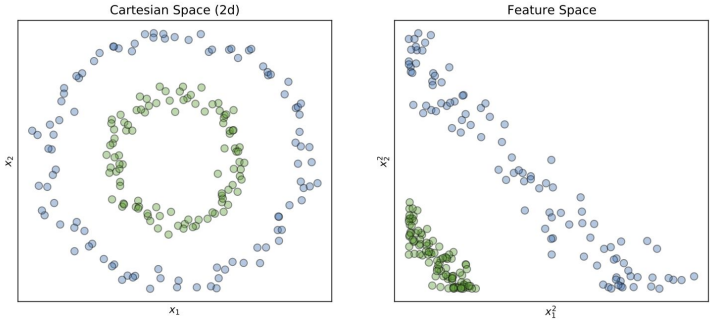
ICME Summer Workshops 2021

Intermediate Topics in Machine Learning and Deep Learning



Session 3.1: Representation Learning

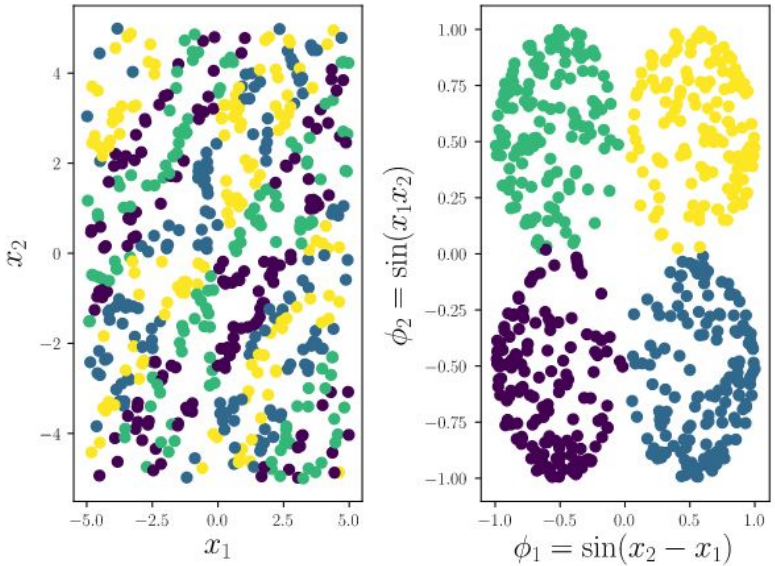
Tuesday, August 17, 9:30—11:00 AM

Instructor: Sherrie Wang

icme-workshops.github.io/intermediate-ml

What is representation learning?

Representation learning is a broad concept in machine learning that refers to automatically discovering representations, or features, from raw data.



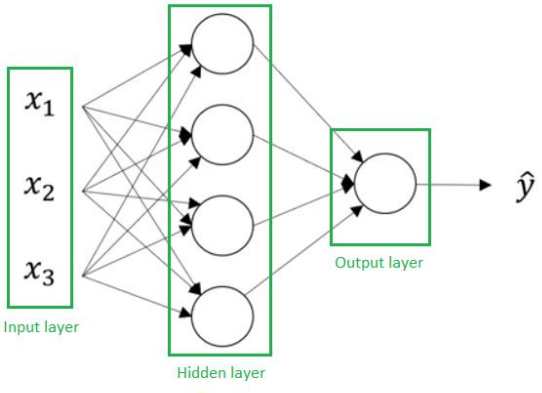
http://williamlacava.com/research/symbolic-representation-learning/

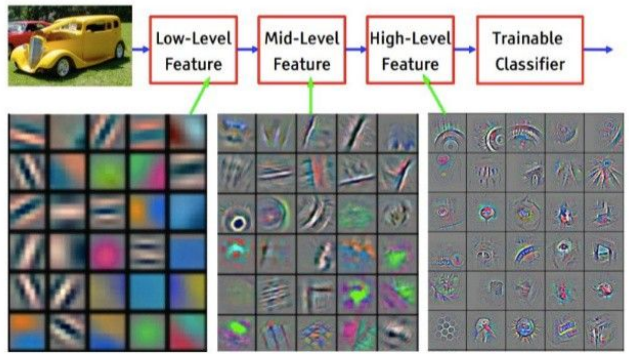
Another term for representation is **feature**, and you will hear the term *feature space* or *embedding space* used to describe the *k*-dimensional space of learned representations.

What is representation learning?

Representation learning can be supervised or unsupervised.

- **Supervised** example: neural networks learn a representation before the last layer of the network (the classifier) that will result in good performance
- **Unsupervised** examples: PCA, autoencoder

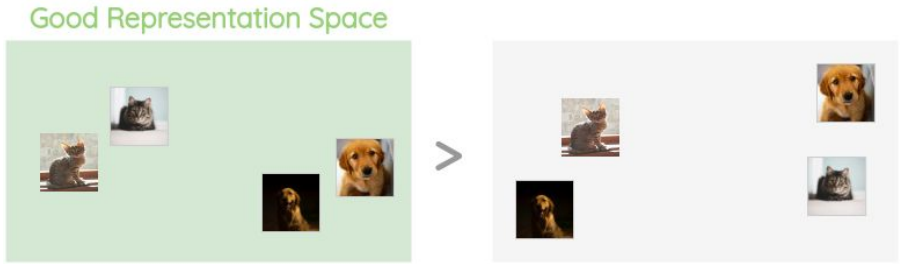




Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

What makes a representation *good*?

1. It makes subsequent (“downstream”) tasks easier.
2. It teases apart the factors of variation in the data into dependent components.
3. It is interpretable.



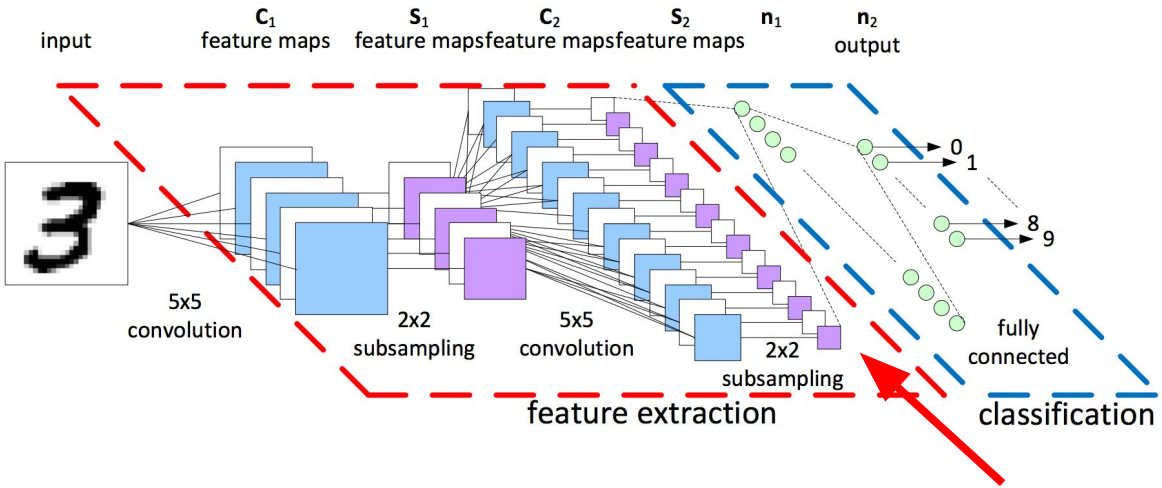
These objectives can sometimes align, but often they compete with each other.

This is related to interpretability, but extends beyond that, since features that are “disentangled” can be manipulated more easily.

Feature extraction from neural networks

Early layers of a neural network can be viewed as a feature extractor.

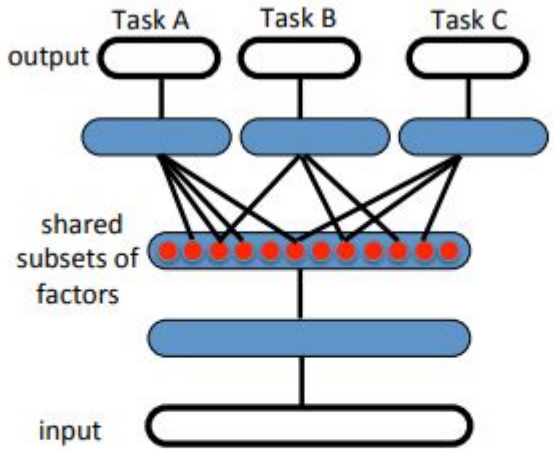
If we're interested in the features themselves, we can take the output of the layer before the classifier for each sample of the data as an embedding of that sample.



Transfer learning and representation learning

Transfer learning can be viewed through the lens of representation learning.

- Pre-training = use the representations learned in a pre-trained model to perform well on a target task.
- Domain adaptation = aligning the representations of the source and target domains to enable a model trained on the source task to transfer to the target task.
- Multi-task learning = share representations across multiple tasks.



Unsupervised representation learning

Given a task and enough labels, supervised learning can learn appropriate representations and solve the task well.

As we saw previously, obtaining labels can be expensive and difficult to scale.

Often, we have a lot more unlabeled data available than labeled data. Can we use them in some way?

ImageNet has 14 million labeled images, but there are 10+ billion images on Google images.

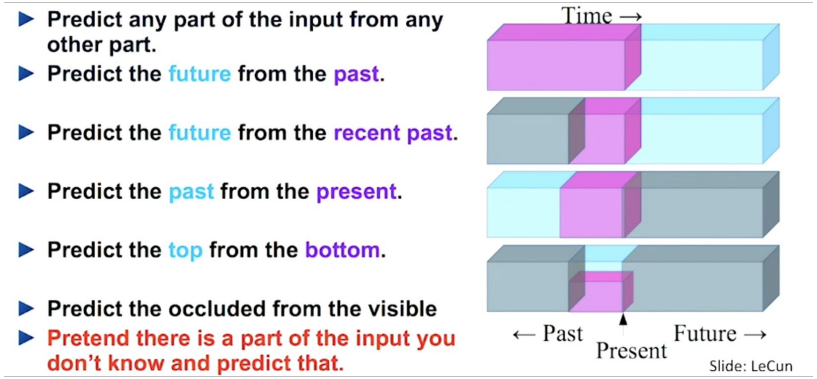
Self-supervised representation learning

What if we can get labels for free for unlabeled data and train on this synthetic task to learn representations?

