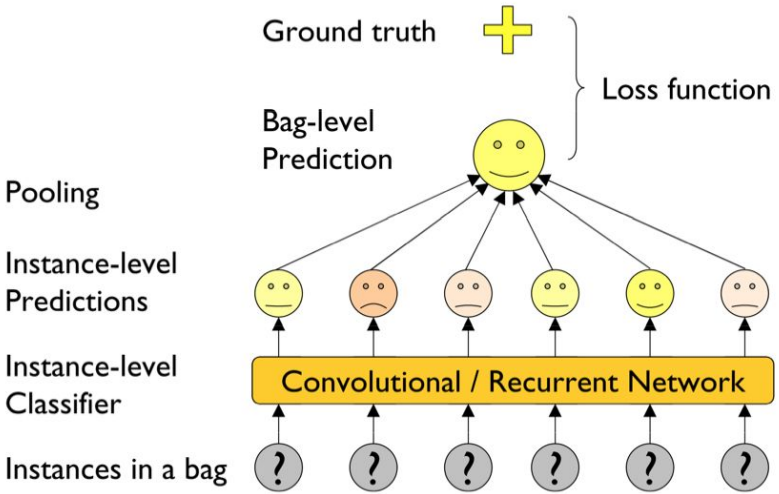
Examples of MIL

There was nothing so very remarkable in that; nor did Alice think it so very much out of the way to **hear the Rabbit say to itself, `Oh dear! Oh dear! I shall be late!'** (when she thought it over afterwards, it occurred to her that she ought to have wondered at this, but at the time it all seemed quite natural); but **when the Rabbit actually took a watch out of its waistcoat-pocket**, and looked at it, and then hurried on, Alice started to her feet, for it flashed across her mind that she had never before seen a rabbit with either a waistcoat-pocket, or a watch to take out of it, and burning with curiosity, she ran across the field after it, and fortunately was just in time to see it pop down a large rabbit-hole under the hedge.

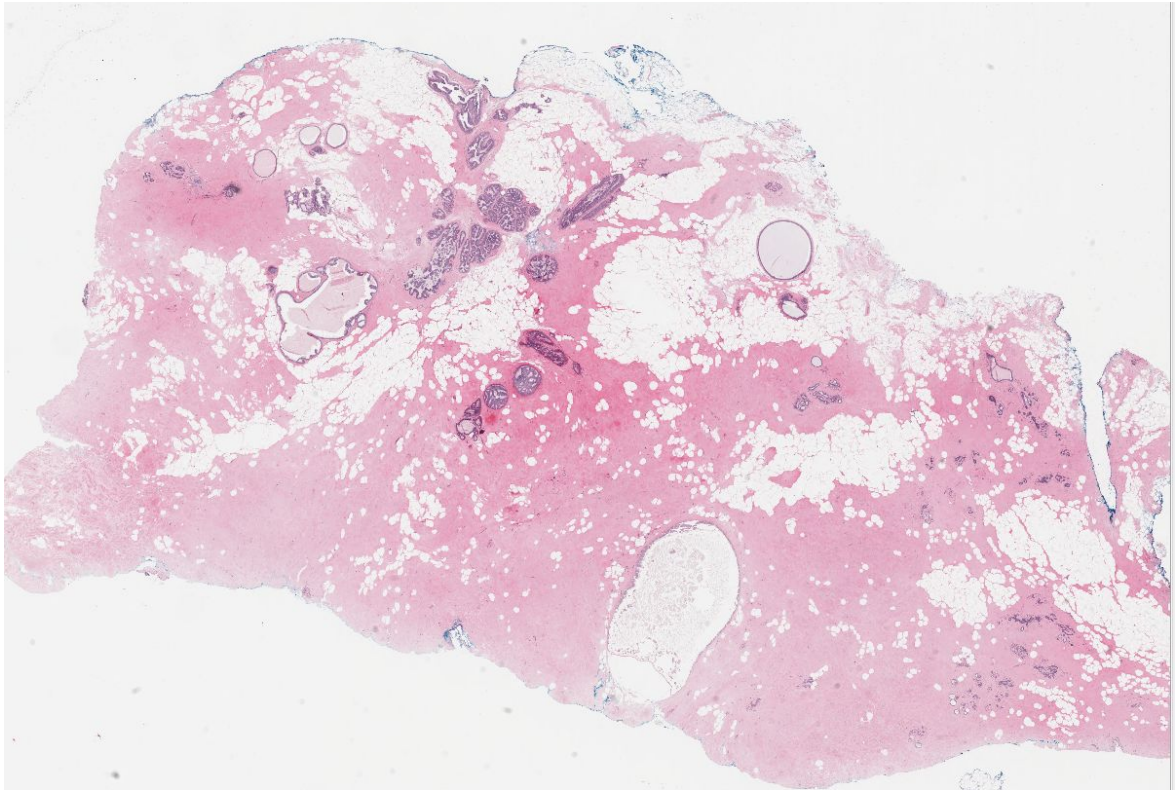
SCI-FI **FANTASY** **ROMANCE**

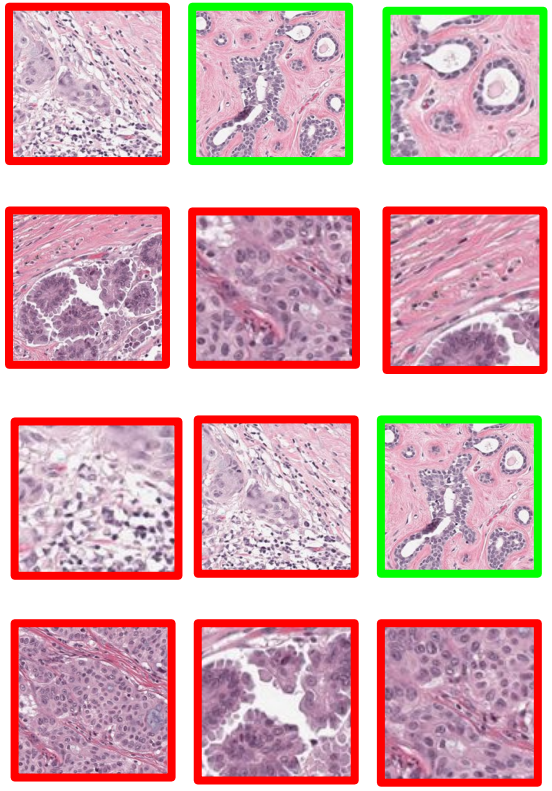
MIL Pooling

Treating each instance independently and then aggregating scores for a bag to generated a single classification



A Comparison Of Five Multiple Instance Learning Pooling Functions For Sound Event Detection With Weak Labeling, Wang et al (2019)





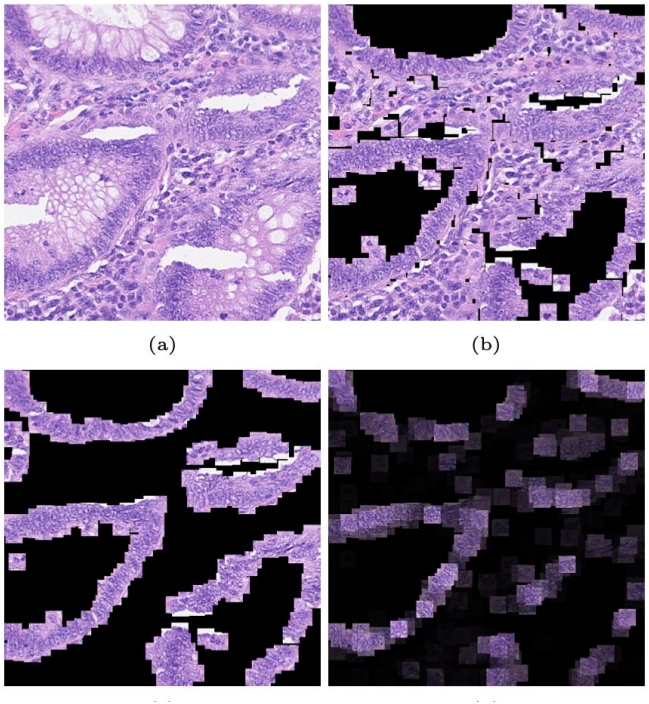
MIL Pooling

- Max Pooling
  - Take the largest y
- Average/Mean Pooling
  - Equal weight to all instances
- Linear Softmax
  - Equal weighted average of y’s
- Exponential Softmax
  - Exponential weighted average of y’s
- Attention
  - Weights for each instance are learned

