Deep Learning Theory and Applications

Learning and Manipulating Latent Space Representations

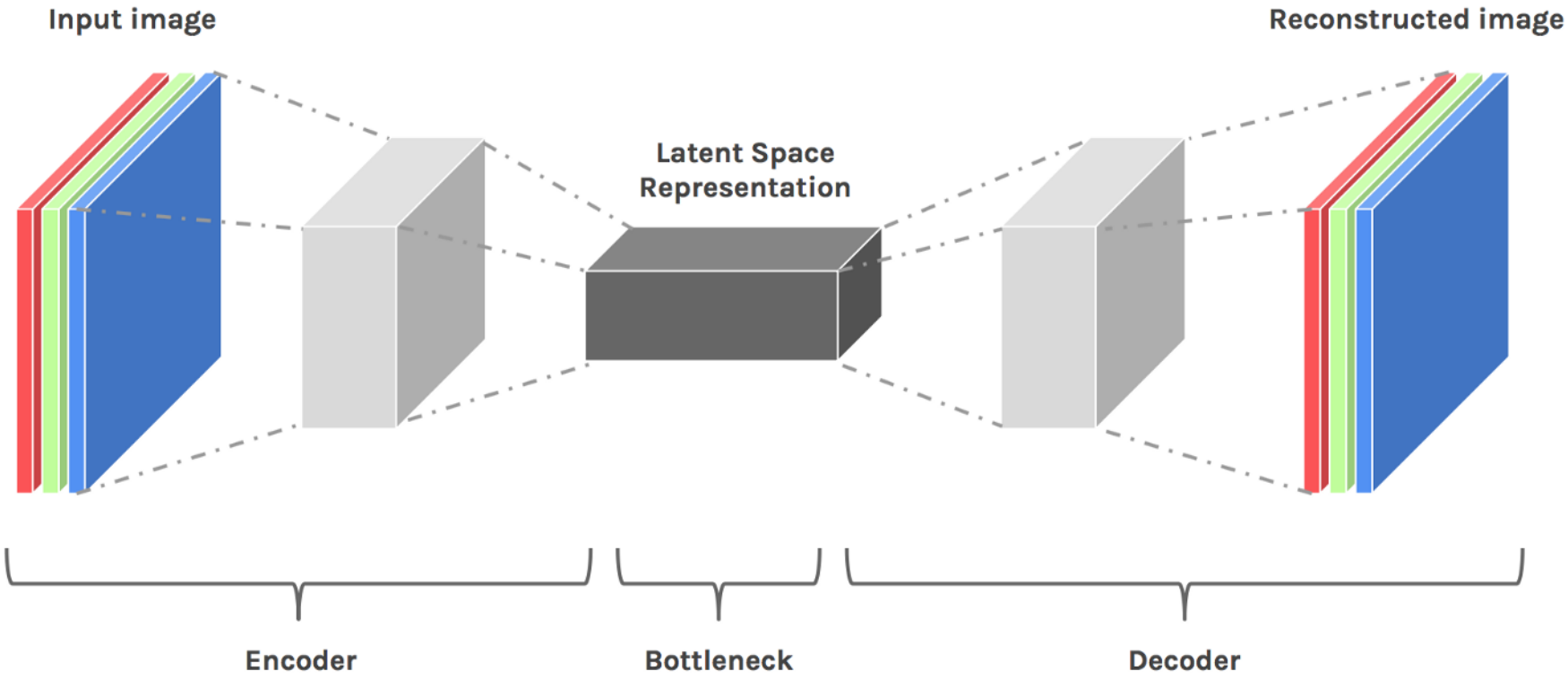
Yale

CPSC/AMTH 663

Outline

1. Representation Learning
2. PCA
3. Anomaly Detection
4. Interpolation
5. Arithmetic
6. Disentangled representations

Autoencoder Internal Representation



“latent space” is the transformed space in which the data lies in the bottleneck layer

Representation learning

- Find the “best” representation of the data
  - I.e., find a representation of the data that preserves as much information as possible while obeying some penalty or constraint that simplifies the representation
- How do we define simple?
  - Low-dimensional
  - Sparse
  - Independent components
- Above criteria aren’t necessarily mutually exclusive
  - E.g., low-dimensional representations often have independent or weakly dependent components

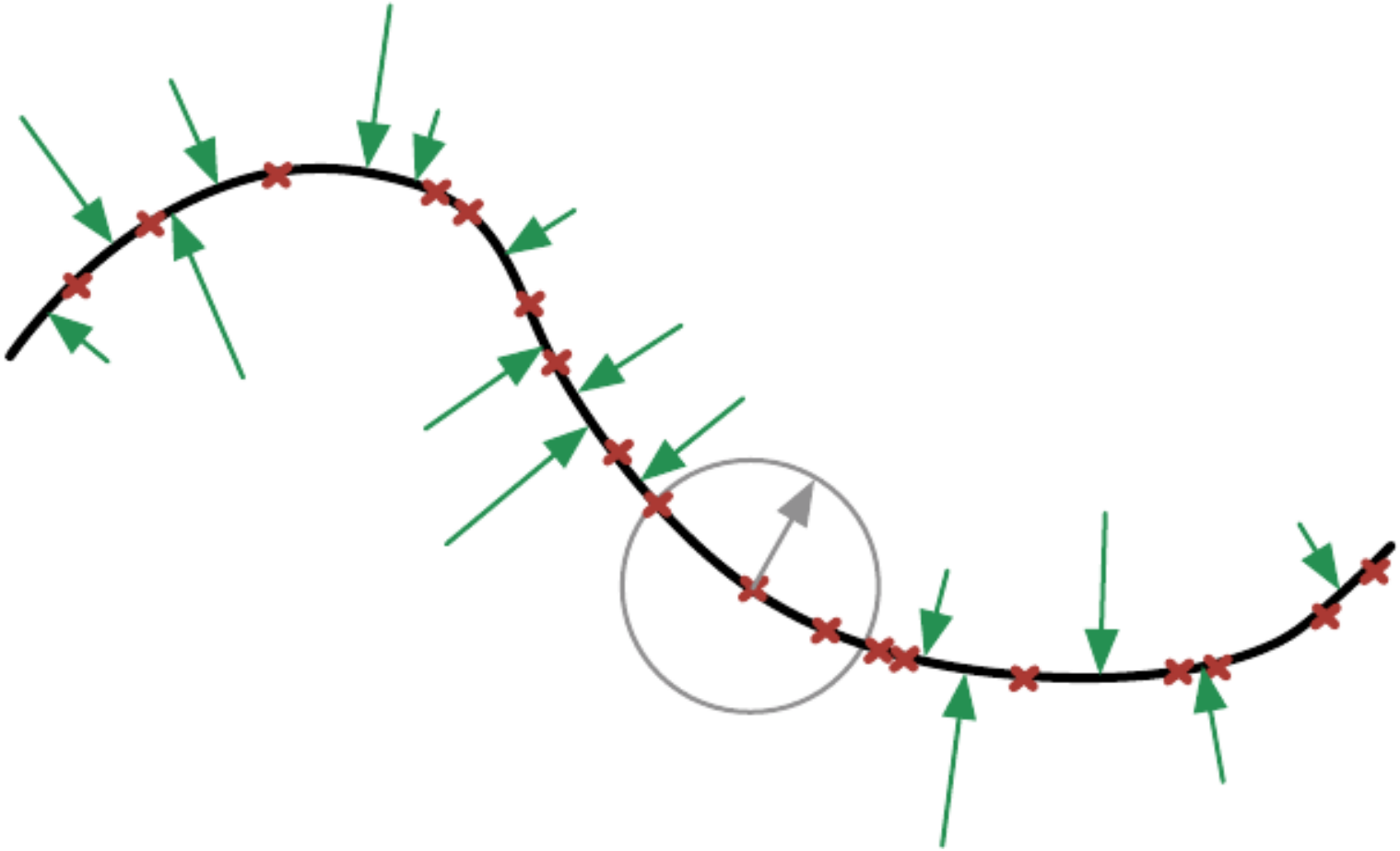
Representation learning

Why is a simple representation useful?

- Interpretability
  - E.g. visualization
- Computational cost
  - Compression
- Performance
  - Preprocessing

Manifold assumption

- Data may be modeled as lying on a low-dimensional manifold



Example

- What is a good lower dimensional representation of this data?

- How can we find it?

Dimensionality Reduction

- How much information is in the data (matrix)?
- Is there redundancy?
- Can we express this data more compactly, using fewer pseudo-features without too much loss of information?
- What should those pseudo-features be?
  - Combinations of original features
  - Linear combinations

Dimensionality Reduction

- Principle Components Analysis
- Multidimensional Scaling
- Related techniques where we derive the solution using eigenvectors of some matrix

- An eigenvector is a direction along which a matrix (linear transformation) is invariant
  - Vectors in that direction are only stretched
  - X is an eigenvector iff $Mx = \lambda x$

How can we re-express this matrix?

- Linear implies linear combinations of features
- Want to maintain maximal information
- How should we measure information?

