Machine Learning & Predictive Analytics

Class 5

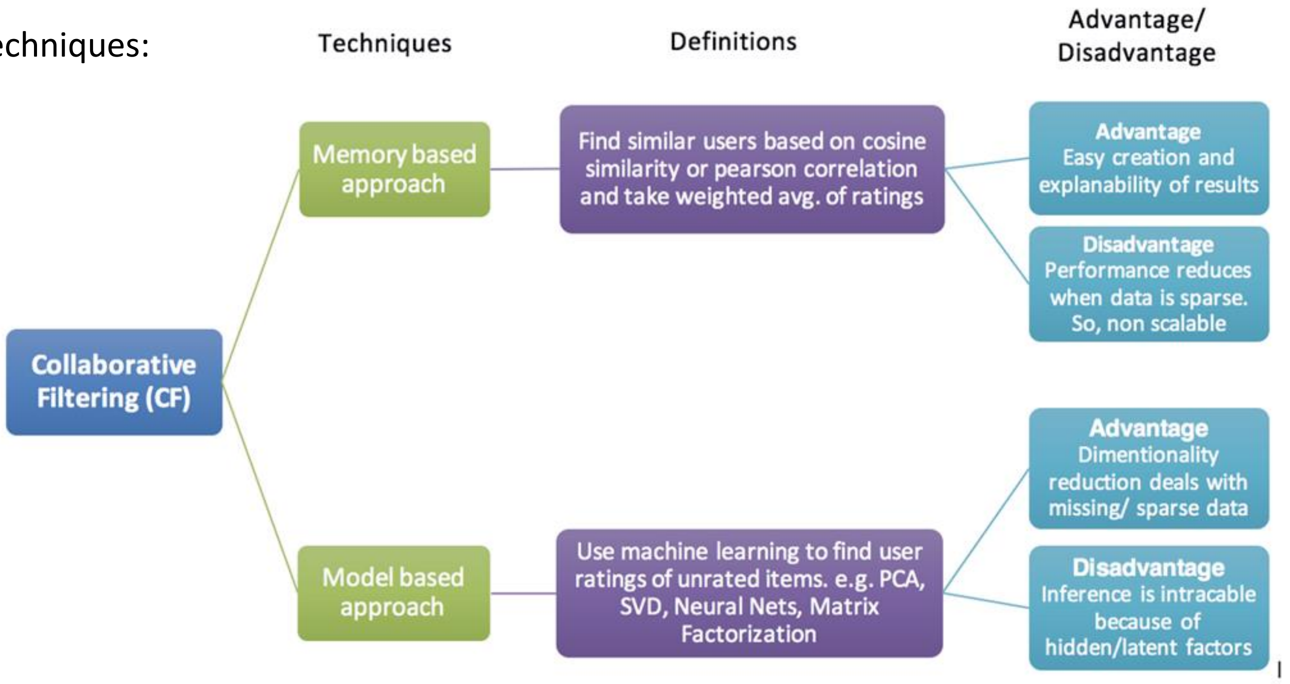
Arnab Bose, Ph.D.

MSc Analytics

University of Chicago

Collaborative Filtering (CF) Recommendation

- Primarily used to predict preferences
- Rooted in information filtering that uses group opinions for recommendations aka Recommender Systems
- Term coined by Dave Goldberg and his colleagues at Xerox PARC
- Techniques:


Techniques
Definitions
Advantage/ Disadvantage
Collaborative Filtering (CF)
Memory based approach
Find similar users based on cosine similarity or pearson correlation and take weighted avg. of ratings
Advantage Easy creation and explanability of results
Disadvantage Performance reduces when data is sparse. So, non scalable
Model based approach
Use machine learning to find user ratings of unrated items. e.g. PCA, SVD, Neural Nets, Matrix Factorization
Advantage Dimentionality reduction deals with missing/ sparse data
Disadvantage Inference is intracable because of hidden/latent factors

https://towardsdatascience.com/various-implementations-of-collaborative-filtering-100385c6dfe0

Similarity Measures (Repeat)