Attention MIL Pooling

Attention-based Deep Multiple Instance Learning, Ilse et al, (2018):

https://arxiv.org/pdf/1802.04712.pdf



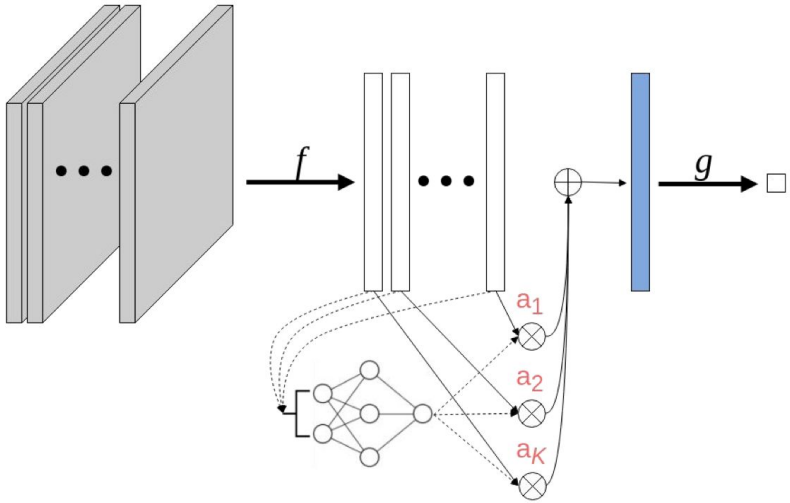
Attention MIL Pooling

Learn weights a which are then combined with instance-level scores:

$$z = \sum_{k=1}^{K} a_k h_k$$

Weights are learned through a MLP network which takes layer outputs as input

We can also have gated attention to learn relationship *between* instances



Attention-based Deep Multiple Instance Learning, Ilse et al, (2018)

Attention MIL Pooling

Table 2. Results on BREAST CANCER. Experiments were run 5 times and an average (± a standard error of the mean) is reported.

METHOD	ACCURACY	PRECISION	RECALL	F-SCORE	AUC
Instance+max	0.614±0.020	0.585±0.03	0.477±0.087	0.506±0.054	0.612±0.026
Instance+mean	0.672±0.026	0.672±0.034	0.515±0.056	0.577±0.049	0.719±0.019
Embedding+max	0.607±0.015	0.558±0.013	0.546±0.070	0.543±0.042	0.650±0.013
Embedding+mean	**0.741**±0.023	**0.741**±0.023	0.654±0.054	0.689±0.034	**0.796**±0.012
Attention	**0.745**±0.018	0.718±0.021	**0.715**±0.046	**0.712**±0.025	0.775±0.016
Gated-Attention	**0.755**±0.016	**0.728**±0.016	**0.731**±0.042	**0.725**±0.023	**0.799**±0.020

Table 3. Results on COLON CANCER. Experiments were run 5 times and an average (± a standard error of the mean) is reported.

METHOD	ACCURACY	PRECISION	RECALL	F-SCORE	AUC
Instance+max	0.842 ± 0.021	0.866 ± 0.017	0.816 ± 0.031	0.839 ± 0.023	0.914 ± 0.010
Instance+mean	0.772 ± 0.012	0.821 ± 0.011	0.710 ± 0.031	0.759 ± 0.017	0.866 ± 0.008
Embedding+max	0.824 ± 0.015	0.884 ± 0.014	0.753 ± 0.020	0.813 ± 0.017	0.918 ± 0.010
Embedding+mean	0.860 ± 0.014	0.911 ± 0.011	0.804 ± 0.027	0.853 ± 0.016	0.940 ± 0.010
Attention	**0.904** ± 0.011	**0.953** ± 0.014	**0.855** ± 0.017	**0.901** ± 0.011	**0.968** ± 0.009
Gated-Attention	**0.898** ± 0.020	**0.944** ± 0.016	**0.851** ± 0.035	**0.893** ± 0.022	**0.968** ± 0.010

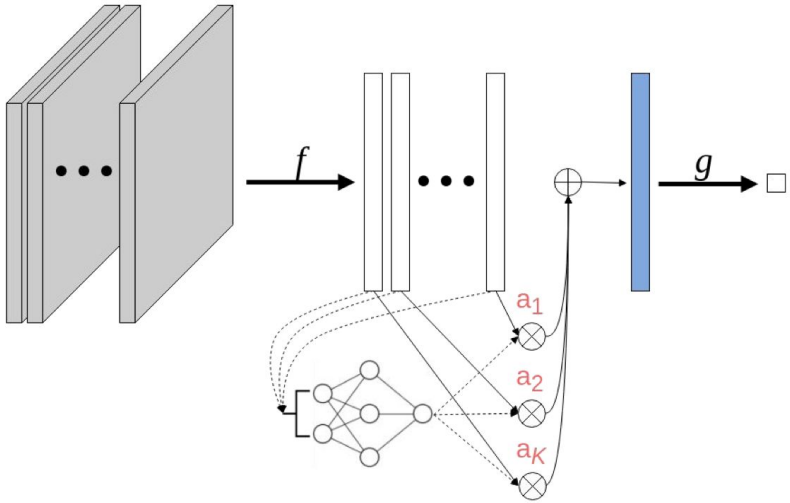
Attention-based Deep Multiple Instance Learning, Ilse et al, (2018)

Practical

Attention MIL Pooling

First implement max-pooling version of this:

1) Build a custom model which take as input multiple images and outputs **one** score
2) Add attention weights
3) Compare aggregated score to label from bag
4) Backprop

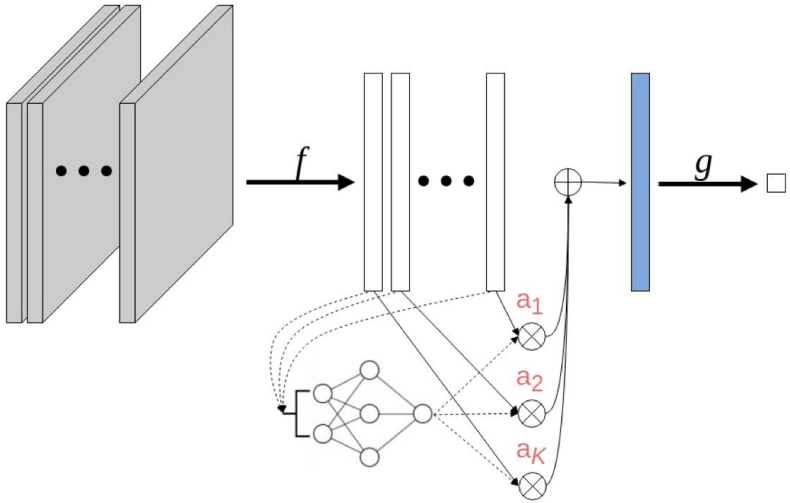


Attention-based Deep Multiple Instance Learning, Ilse et al, (2018)