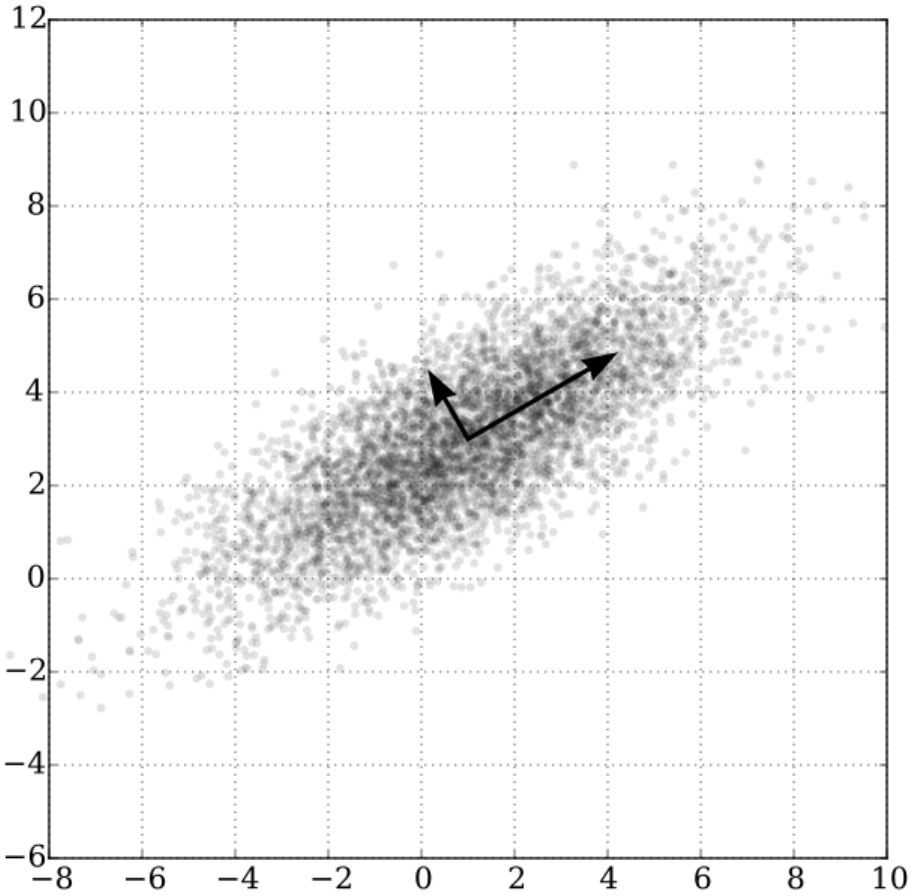
Variance as a Proxy for Information

- Can we maintain maximal variance in a few dimensions?
- Maximizing variation also implies minimizing redundancy in the re-expression or co-variance
- Notice we said “linear”, that implies we’re looking for a rotation and stretch
- This leads to PCA

Principle Components Analysis

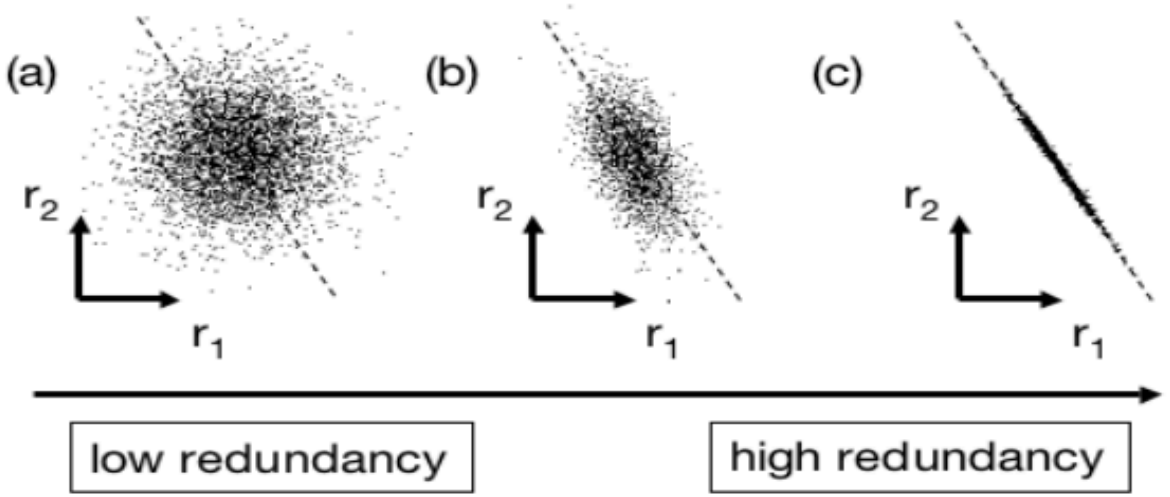
- The first new dimension will contain the maximal variance (and the second one the next most) …
- The new dimensions should be uncorrelated
- Just these two conditions are enough to drive PCA!

Maximal Variance



$$Var(X) = E[(X - E(X))^2]$$

Minimal Redundancy



Covariance Matrix

$$\begin{aligned}\operatorname{cov}(X, Y) &= \operatorname{E}[(X - \operatorname{E}[X])\,(Y - \operatorname{E}[Y])] \\ &= \operatorname{E}[XY - X\operatorname{E}[Y] - \operatorname{E}[X]Y + \operatorname{E}[X]\operatorname{E}[Y]] \\ &= \operatorname{E}[XY] - \operatorname{E}[X]\operatorname{E}[Y] - \operatorname{E}[X]\operatorname{E}[Y] + \operatorname{E}[X]\operatorname{E}[Y] \\ &= \operatorname{E}[XY] - \operatorname{E}[X]\operatorname{E}[Y].\end{aligned}$$

$$\Sigma_{ij} = \operatorname{cov}(X_i, X_j) = \operatorname{E}[\,(X_i - \mu_i)(X_j - \mu_j)\,]$$

$$\Sigma = \begin{bmatrix} \operatorname{E}[(X_1 - \mu_1)(X_1 - \mu_1)] & \operatorname{E}[(X_1 - \mu_1)(X_2 - \mu_2)] & \cdots & \operatorname{E}[(X_1 - \mu_1)(X_n - \mu_n)] \\ \operatorname{E}[(X_2 - \mu_2)(X_1 - \mu_1)] & \operatorname{E}[(X_2 - \mu_2)(X_2 - \mu_2)] & \cdots & \operatorname{E}[(X_2 - \mu_2)(X_n - \mu_n)] \\ \vdots & \vdots & \ddots & \vdots \\ \operatorname{E}[(X_n - \mu_n)(X_1 - \mu_1)] & \operatorname{E}[(X_n - \mu_n)(X_2 - \mu_2)] & \cdots & \operatorname{E}[(X_n - \mu_n)(X_n - \mu_n)] \end{bmatrix}.$$

Covariance as An Operator

- Covariance matrix maps a linear combination c onto a vector of covariances
- Covariance between two linear combinations:

$$d^T \sum c = \mathrm{cov}(d^T X, c^T X)$$

- Variance of c:

$$Var(c) = c \sum c$$

PCA

- The first k principle components are the kth largest eigenvectors of Σ
- If these eigenvectors are chosen to have unit norm then the variance captured by the eigenvectors is equal to the eigenvalue

Hint at derivation

- Find first vector c such that var(c) is maximal, but c has unit norm

$$\text{argmax}_c \, Var(c) = c^T \sum c$$

$$c^T c = 1$$

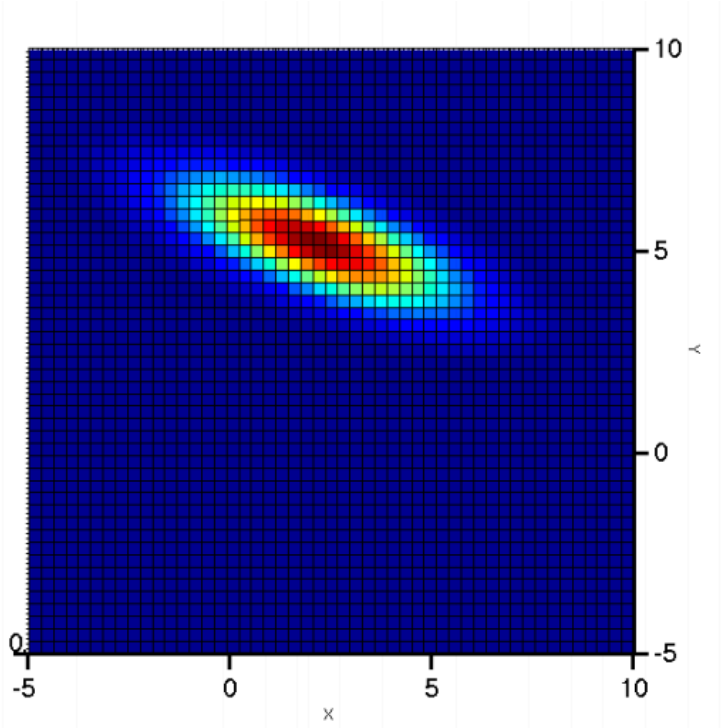
- Use a lagrange multiplier and maximize: $c^T \sum c - \lambda(c^T c - 1)$
- Differentiate wrt c and set to 0

