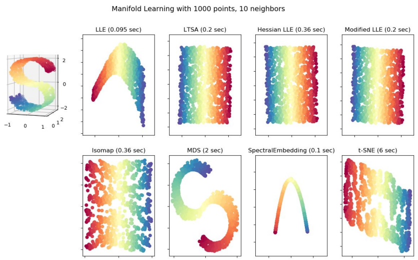
Autoencoder

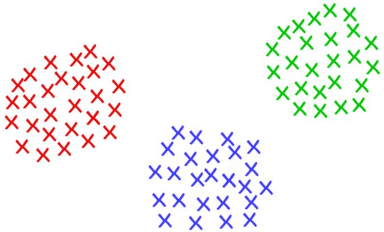
Autoencoder

Unsupervised Learning

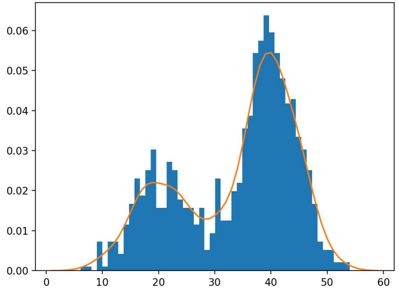
- Given the data x without labels
- Goal: Learn hidden structure(low dimension) from



Representation Learning
Data lies on a low-dimensional manifold



Clustering
Group data points based their similarity



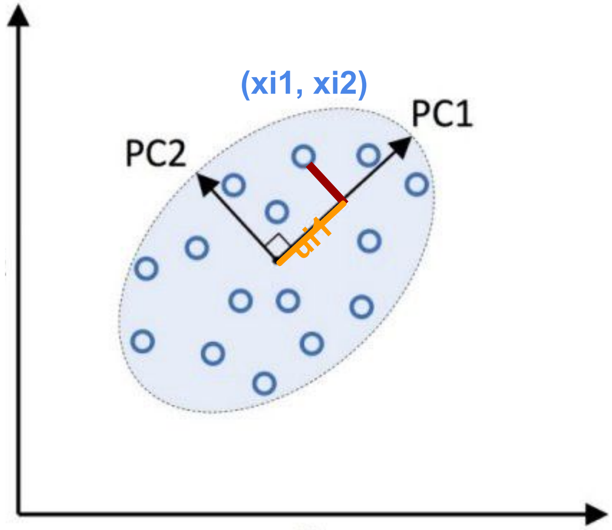
Density Estimation
Estimate data probability p(x) from data x1, x2, ...,, xn

Principal Component Analysis: Maximize Variance

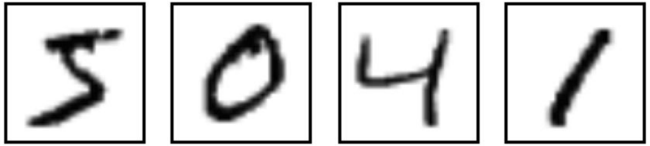
PCA aims to find the directions of maximum variance in high-dimensional data and projects it onto a new subspace with equal or fewer dimensions than the original one

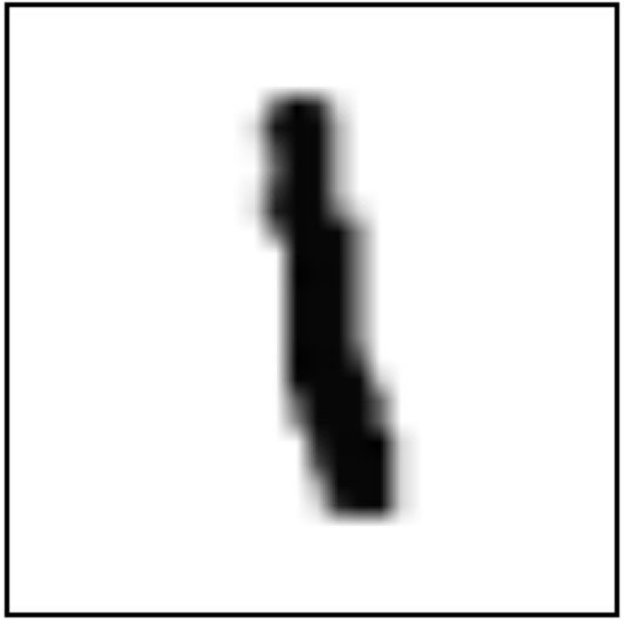
Original Space | Projection Matrix | New/Latent Space

$$\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ \vdots & \vdots \\ x_{n1} & x_{n2} \end{bmatrix} \times \underbrace{\begin{bmatrix} w_{11} \\ w_{21} \end{bmatrix}}_{\text{PC1}} = \begin{bmatrix} u_{11} \\ u_{21} \\ \vdots \\ u_{n1} \end{bmatrix}$$



MNIST Dataset

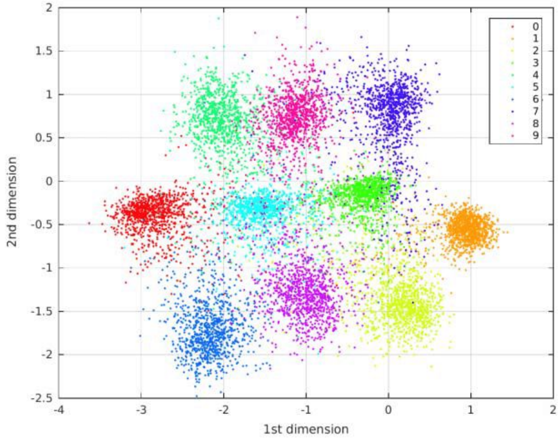




$$\cong \begin{bmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&.6&.8&0&0&0&0&0&0\\
0&0&0&0&0&0&.7&1&0&0&0&0&0&0\\
0&0&0&0&0&0&.7&1&0&0&0&0&0&0\\
0&0&0&0&0&0&.5&1&.4&0&0&0&0&0\\
0&0&0&0&0&0&0&1&.4&0&0&0&0&0\\
0&0&0&0&0&0&0&1&.4&0&0&0&0&0\\
0&0&0&0&0&0&0&1&.7&0&0&0&0&0\\
0&0&0&0&0&0&0&1&1&0&0&0&0&0\\
0&0&0&0&0&0&0&.9&1&.1&0&0&0&0\\
0&0&0&0&0&0&0&.3&1&.1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{bmatrix}$$

PCA for MNIST Visualization

- Each image has 28 by 28 pixels -> 28 by 28 matrix -> 784 dimensional vector
- Using PCA, find a project matrix $\mathbf{W} \in R^{784 \times 2}$
- After projection, each image can be encoded into a 2-Dimensional space



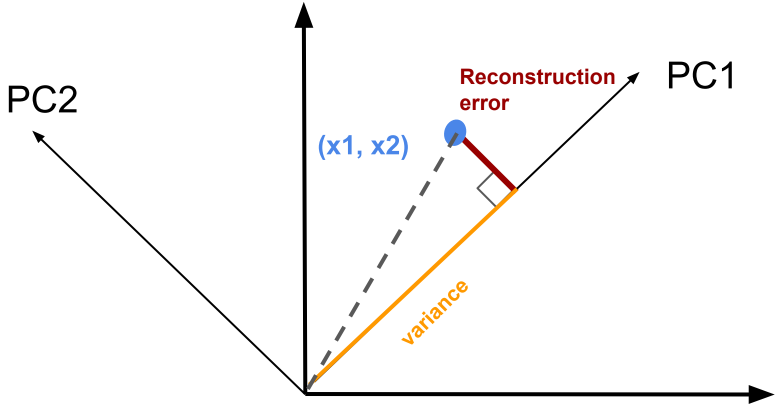
Principal Component Analysis: Minimize Reconstruction Error

PCA aims to find a linear subspace that minimize the distance of the projection in a least-square sense

minimize $\mathbf{W}$ $$||\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{W}^T||_F^2$$