Attention MIL Pooling

Replace the Max-pooling layer with a 2-layer MLP

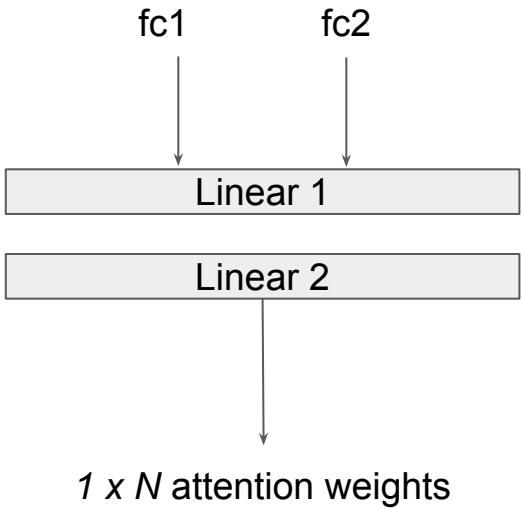
```
def forward(x):
    …
    fc1 = dense(...)(conv2)
    fc2 = dense(...)(fc1)
    out = dense(10)(fc2)

    attention = ...

    return  attention * out
```

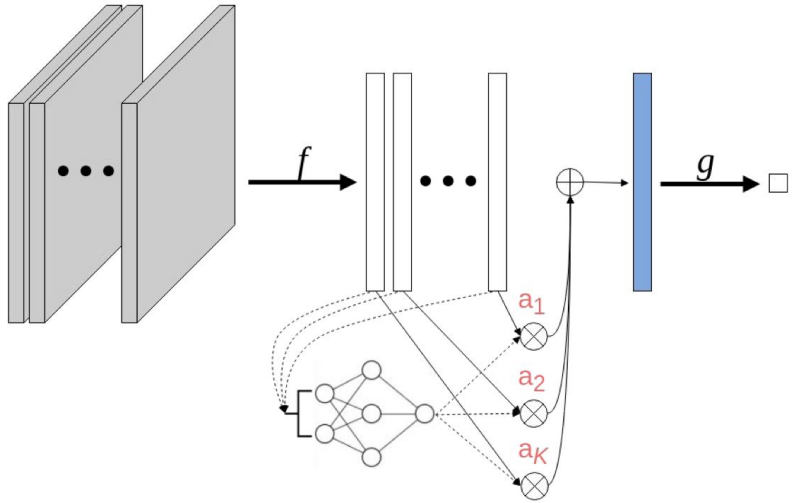


Attention-based Deep Multiple Instance Learning, Ilse et al, (2018)

Attention MIL Pooling

Replace the Max-pooling layer with a 2-layer MLP





Attention-based Deep Multiple Instance Learning, Ilse et al, (2018)

Multiple Instance Learning

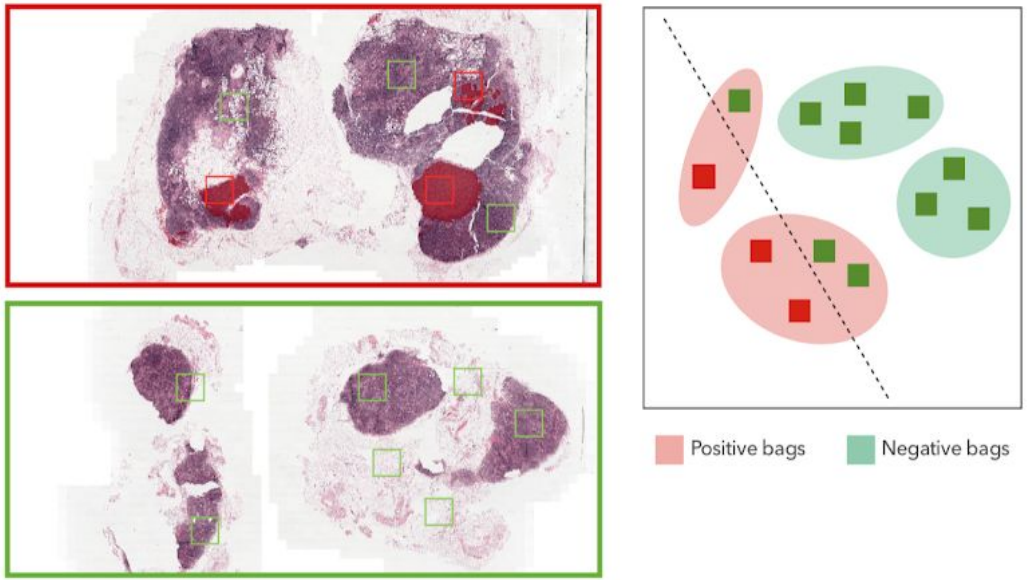
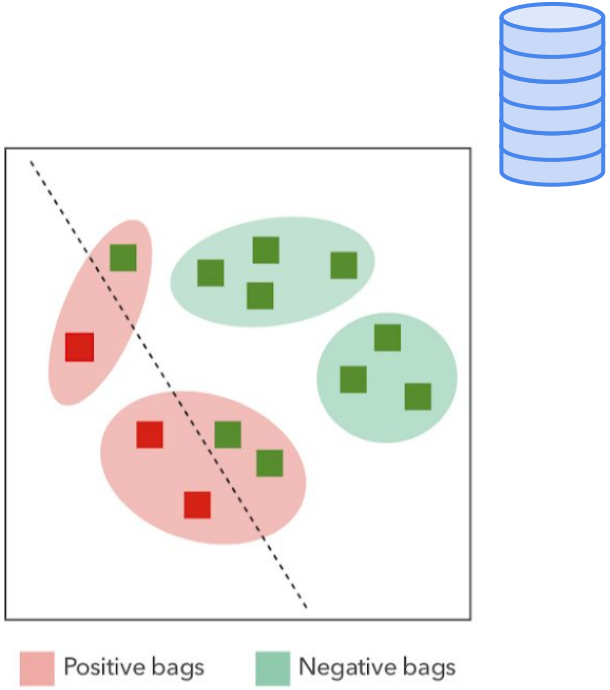
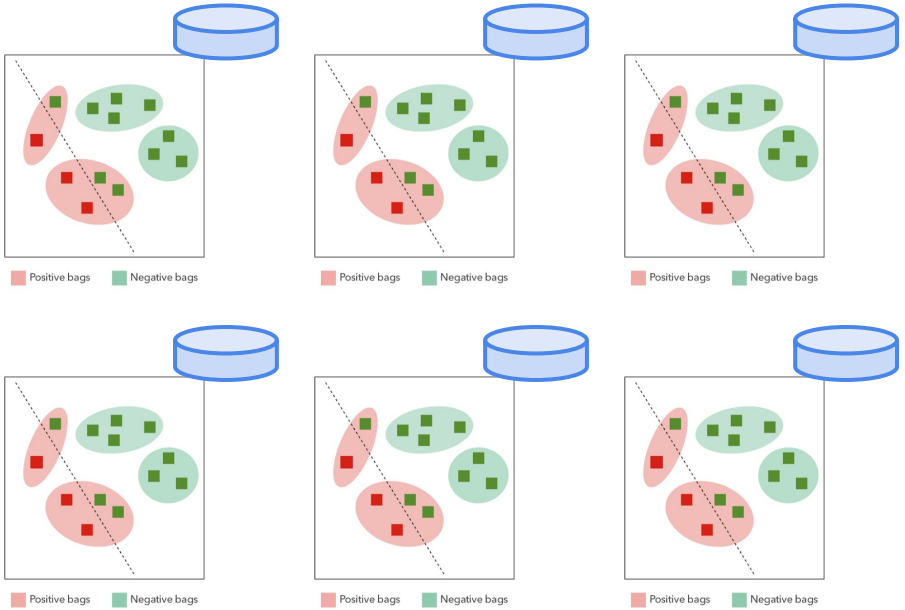


