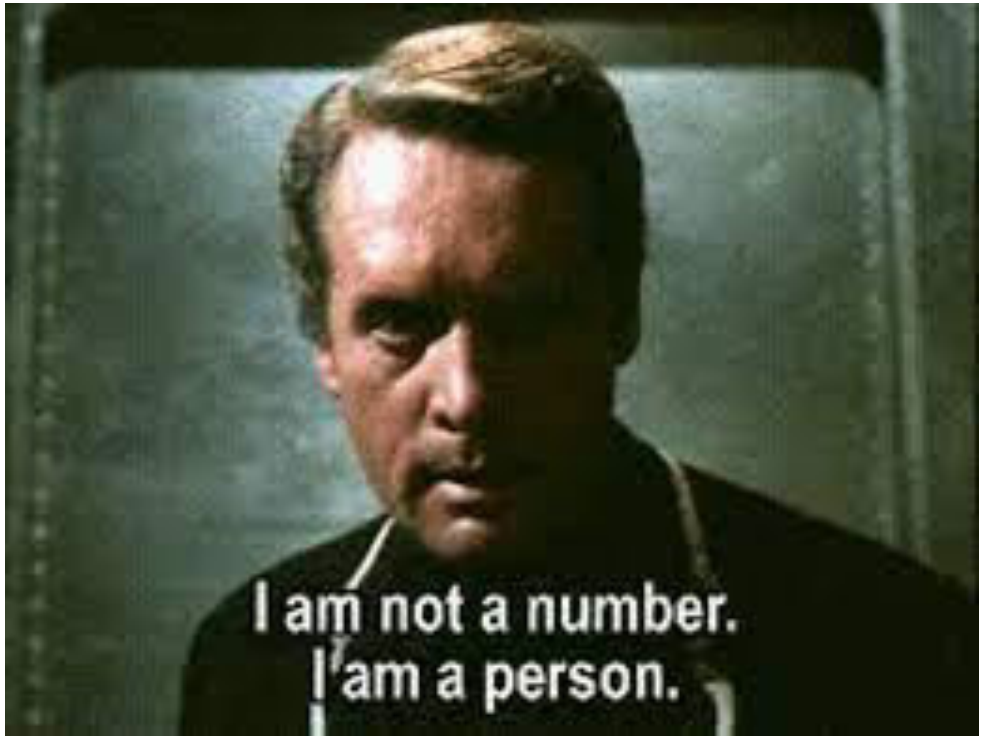
Representation Learning

Week 8

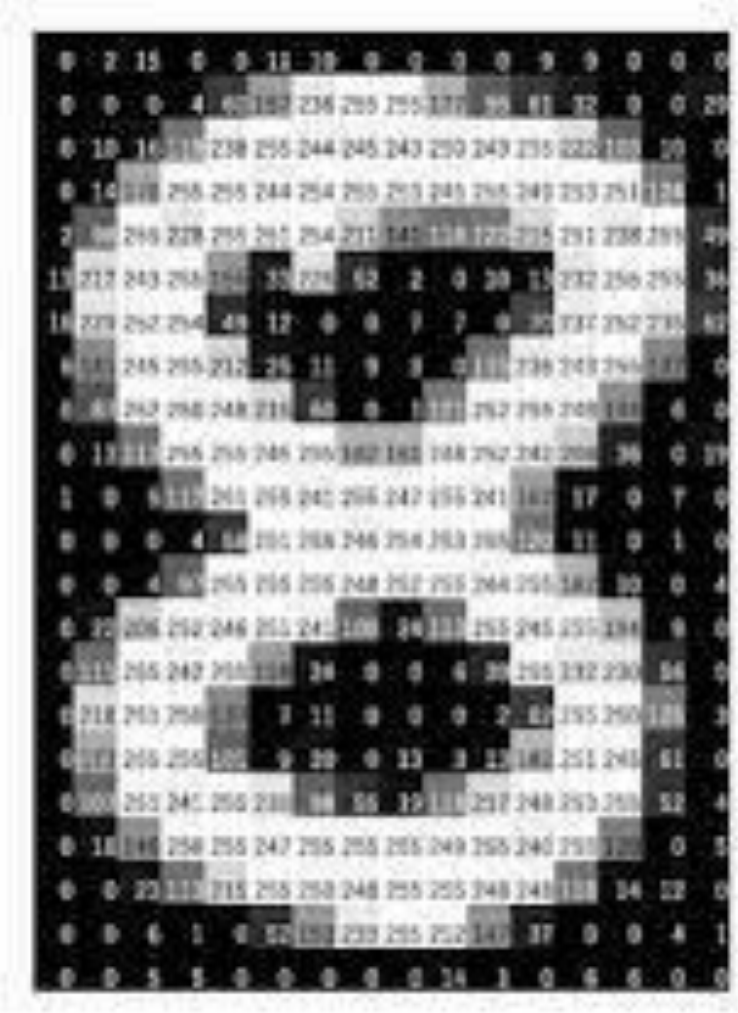
What Our Datum Really Is

- A datum of interest is typically something in the real world
  - A physical object
  - A place
  - A user
  - An idea



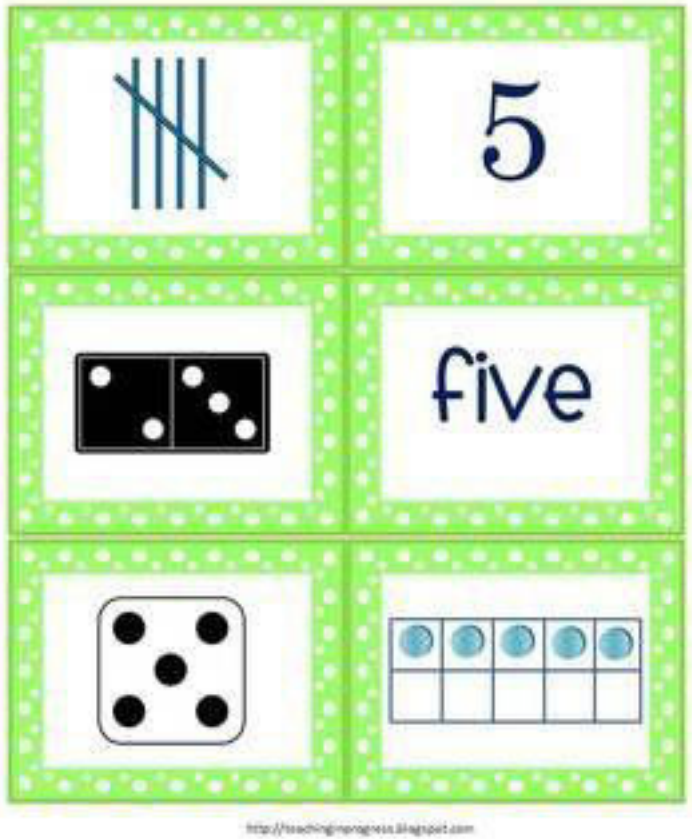
Numerical Representation

- Computers don’t see the real data
- Computers see numbers
- Use numbers represent the real datum
- Simplest example: use raw observarional data
  - Pixels
  - Point clouds
  - Letters



Countless Representations

- There are many way to represent the same daum
- In fact, infinite ways
- How do we select the best one?
- What are the criteria for a good representation?