subject to $$\mathbf{W}^T\mathbf{W} = I$$

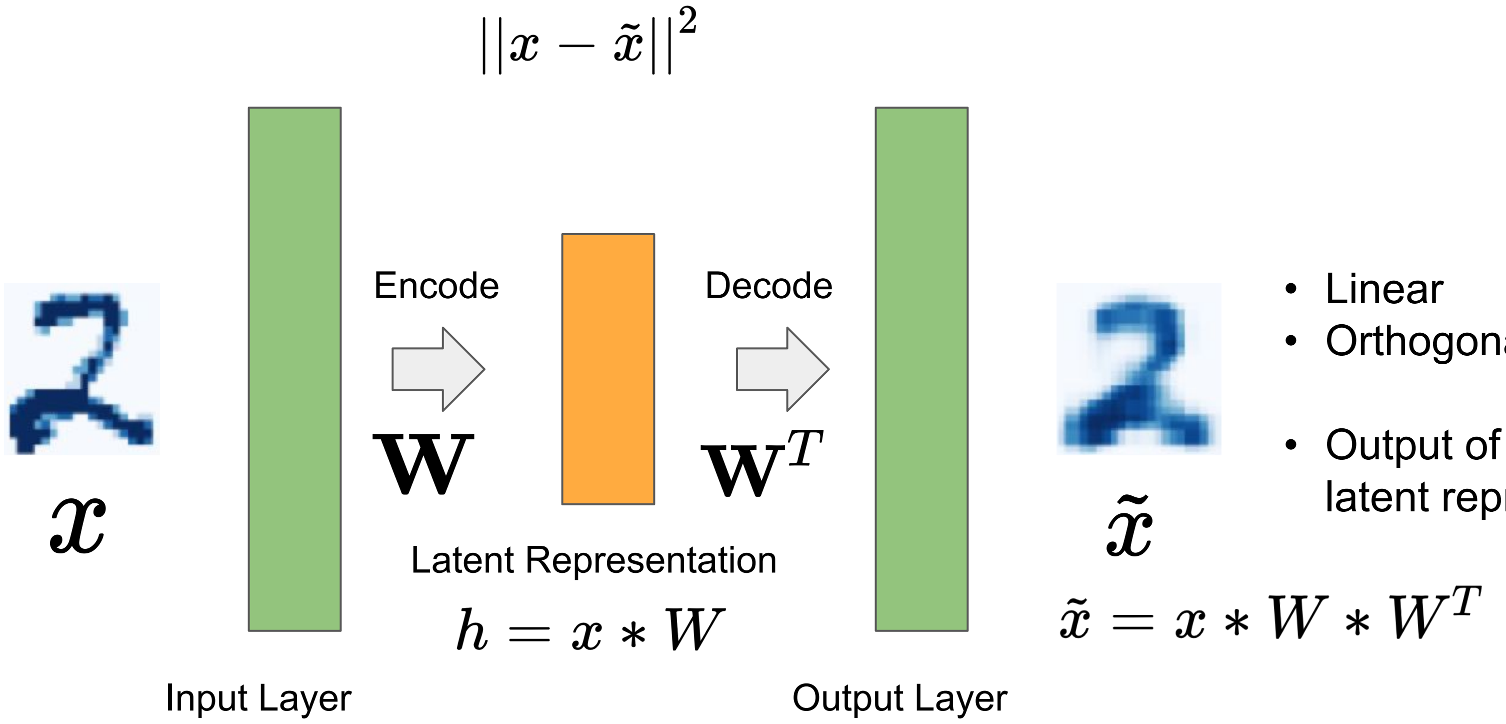
W's shape is (d, h) and h < d



Reconstruction Error + Variance = Constant

minimize *maximiz*

Principal Component Analysis



- Linear
- Orthogonal Columns in W

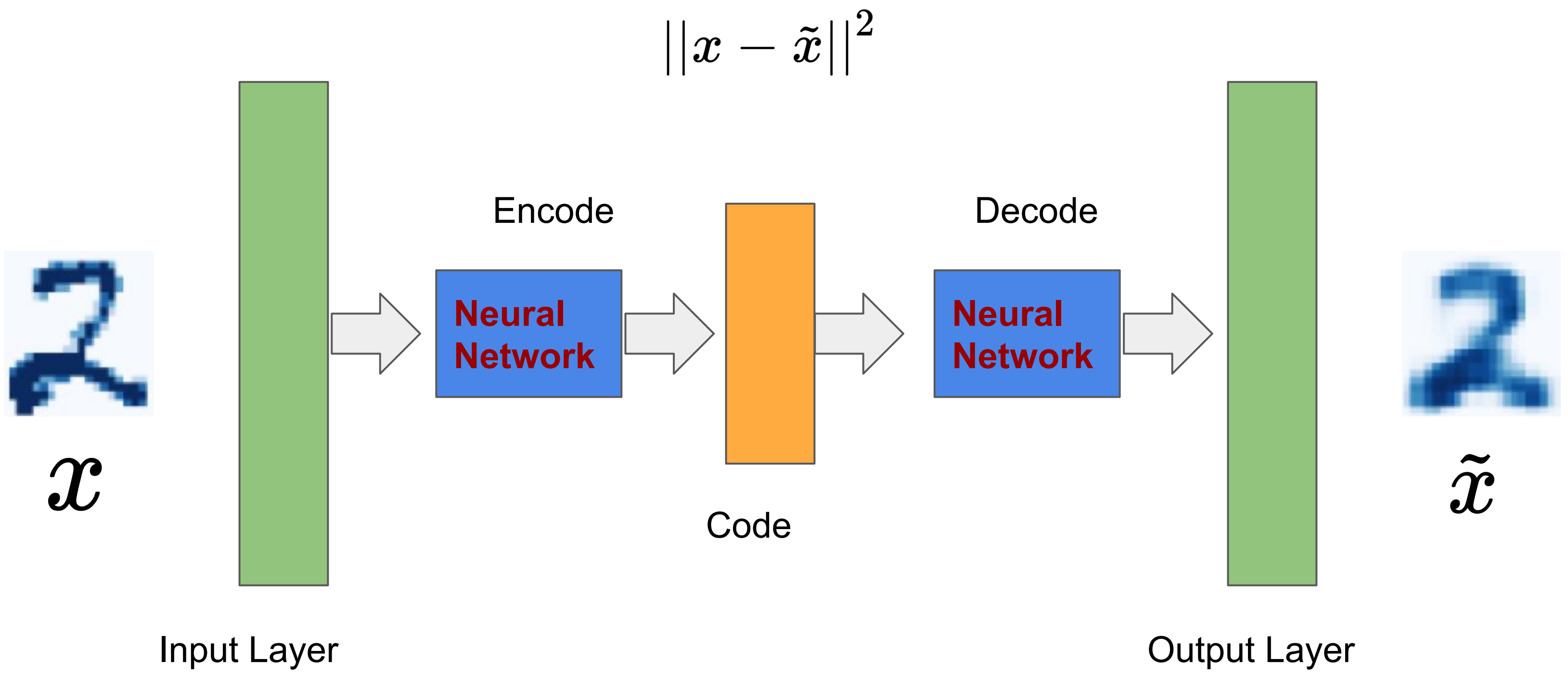
- Output of hidden layer is the latent representation

More Details on PCA: http://www.cs.cornell.edu/courses/cs4786/2016sp/lectures/lec03.pdf

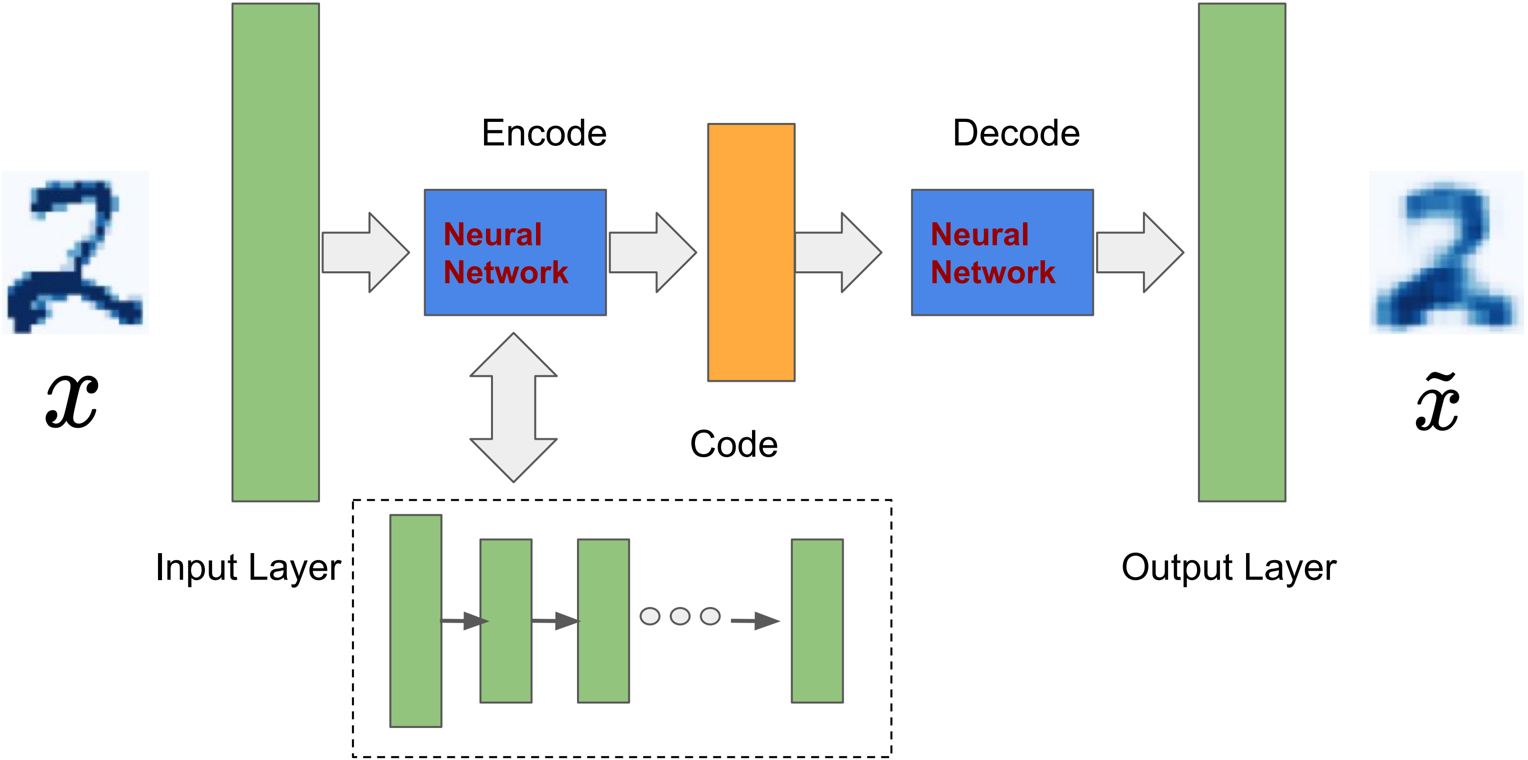
- Non-linear relationship between original representation and latent features

- Which machine learning model to use for **nonlinear approximation**?