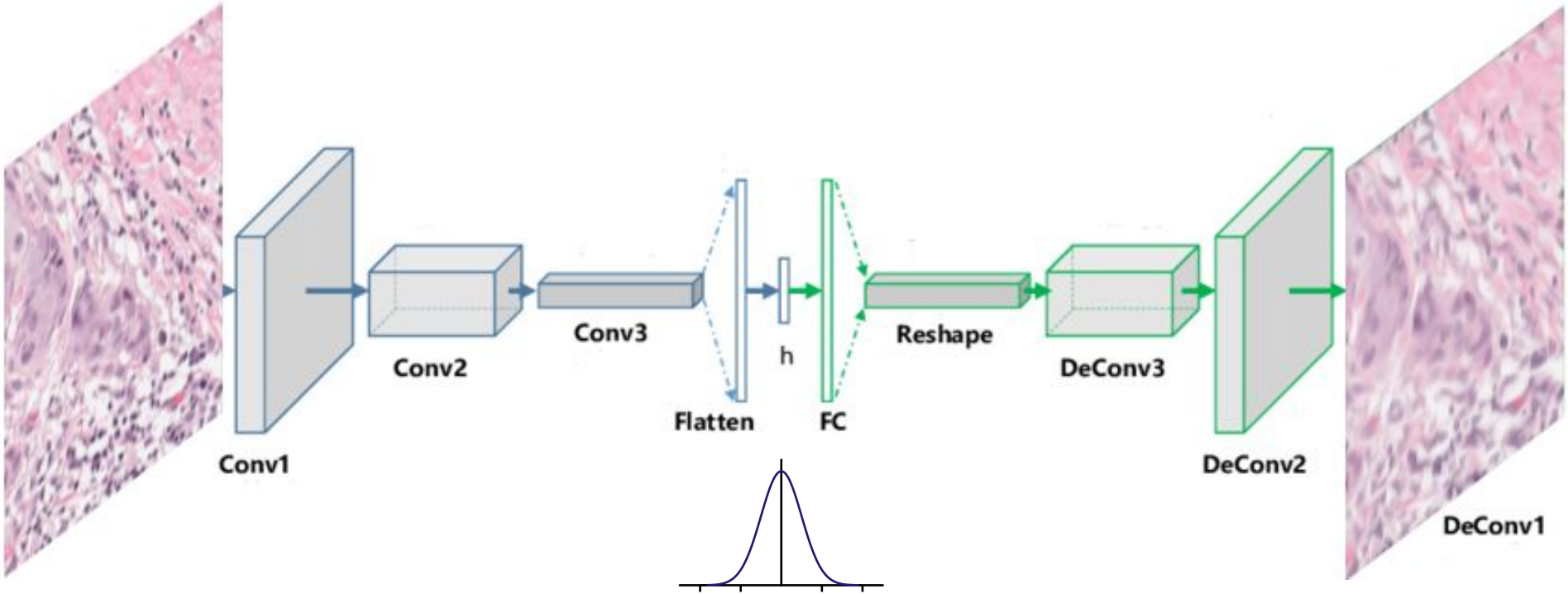
Figure 2: Illustrative example of our multiple instance learning setup where the task is to distinguish between cancerous (red) and healthy (green) patches.

S Akbar, AL Martel. “Cluster-Based Learning from Weakly Labeled Bags in Digital Pathology”, ML4H NeurIPS 2018

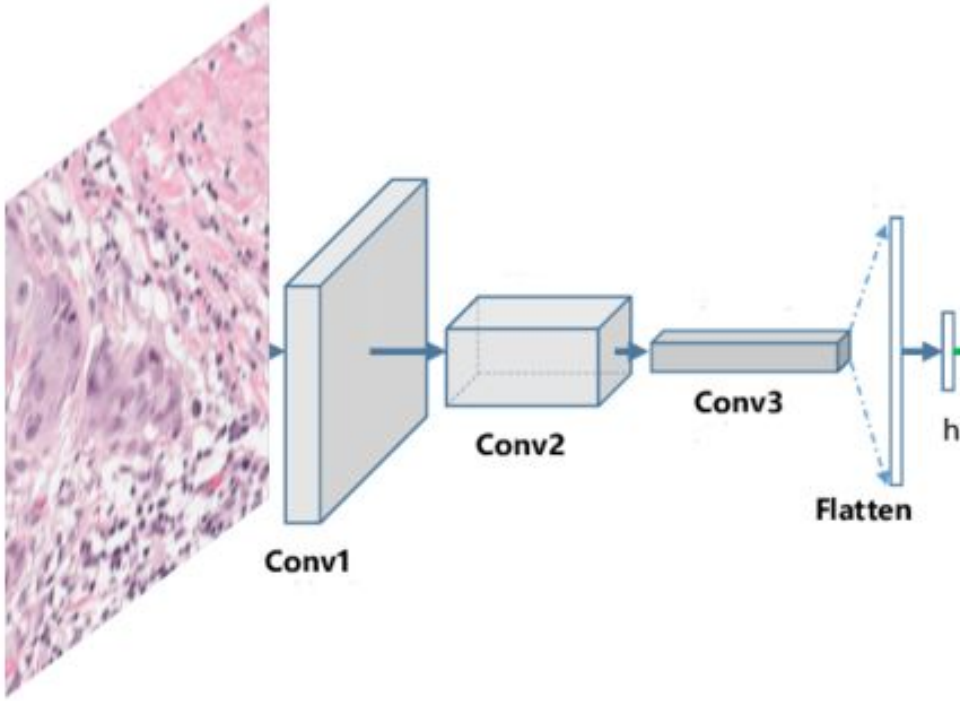

Positive bags
Negative bags


Positive bags
Negative bags
Positive bags
Negative bags
Positive bags
Negative bags
Positive bags
Negative bags
Positive bags
Negative bags
Positive bags
Negative bags