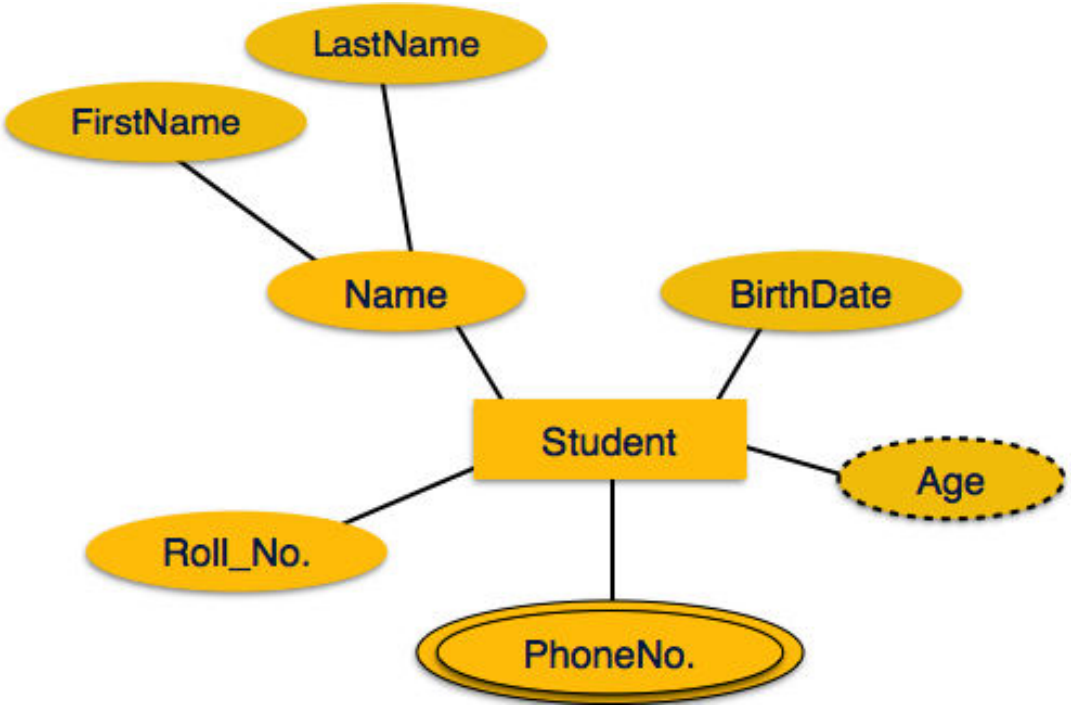


Mathematical Formulation

- A procedure such that:
  - for every data point **x**, we have vector **e** that represents it
- Very general
- Can map every **x** to 0
- Can map each **x** to a different random number
- Procedure is not neccesarily a parametric function
- Need more constraints…

Information Preservation

- Representation should present the important information
- What's imporant?
- Option 1: Everything
- Option 2: Only the semantic aspects
  - What's semantic?



Low-dimension

- We often want a compact representation
- Like compression – there is a tradeoff
  - Low dimension - high compression – high information loss
  - High dimension - low compression – low information loss

Meaningful Distance on Semantic Attributes

- Assume there is a set of attributes we care about
- Does low (l1,l2,cosine) distance = similar value of semantic attributes?

$$\|e_{dog} - e_{anotherdog}\|^2 < \|e_{dog} - e_{donkey}\|^2$$

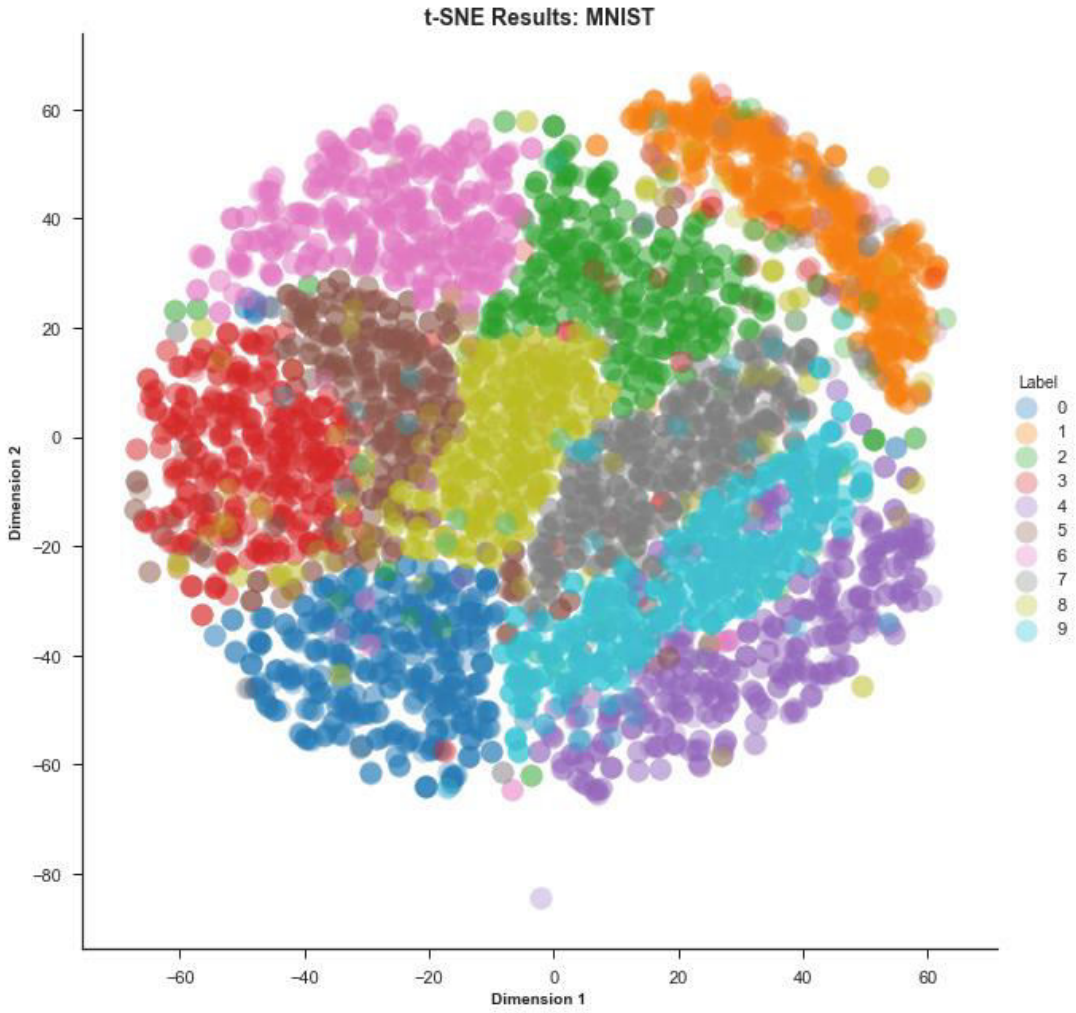
Informativeness on Semantic Attributes

- Assume there is a set of attributes we care about
- Can linear classifier predict attribute values?