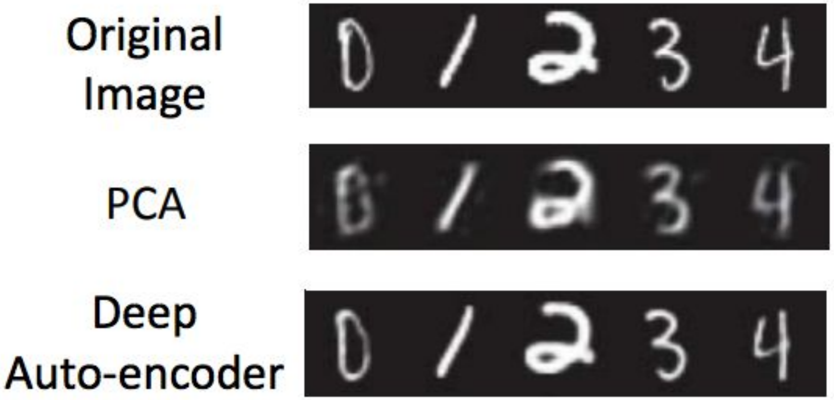
Autoencoder: NonLinear

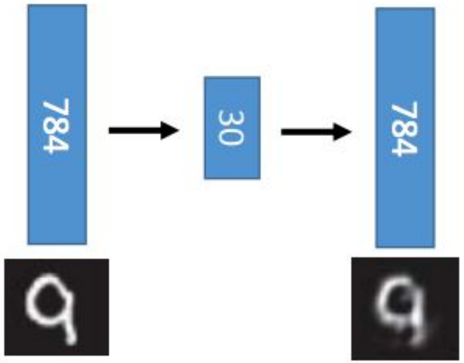


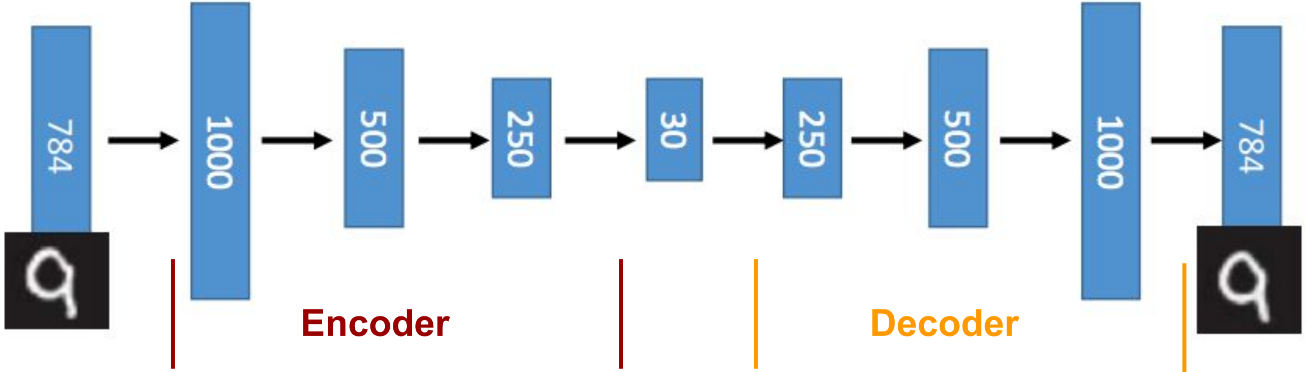
Deep Autoencoder



Deep Autoencoder vs PCA

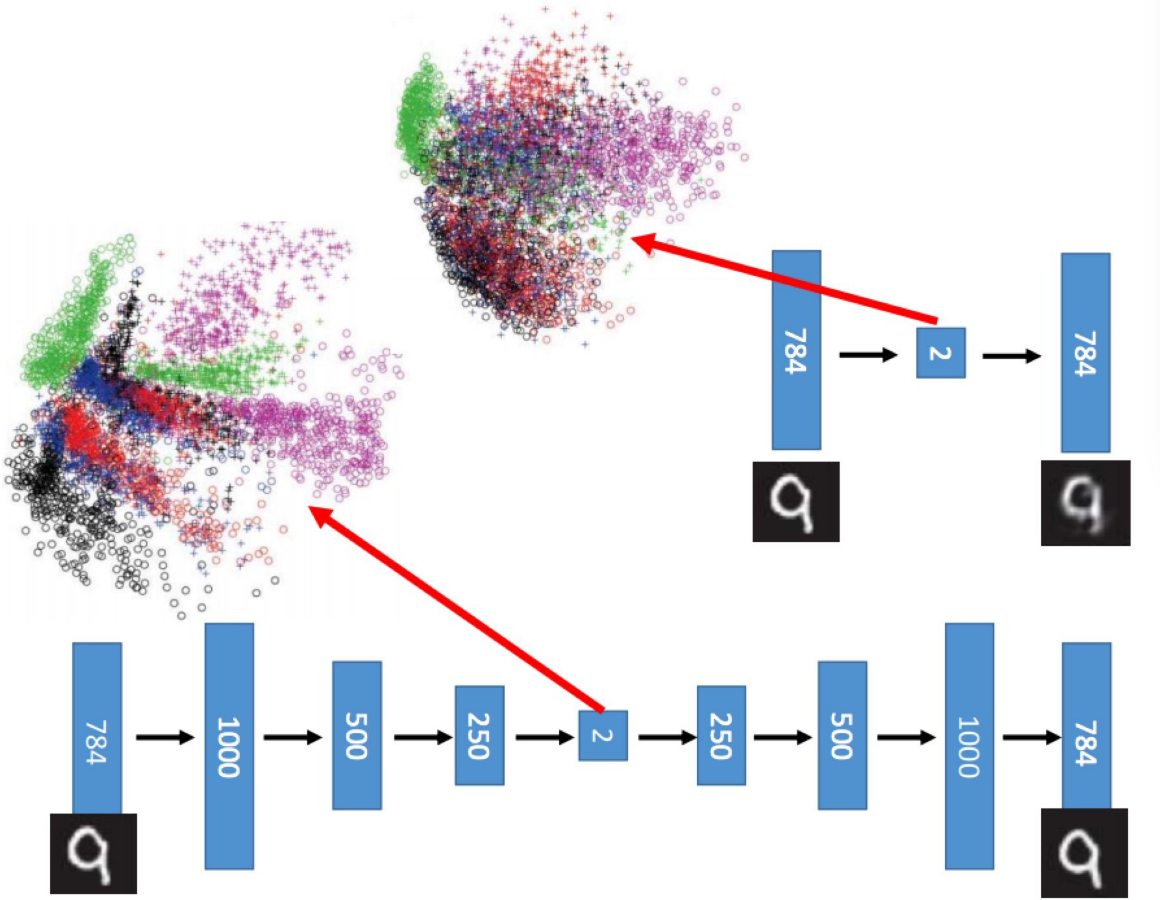




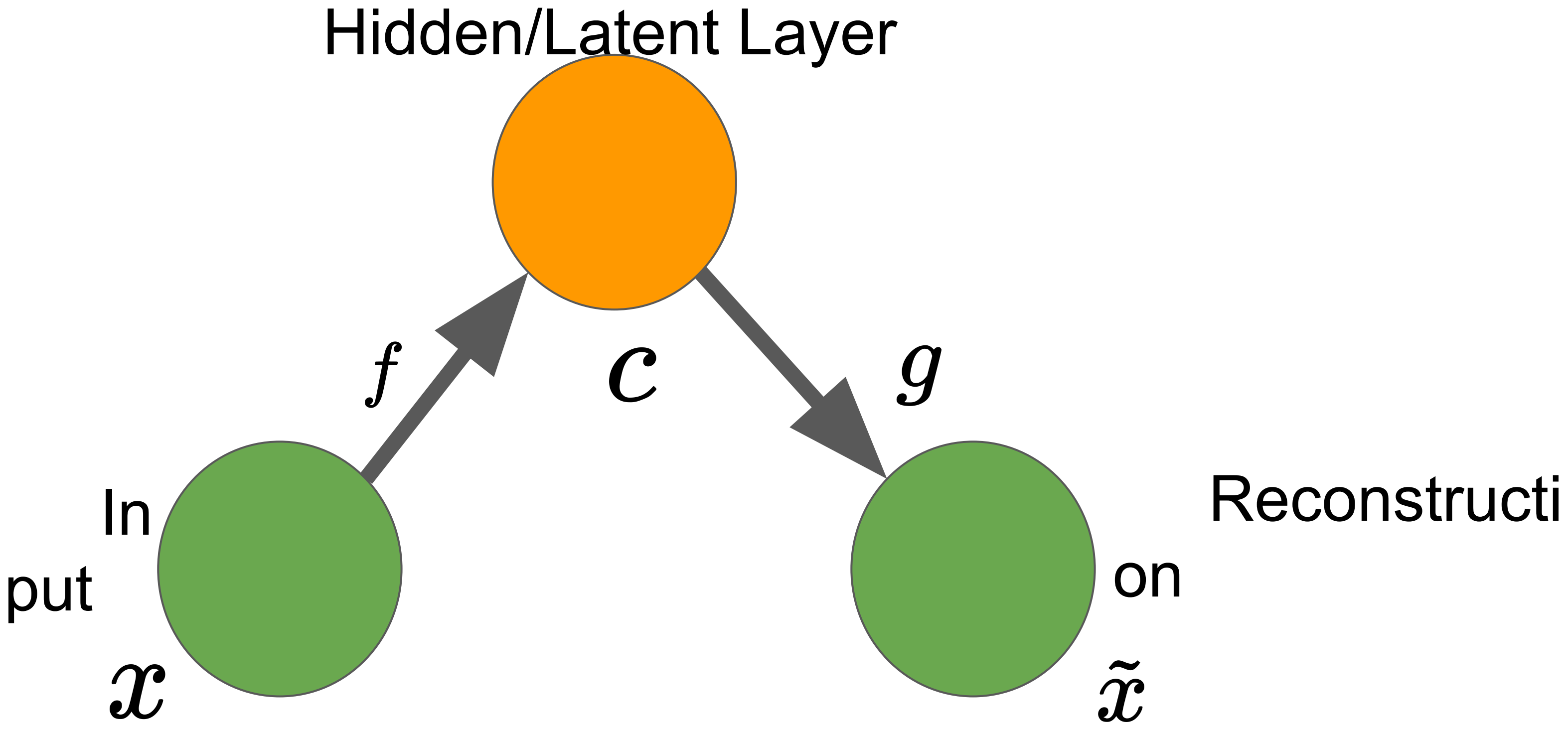


Symmetric Structure

Deep Autoencoder vs PCA



Structure of Autoencoder

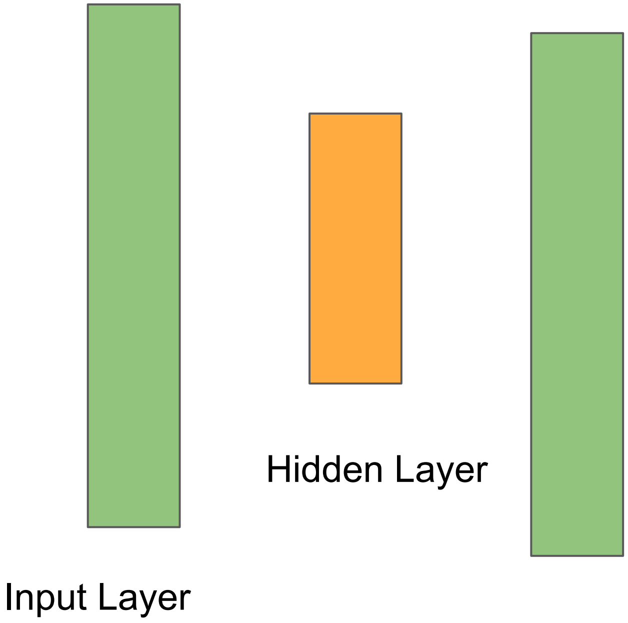


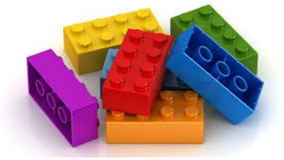
Undercomplete Autoencoder

- Simply copy input to output without learning anything useful
    - The autoencoder just mimic the identity function
    - Reconstruct the training data perfectly
    - Overfitting
- To avoid the above issue, we should use undercomplete autoencoder
    - The hidden layer size c is small compared to the original feature dimensionality

Sandwich Architecture in Autoencoder

- Forcing c (hidden layer size) is less than d (the input layer size)
    - Learn the important features
    - Information bottleneck:
        - A kind of trade-off between compression and retaining information





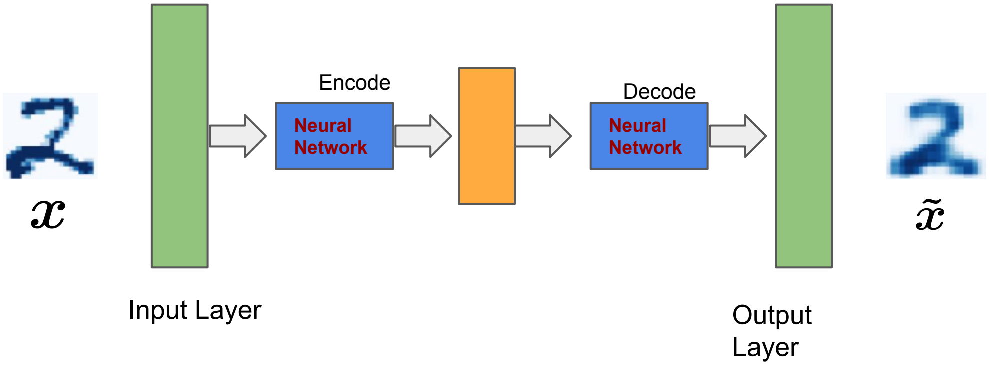
Original **6** Bricks

Can we use only **4** bricks to rebuild the previous shape?

Optimization Targets

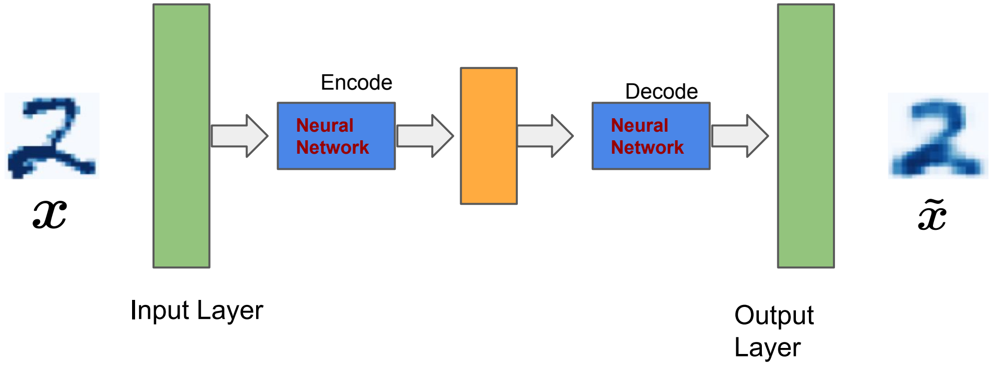