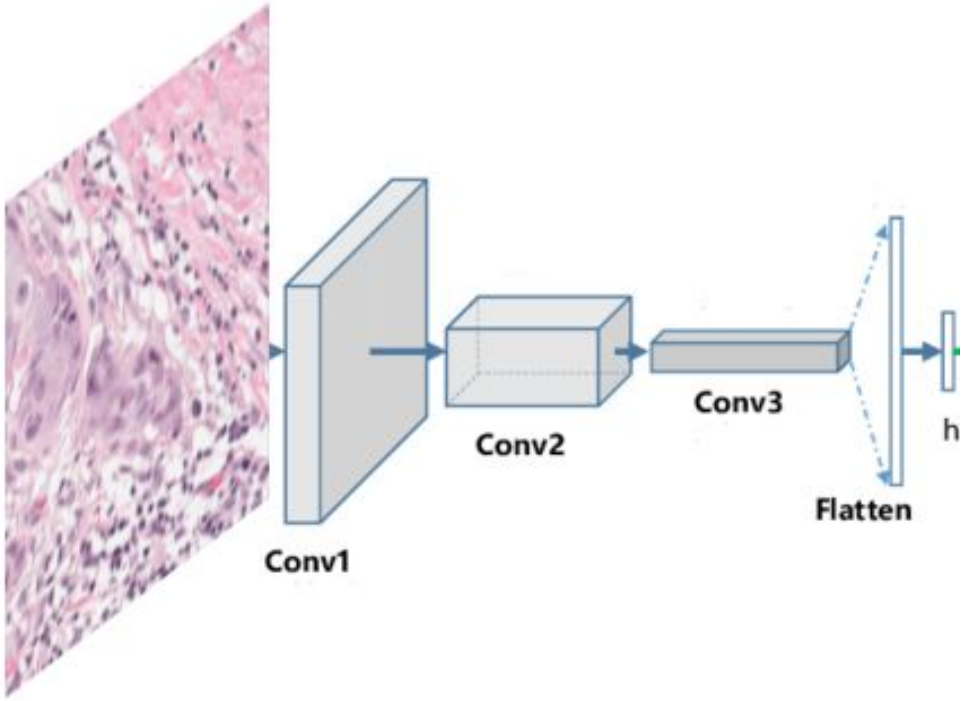
Variational autoencoder

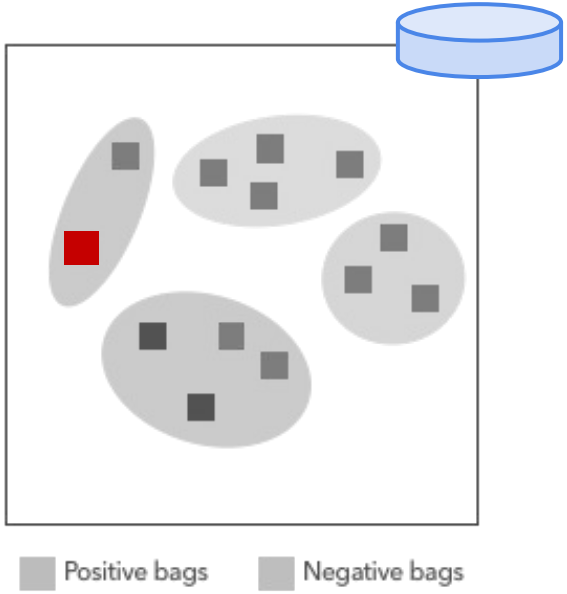


Variational autoencoder

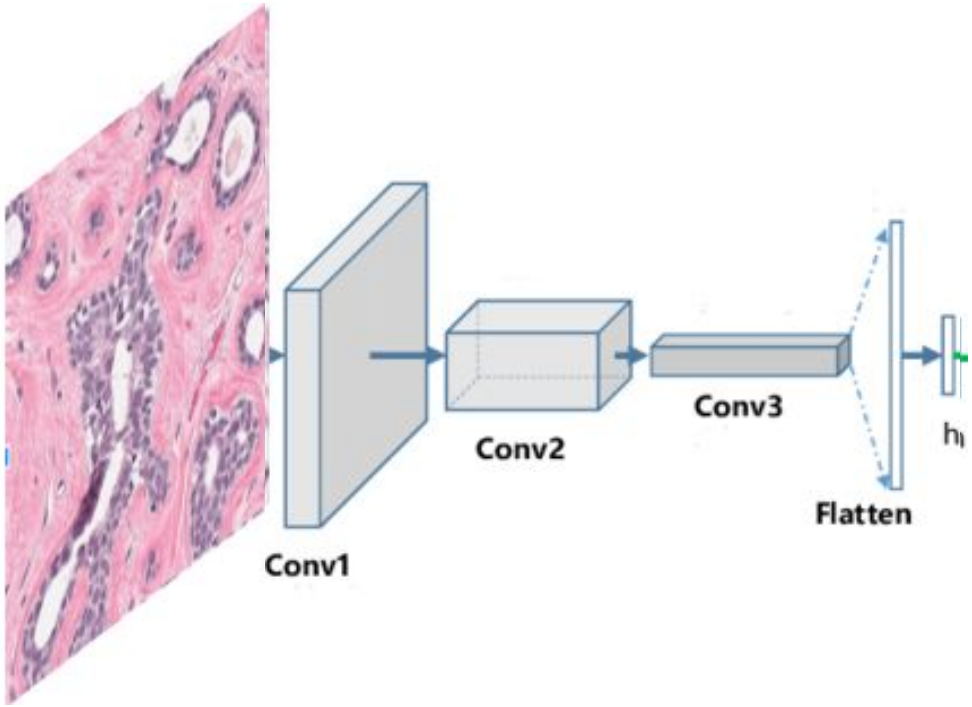


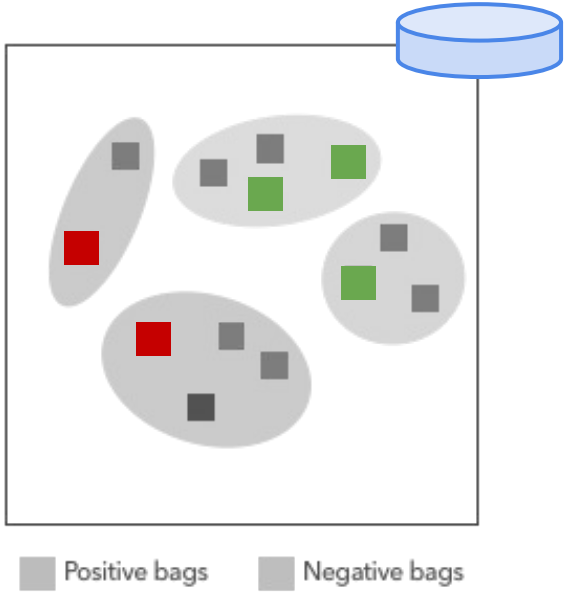
Variational autoencoder

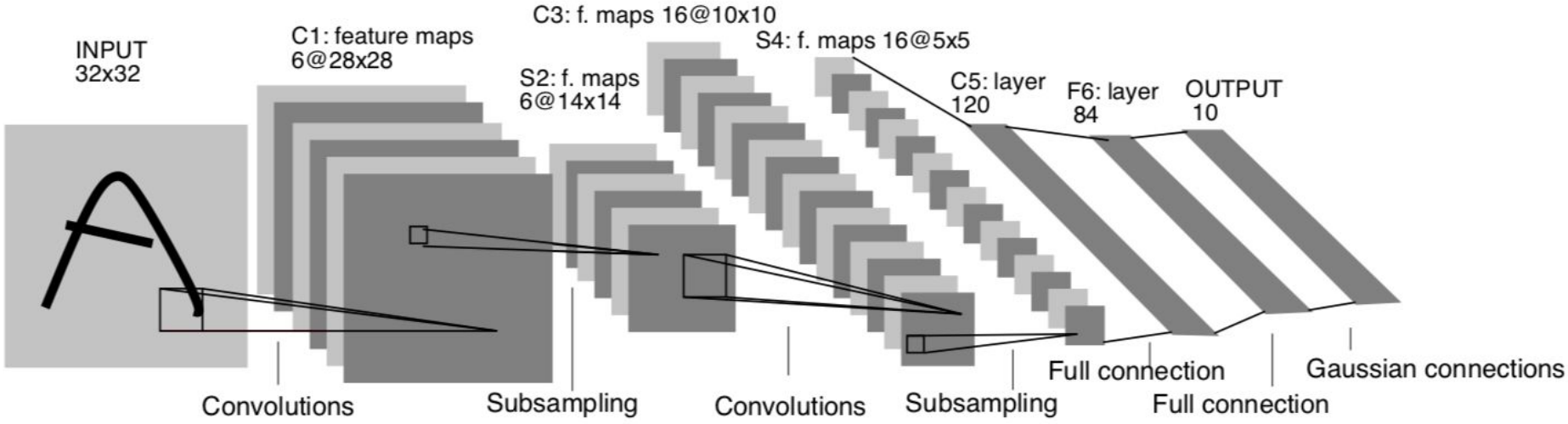




Variational autoencoder






INPUT
32x32
C1: feature maps
6@28x28
S2: f. maps
6@14x14
C3: f. maps 16@10x10
S4: f. maps 16@5x5
C5: layer
120
F6: layer
84
OUTPUT
10
Convolutions
Subsampling
Convolutions
Subsampling
Full connection
Full connection
Gaussian connections