$$y = w \cdot e + c$$

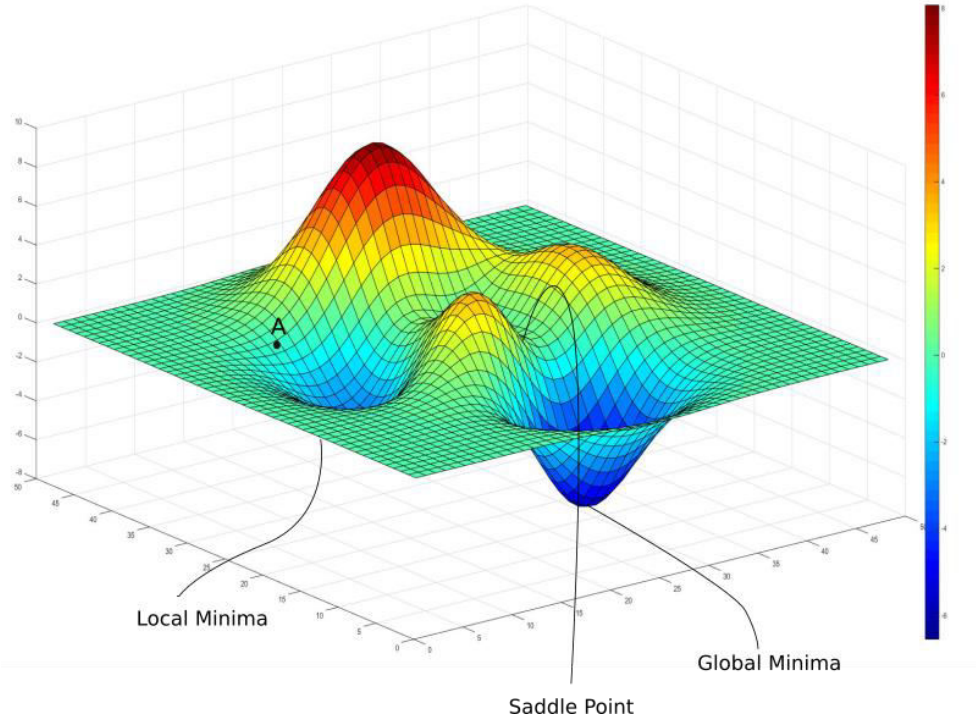
Visualization

- Can we easily visualize the representation?
- Typically means:
  - 2D representation
  - Distance corresponds to semantic attributes



Pretraining – Not Really RL

- In DL –preliminary training stage on large, but unrelated dataset
- A good starting point for training on our target dataset
- Not really representation learning
- Often confused together



Classical Machine Learning Ideas

- Traditional literature:
  - Reducing dimension
  - Preserving as much information as possible
- Challenge: measuring how much important information is preserved

Distance Preservation: Schema

- Calculate distance between each pairs of samples $d_{ij} = d(x_i, x_j)$
- Come up with embeddings for each point e.g. e_i for xi
- Distances between embeddings and samples should be similar

Simple Case: Classical Dimensional Scaling

- Measure distance between sample pairs using L2
- Measure distance between embedding pairs using L2
- Minimize difference between distances
- We will show this is equivalent to PCA

cMDS = PCA

Assume $\|e_i - e_j\|^2 = d_{ij} \equiv \|x_i - x_j\|^2$ then if the embeddings have mean 0,

$$e_i \cdot e_j = -\frac{1}{2}[d_{ij} - \frac{1}{n}\sum_i d_{ij} - \frac{1}{n}\sum_j d_{ij} + \frac{1}{n^2}\sum_{ij} d_{ij}]$$

or in matrix notation:

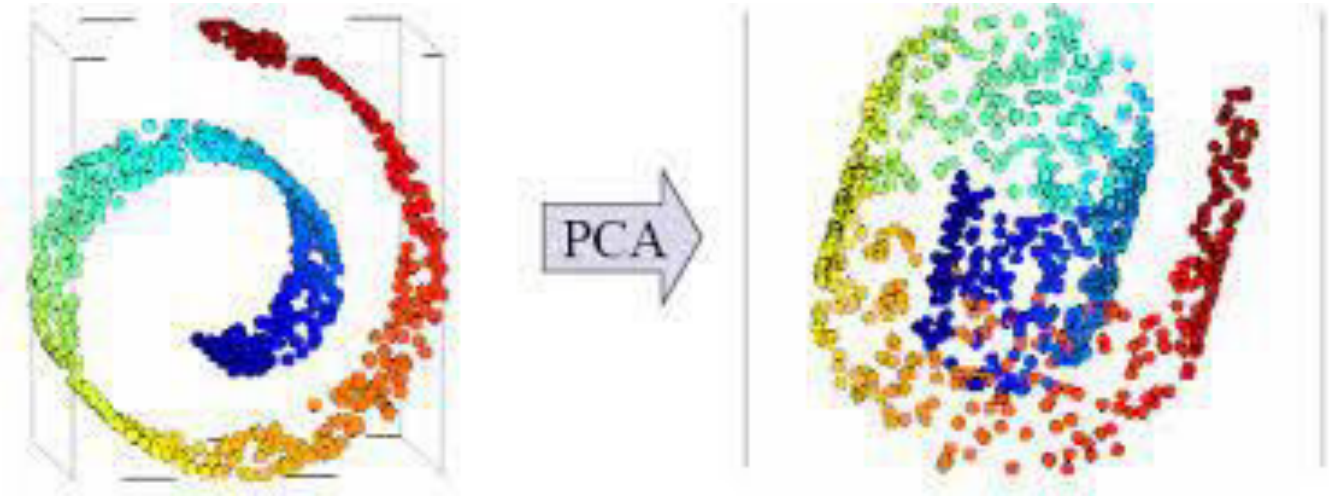
$$E^T E = K \equiv -\frac{1}{2}(I - \frac{1}{n}1)^T D(I - \frac{1}{n}1)$$