- For Autoencoder, the training objective is to minimize $||x - \tilde{x}||^2$

- Hidden representation is what we really want to learn



Unsupervised or Self-supervised?

- Autoencoder is one kind of self-supervised learning
- Input is x, target is x
- **Pretend there is part of the input you do not know and predict that**
- Word2vec



Build Autoencoders in Keras

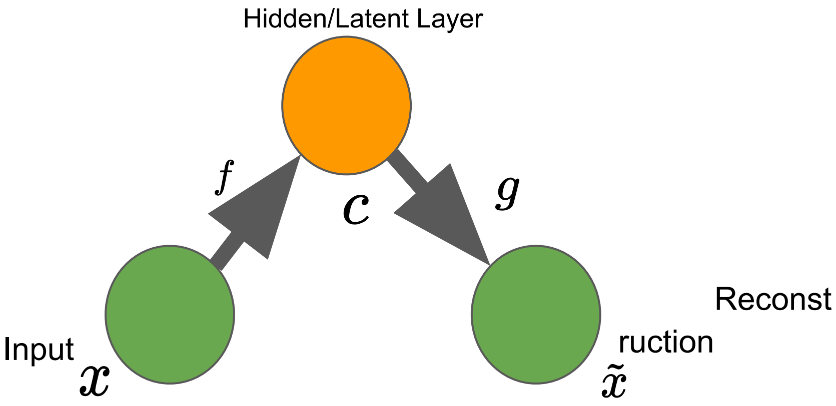
https://blog.keras.io/building-autoencoders-in-keras.html

Regularized Autoencoder

Add constraints in case the identity transformation is learned, i.e., overfitting

Sparse Autoencoders

- Constrain on c that penalizes it from dense
- Regularization **on output of encoder, not parameters**



$$L(x, g(f(x))) + \Omega(c)$$

- `kernel_regularizer` : instance of `keras.regularizers.Regularizer`
- `bias_regularizer` : instance of `keras.regularizers.Regularizer`
- `activity_regularizer` : instance of `keras.regularizers.Regularizer`

Example

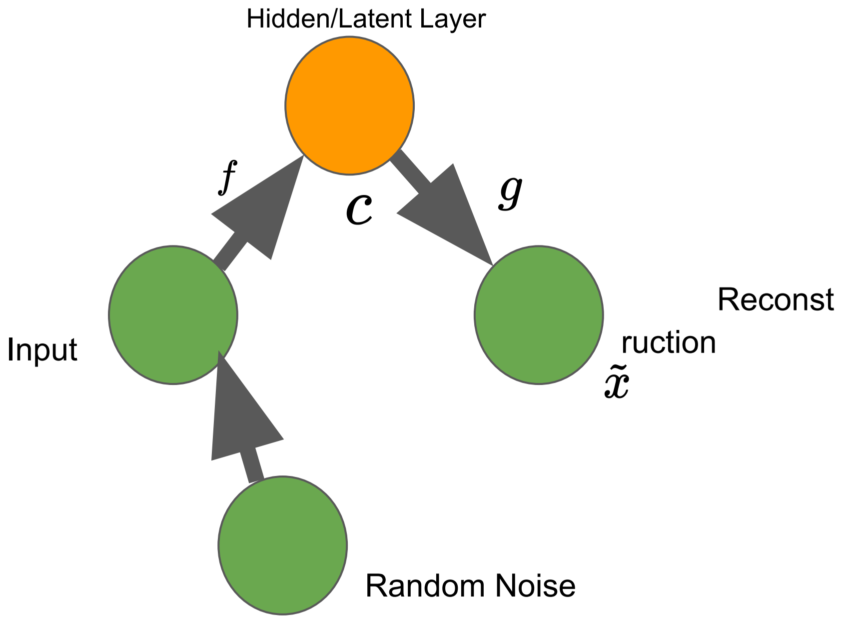
```
from keras import regularizers
model.add(Dense(64, input_dim=64,
                kernel_regularizer=regularizers.l2(0.01),
                activity_regularizer=regularizers.l1(0.01)))
```

Denoising Autoencoders

- Add noise into original data points
- Still reconstruct the original data points