$$\frac{d(c^T \sum c - \lambda(c^T c - 1))}{dc} = 0$$

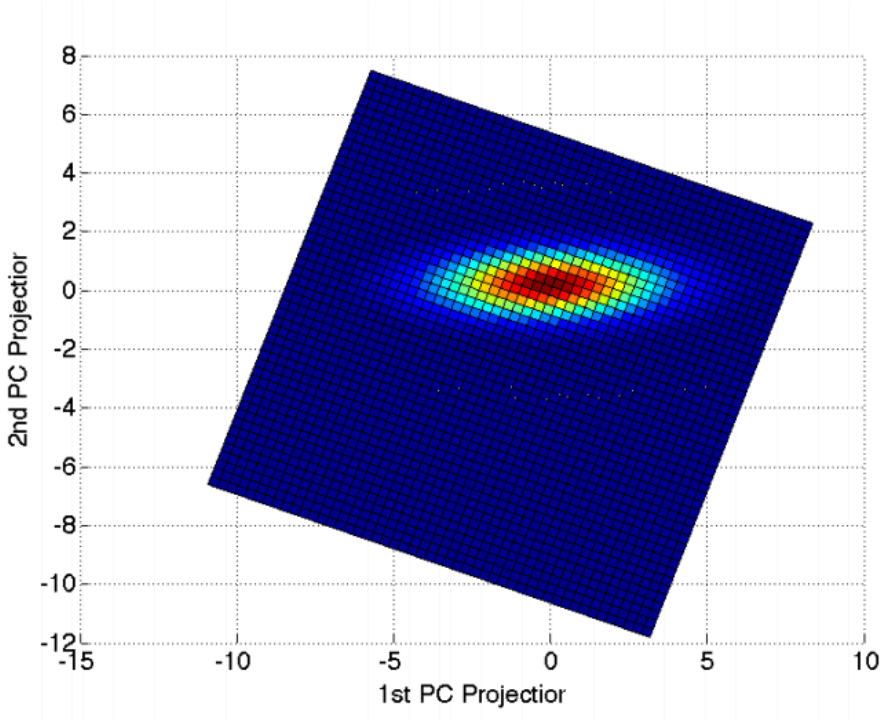
$$\sum c - \lambda c = 0$$

$$\sum c - \lambda c = 0$$

PCA Before and After



Gaussian PDF

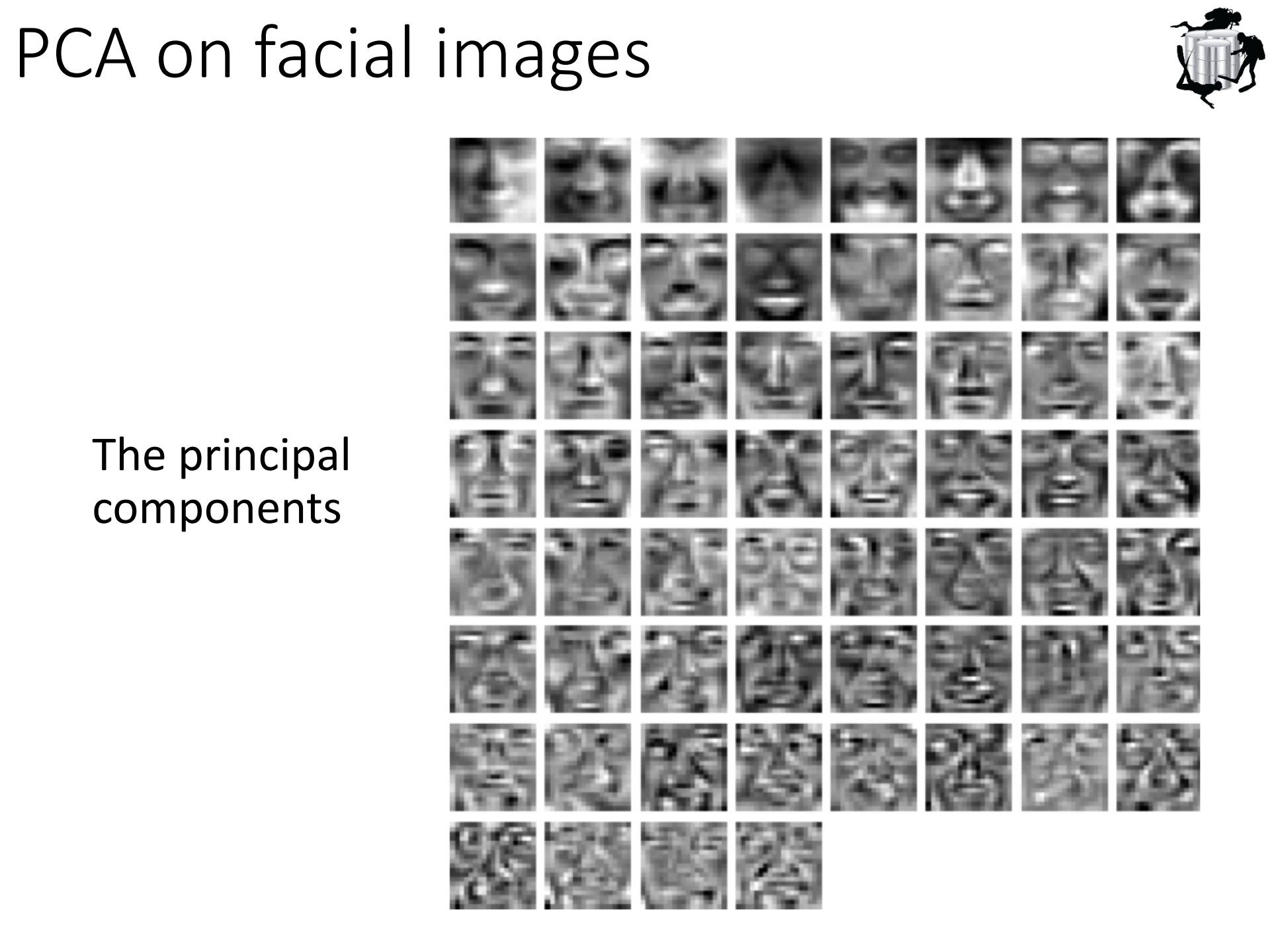


Projection on to PCA vectors

PCA on facial images

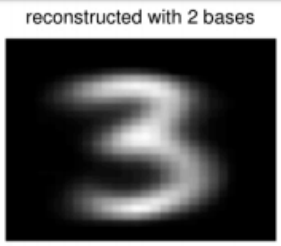
- PCA applied to 2429 19×19 images from CBCL dataset
- Reconstruction with only 3 components:

PCA on facial images

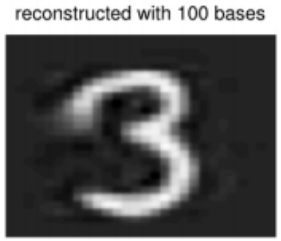


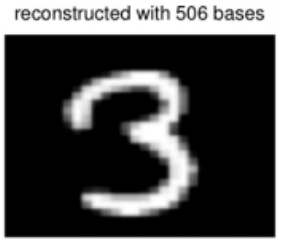
The principal components

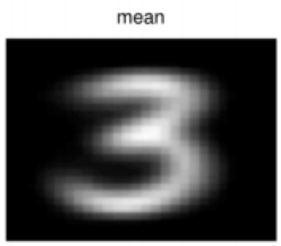
PCA on MNIST

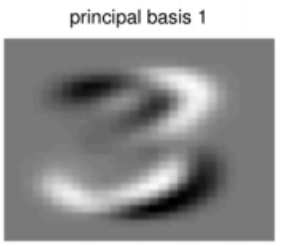

reconstructed with 2 bases

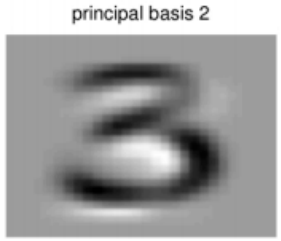

reconstructed with 10 bases

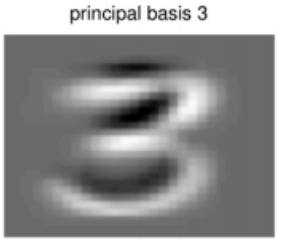

reconstructed with 100 bases


reconstructed with 506 bases

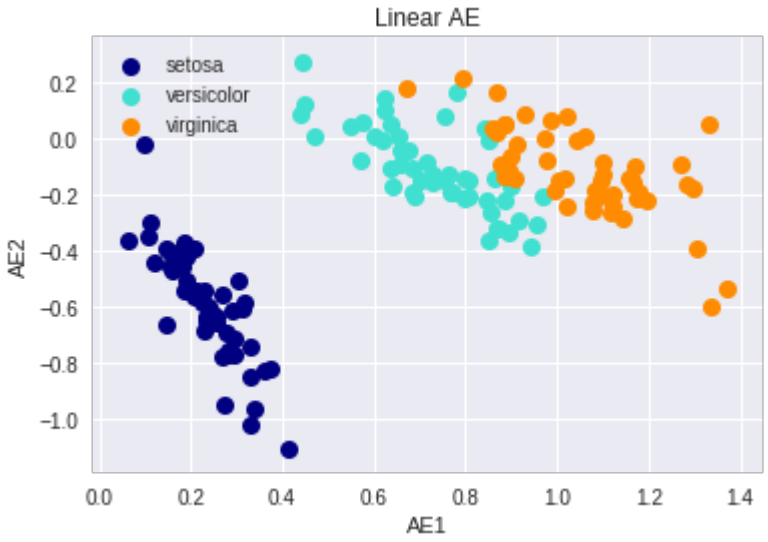

mean


principal basis 1


principal basis 2

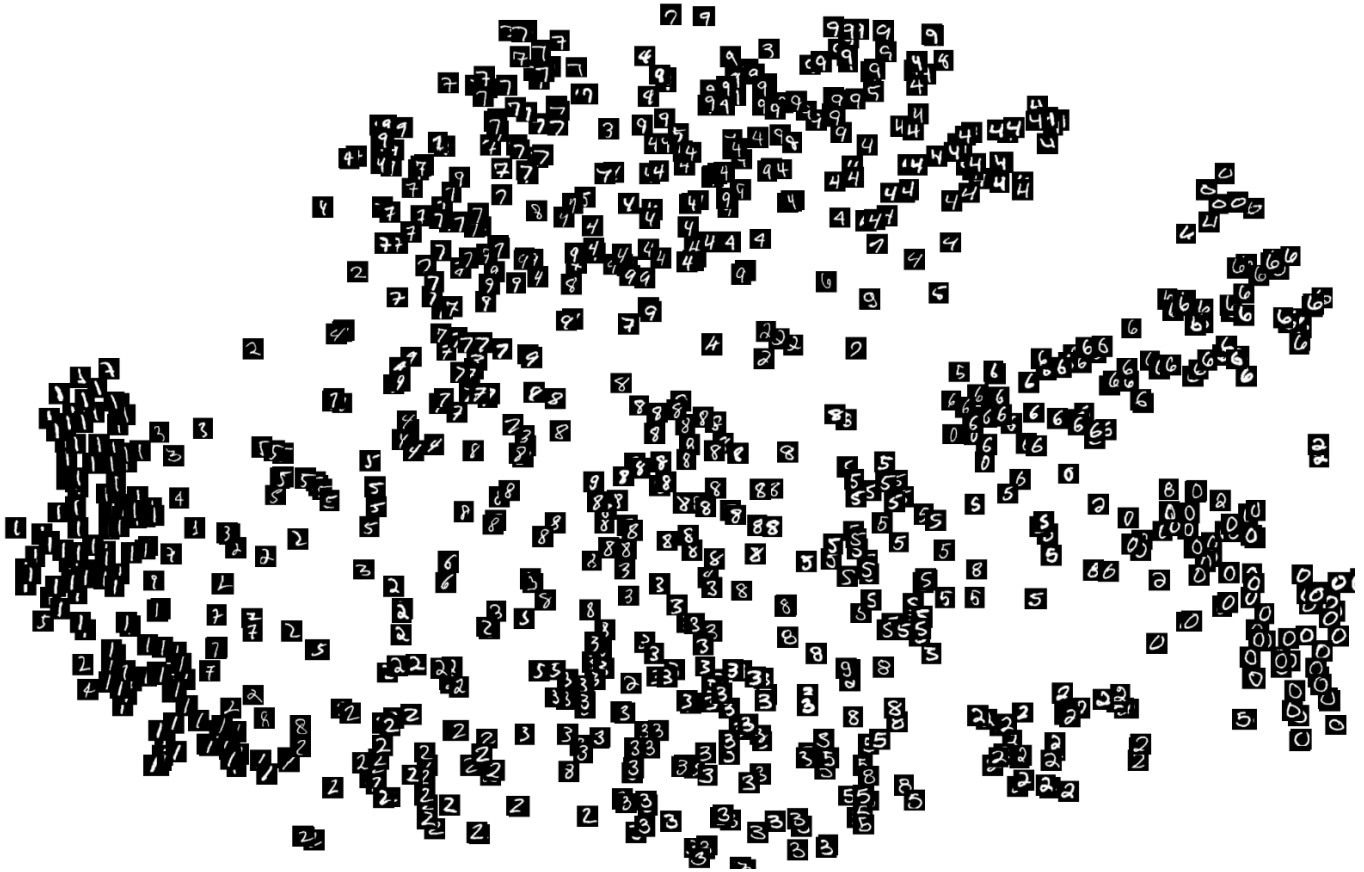

principal basis 3

Linear autoencoders equivalent to PCA

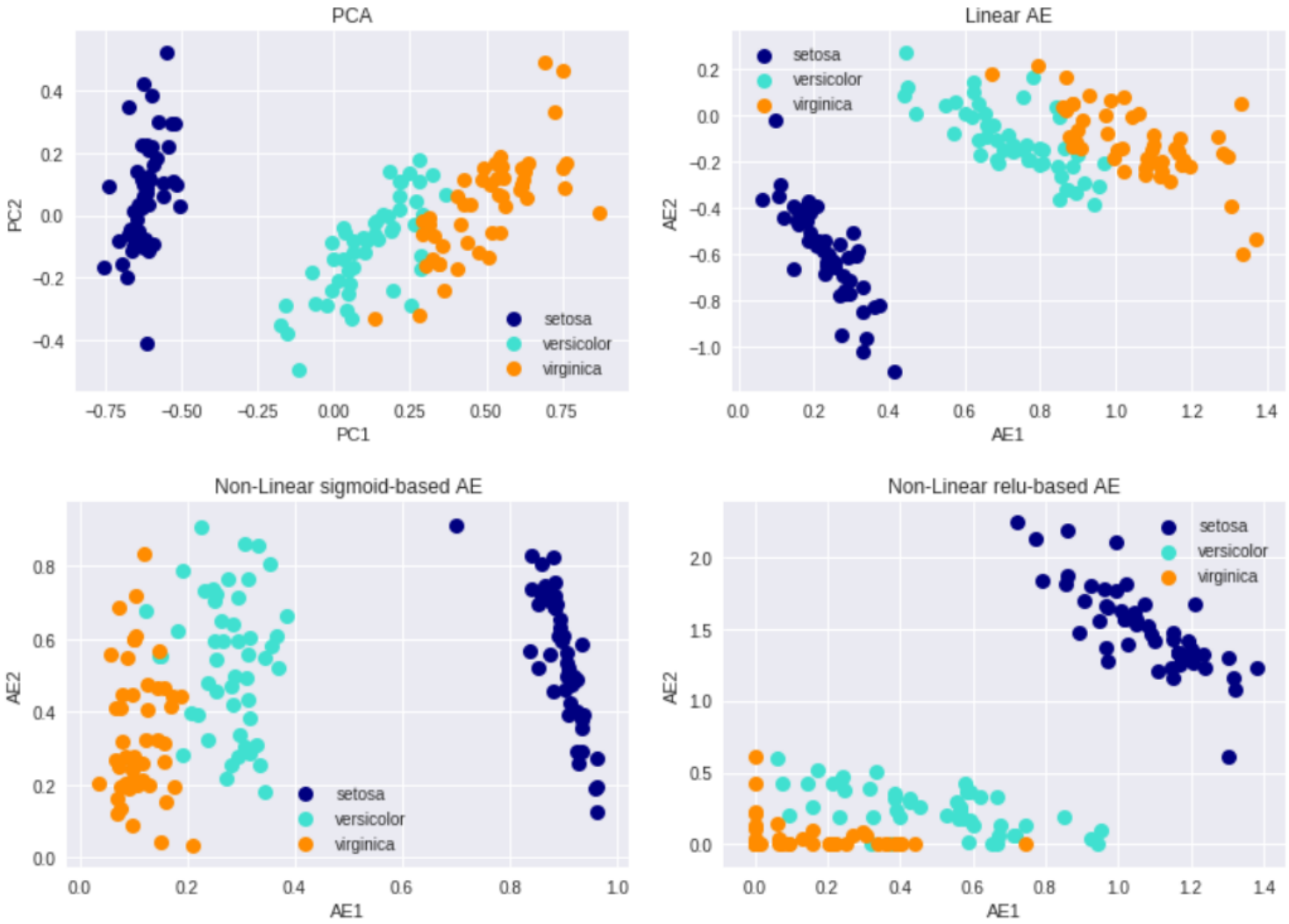


Non-Linear Embedding tSNE

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2/2\sigma_i^2)},$$



Non-Linear Autoencoders come up with non-linear embeddings

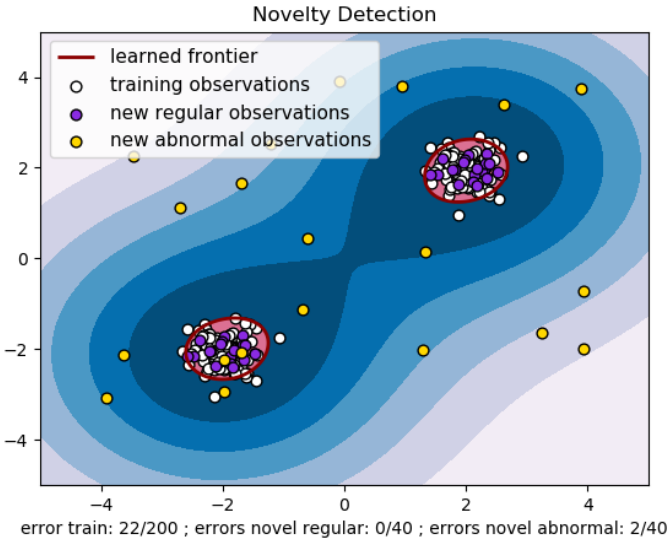


A comparison of all four plots.