As we can write: $K = V^t \Lambda V$

We have: $K = \Lambda^{\frac{1}{2}} V$

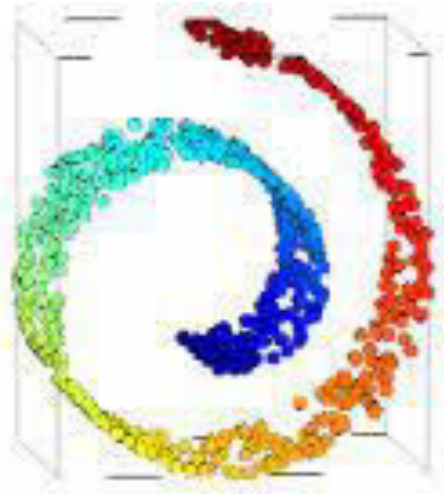
So What's Wrong with PCA

- Several issues:
  - A linear function – not powerful enough for complex data
  - More influence by large distances than small ones
    - Clusters are separated but local structure not preserved



Local Embedding Methods

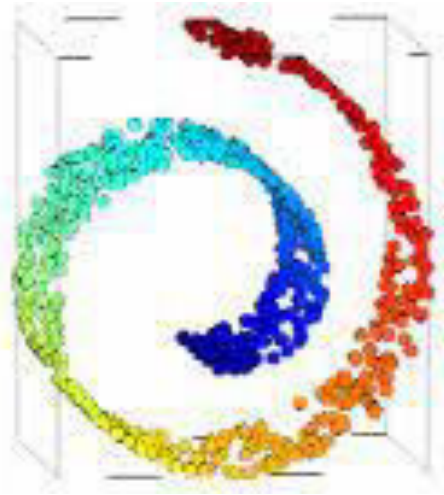
- Let’s reconsider the swiss roll
- Data lies on a 2D manifold
- How to describe it’s geometry?
- Idea: local structure!



Laplacian Eigenmaps

- For each point x_i find its k NNs
- For each NN j, compute weight $w_{ij} \propto e^{-d_{ij}}$
- Learn embedding ei for each xi such that

$$\sum_{ij} \|e_i - e_j\|^2 w_{ij} \quad s.t. \quad cov(E) = I$$



- Constraint is in theory a little more complex
- Constraint ensures solution does not collapse to 0

Lots of Issues with Local Methods

- Discuss in class

Stochastic Neighborhood Embeddings

- A mixture of global and local approaches
- Define:

$$p_{i|j} = \frac{e^{-\frac{1}{2\sigma^2}\|x_i - x_j\|^2}}{\sum_{k \neq i} e^{-\frac{1}{2\sigma^2}\|e_i - e_k\|^2}}$$

$$q_{i|j} = \frac{e^{-\|e_i - e_j\|^2}}{\sum_{k \neq i} e^{-\|e_i - e_k\|^2}}$$

- The loss is given by:

$$C = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}},$$

Symmetric SNE

- SNE is not symmtric, but easily fixed

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$$

$$q_{ij} = \frac{e^{-\|e_i - e_j\|^2}}{\sum_{l \neq k} e^{-\|e_l - e_k\|^2}}$$

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}.$$

tSNE: Symmetric SNE with Student's t

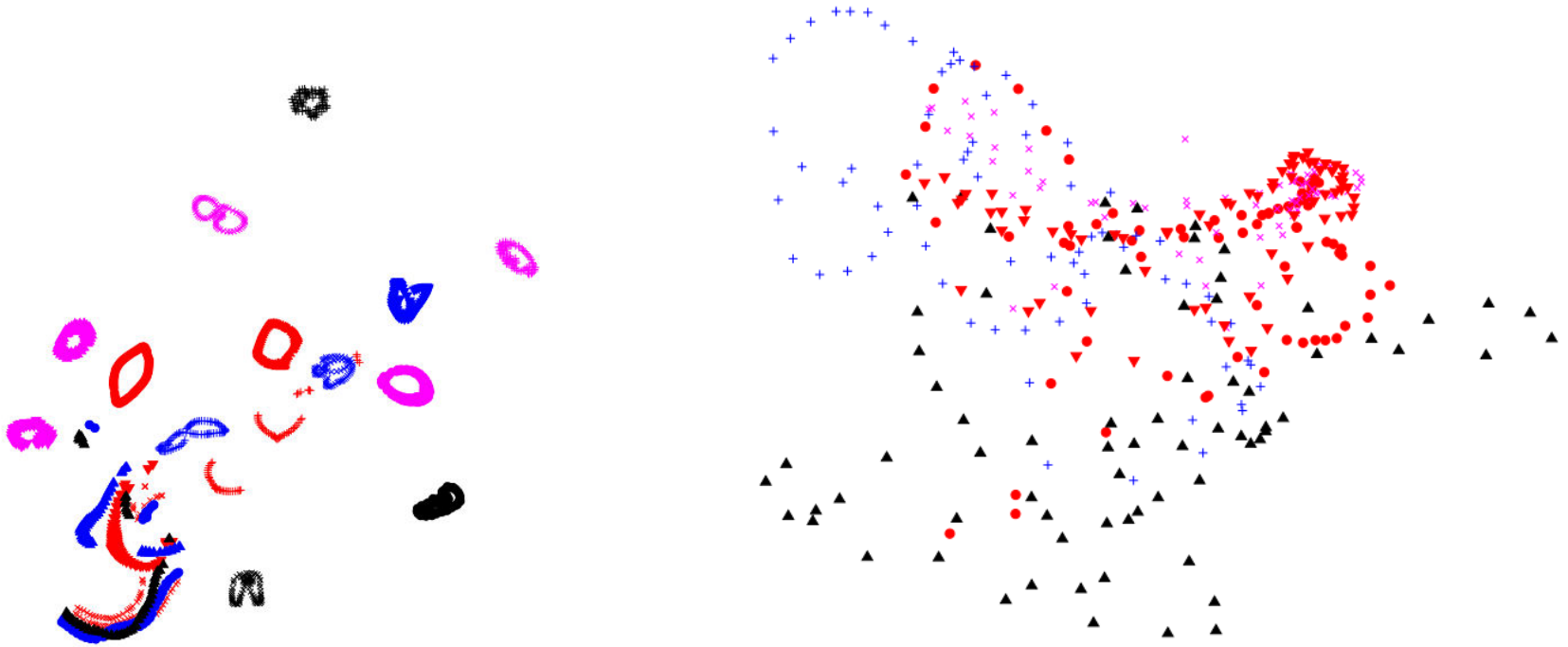
- The Gaussian distribution makes far embedding decay too fast
- Replace with Student's t

$$q_{ij} = \frac{e^{-\|e_i - e_j\|^2}}{\sum_{l \neq k} e^{-\|e_l - e_k\|^2}}$$

$$q_{ij} = \frac{(1+\|e_i - e_j\|^2)^{-1}}{\sum_{l \neq k} (1+\|e_i - e_j\|^2)^{-1}}$$