The loss function

$$\mathrm{CCE}(y, p) = -\sum_i y_i log(p_i)$$

$$\mathcal{L} = \alpha \mathrm{CCE}(y, p) + (1 - \alpha)\mathrm{CCE}(\hat{y}, p)$$

S Akbar, AL Martel. “Cluster-Based Learning from Weakly Labeled Bags in Digital Pathology”, ML4H NeurIPS 2018

Camelyon Challenge

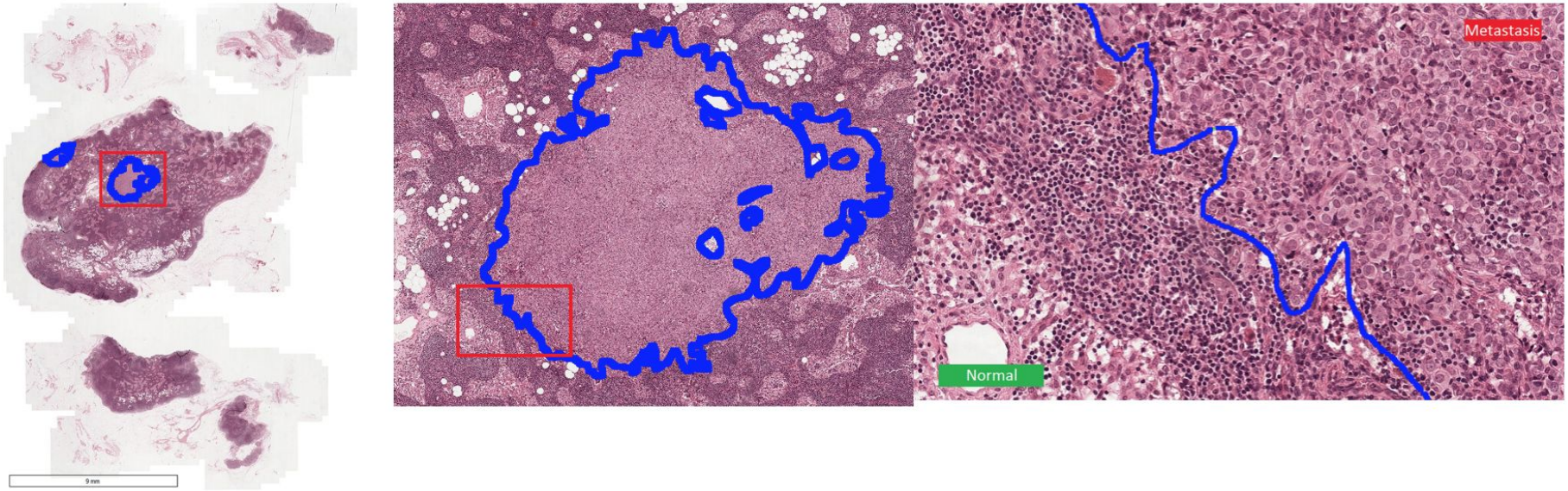
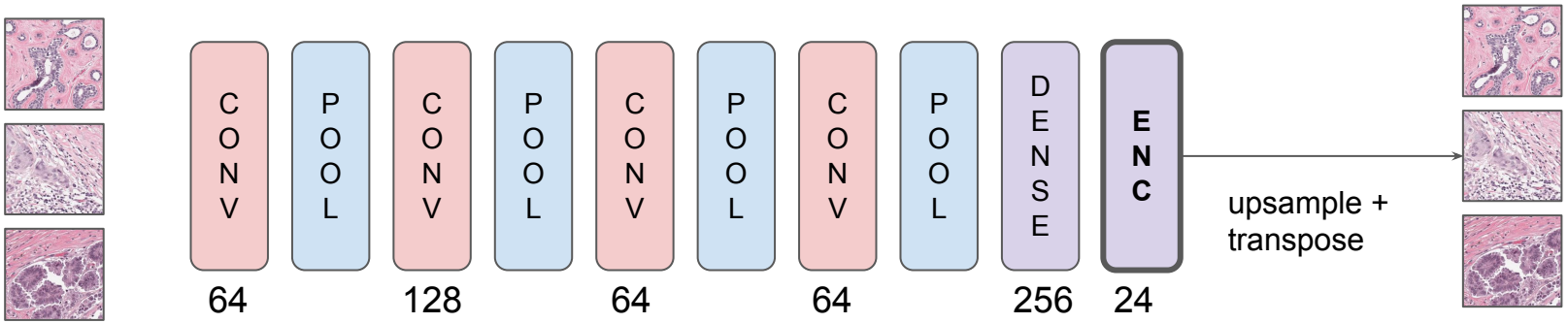


Image obtained from https://camelyon17.grand-challenge.org/

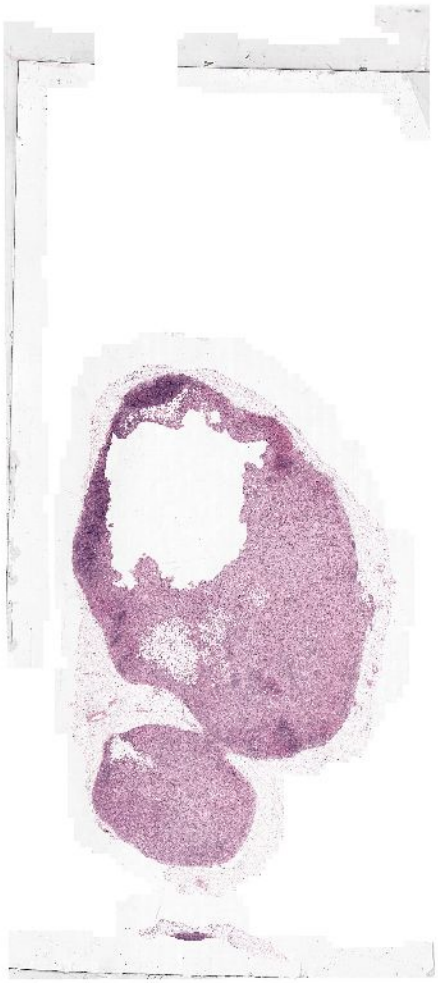
Camelyon Challenge

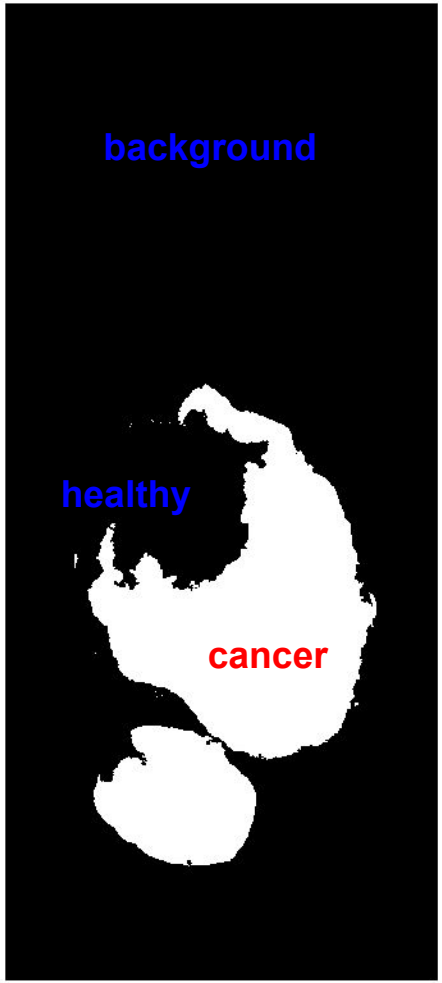
Variational autoencoder (trained with 135,000 patches):