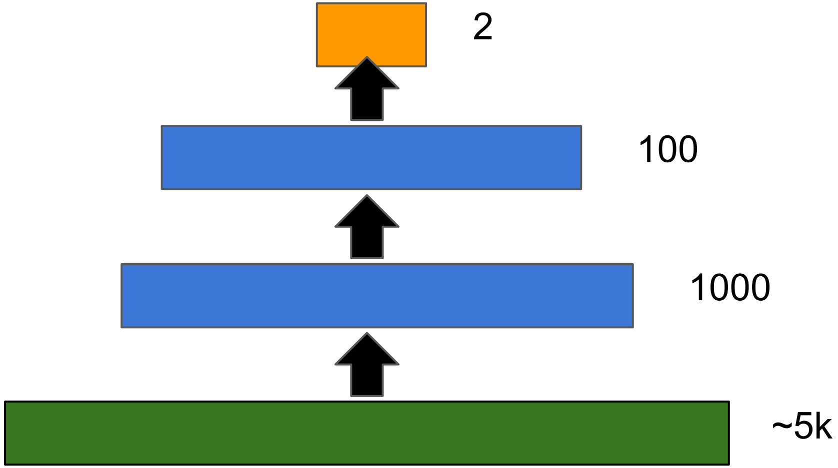
$$L(x, g(f(\bar{x})))$$

Corrupted copy of x

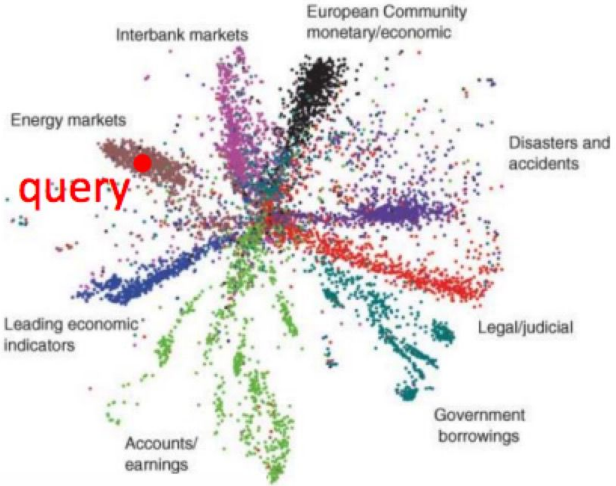


Applications of Autoencoders

Better Representation



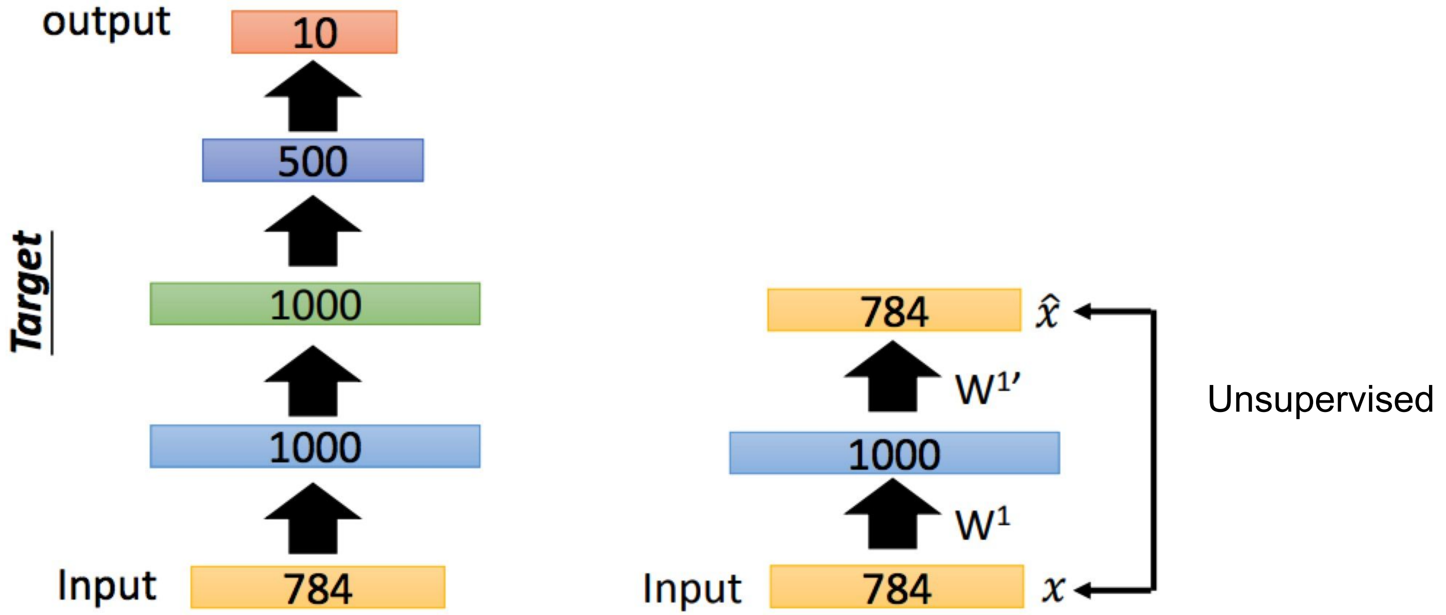
Bag-of-Word



LSA: project documents to 2 latent topics

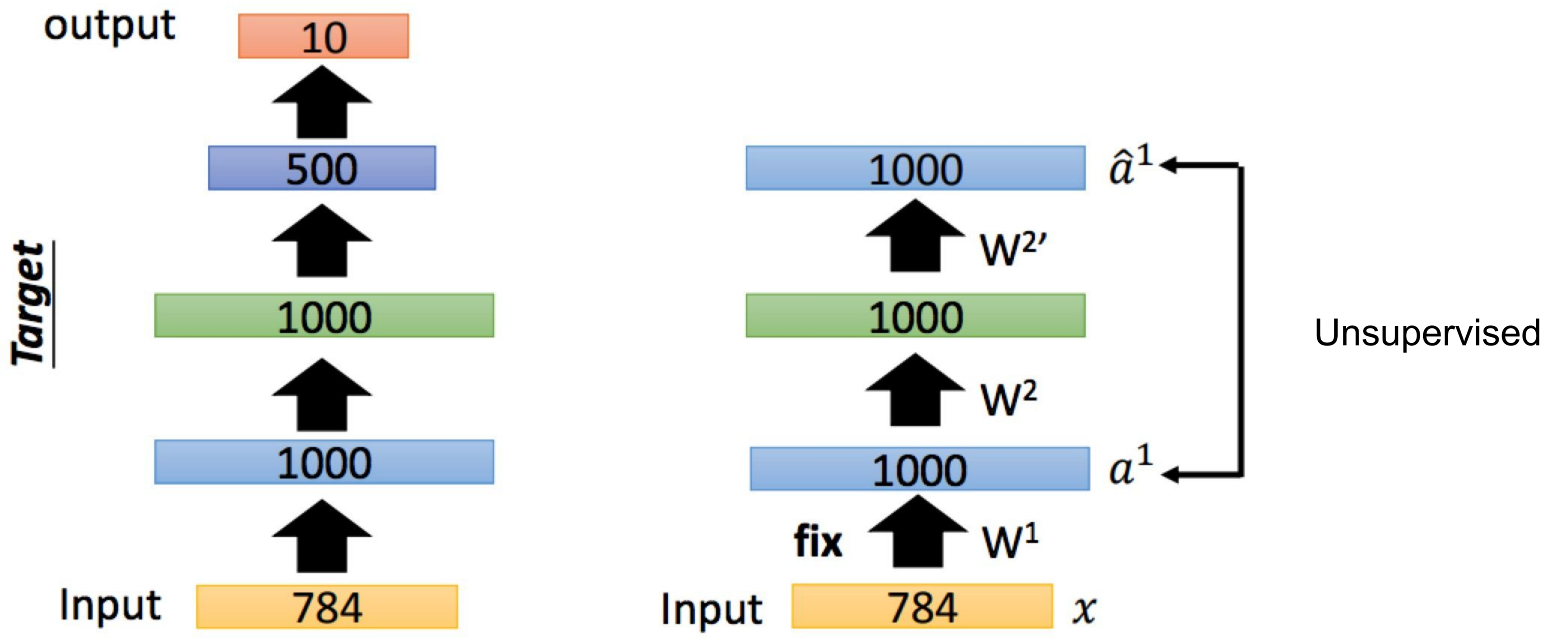
Pre-training Deep Neural Network

- Greedy Layer-wise Pre-training for W1



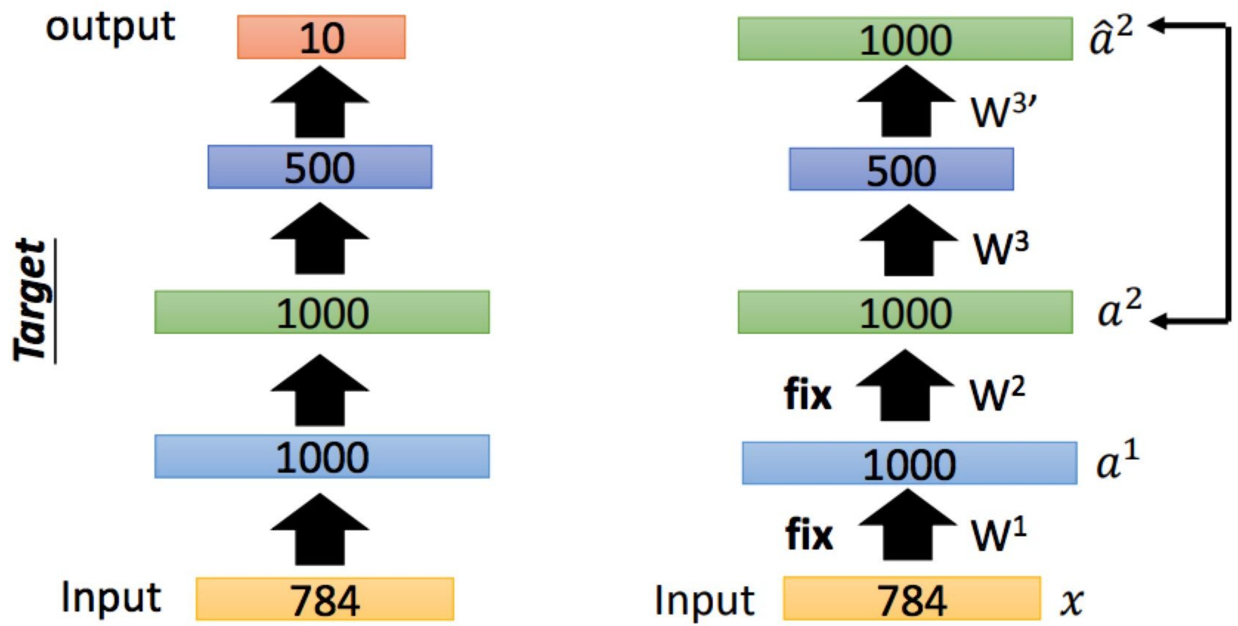
Pre-training Deep Neural Network

- Greedy Layer-wise Pre-training for W2



Pre-training Deep Neural Network

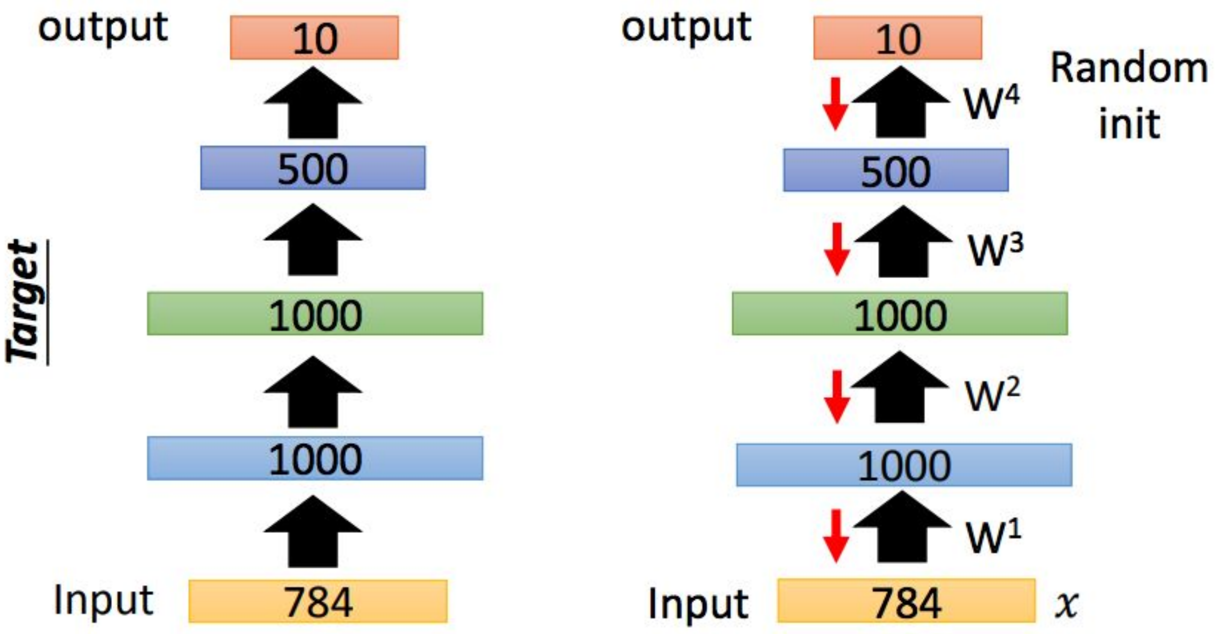
- Greedy Layer-wise Pre-training for W3



Unsupervised

Pre-training Deep Neural Network

- Fine-tune by backpropagation

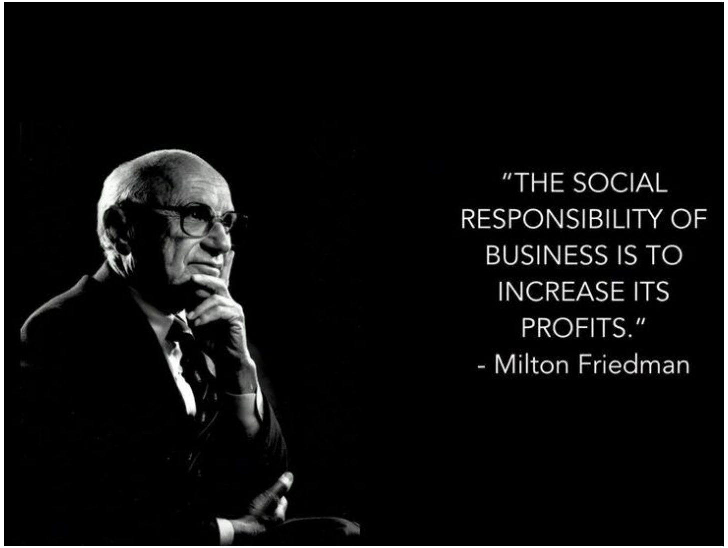


supervised

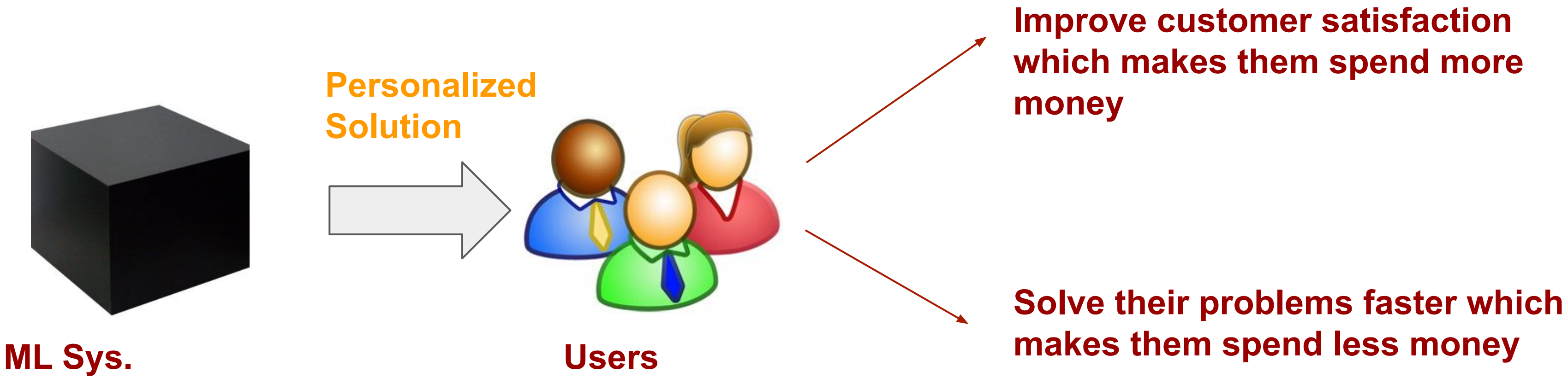
Project Scoping

Goals

- An ML project should be aimed at increasing profits directly or indirectly.
  - Increasing sales
  - Cutting costs
  - Increasing satisfaction
  - Increasing time spent on a website
- Do we have non-profits projects? Yes
  - Climate change
  - Public health
  - Education



Case Study


Personalized Solution
ML Sys.
Users
Improve customer satisfaction which makes them spend more money
Solve their problems faster which makes them spend less money

Case Study：Movie Recommendation

When building a recommendation system for movie

- Maximize Engagement
- Maximize Revenue from sponsored content
    - Click more, ads fee more
- Minimize the spread of restricted content

How to set goals?

- Goals: General purpose of a project
  - Maximize users' engagement while minimizing the spread of violent content and maximize revenue from sponsored content