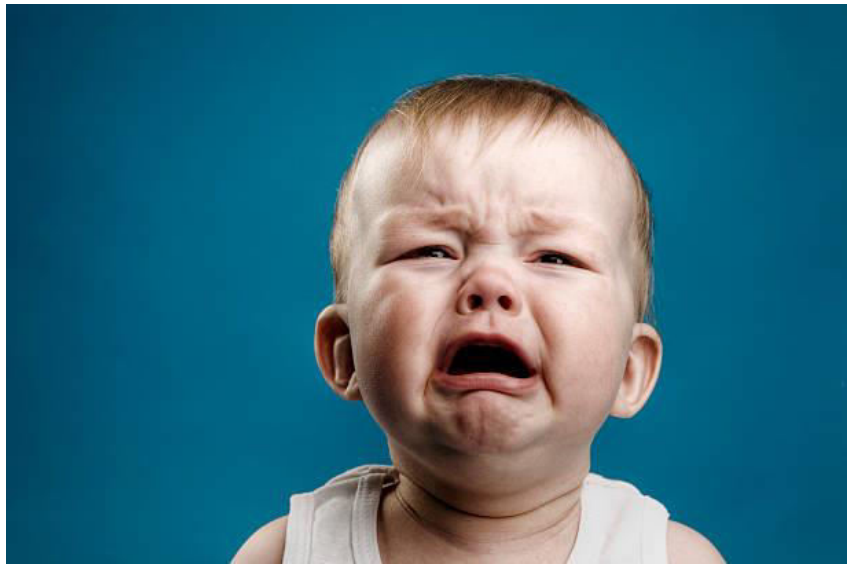
tSNE is Great for Visualization

- Combines global and local structure
- Comarison to local method LLE:



Issues with Classical Methods

- What to do with test samples?
- Is computing neighbors with L2 really a good idea?
- Not really suitable for high d embeddings? (why?)
- Classically, local methods did not work really well (t-SNE did in low d)

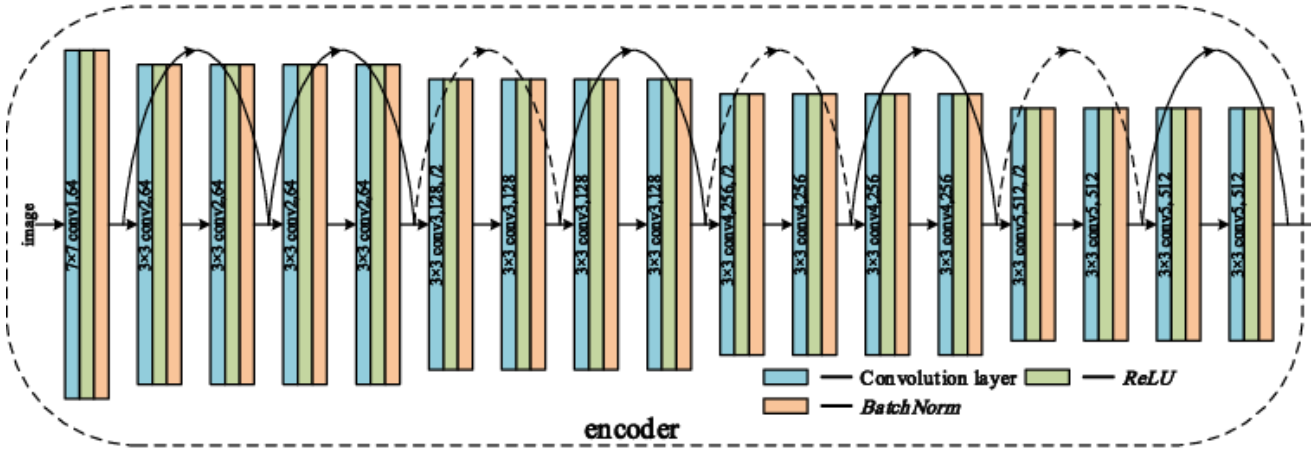


Two Modification are All You Need

- Modern method only modify two aspects
  1. Mapping from x to e through a parametric function (e.g. deep nets)
  2. Neighbors generated, not retrieved
- These simple changes are enough to make a massive impact!

Amortizarion

- Classical method optimized all embeddings directly
  - Cannot generlize to new samples
  - Prone to overfitting for large d
- Modern methods learn the mapping, this solves the issues
- Typically a ResNet or Transformer (like anywhere else In the course)



Augmentations

- Local methods want similar samples to have similar embeddings
- What is similar? L2 distance is terrible at high d
- Idea: let user decide which variation is not important
- For sample: generate samples that differ in unimportant aspects

Examples of Image Augmentations

Augmentations: the Good and the Ugly

- Good:
  - Can model much more semantic variation than L2
  - Does not require kNN search
- Bad:
  - Need to know what variation is not important
  - Need to be able to sample such augmentations (when is that hard?)

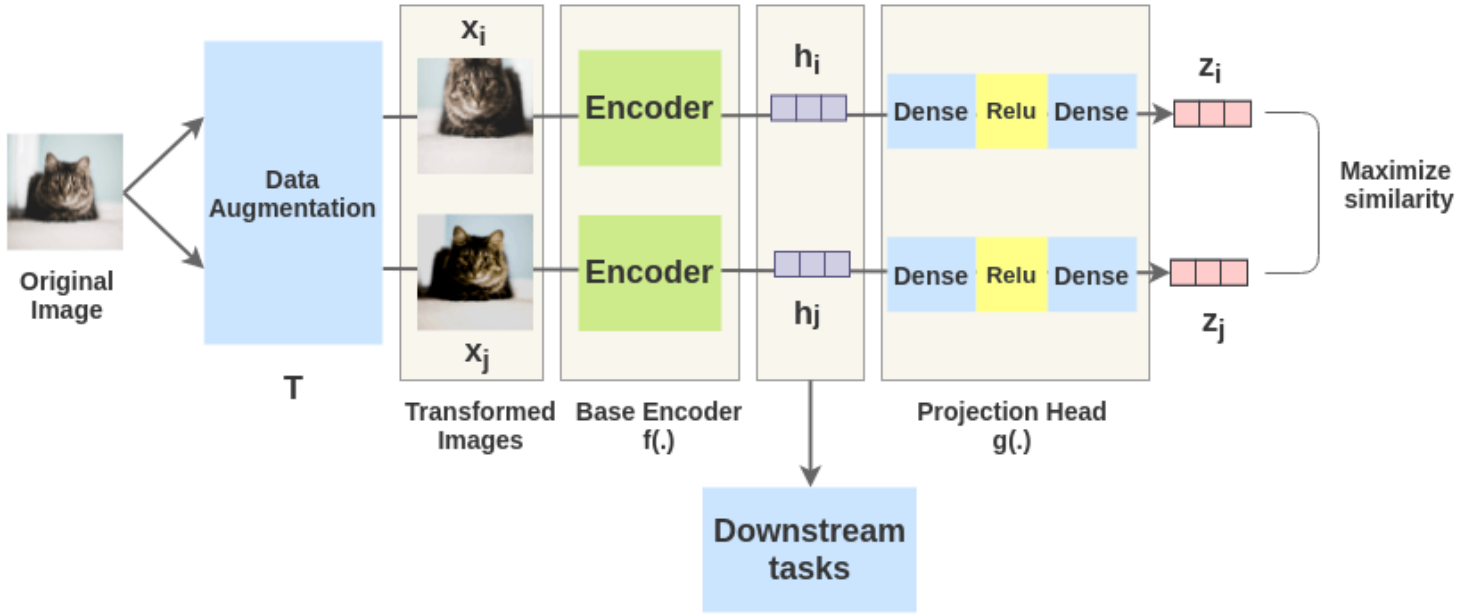
Contrastive Learning – SNE Interpretation