Anomaly Detection

Goals

- **Identify** anomalous points
- Project “anomalies” back to “normal” space



General Idea

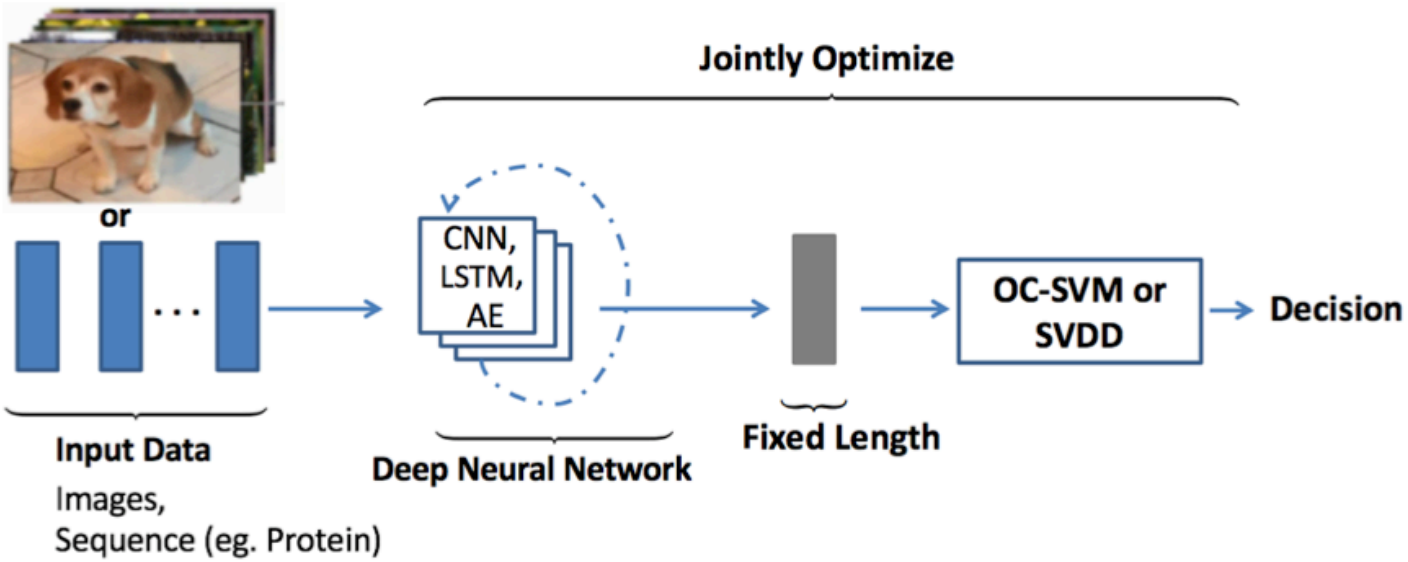
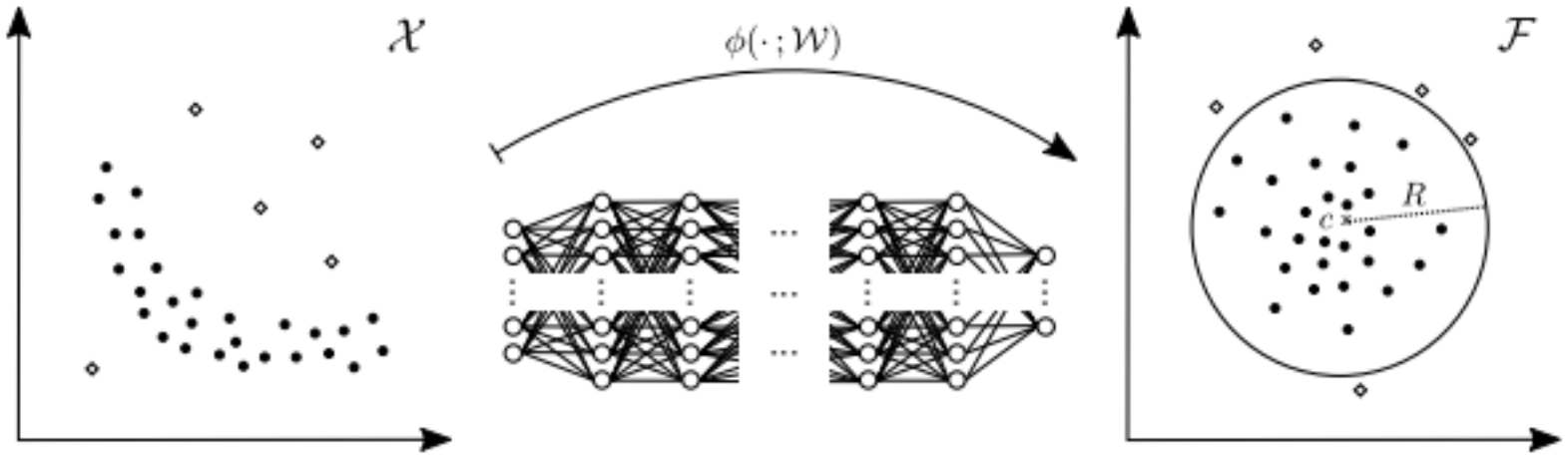


Figure 7: Deep Hybrid Model Architecture.

Deep On—Class Classification



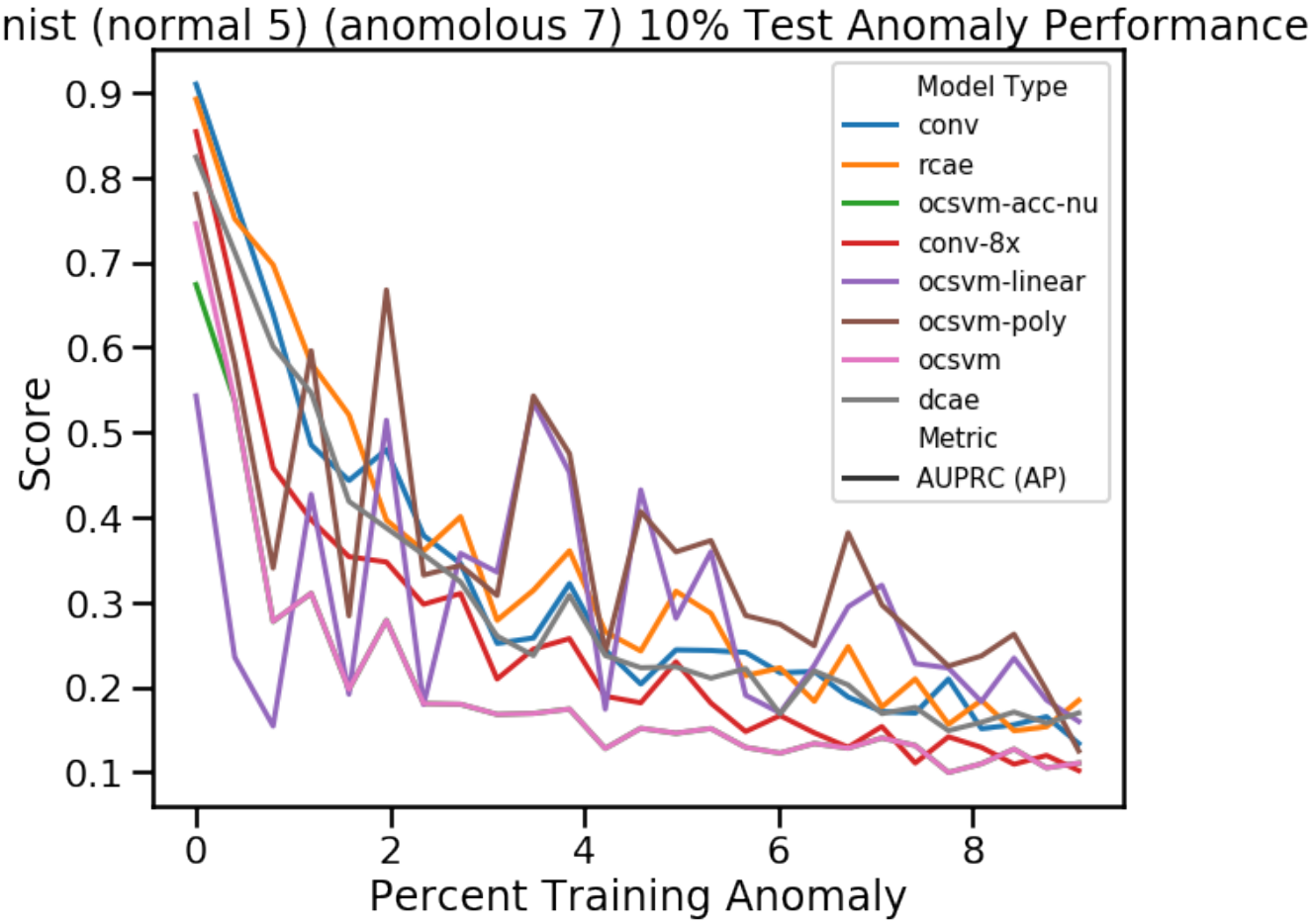
[Ruff et al. 2018]

$$\min_{R,\mathcal{W}} \quad R^2 + \frac{1}{\nu n}\sum_{i=1}^{n}\max\{0, \|\phi(\boldsymbol{x}_i;\mathcal{W}) - \boldsymbol{c}\|^2 - R^2\} + \frac{\lambda}{2}\sum_{\ell=1}^{L}\|\boldsymbol{W}^\ell\|_F^2.$$

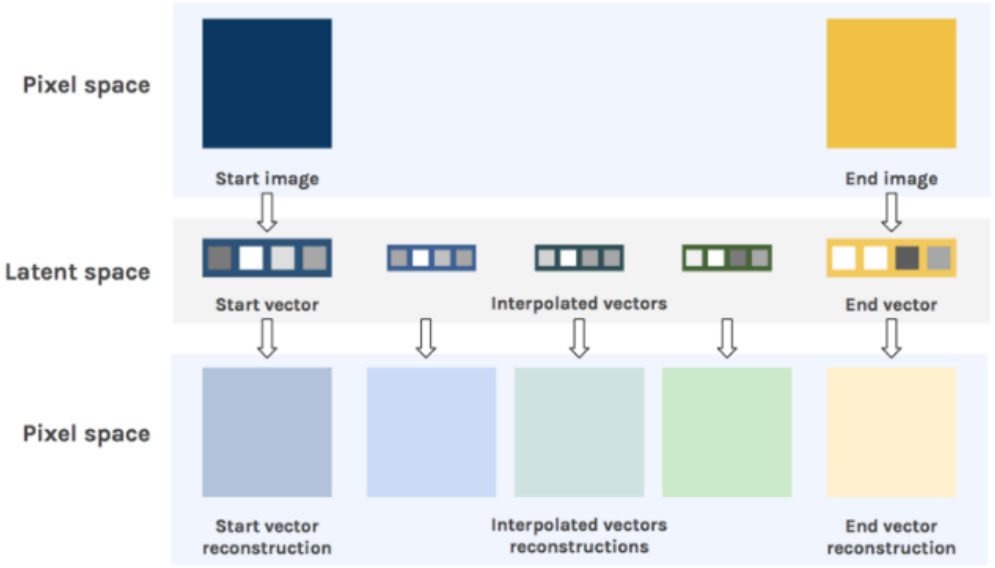
Anomaly score

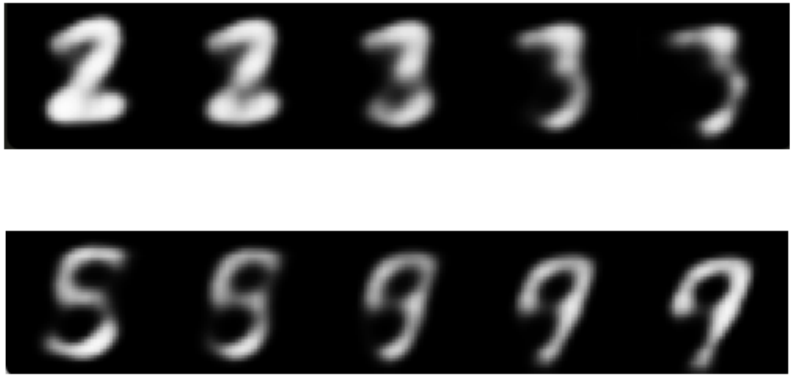
CIFAR

What if Anomalies are part of data?