Achieve this by having the task be: predict part of the input using the rest.

This is **self-supervised learning**.

This strategy is used a lot with language data.



https://lilianweng.github.io/lil-log/2019/11/10/self-supervised-learning.html

Self-supervised representation learning

The self-supervised task is called the *pretext task*.

We don’t actually care about the pretext task itself.

However, with these features learned through self-supervision, we hope to do well on a downstream task with fewer labels.

https://amitness.com/knowledge-transfer/

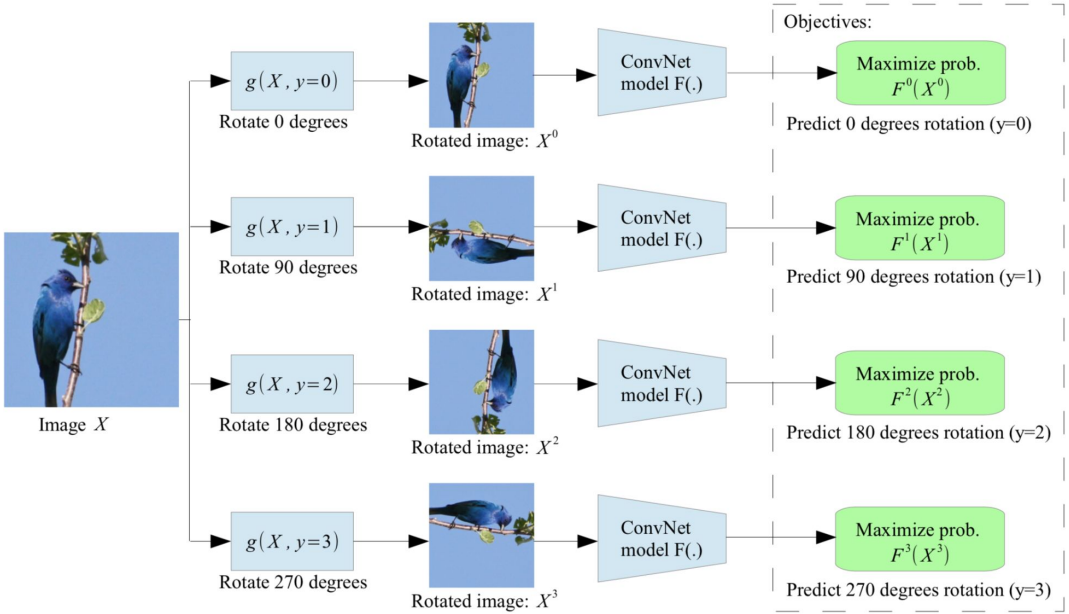
Self-supervision for image data

Common workflow:

- Train a model on one or more pretext tasks with unlabeled images.
- Use an intermediate feature layer of this model as input to a logistic regression classifier on ImageNet classification.
- The classification accuracy quantifies how good the representation is.

Pretext tasks for images

Predict rotation of an entire image



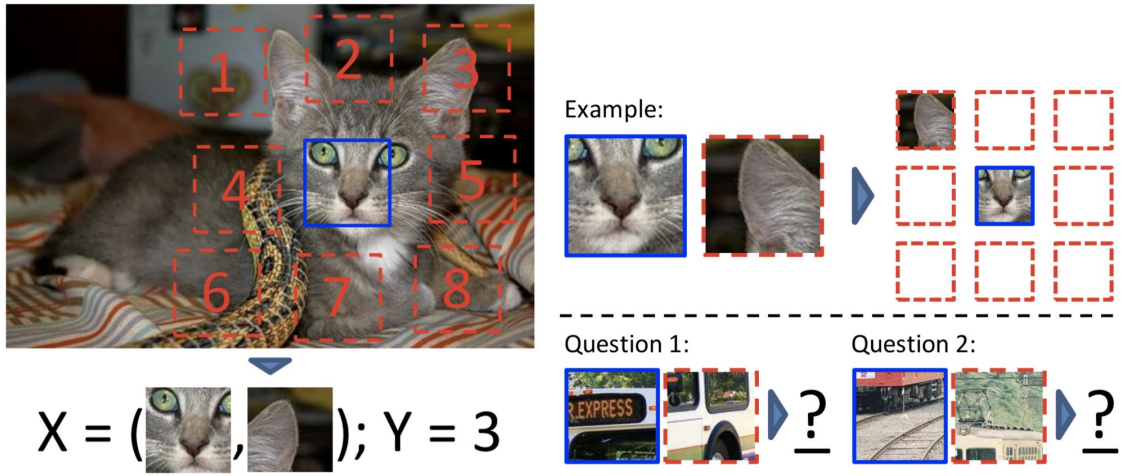
To perform the pretext task, the model has to learn high level object parts like heads, noses, eyes, and the relative positions of those parts.

Method	Conv4	Conv5
ImageNet labels from (Bojanowski & Joulin, 2017)	59.7	59.7
Random from (Noroozi & Favaro, 2016)	27.1	12.0
Tracking Wang & Gupta (2015)	38.8	29.8
Context (Doersch et al., 2015)	45.6	30.4
Colorization (Zhang et al., 2016a)	40.7	35.2
Jigsaw Puzzles (Noroozi & Favaro, 2016)	45.3	34.6
BIGAN (Donahue et al., 2016)	41.9	32.2
NAT (Bojanowski & Joulin, 2017)	-	36.0
(Ours) RotNet	50.0	43.8

Gidaris *et al.* (2018) "Unsupervised Representation Learning by Predicting Image Rotations"

Pretext tasks for images

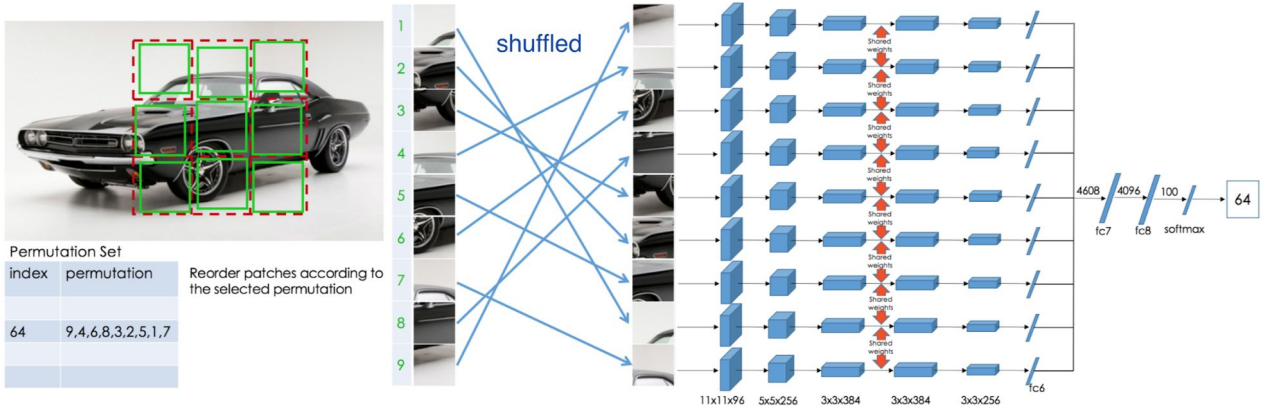
Sample random patches of an image and ask the model to figure out the relative position between the patches.



Doersch *et al*. (2015) “Unsupervised Visual Representation Learning by Context Prediction”

Pretext tasks for images

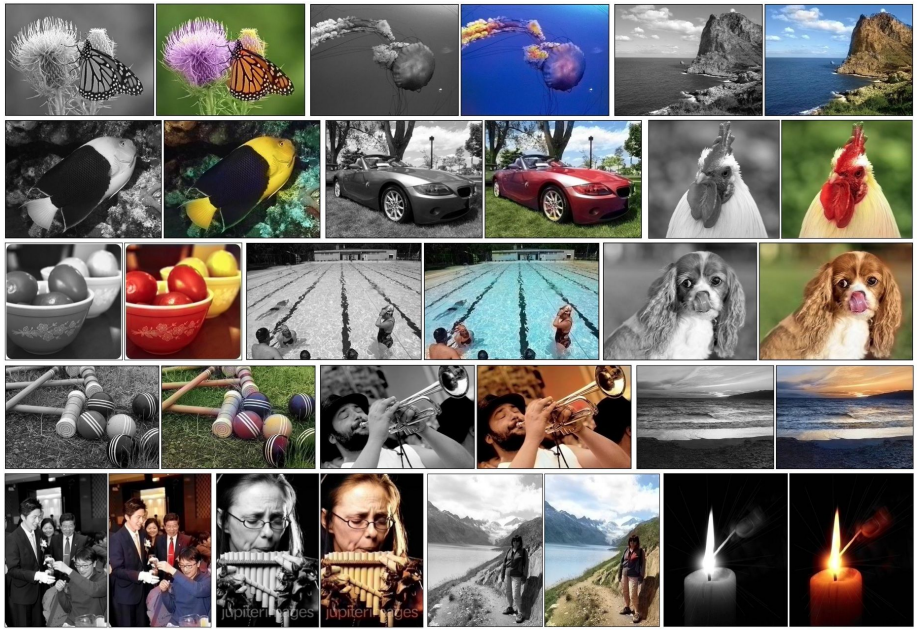
Sample random patches of an image and create a “jigsaw puzzle” for the model to solve.



Noroozi and Favaro (2016) “Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles”

Pretext tasks for images

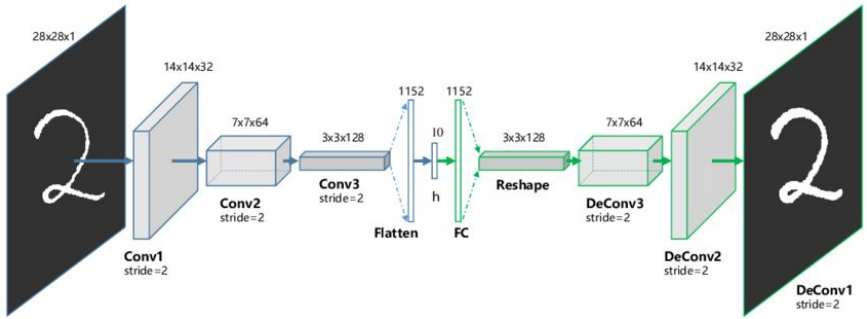
Colorization: predict the color version of the image from the grayscale version.