- Augment image x_i
- Compute probability of augmented image to belong to each image
- Embeddings unit normalized
- Groundtruth $p_{i|j}$ is now 1-hot

$$KL(Q|P) = \sum_i \frac{e^{e_i' \cdot e_i}}{\sum_{j \neq i} e^{e_i' \cdot e_j}}$$

Contrastive Learning – Classifier Interpret.

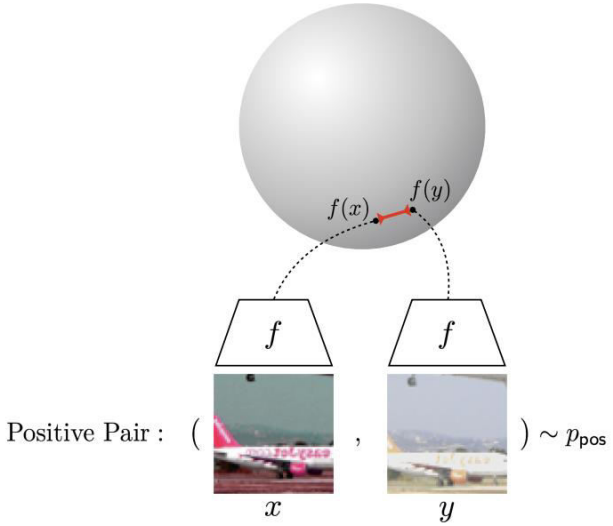
- Define each image x_i as a class
- All augmentations of x_i belong to class I
- Train a classifier to map augmented image to the correct class
- In practice, we subsample the denominator for faster runtime



Intuition

- Objective:
  - Augmentations fall in the same position
  - Feature hypersphere unifromly populated

$$\mathcal{L}_{\text{contrastive}}(f;\tau,M) = \mathop{\mathbb{E}}_{(x,y)\sim p_{\text{pos}}}\left[-f(x)^{\mathsf{T}} f(y)/\tau\right] + \mathop{\mathbb{E}}_{\substack{(x,y)\sim p_{\text{pos}} \\ \{x_i^-\}_{i=1}^{M} \overset{\text{i.i.d.}}{\sim} p_{\text{data}}}}\left[\log\left(e^{f(x)^{\mathsf{T}} f(y)/\tau} + \sum_i e^{f(x_i^-)^{\mathsf{T}} f(x)/\tau}\right)\right].$$



Alignment: Similar samples have similar features. (Figure inspired by Tian et al. (2019).)

Uniformity: Preserve maximal information.

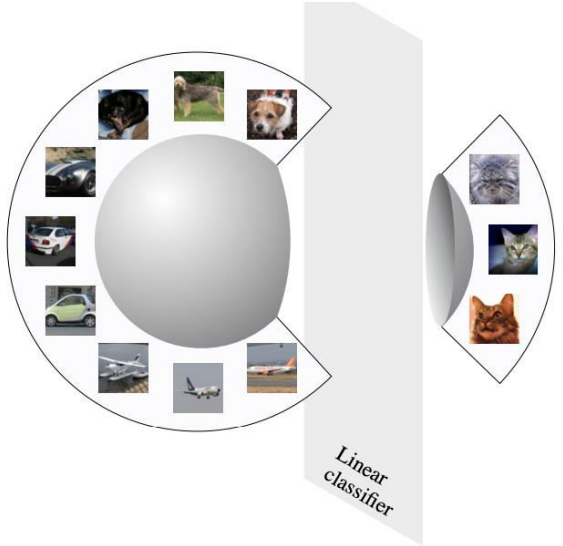
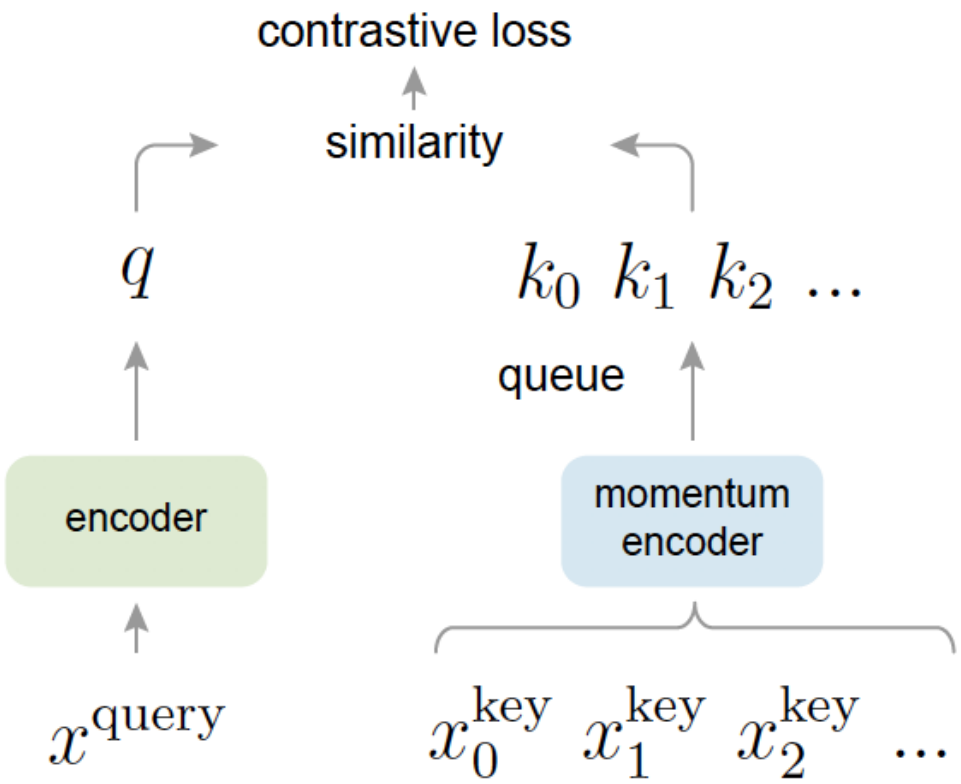


Figure 2: **Hypersphere:** When classes are well-clustered (forming spherical caps), they are linearly separable. The same does not hold for Euclidean spaces.