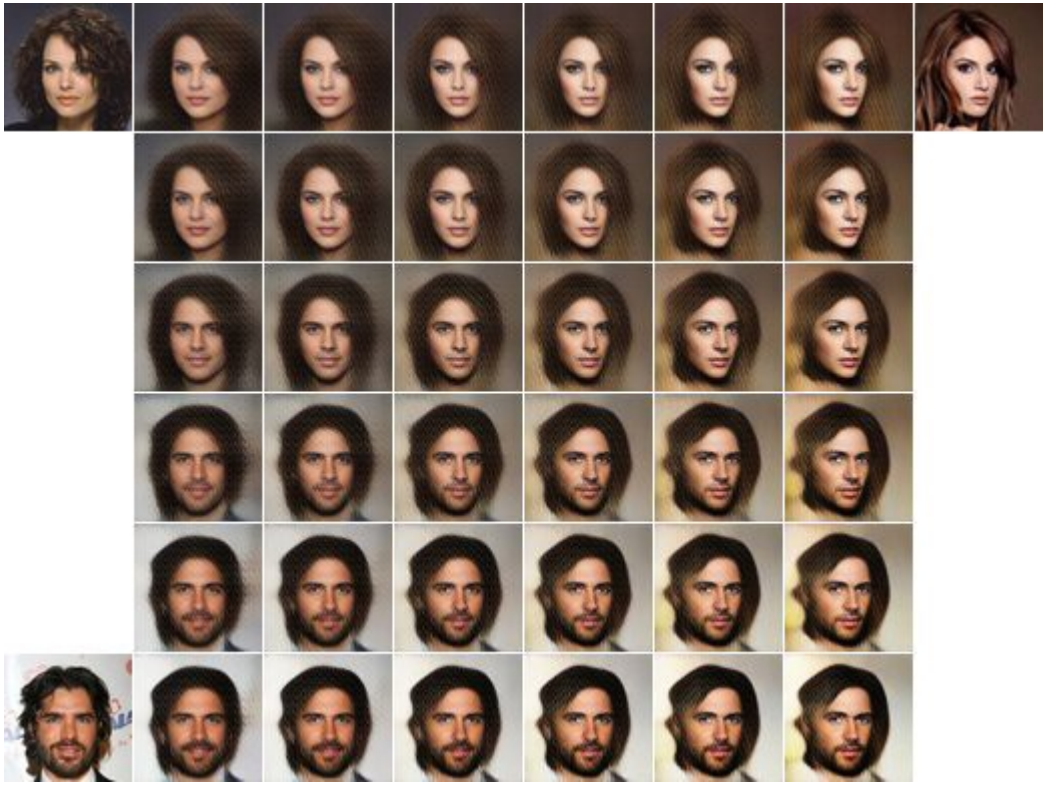
Interpolation in Latent Space



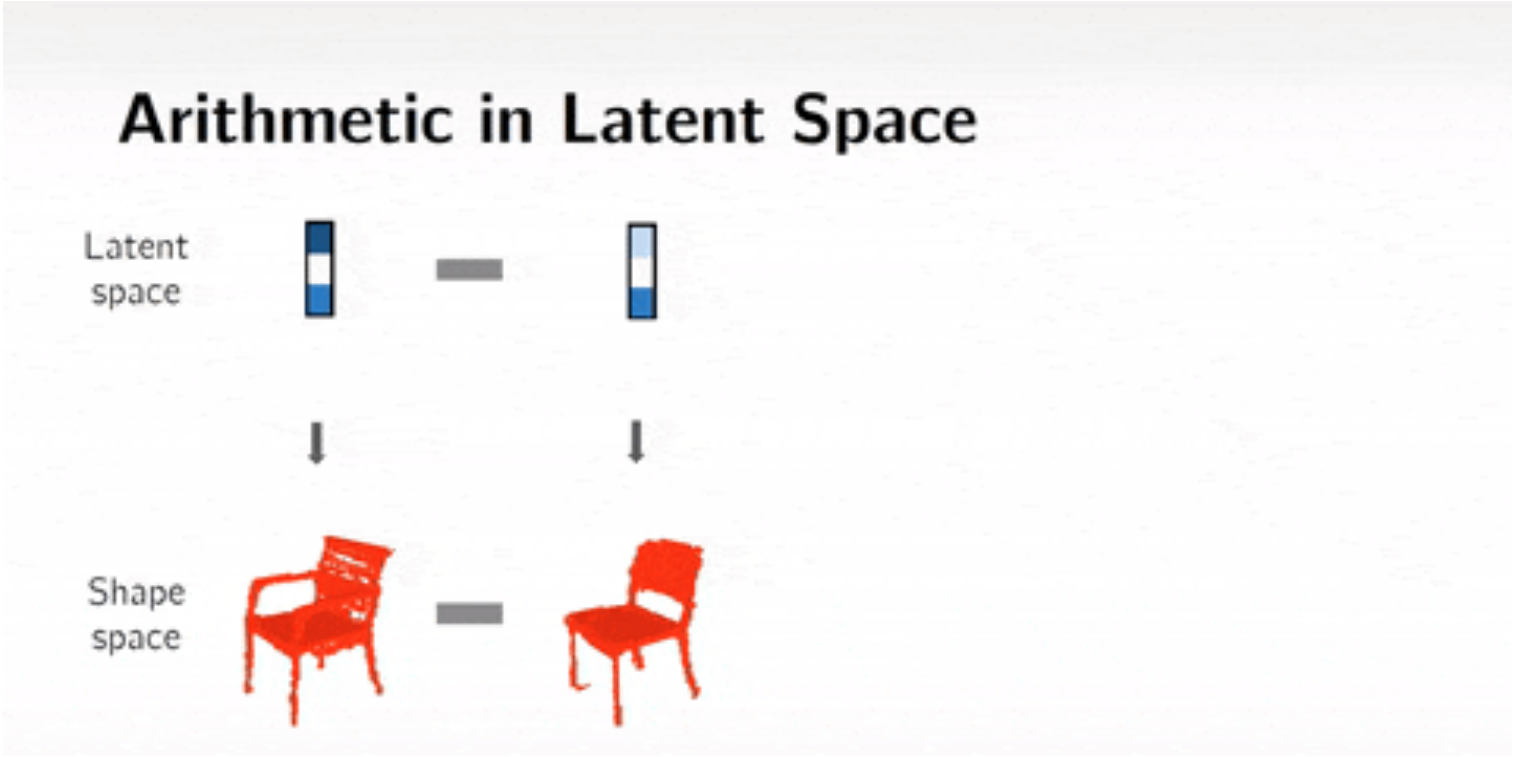


Interpolation in **latent space**

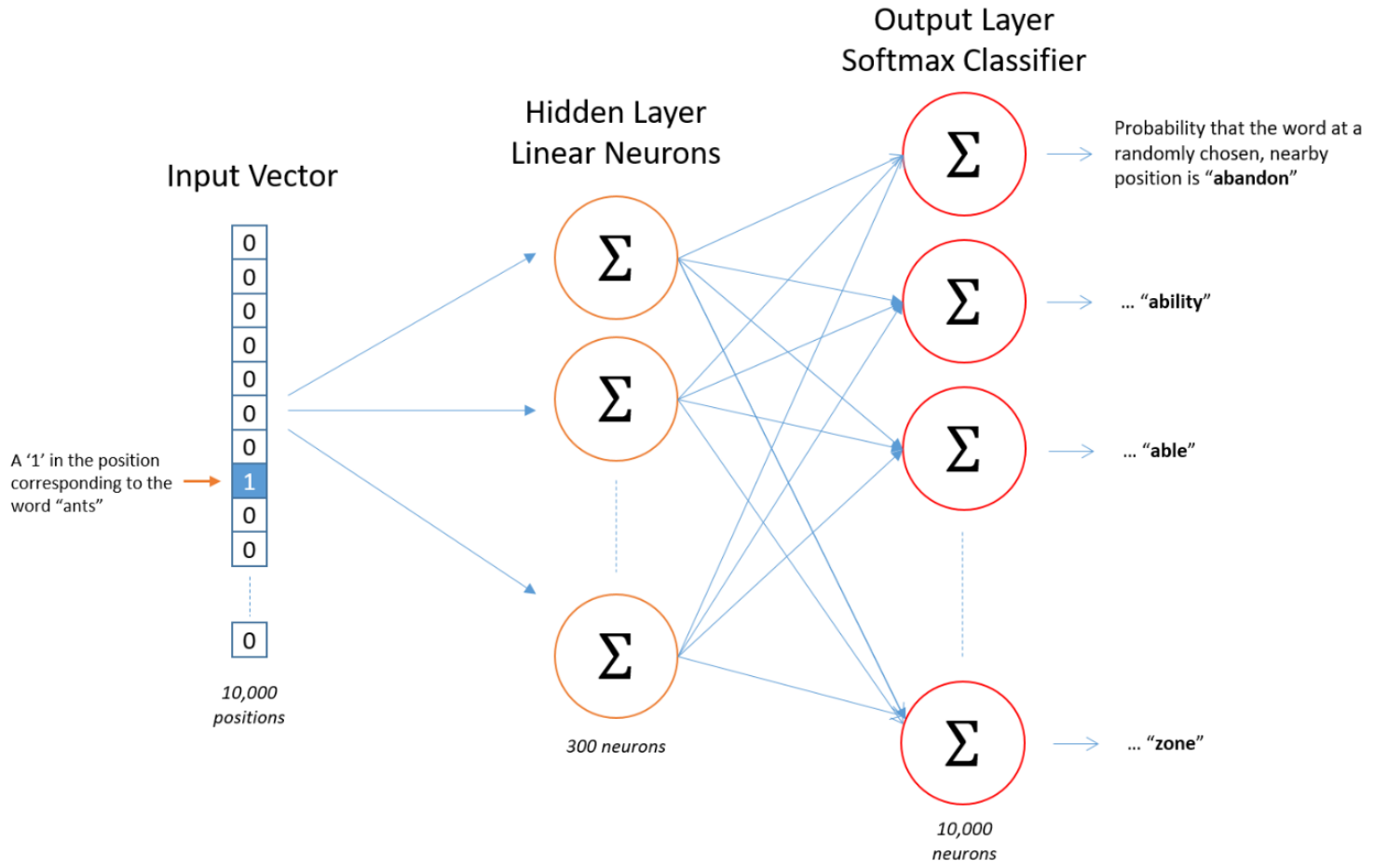
Interpolating Between Images



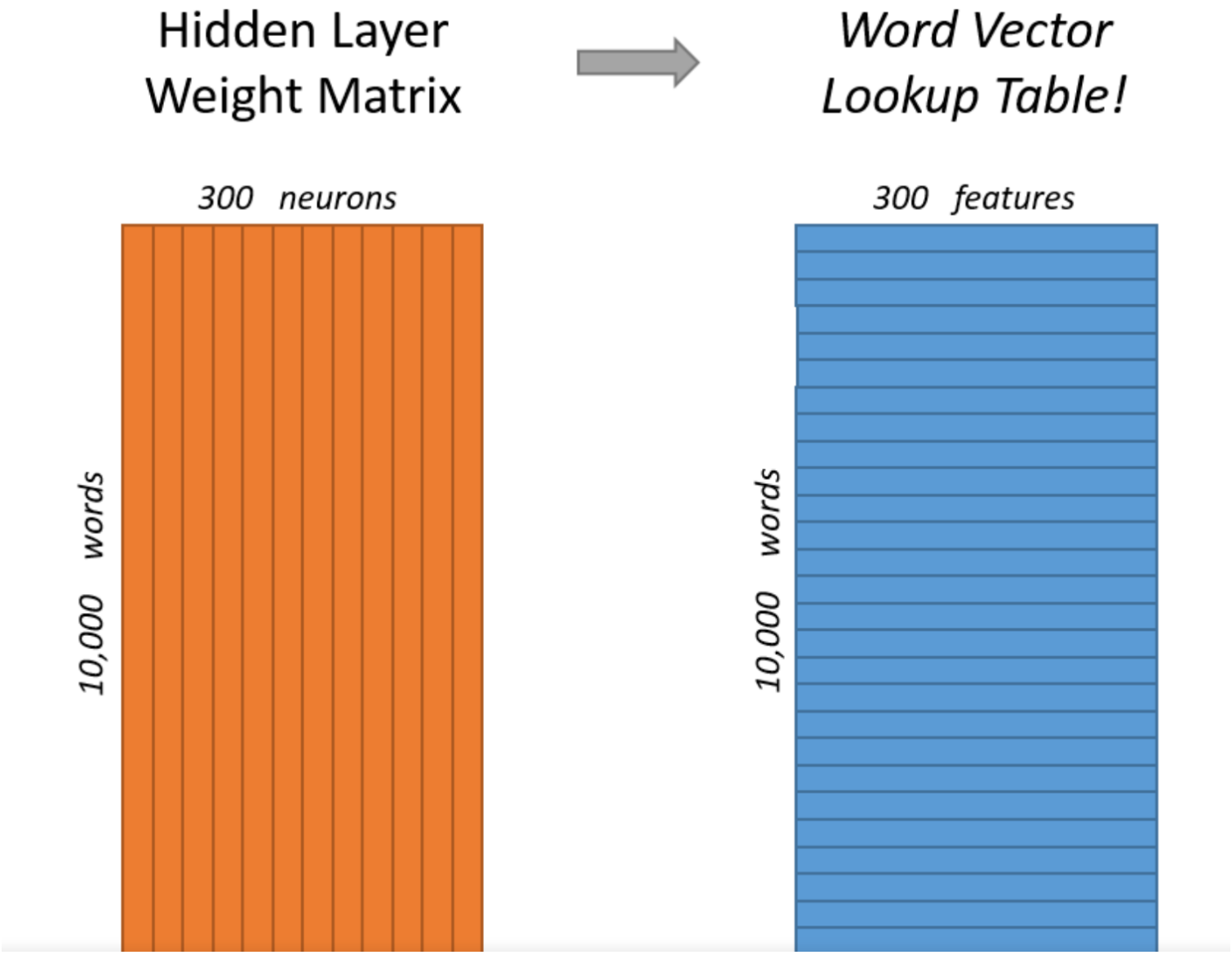
Arithmetic in Latent Space



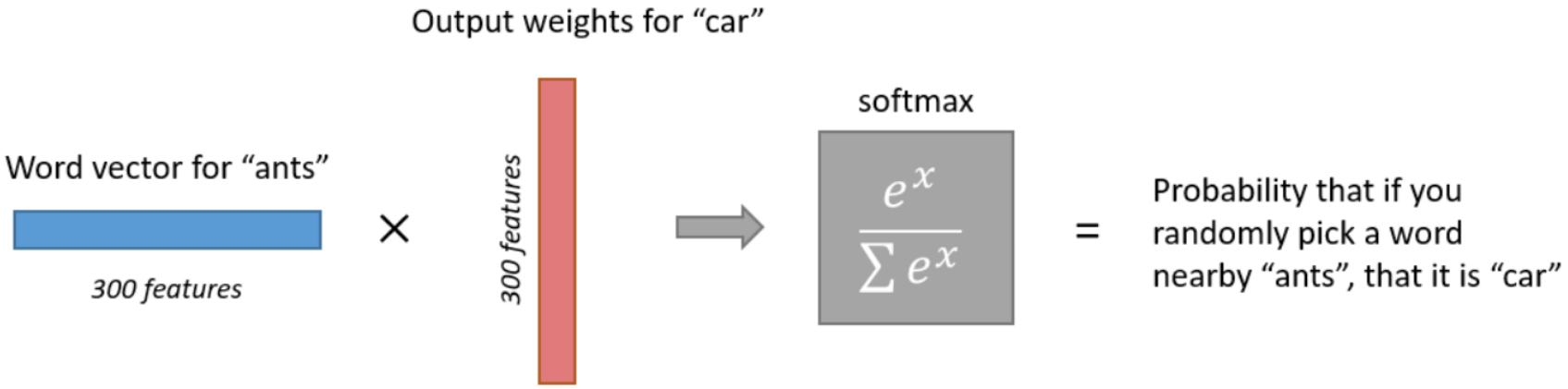
Word vector Embedding



Word vectors

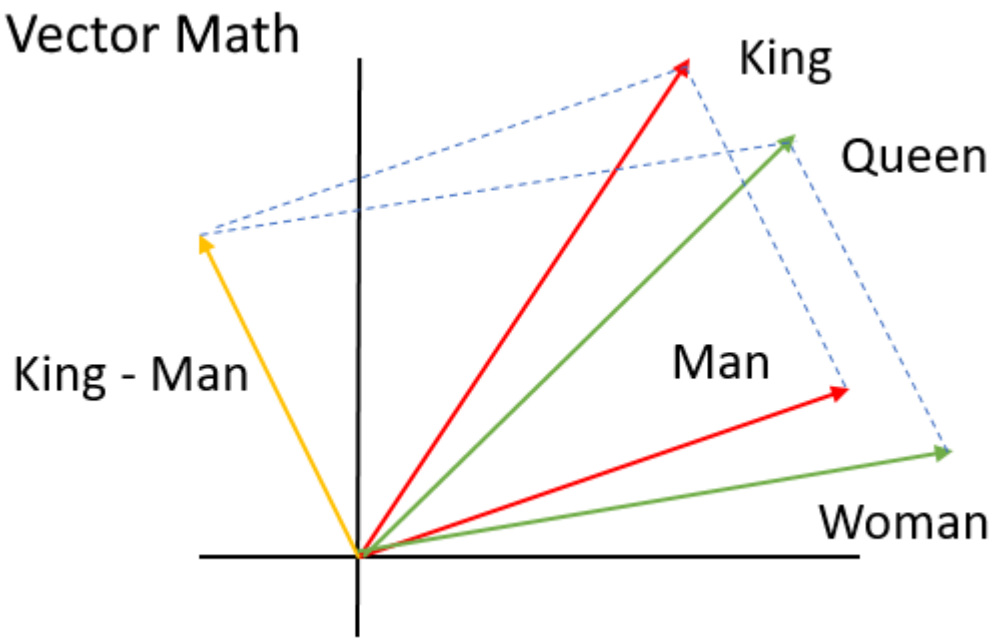


Output calculation

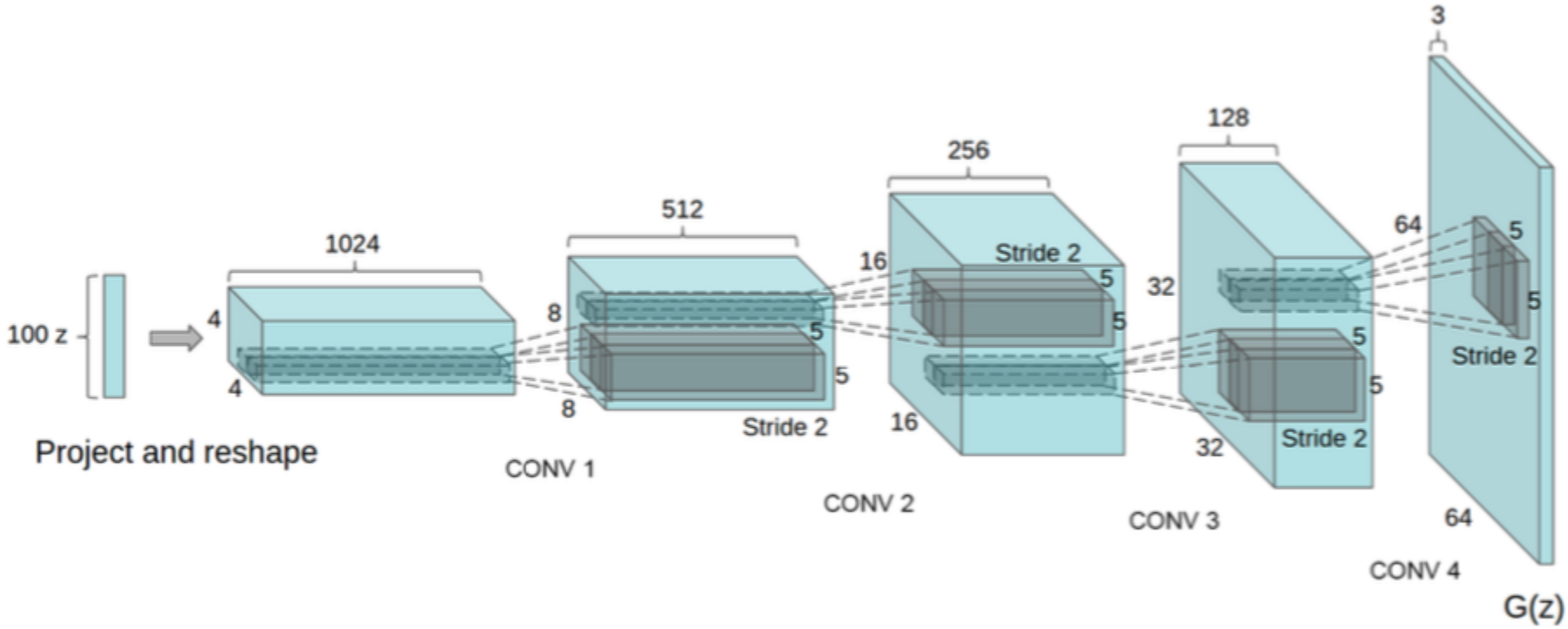


Training on negative samples improves this.

Word-vector Arithmetic



Arithmetic on Images (DCGAN)

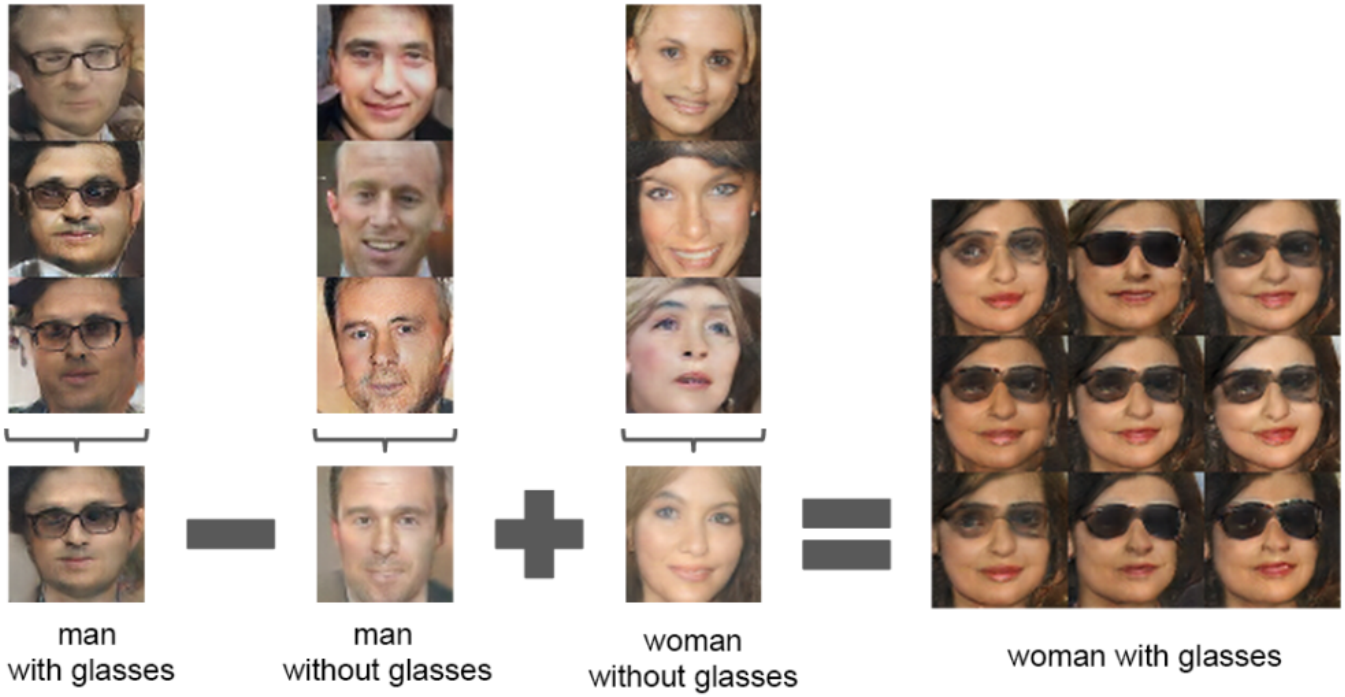


Radford et al 2015

Arithmetic on input noise

- Take z vector for MAN with glasses