- Objectives: Specific steps on how to achieve the above goals
  - Filter out unclasificated movies
  - Rank movies by quality
  - Rank movies by their ads fee
  - Rank movies by engagement: how likely users will watch it

How to combine these two targets via ML systems?

Multi-objective System

Due to sparsity

- Rank movies by quality
  - Predict films' rating
  - Minimize Rating_loss: loss between predicted rating and true rating
- Rank movies by engagement: how likely users will watch it
  - Predict watch times
  - Minimize Engagement_loss: loss between predicted watch times and true times

Solution A: one model with combined loss

- Train one model

The loss is defined as:

$$\text{Loss} = \alpha * \text{Rating\_loss} + \beta * \text{Engagement\_loss}$$

Solution B: combine different models

- Train two models
  - Model A: rating_loss
  - Model B: engagement_loss

Rank posts by \alpha*Pred_ModelA + \beta*Pred_ModelB

Decouple different objectives

- Easier for training
- Easier to tweak our systems
    - No need to retrain the whole system if weights for different objectives are changed.
- Easier for maintenance
    - Different objectives might need different maintenance schedules

Recommendation System

Arjun Narayan
@narayanarjun

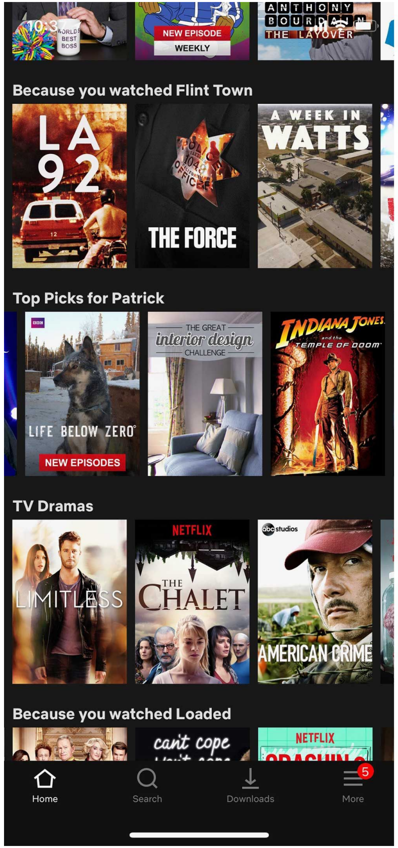
Follow

The two best performing public stocks of the decade - Netflix (+3700%) and Domino's Pizza (+3000%) - perfectly epitomize the 2010s. You either build the world's most advanced machine learning content recommender system, or make a better pizza sauce, there's no middle ground.