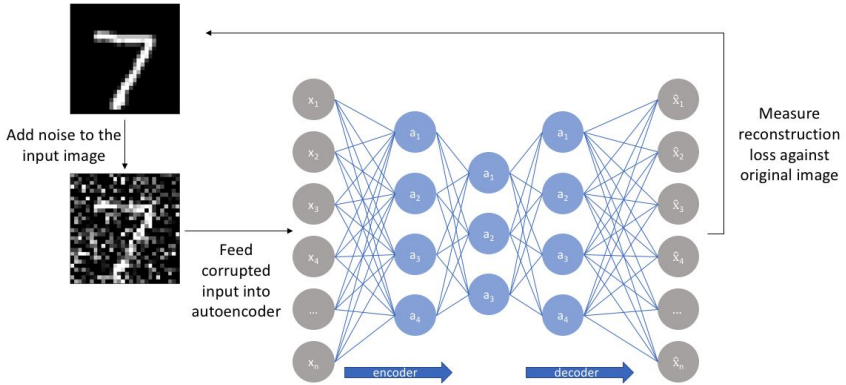


Zhang *et al*. (2016) “Colorful Image Colorization”

Pretext tasks for images

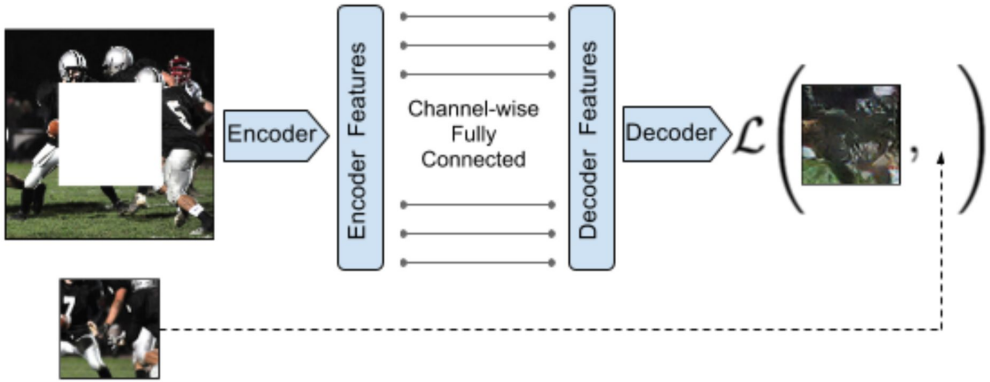
Generative modeling: reconstruct the original input while learning a good latent representation.





Pretext tasks for images

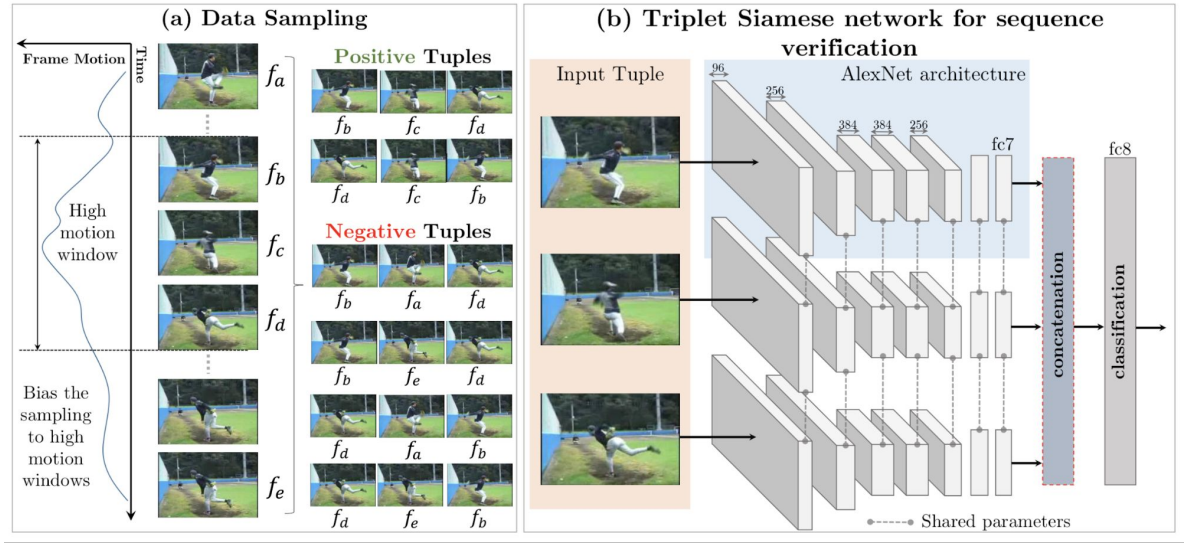
Inpainting: fill in a missing piece in the image (“context encoder”)



Pathak *et al.* (2016) “Context Encoders: Feature Learning by Inpainting”

Pretext task for video

Determine whether a sequence of frames from a video is placed in the correct temporal order.



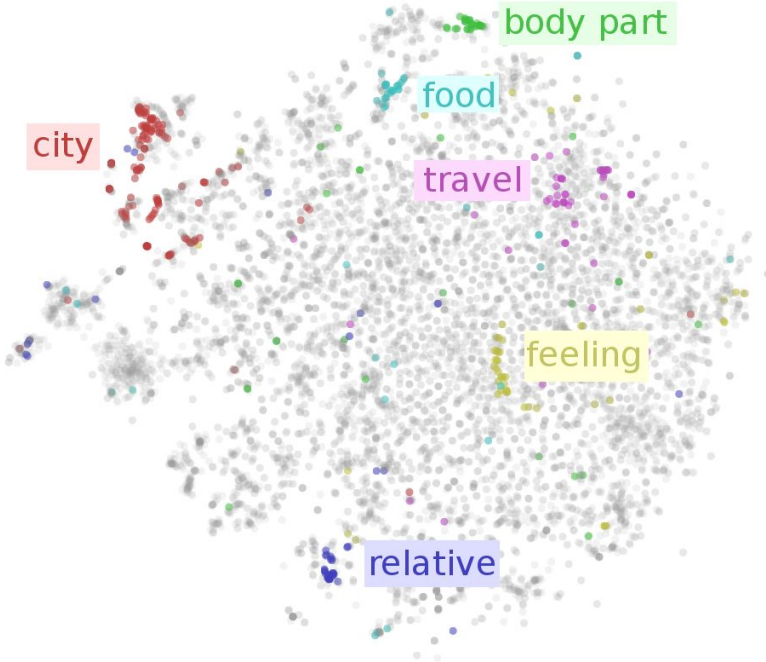
Misra *et al.* (2016) "Shuffle and Learn: Unsupervised Learning using Temporal Order Verification"

Learning word embeddings

Unlike image pixels, words are not already represented by a list of numbers.

How do we give each word a representation?

Self-supervision using a corpus (a collection of written texts).



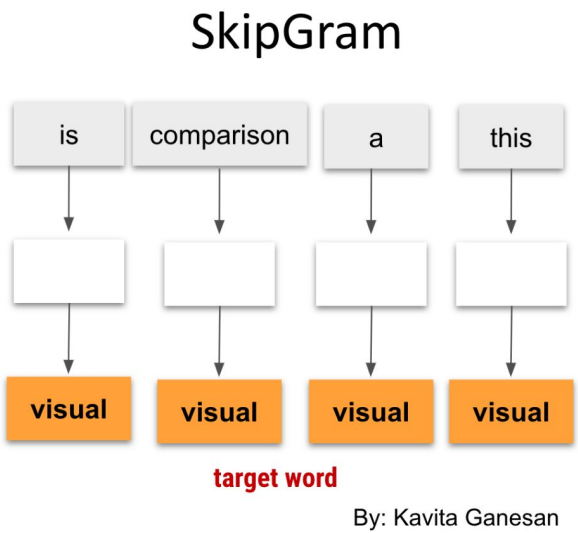
Word2vec

One of the most popular methods for learning word embeddings.

Suppose we have the sentence:

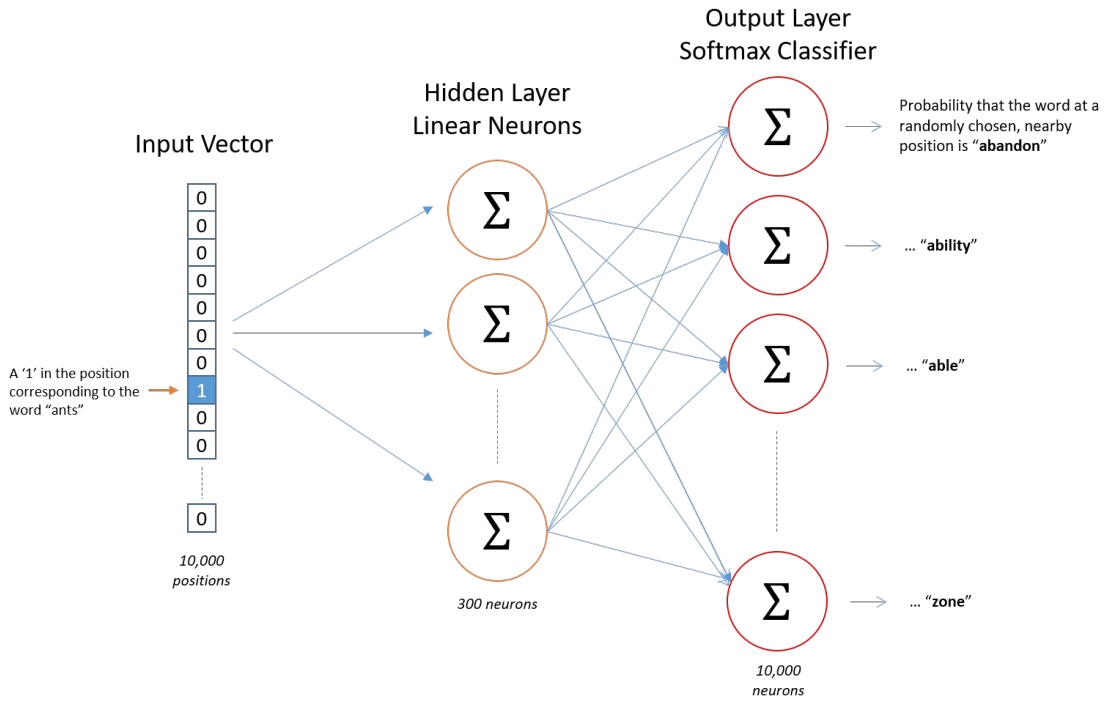
This is a visual comparison.