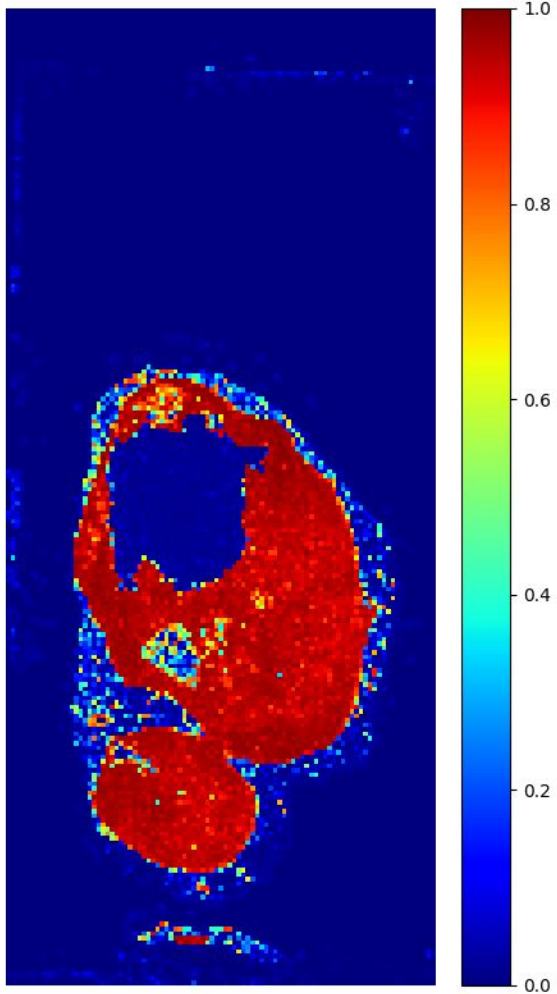


CNN: InceptionNet (Szegedy *et al*, CVPR, 2016)

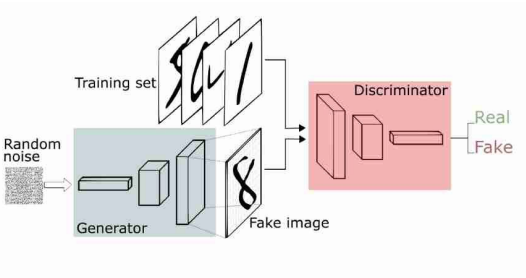
256 X 256 RGB images; 270 digital slides training, 49 digital slides testing



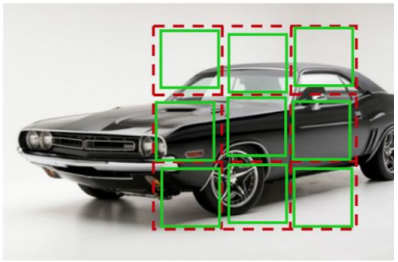

background
healthy
cancer


1.0
0.8
0.6
0.4
0.2
0.0

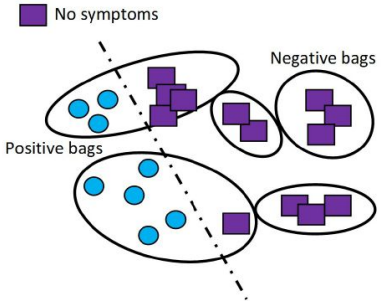
Weak supervision techniques



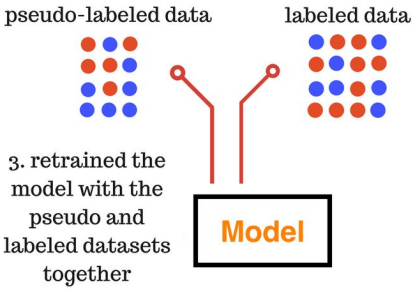
Generative Models
(data is modeled)



Self Training
(data provides the supervision)

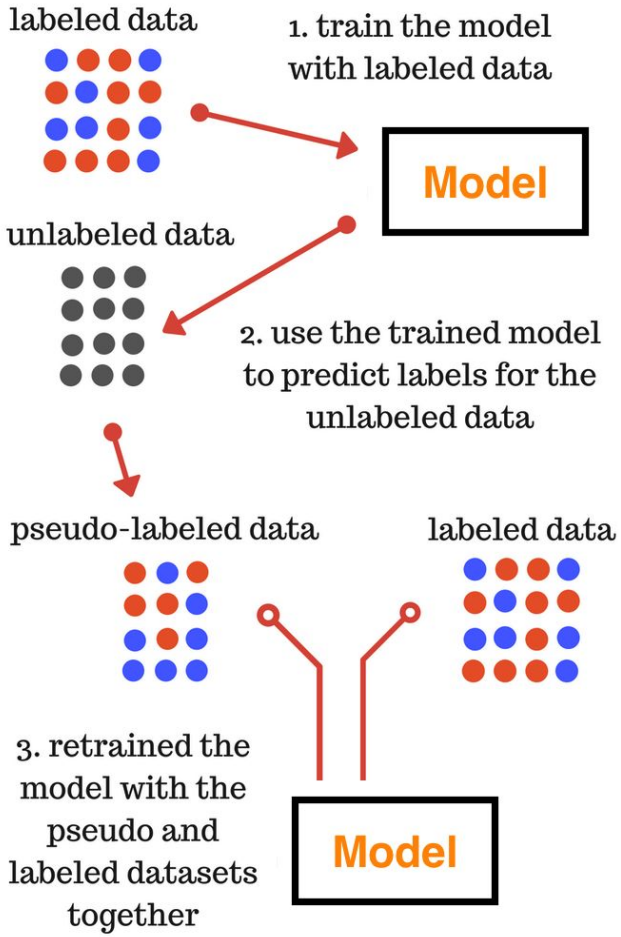


Multi-Instance Learning
(labels are coarse/noisy)



Psuedo-labeling
(labels are generated)

Psuedo-labeling:



MixUp

Create “soft psuedo-labels” for unlabeled data

- Storing softmax probabilities for every batch and every epoch
- Adapt psuedo-labels according to a record of past predictions
- Regularizer at beginning of training to limit confident in these predictions

MixUp

- Data augmentation
  - Sampling from a beta distributions
- Adds a little bit of randomization to avoid overfitting

Co-training

Using two networks (called views) trained simultaneously to agree on their predictions and disagree on their errors

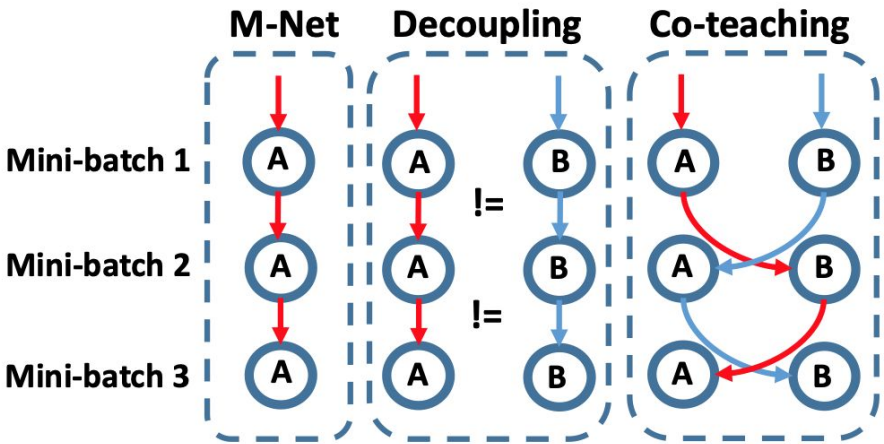
Models are complementary to one another and can help “correct” each other

Adapted from Blum and Mitchell’s work in 1998

Co-teaching

Select small loss instances and feed it into the other network

Intuition: “small loss instances more likely to be correct, therefore network resistant to noisy labels



Co-teaching: Robust Training of Deep Neural Networks with Extremely Noisy Labels, Han et al (2018)

Co-training

From 2009:

When Does Co-Training Work in Real Data?