MOCO

- Tricks to reduce memory requirements
- Do not calculate embeddings for every batch
- Instead, keep the values of all embeddings in memory
- Momentum encoder ensure that embeddings change slowly



VICReg: Modern Laplacian Eigenmaps

- Another method by Lecun
- This one is basically laplacian eigenmaps with augmentations

$$\mathcal{L}_{\text{vic}} = \alpha \sum_{k=1}^{K} \max\left(0, 1 - \sqrt{\text{Cov}(\boldsymbol{Z})_{k,k}}\right) + \beta \sum_{j=1, j\neq k}^{K} \text{Cov}(\boldsymbol{Z})_{k,j}^2 + \frac{\gamma}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} (\boldsymbol{G})_{i,j} \|\boldsymbol{Z}_{i,.} - \boldsymbol{Z}_{j,.}\|_2^2.$$

Classical Idea: CCA

- Canonical Correlation Analysis – CCA
- Idea: given two views of the same data, learn common feature space
- Example: image and video descriptions of event
- Criterion: assume the features are x_i and y_i learn functions f and g s.t.:
  - $Corr(f(x_i),g(y_i)) = I$

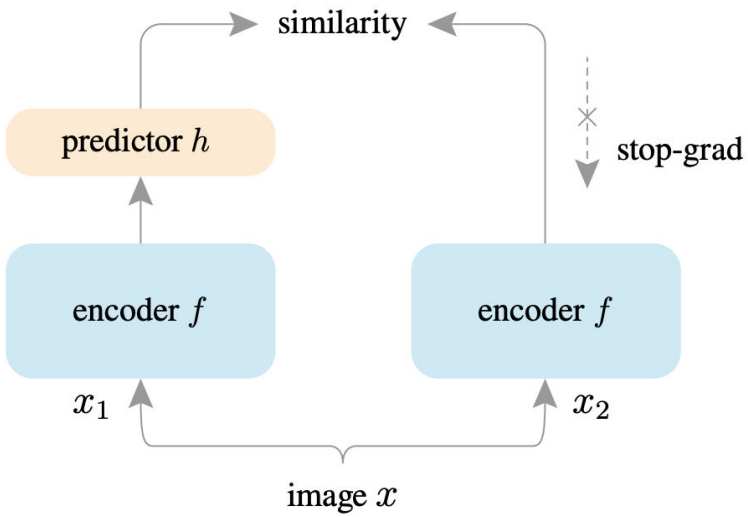
Barlow Twins: CCA with Augmentations

- We denote each sample xi and its augments x'_i
- Learn function f s.t.
  - Corr(f(xi),f(x'_i)) = I
- Does not really depend on batch size

$$\mathcal{L}_{\mathrm{BT}} = \sum_{k=1}^{K}((\boldsymbol{C})_{k,k} - 1)^2 + \alpha \sum_{k' \neq k} (\boldsymbol{C})_{k,k}^2, \ \alpha > 0.$$

SimSiam: No Negative Samples

- Exploring Simple Siamese Representation Learning, Chen and He, CVPR'21
- Main idea: drop the negative samples, fix one view



SimSiam Idea

- Extract features from two augmented views
- Learn predictor network to predict one from from the other

Algorithm 1 SimSiam Pseudocode, PyTorch-like

```
# f: backbone + projection mlp
# h: prediction mlp

for x in loader: # load a minibatch x with n samples
   x1, x2 = aug(x), aug(x) # random augmentation
   z1, z2 = f(x1), f(x2) # projections, n-by-d
   p1, p2 = h(z1), h(z2) # predictions, n-by-d

   L = D(p1, z2)/2 + D(p2, z1)/2 # loss

   L.backward() # back-propagate
   update(f, h) # SGD update

def D(p, z): # negative cosine similarity
   z = z.detach() # stop gradient

   p = normalize(p, dim=1) # l2-normalize
   z = normalize(z, dim=1) # l2-normalize
   return -(p*z).sum(dim=1).mean()
```

Multi-Modal Supervision

- Self-supervised learning needs distance guidance
- So far, augmentation – generative method
- Sometimes better free supervision exists
- Internet is full of (image, text caption) pairs
- Idea: replace or extend guidance to SSL methods using this

CLIP: ML Trick

- Double contrastive learning
- One contrastive loss predicting the correct image given the caption
- One contrastive loss predicting correct caption given image

CLIP: DL Trick

- CLIP by OpenAI implements this ideas
- DL trick: train different encoders for the image and text
- Image encoder can be ResNet or ViT
- Text encoder is a standard trasformer
- Train on lots (300M pairs) of data – there are now much bigger models

Masked Autoencoding

- Create hard labelled task for free
- Example: for a sentence, remove some word. Task is to predict word given the rest of the sentence.
- Very simple, but effective representation learner
- Similarly to AR models, here we estimate:

$$p(x_i|x_{-i})$$

BERT

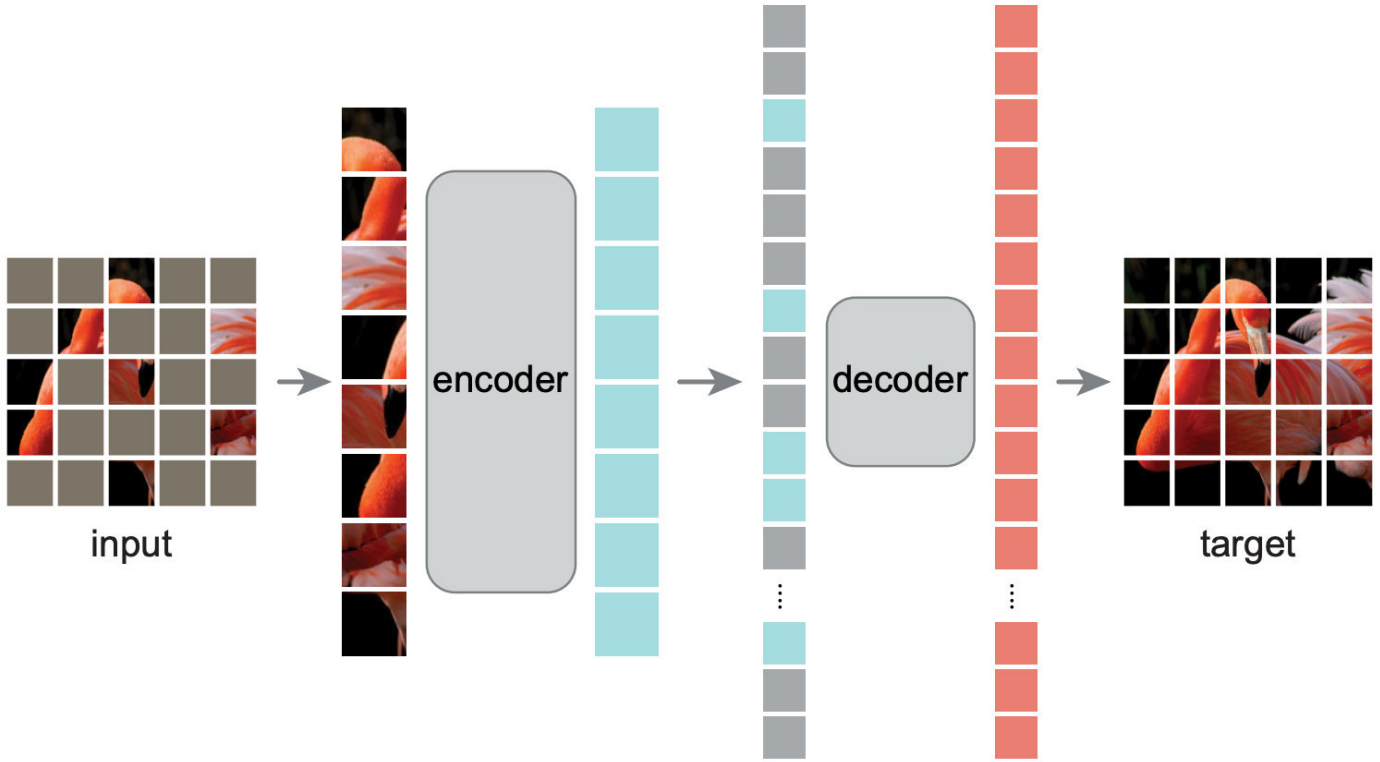
- Given sentence, remove word then learn to predict it
- A representation succesful at this task must be very informative
- Technical details:
  - Use transformer
  - Replace of 15 of tokens:
  - 80% prob replace by “mask” token
  - 10% replace by wrong word
  - 10% word unchanged