Using a skip-gram model, we want the neural network to predict a target word (“visual”) using words to its left and right (context words, “is”, “a”, “comparison”).



Word2vec

The neural network's input is a one-hot vector representing the input word, and the label is a one-hot vector representing the target word. The network's output is a probability distribution over all words in the corpus.

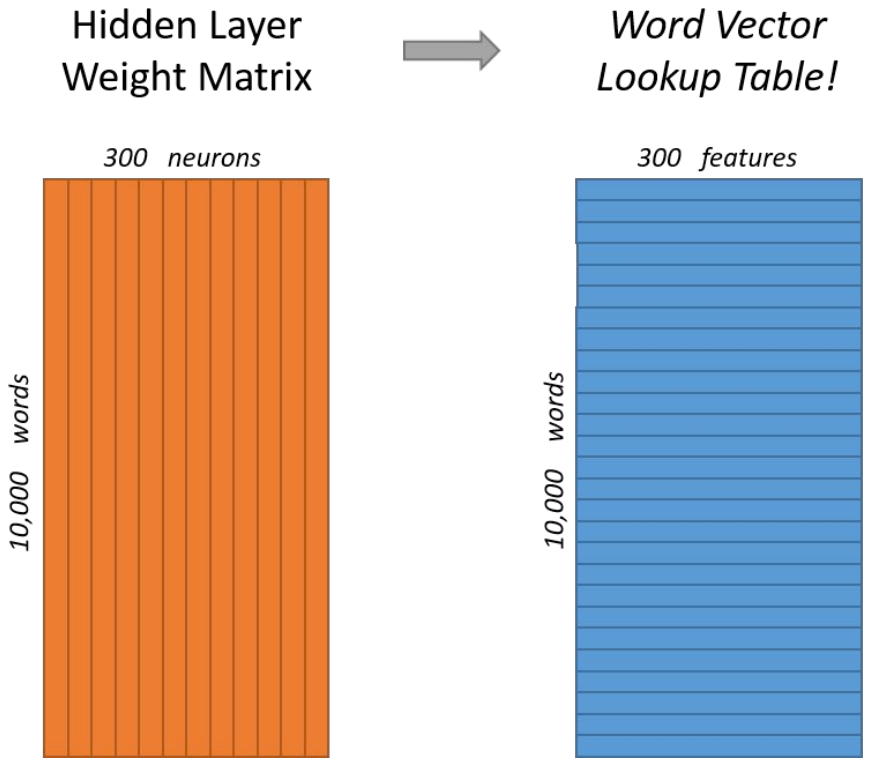


https://israelg99.github.io/2017-03-23-Word2Vec-Explained/

Word2vec

Word2vec is a shallow 2-layer neural network. The hidden layer's weight matrix can therefore be interpreted as a lookup table for the word vectors.

$$[0 \quad 0 \quad 0 \quad 1 \quad 0] \times \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix} = [10 \quad 12 \quad 19]$$

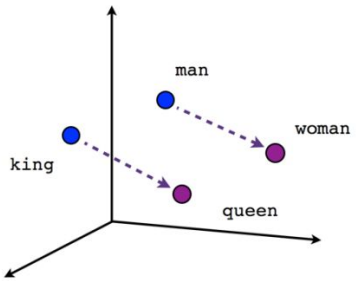


https://israelg99.github.io/2017-03-23-Word2Vec-Explained/

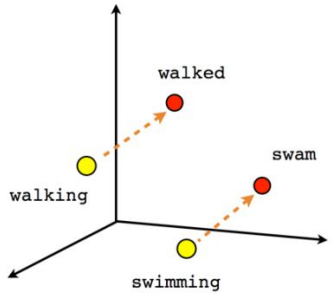
Word2vec

The cool part: this simple algorithm captures the relationship between words in the embedding space!

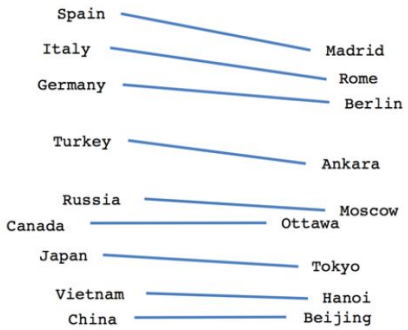
For example: “Brother” - “Man” + “Woman” = “Sister” ← closest vector



Male-Female



Verb tense



Country-Capital

Mikolov *et al.* (2013) “Efficient Estimation of Word Representations in Vector Space”

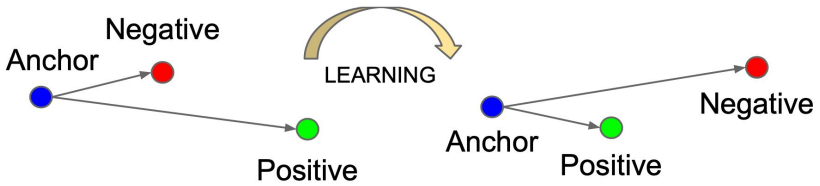
Contrastive learning

Learn an embedding space in which similar samples stay close to each other while dissimilar ones are far apart.

Can be supervised or unsupervised. (Unsupervised is part of self-supervised learning.)

In the supervised case, uses **contrastive loss**.

$$\mathcal{L}_{\text{cont}}(\mathbf{x}_i, \mathbf{x}_j, \theta) = \mathbb{1}[y_i = y_j] \|f_\theta(\mathbf{x}_i) - f_\theta(\mathbf{x}_j)\|_2^2 + \mathbb{1}[y_i \neq y_j] \max(0, \epsilon - \|f_\theta(\mathbf{x}_i) - f_\theta(\mathbf{x}_j)\|_2)^2$$



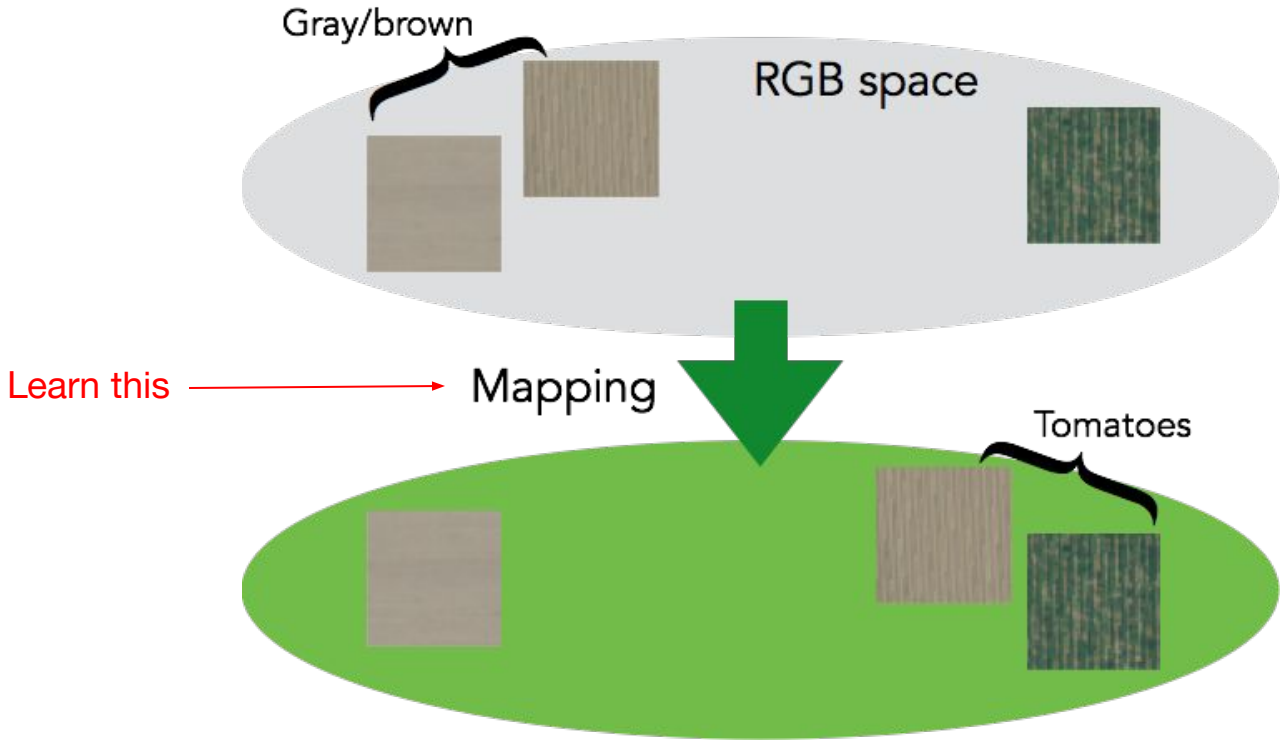
https://lilianweng.github.io/lil-log/2021/05/31/contrastive-representation-learning.html

Tile2Vec: Unsupervised representation learning for spatially distributed data

- Satellite imagery often exhibits high within-class variance in RGB / spectral space
- Can we use the structure inherent in Earth observation to learn a better feature space?