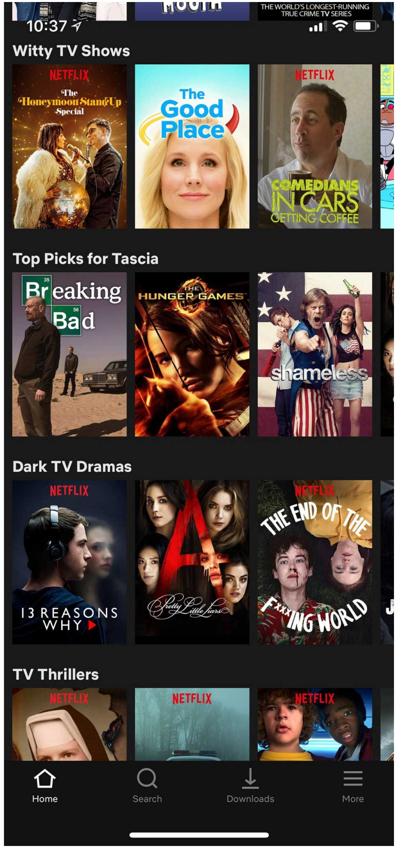
1:20 PM - 27 Dec 2019

3,926 Retweets **20,086** Likes

183 3.9K 20K

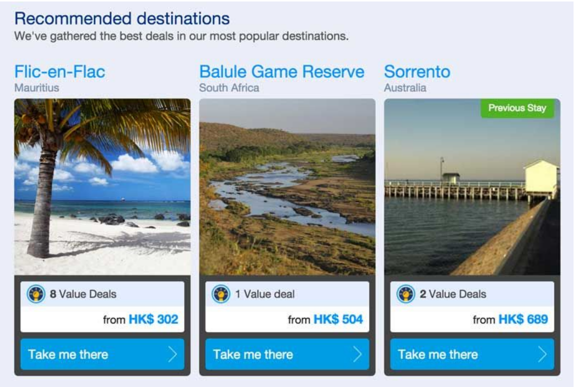


NEW EPISODE
WEEKLY
Because you watched Flint Town
LA 92
THE FORCE
A WEEK IN WATTS
Top Picks for Patrick
LIFE BELOW ZERO
NEW EPISODES
THE GREAT interior design CHALLENGE
INDIANA JONES and the TEMPLE OF DOOM
TV Dramas
LIMITLESS
NETFLIX
THE CHALET
AMERICAN CRIME
Because you watched Loaded
can't cope
NETFLIX
Home
Search
Downloads
More



10:37
Witty TV Shows
NETFLIX
The Honeymoon Stand Up Special
The Good Place
NETFLIX
COMEDIANS IN CARS GETTING COFFEE
Top Picks for Tascia
Breaking Bad
THE HUNGER GAMES
shameless
Dark TV Dramas
NETFLIX
13 REASONS WHY
Pretty Little Liars
NETFLIX
THE END OF THE F***ING WORLD
TV Thrillers
NETFLIX
NETFLIX
NETFLIX
Home
Search
Downloads
More

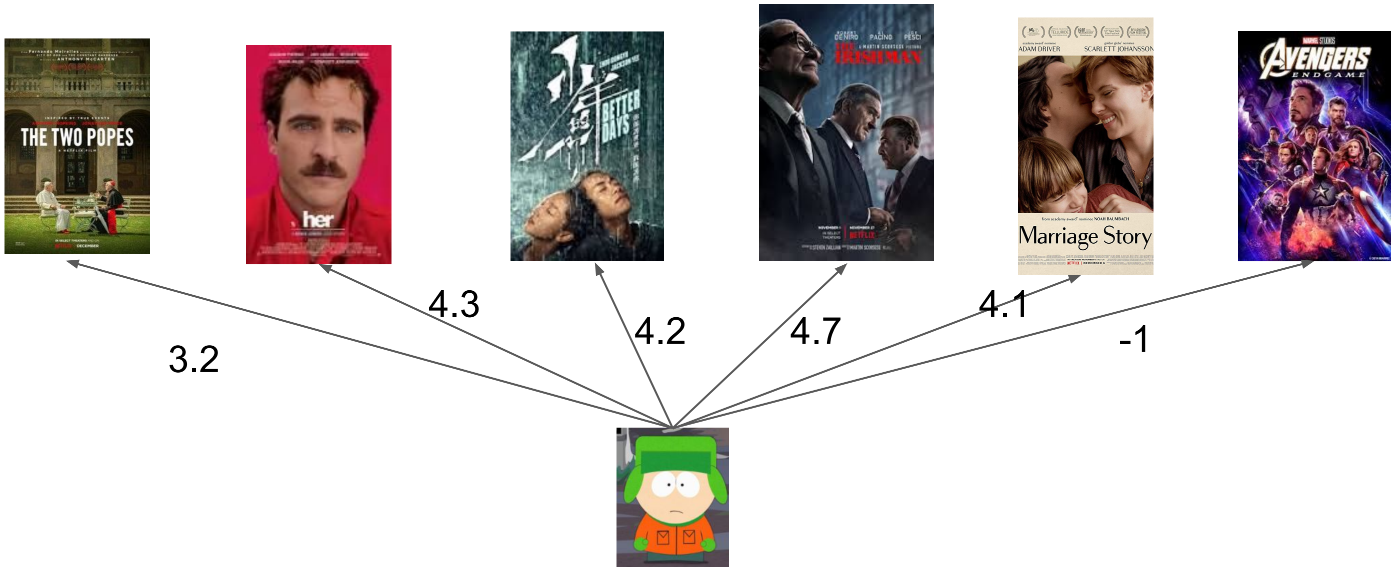
amazon.com
Recommended for You
Amazon.com has new recommendations for you based on items you purchased or told us you own.
BIG
The Little Big Things: 163 Ways to Pursue EXCELLENCE
Fascinate: Your 7 Triggers to Persuasion and Captivation
Sherlock Holmes [Blu-ray]
Alice in Wonderland [Blu-ray]



Recommended destinations
We've gathered the best deals in our most popular destinations.
Flic-en-Flac
Mauritius
8 Value Deals
from HK$ 302
Take me there
Balule Game Reserve
South Africa
1 Value deal
from HK$ 504
Take me there
Sorrento
Australia
Previous Stay
2 Value Deals
from HK$ 689
Take me there

Core Problem in Rec. Sys.

- Filter Information for users
- Personalization is the key:
  - Given a certain user, compute the score that quantifies how strongly a user u likes/prefer items i.

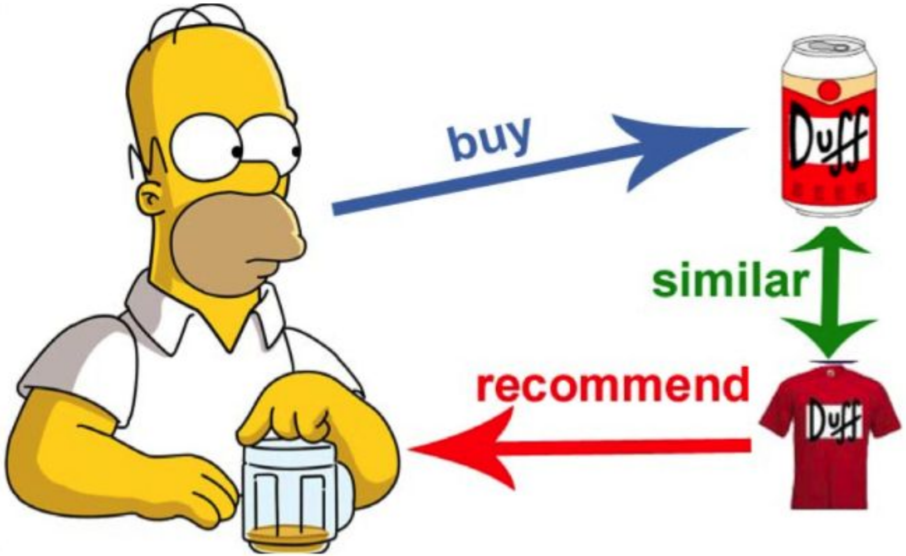


Content-based Method

- Define the similarity from items' content
  - Name: cosine similarity
  - Category
  - Rating
  - Description
  - Etc

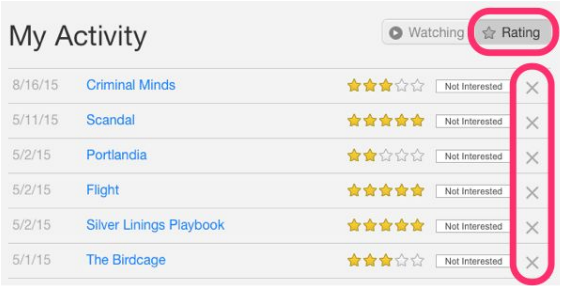
- Combine them into a final score

- Ranked items based on their similar scores compared to users' purchased item

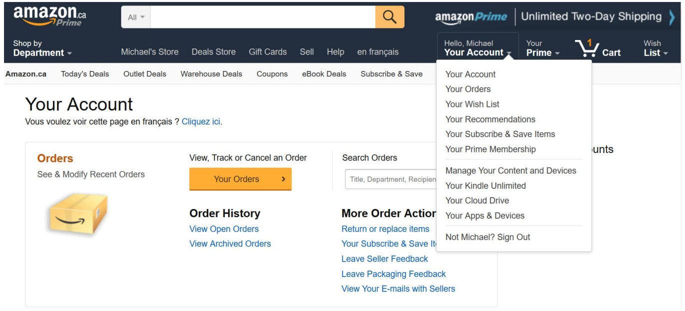


User Behaviour

- Content-based methods: only look at the items' information
- The Insights behind the huge interaction behind users and items



Ratings in Netflix



Order History

User-Item Matrix

- Content-based methods: only look at the items' information
- The Insights behind the **interaction** behind users and items

Item Vector

User Vector

	Item 1	Item 2	Item 3	…….	Item k-1	Imte k
User 1	1	0	0		3	1
User 2	0	3	1		0	2
...						
User n-1	0	2	0		1	1
User n	0	0	0		0	0

User-based CF

- Find the similarity score betweens users
- Recommend products which these similar uses have liked or bought previously

The rating of item i given by user v

$$P_{u,i} = \frac{\sum_{v \in U} (r_{v,i} * s_{u,v})}{\sum_{v \in U} s_{i,v}}$$

The similarity between users u and v

The prediction of an item i for user u

User Space

$$s_{u,v} = cos(\vec{u}, \vec{v}) = \frac{\vec{u} * \vec{v}}{||\vec{u}|| ||\vec{v}||}$$

Cosine similarity used a lot in information retrieval

Item-based CF

- Find the similarity between each item pair
- Recommend similar items which were liked or purchased by the users in the past

The rating of item m given by user u

$$P_{u,i} = \frac{\sum_{m \in I} (r_{u,m} * s_{i,m})}{\sum_{m \in I} s_{i,m}}$$

The similarity between items i and m

The prediction of an item i for user u

Item Space

$$s_{i,m} = cos(\vec{i}, \vec{m}) = \frac{\vec{i} * \vec{m}}{||\vec{i}||\,||\vec{m}||}$$