- Subtract the vector for MAN
- Add vector for woman

Image Pixel Space



Results of doing the same arithmetic in pixel space

Problem

- Have to find noise vectors that do the right thing
- There are no knobs to turn
- How do we create knobs?
- By creating a latent code

Disentangled Representations

Entangled

Disentangled

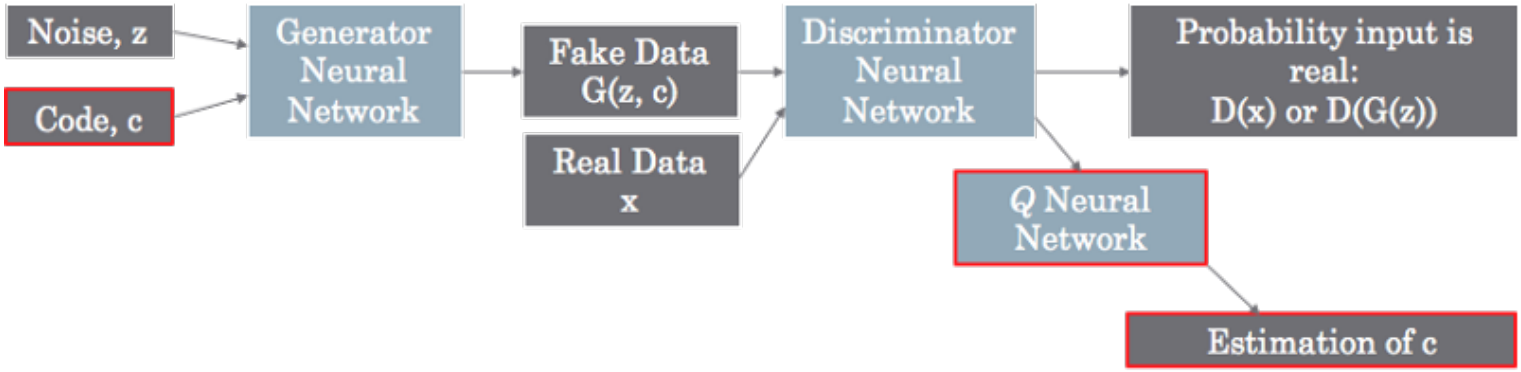
InfoGAN

- Split input into latent code and noise
- Maximize mutual information between code and Generated output

$$\min_G \max_D V_I(D,G) = V(D,G) - \lambda I(c; G(z,c))$$

zHard to do explicitly!

InfoGAN Architecture



$$\min_{G,Q} \max_{D} V_{\text{InfoGAN}}(D, G, Q) = V(D, G) - \lambda L_I(G, Q)$$

Manually tuning latent code *c* produces meaningful changes in output

(a) Varying c_1 on InfoGAN (Digit type)

(b) Varying c_1 on regular GAN (No clear meaning)

(c) Varying c_2 from -2 to 2 on InfoGAN (Rotation)

(d) Varying c_3 from -2 to 2 on InfoGAN (Width)

Manually tuning latent code *c* produces meaningful changes in output

(a) Azimuth (pose)

(b) Elevation

(c) Lighting

(d) Wide or Narrow

Manually tuning latent code *c* produces meaningful changes in output

(a) Rotation

(b) Width

Further reading

- https://towardsdatascience.com/infogan-generative-adversarial-networks-part-iii-380c0c6712cd
- https://towardsdatascience.com/generative-adversarial-networks-part-ii-6212f7755c1f