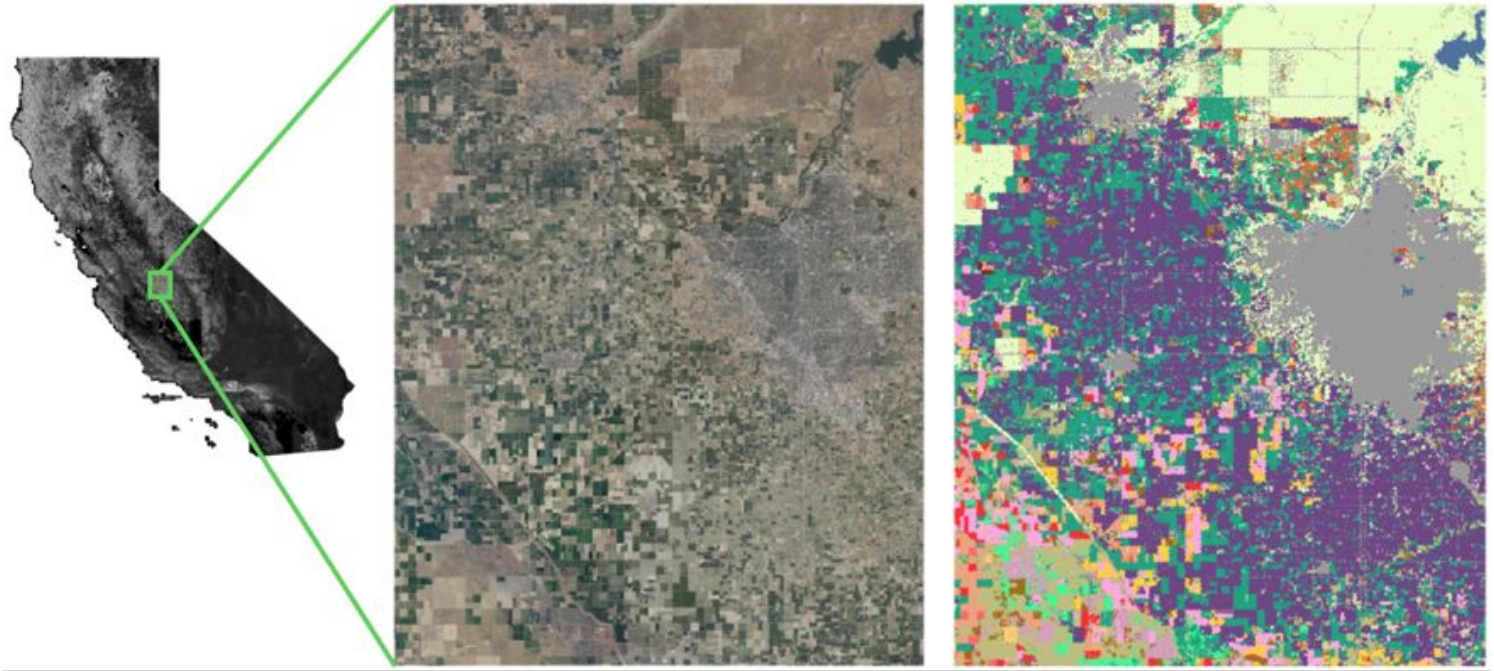
Jean, Wang, et al. Tile2vec: Unsupervised representation learning for spatially distributed data. *AAAI* (2019).

Learning a better feature space



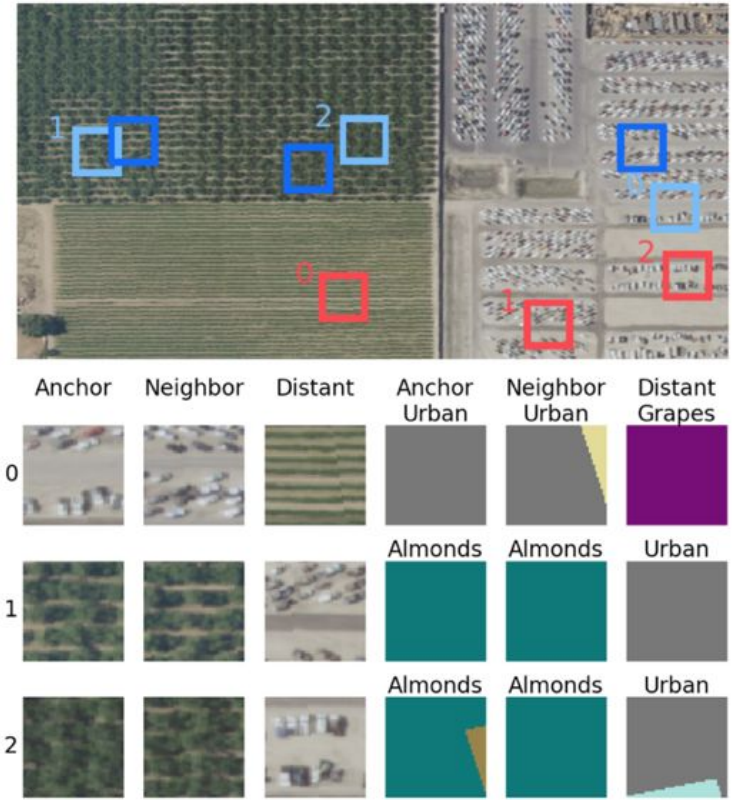
Jean, Wang, et al. Tile2vec: Unsupervised representation learning for spatially distributed data. *AAAI* (2019).

Study area and land cover dataset



Jean, Wang, et al. Tile2vec: Unsupervised representation learning for spatially distributed data. *AAAI* (2019).

Algorithm: Tile2Vec triplet sampling



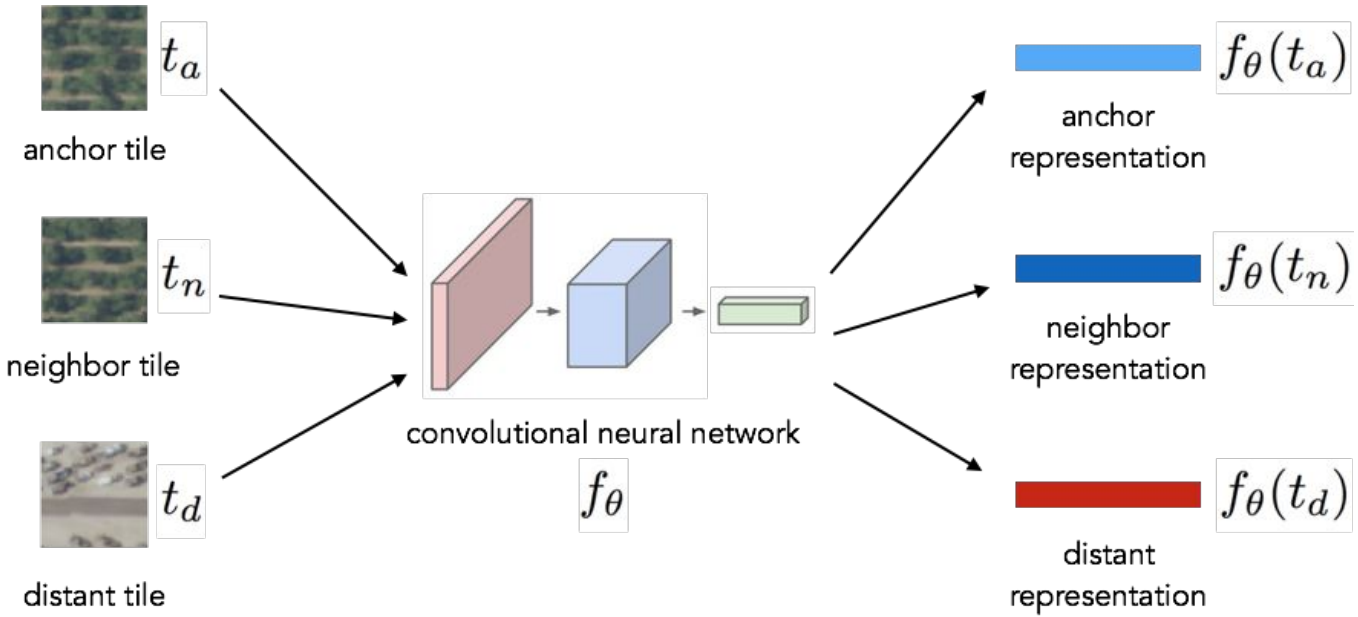
Tobler's First Law of Geography:

"Everything is related to everything else,

but near things are more related than distant things."

Jean, Wang, et al. Tile2vec: Unsupervised representation learning for spatially distributed data. *AAAI* (2019).

Algorithm: Tile2Vec model training

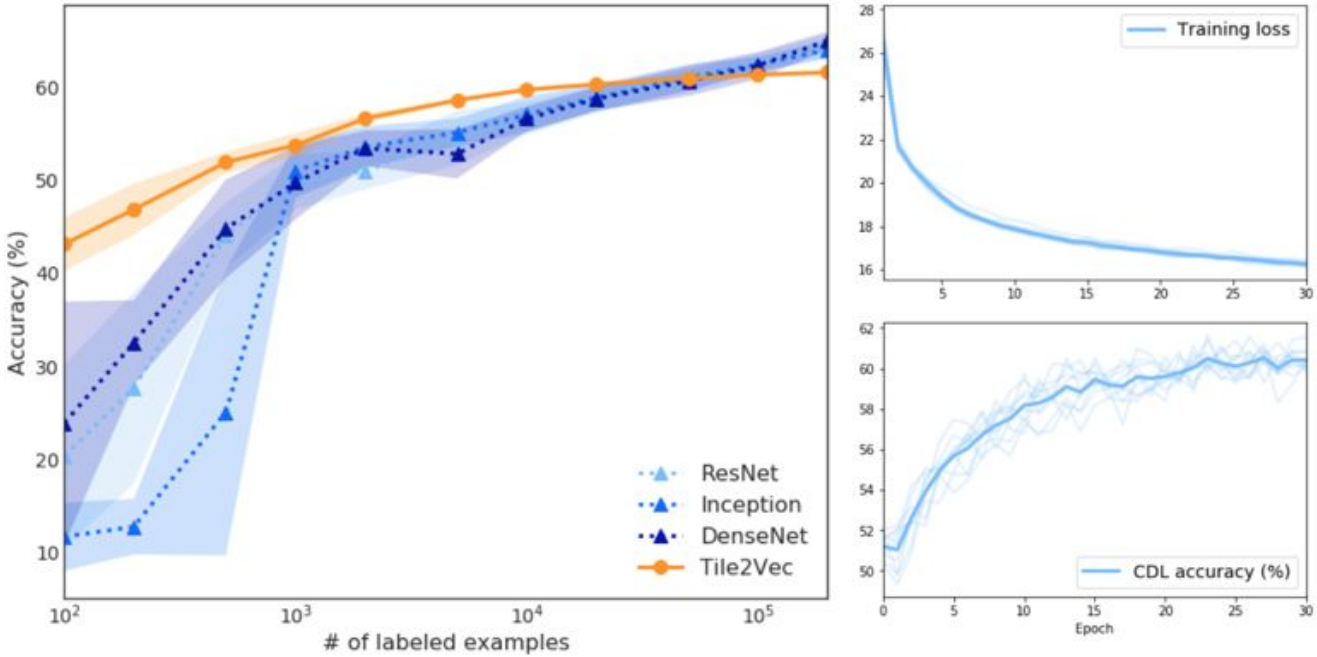


$$L(t_a, t_n, t_d) = [||f_\theta(t_a) - f_\theta(t_n)||_2 - ||f_\theta(t_a) - f_\theta(t_d)||_2 + m]_+$$

Jean, Wang, et al. Tile2vec: Unsupervised representation learning for spatially distributed data. *AAAI* (2019).