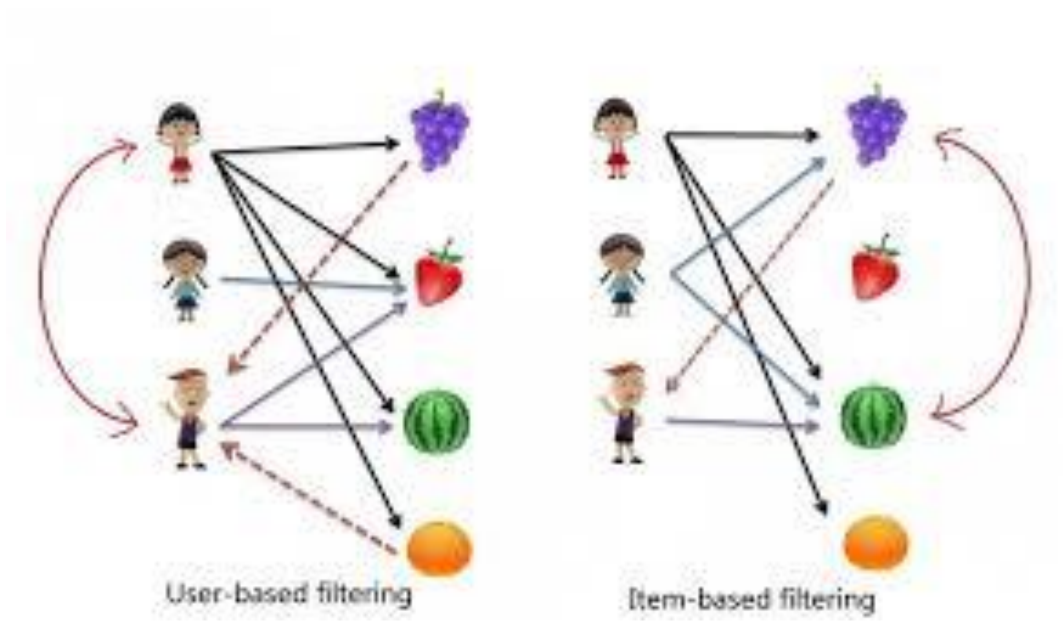
1. Represent dataset in a feature space- if two data are close to each other in terms of similarity, they should map to close points in the feature space
   - N-dim coordinate system or vector representation
2. Measure similarity between data points in the feature space using a metric that should satisfy
   - Non-negativity: $metric(a,b) \geq 0$
   - Identity: $metric(a,b) = 0 \Rightarrow a = b$
   - Symmetry: $metric(a,b) = metric(b,a)$
   - Triangular Inequality: $metric(a,b) \leq metric(a,c) + metric(b,c)$
3. Metric examples
   - Distance metric – descriptive features of the data can be represented in a coordinate system (eg. Euclidean, Manhattan, Minkowski)
   - Similarity index – descriptive features of the data are related to each other (eg. cosine similarity, Jaccard)
4. Note that points 1 and 2 are combined in Nearest Neighbor model

Collaborative Filtering – Memory based Approach

- 2 types:
  - User based – database of preferences for items (consumption patterns) by users and a new user is matched against the database to find neighbors who have similar preferences and recommends items preferred by the neighbors to the new user
  - Item based – invented by Amazon; based on similar items rather than similar neighbors, rates items to similar items and combines the similar items into a recommendation. Done by monitoring how many users that bought item X also bought item Y. Note since the relationship between items is relatively static, the item-based recommendation may be able to provide the same quality with less online computation

Collaborative Filtering – Memory based Approach

- Note that the definition of “similar” varies from system to system and one aspect of making good recommendations is having an appropriate measure of similarity (cosine, nearest neighbor, word/letter distances)



http://www.salemmarafi.com/code/collaborative-filtering-with-python/

Techniques – Memory based CF

- Use the entire data
- Based on calculating similarity between users + items

For example Pearson correlation similarity of 2 users x, y is defined as

$$similarity(x,y) = \frac{\sum_{i \in I_{xy}} \left(r_{x,i} - \overline{r_x}\right)\left(r_{y,i} - \overline{r_y}\right)}{\sqrt{\sum_{i \in I_{xy}} (r_{x,i} - \overline{r_x})^2} \sqrt{\sum_{i \in I_{xy}} (r_{y,i} - \overline{r_y})^2}}$$