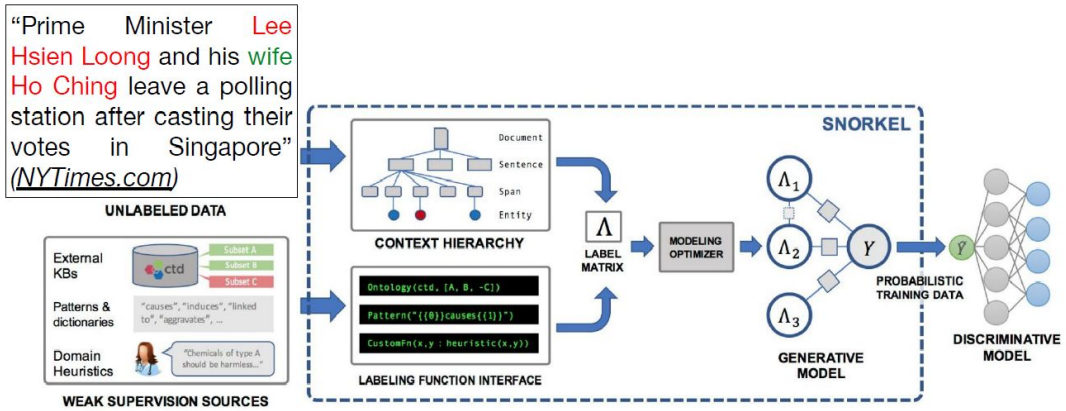
Jun Du, Charles X. Ling, *Senior Member, IEEE,* and Zhi-Hua Zhou, *Senior Member, IEEE*

"...given only limited training examples, even the most effective splitting method (random-restart hill climbing) still cannot successfully reconstruct the true two views."

Snorkel

Defines labeling to functions

Given unlabeled data as well as “labeling functions”, we can programmatically build training data



https://www.snorkel.org

Snorkel

Keyword matches:

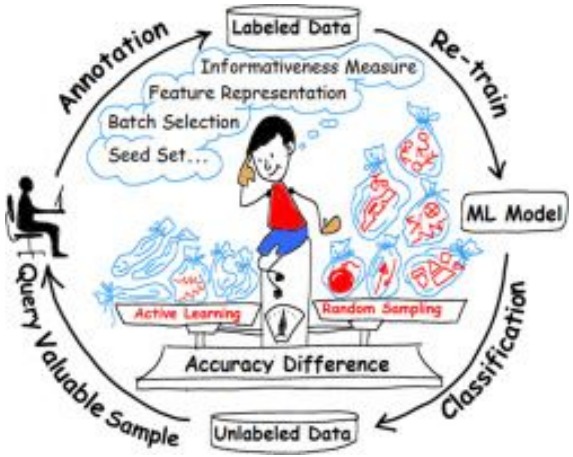
```
from snorkel.labeling import labeling_function


@labeling_function()
def lf_keyword_my(x):
    """Many spam comments talk about 'my channel', 'my video', etc."""
    return SPAM if "my" in x.text.lower() else ABSTAIN
```

Regular expressions:

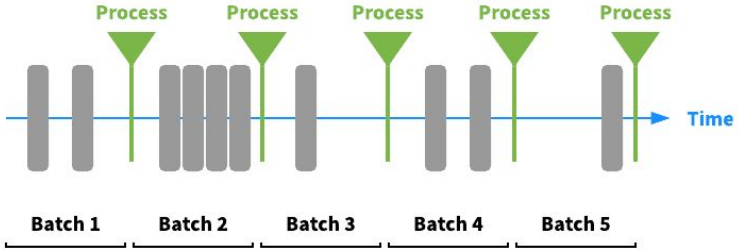
```
import re


@labeling_function()
def lf_regex_check_out(x):
    """Spam comments say 'check out my video', 'check it out', etc."""
    return SPAM if re.search(r"check.*out", x.text, flags=re.I) else ABSTAIN
```

https://www.snorkel.org/get-started/

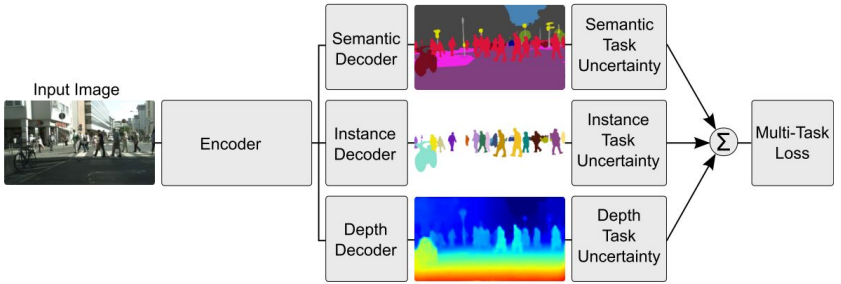
Other learning techniques



Active Learning



Online Learning



Multi-task Learning

Self Training

Pros	Cons
Simple to understand No prior knowledge required “Free” additional knowledge	Mistakes can re-enforce themselves

Multi Instance Learning

Pros	Cons
Useful in many scenarios when labelling is noisy and/or costly	Aggregating instance-level predictions is task-dependent
Appropriate for non-imaging data	Additional time needed to reformulate problem
Useful when some knowledge is necessary	

Psuedo-labeling

Pros	Cons
Useful if you have some very accurate labels	Error in labeled subset propagate
Learn structure of data	Unclear whether it always improves performance
Ability to grow training set	

Extra Material

Pro and Cons

Autoencoders

✓ Simple ✓ Stack multiple layers ✓ Intuitive	✗ Each layer is trained greedy ✗ No global optimization ✗ Reconstruction may be the ideal metric for learning

Generative Models (e.g. VAE)

✓ Global training ✓ Learning meaningful representation of data ✓ Better performance that AE	✗ Hard to train: conversion problem ✗ More computationally expensive than AE ✗ Slightly more parameters to learn, increasing complexity

Deep Clustering

✓ Both benefits of clustering and reconstruction ✓ Deep disentangled features ✓ Visualizable	✗ Complex implementation ✗ Initialization is poor (often need pretrained models) ✗ Need to choose k

Extra Material

SemiSL Important Links

Reading Material

Unsupervised Learning (in general):

- Weak Supervision: The New Programming Paradigm for Machine Learning
- UFLDL Tutorial: Some of the basics and background before diving into deep learning

Autoencoders, Variational Autoencoders:

- Extracting and Composing Robust Features with Denoising Autoencoders
- Overview of all the material I will be covering in note form: https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html
- Generating Sentences from a Continuous Space: An example of VAEs in NLP

Clustering

- A Survey of Clustering With Deep Learning: From the Perspective of Network Architecture: a great overview of the field!
- Variational Deep Embedding: An Unsupervised and Generative Approach to Clustering
- Unsupervised Deep Embedding for Clustering Analysis
- Joint Unsupervised Learning of Deep Representations and Image Clusters: some earlier work in this field (2016)

Thank You!