Comparison with supervised, end-to-end training

- When Tile2Vec features are used with a logistic regression classifier, results outperform supervised CNNs trained directly on labels up to 50k labeled samples



Jean, Wang, et al. Tile2vec: Unsupervised representation learning for spatially distributed data. *AAAI* (2019).

ICME Summer Workshops 2021

Intermediate Topics in Machine Learning and Deep Learning

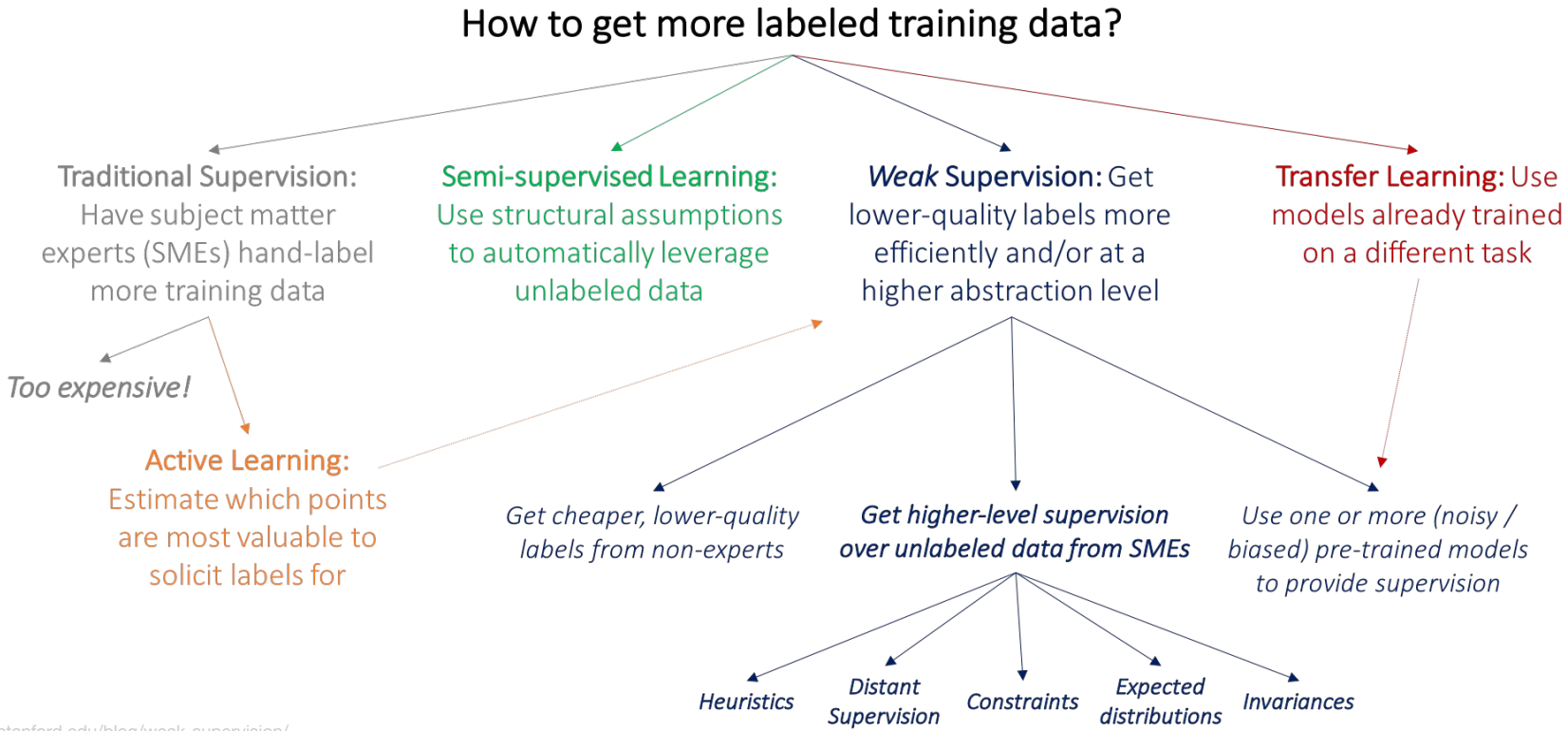
Session 3.2: Weak Supervision

Tuesday, August 17, 9:30—11:00 AM

Instructor: Sherrie Wang

icme-workshops.github.io/intermediate-ml

Back to the label scarcity problem



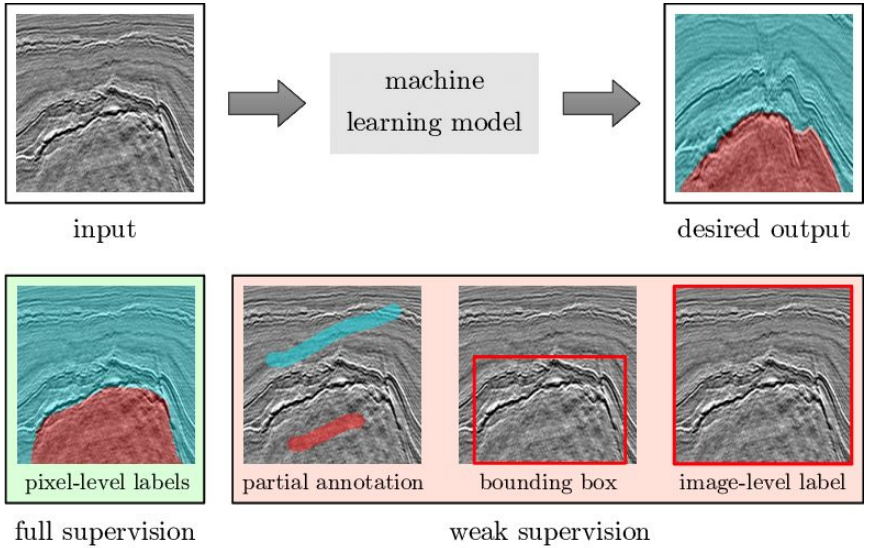
http://ai.stanford.edu/blog/weak-supervision/

Weak supervision

High-quality labels are expensive and time-consuming to generate, but often low-quality labels already exist or are much cheaper / faster to generate.

Can we make use of these low-quality labels?

Low quality = labels exist at some higher level than desired, or contain noise

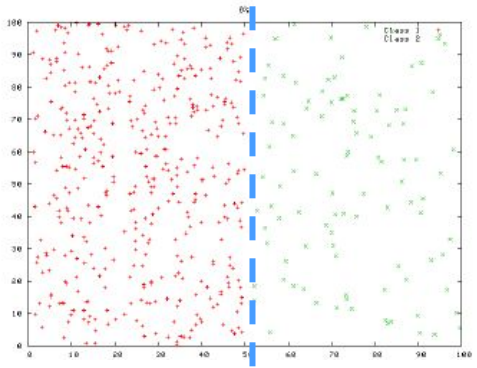


Weak supervision

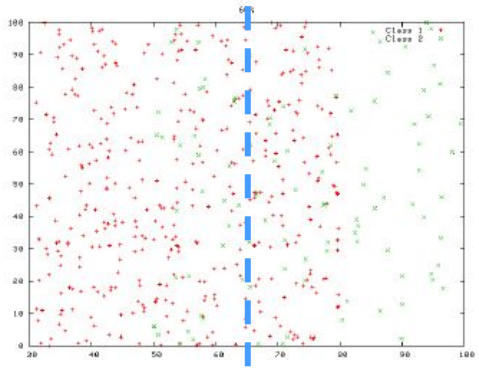
If the main problem with labels is that they are noisy, and the noise is random -- can still use them to train a model.

For model evaluation, you want a clean, high-quality set of labels.