Where $r_{x,i}$ is the rating by user x for item i that belongs to set of items I_{xy} rated by both users

$\overline{r_x}$ is the average item rating for user x

- Advantage – quality of predictions are using a simple approach
- Disadvantage – overfit the data and computationally intensive so could be slow; also scalability hindered when there is sparse data

User-Based or Item-Based Filtering – Data

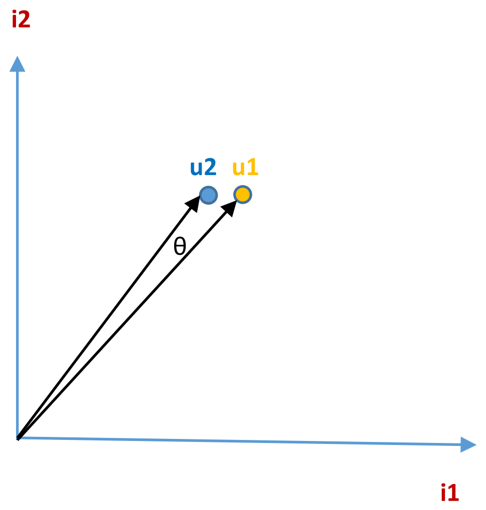
\Items Users	i1	i2	I3
U1	4	5	2
U2	3	5	?
U3	1	2	5
U4	2	?	5
U5	5	5	?

User-Based or Item-Based Filtering – Similarity

\Items Users	i1	i2	I3
U1	4	5	2
U2	3	5	?
U3	1	2	5
U4	2	?	5
U5	5	5	?

user-user

item-item



User-Based or Item-Based Filtering?

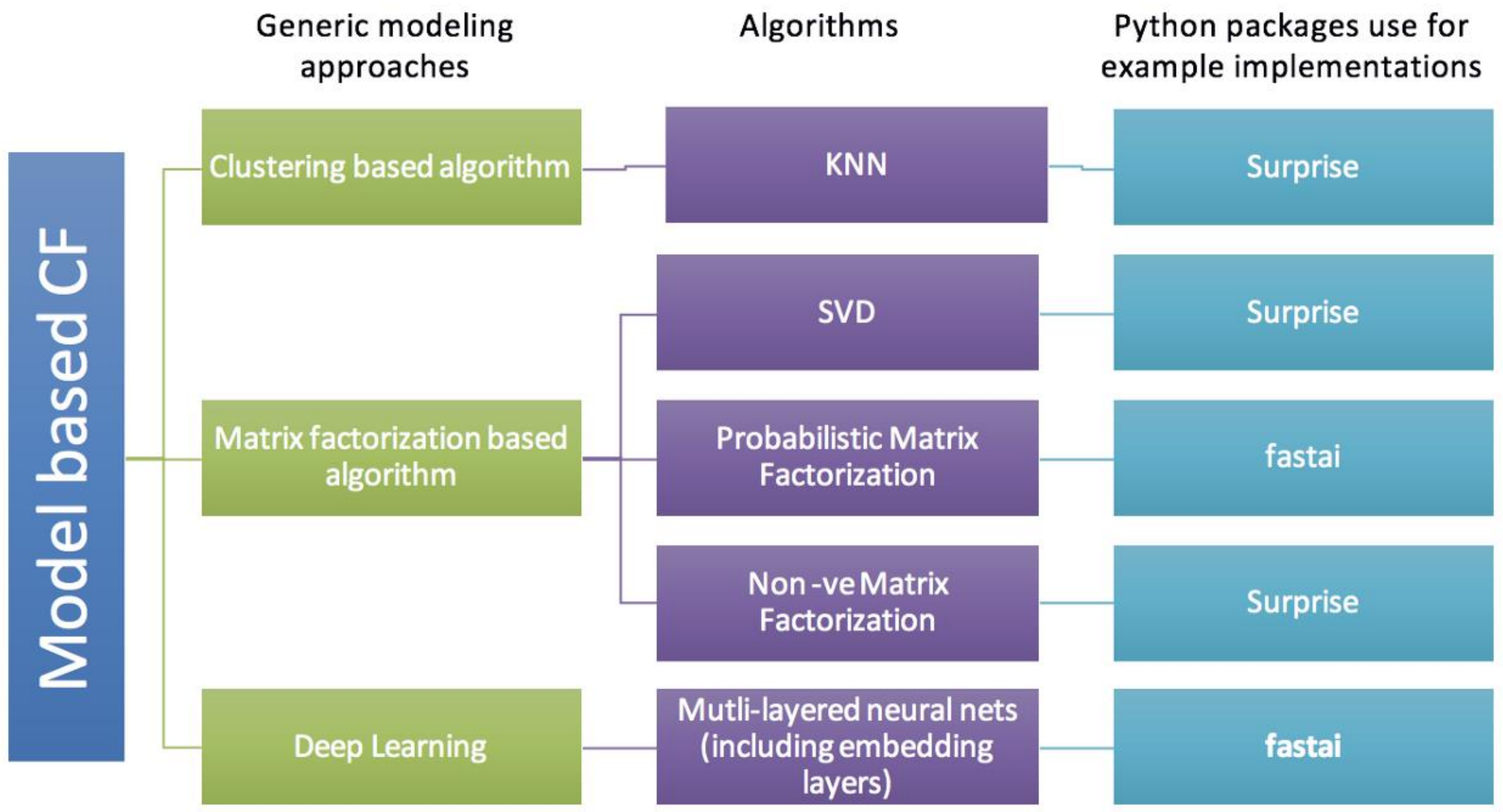
1. Item-based is significantly faster for large datasets where comparing a user with every other user and then comparing every item each user has rated can be very slow.
2. Item-based has the overhead of maintaining item similarity table (that is derived from user similarity table).
3. Item-based does not need to be recalculated often as comparisons between items do not change as regularly as comparisons between users.
4. User-based is simpler to implement.