Masked Autoencoders

- Idea: mask out most of the image, and try to reconstruct it
- Kaiming He and other FAIR team

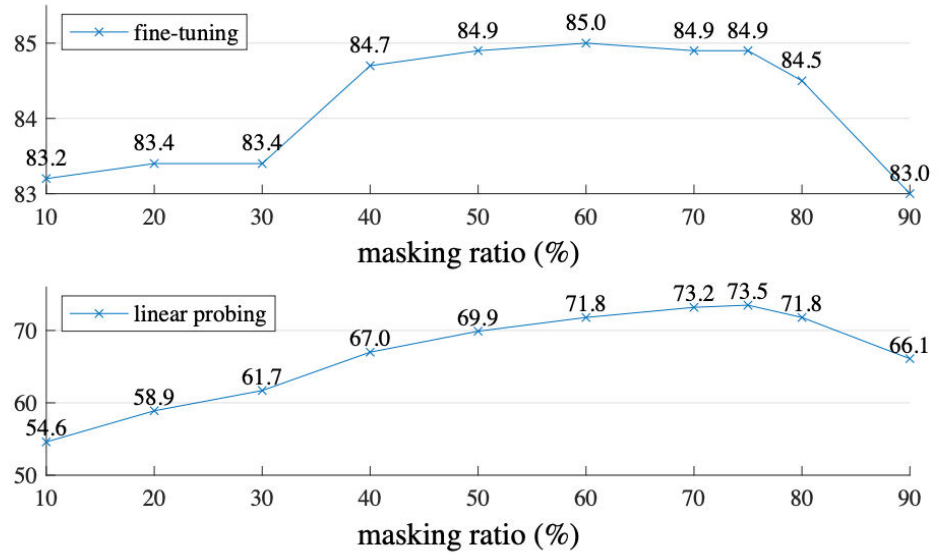
Architecture

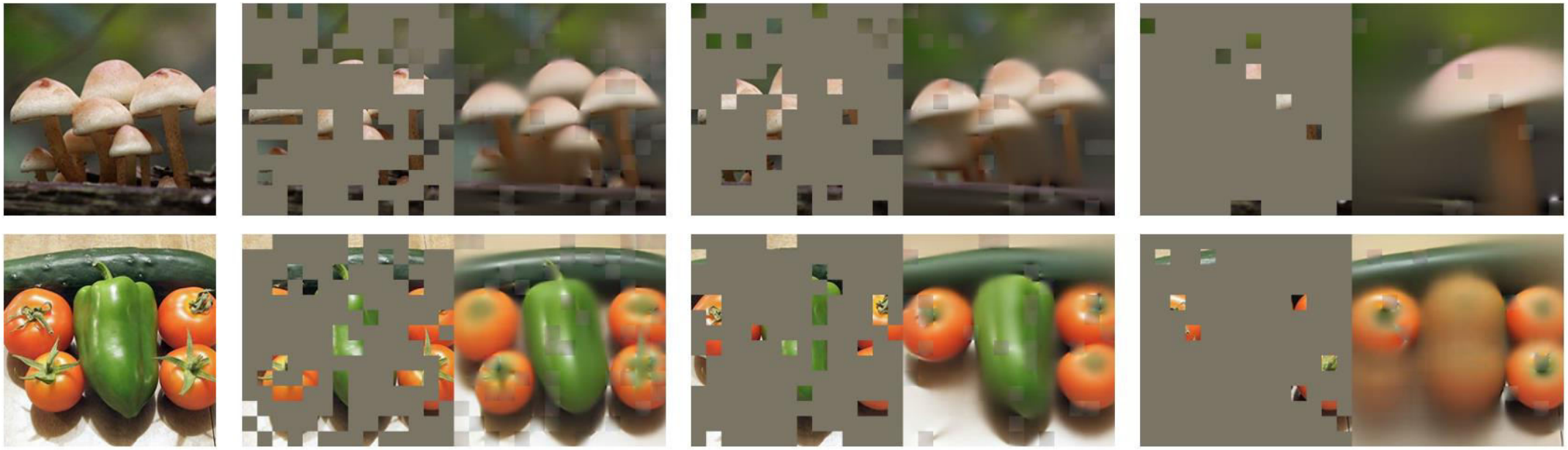
- Transformer encoder only operates on visible patches
- Decoder operates on both visible and masked token
- Reconstruction loss



High Masking Ratio

- Images are more sparse than NLP
- Masking of 75-85% vs 15% in BERT
- For video, ratios are around 95%!





Other Findings

- Working on raw pixels is as good as more complex ideas
- Random masking at least as good as other ideas
- Results are much better than contrastive methods for fiinetuning
- Results are not as good as contrastive for linear probing/ knn
- Great scaling behavior to very large transformers
- Operates well even on small datasets (10k images)

BEIT

- BERT for images
- First tokenize image using some vector quantizer
- Then run discrete BERT on the discrete image representation
- Results in line with MAE

Overview of Pros and Cons of Representations

- Contrastive
- MoCo
- Barlow Twins
- SimSiam
- VICReg
- BERT
- BEIT
- MAE