High quality labels



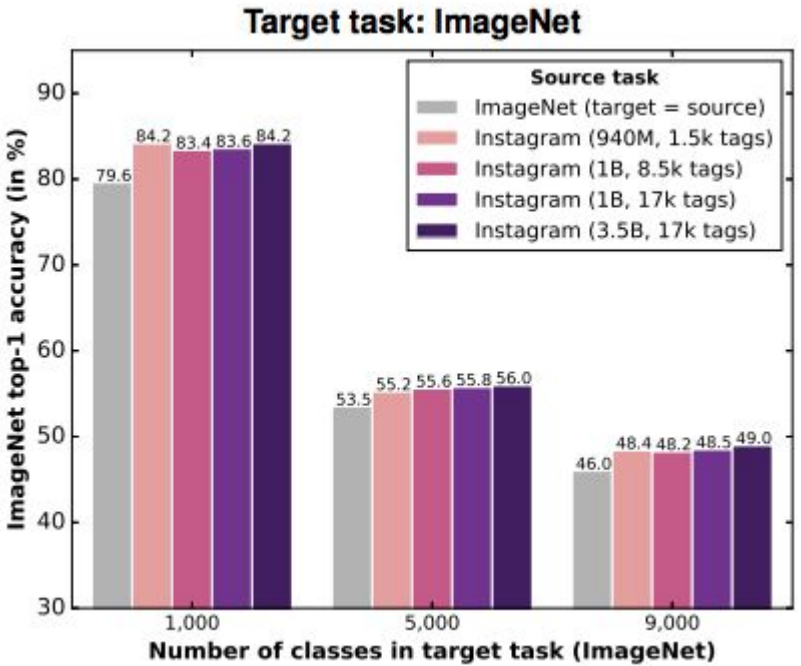
Noisy labels



Example: Using hashtags as weak labels for images

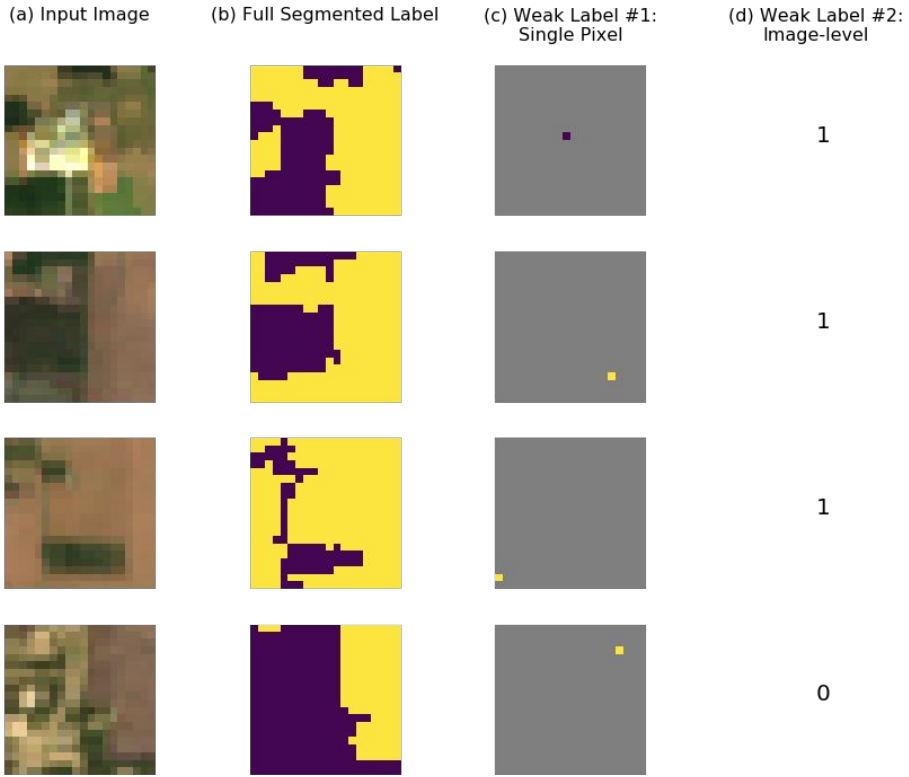
Researchers from Facebook used Instagram images and hashtags to pre-train a model for image classification.

When the model's features were used to classify ImageNet, performance was excellent.



Mahajan *et al*. (2018) "Exploring the Limits of Weakly Supervised Pretraining"

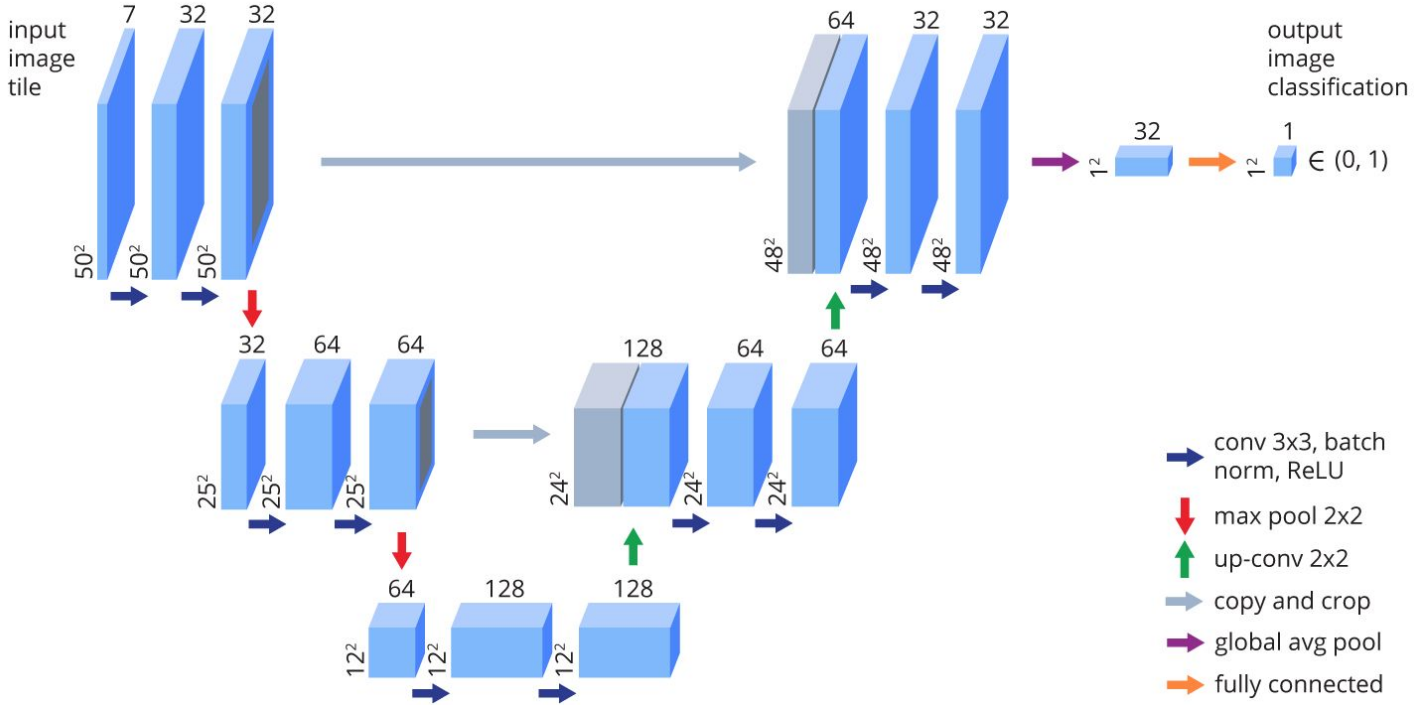
Example: Image-level and single pixel labels for cropland segmentation



- Can we use image and single pixel labels to supervise segmentation?
- How many labels do we need?

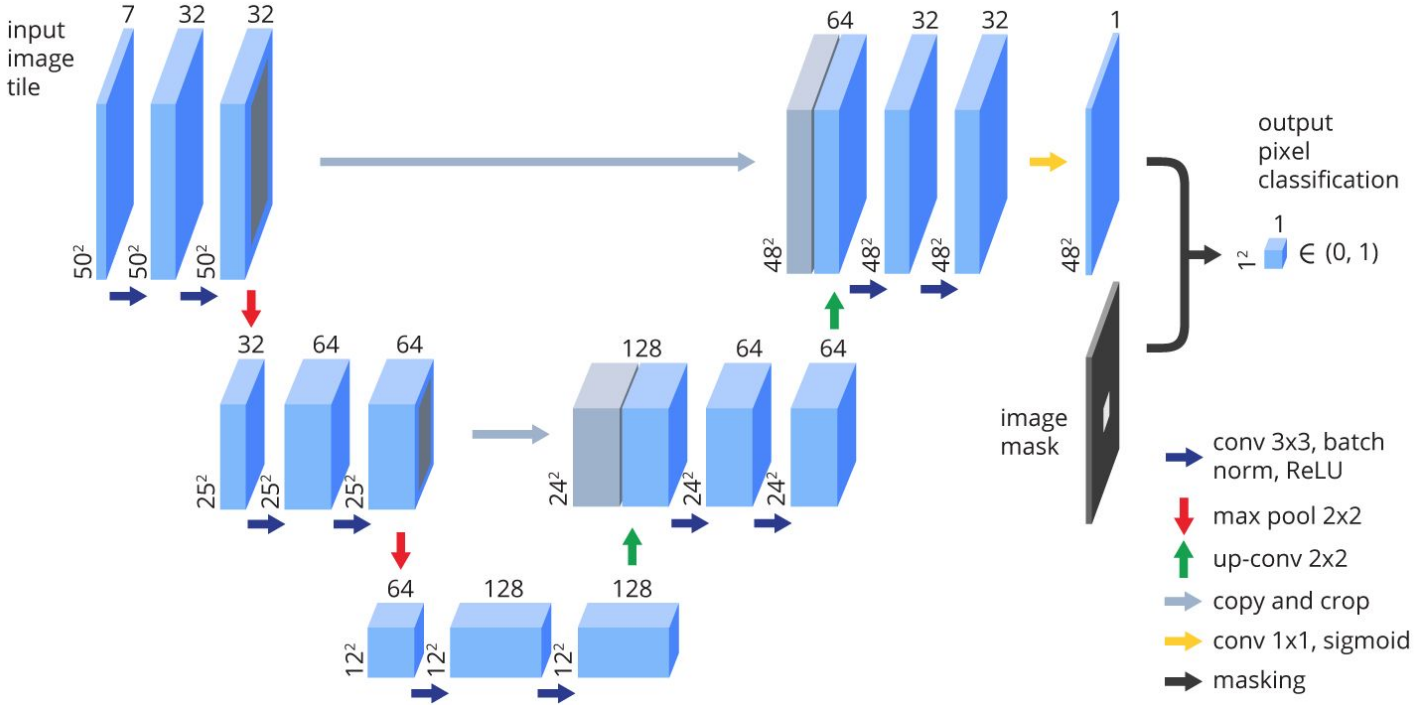
Wang *et al.* Weakly supervised deep learning for segmentation of remote sensing imagery. *Remote Sensing* (2020).