Data Sparsity

movieId	1	2	3	4	5	6	7	9	10	11	...	106487	106489	106782	106920	109374
userId																
316	-0.829457	NaN	NaN	NaN	NaN	NaN	-1.329457	NaN	-0.829457	NaN	...	NaN	NaN	NaN	NaN	NaN
320	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN
359	1.314526	NaN	NaN	NaN	NaN	1.314526	NaN	NaN	0.314526	0.314526	...	NaN	NaN	NaN	NaN	NaN
370	0.705596	0.205596	NaN	NaN	NaN	1.205596	NaN	NaN	NaN	NaN	...	-1.294404	-0.794404	0.705596	0.205596	NaN
910	1.101920	0.101920	-0.39808	NaN	-0.39808	-0.398080	NaN	NaN	NaN	0.101920	...	NaN	NaN	-0.398080	NaN	NaN

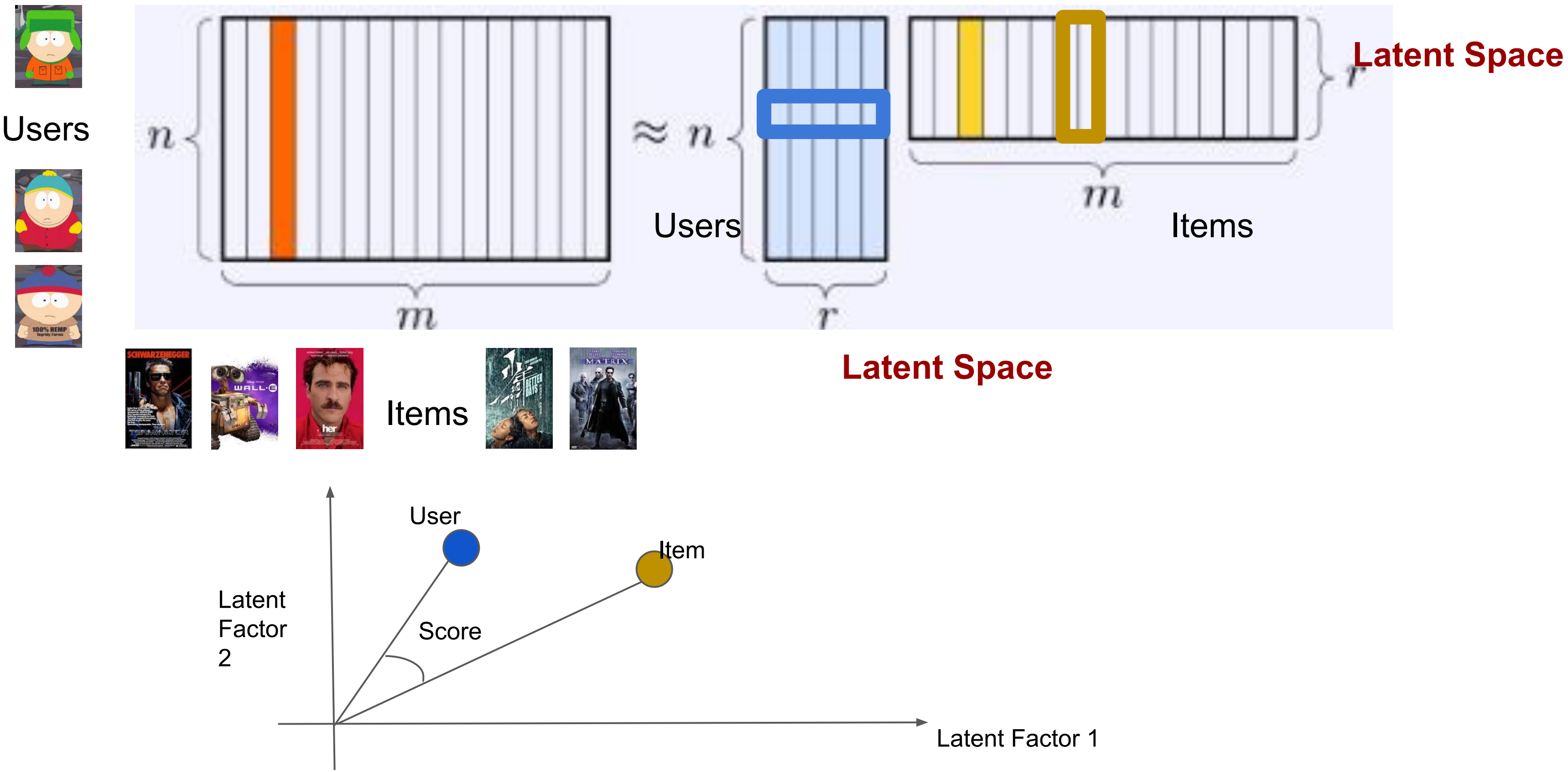
Similarities between users and items are zero

The core problem behind recommendation sys. is to **fill these zero entries**, i.e., *infer the user's preference over the item.*

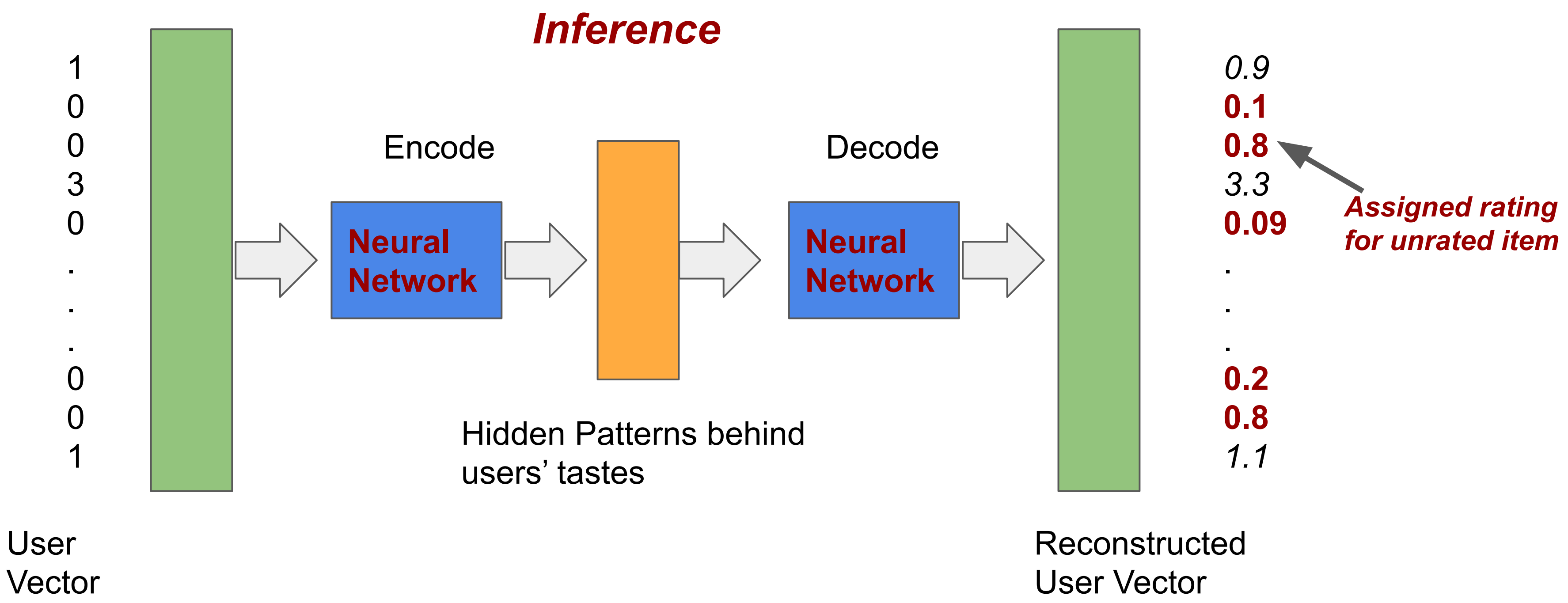
- Data Preprocessing:
  - Use the mean value of the row
  - Use the mean value of the column

- Matrix Factorization
  - Singular Value Decomposition
  - Non-Negative Matrix Factorization
  - Auto-encoder

NMF for Rec.


Users
n
m
≈ n
Users
r
Latent Space
m
Items
Items
Latent Space
User
Item
Score
Latent Factor 2
Latent Factor 1

Autoencoder for Rec.



Pros & Cons of CF

- Pros:
    - Capture latent user and item factors
    - Can handle sparsity
    - Scalable computation (ALS)

- Cons:
    - Biases (Temporal and Popularity)
    - Cold Start Problem
    - No Context-awareness

How to evaluate Rec. Sys.

- Offline Evaluation:
  - Train/Test Splitting
  - RMSE
  - Recall
  - Etc

- Online Evaluation:
  - A/B Testing
  - Click-Through Rate (CTR)
  - Conversion Rate (CR)
  - Etc

Hypothesis Testing

- New algorithm A and old algorithm B
- Test users are drawn independently from some population.
- Performance measures of the algorithms for each test user is able to give us the independent comparisons.
- Given such paired per-user performance measures for algorithms A and B, we can count the number of users for whom A outperforms B (nA) and the number of users for whom B outperforms A (nB).
- If nA > nB, can we algorithm A is indeed better than algorithm B?

Sign Test

- H0: A is no better than B
- H1: A is truly better than B
- Firstly, we need to compute the significance level or p-value.
- P-value can be regarded as the probability that the obtained result were due to luck or A is not truly better than B.
- Here, it will be estimated as the probability of at least nA out of nA+nB 0.5 probability Binomial trials succeeding.

$$0.5^{(nA+nB)} \sum_{k=nA}^{nA+nB} \frac{(nA+nB)!}{k!(nA+nB-k)!}$$

https://en.wikipedia.org/wiki/Sign_test

https://www.jmlr.org/papers/volume7/demsar06a/demsar06a.pdf