Image labels: adding class activation map to U-Net

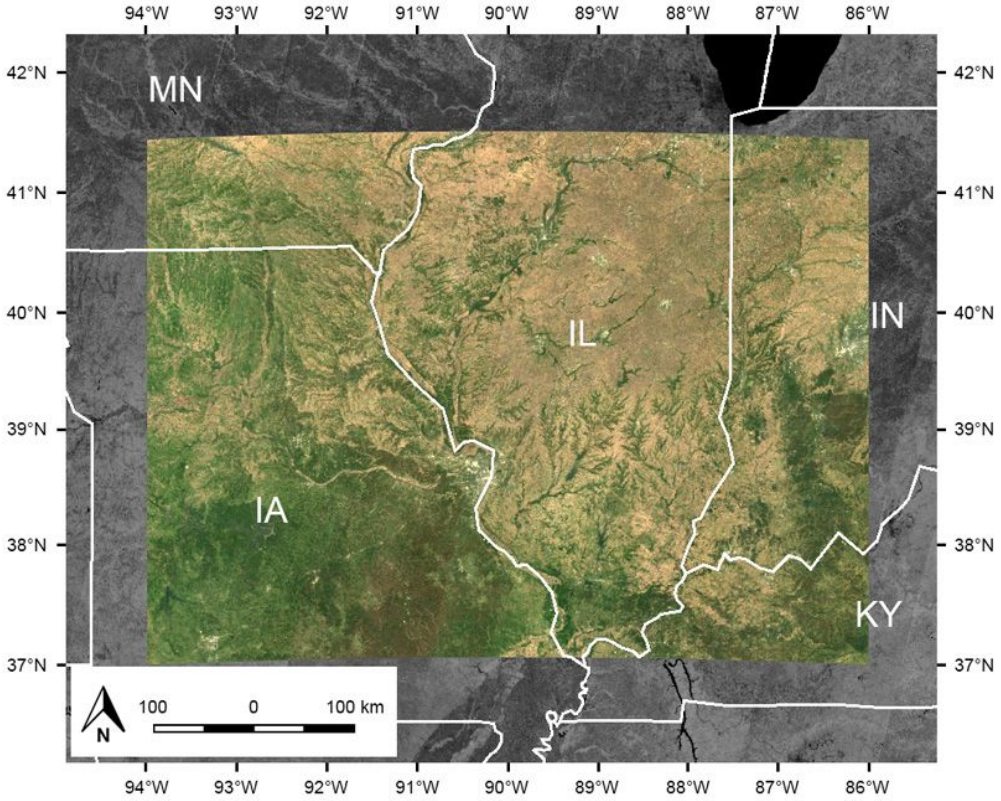


Zhou *et al*. Learning Deep Features for Discriminative Localization. CVPR (2016).

Single pixel labels: masking U-Net output

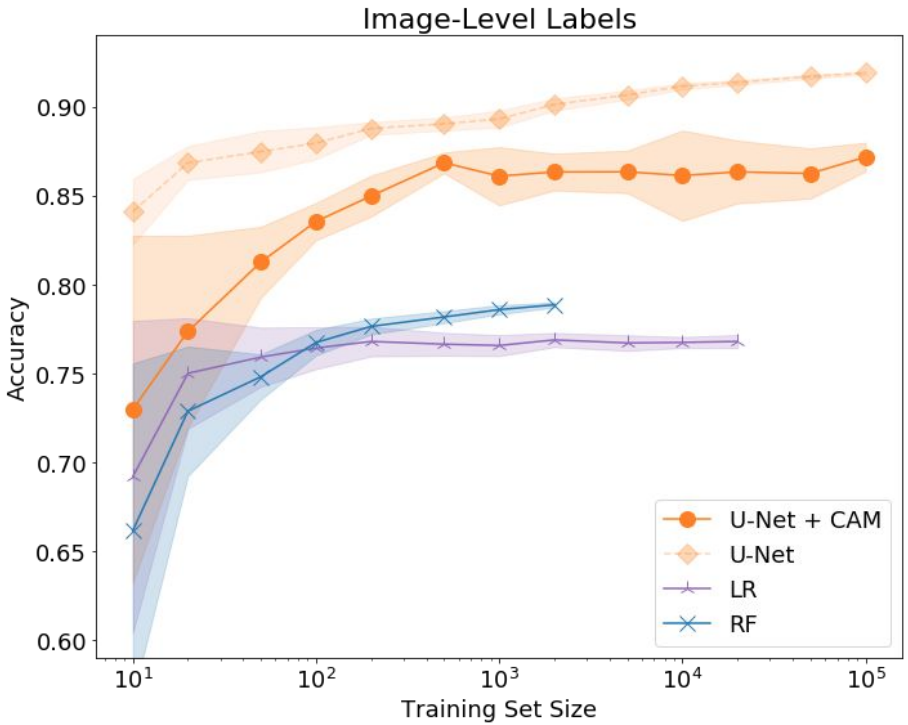


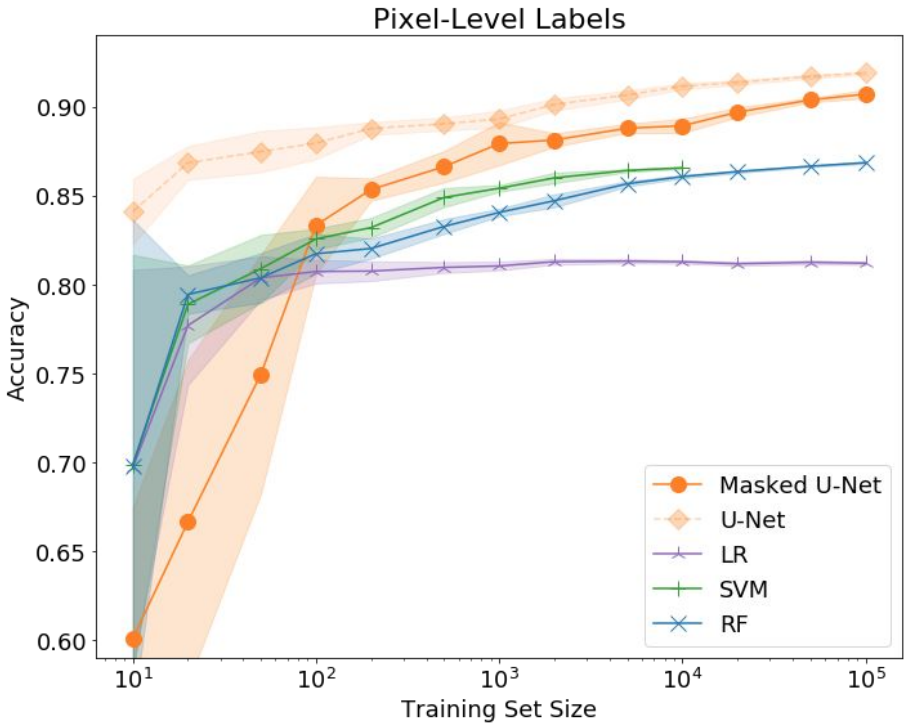
Study area and cropland segmentation



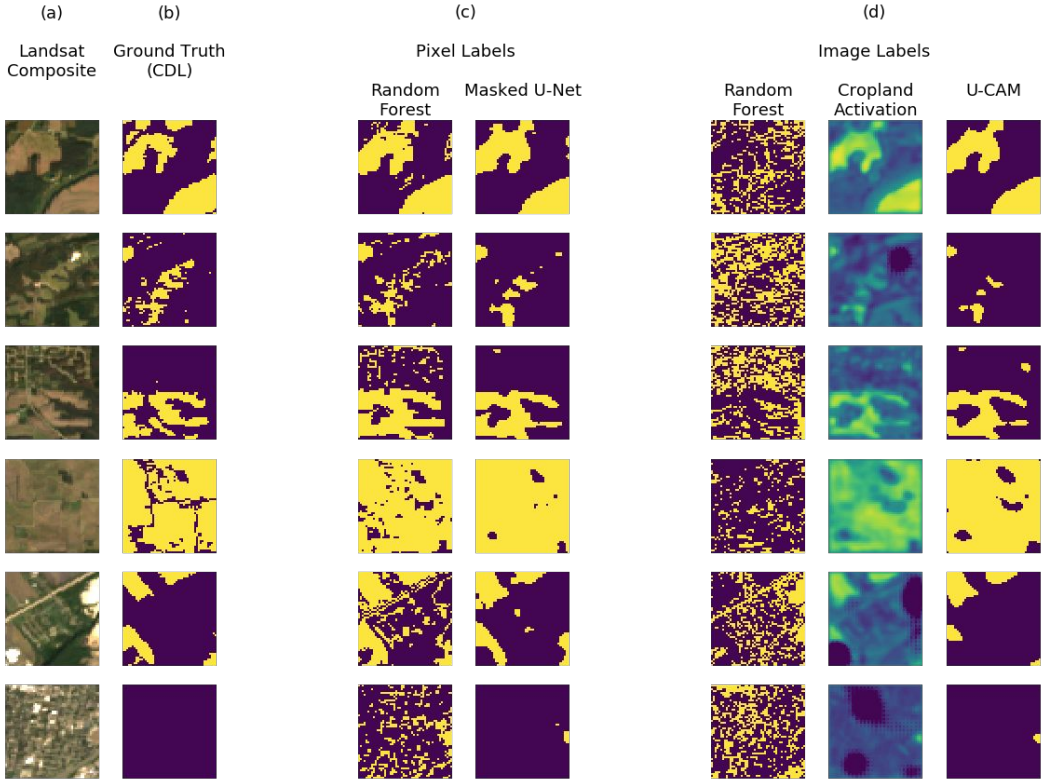
- Satellite image over the Midwestern US
- USDA's Cropland Data Layer (CDL) as ground truth
- Tiles of 50 x 50 pixels (200,000 images)

Conclusion: Weak labels can supervise segmentation with relatively few labels





Example segmentation



Thanks for attending this workshop!

This is the first iteration of the Intermediate ML and DL workshop.

Please let us know which topics were the most interesting/useful and which ones you wish had been covered in your reviews.

It won’t be possible to cover all intermediate topics, but hopefully we can converge on the most interesting topics over time!