Techniques – Model based CF

Model based – Bayesian networks, clustering models, SVD, LDA

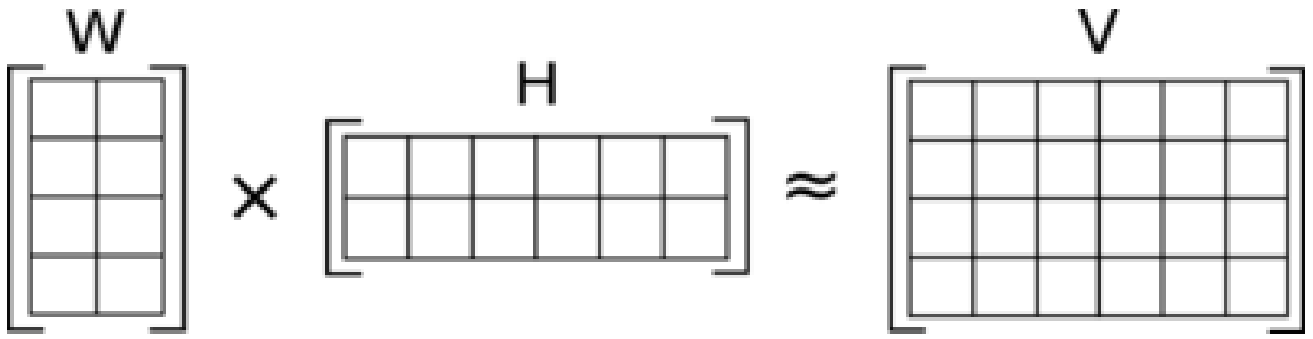
- Advantage – fast, scalable models that avoid overfit
- Disadvantage – quality may not be as good as memory-based

Model based CF – Approach


Generic modeling approaches
Algorithms
Python packages use for example implementations
Model based CF
Clustering based algorithm
KNN
Surprise
SVD
Surprise
Matrix factorization based algorithm
Probabilistic Matrix Factorization
fastai
Non -ve Matrix Factorization
Surprise
Deep Learning
Mutli-layered neural nets (including embedding layers)
fastai

https://towardsdatascience.com/various-implementations-of-collaborative-filtering-100385c6dfe0

Model based CF – Non-negative Matrix Factorization (NMF)



https://en.wikipedia.org/wiki/Non-negative_matrix_factorization

http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.NMF.html

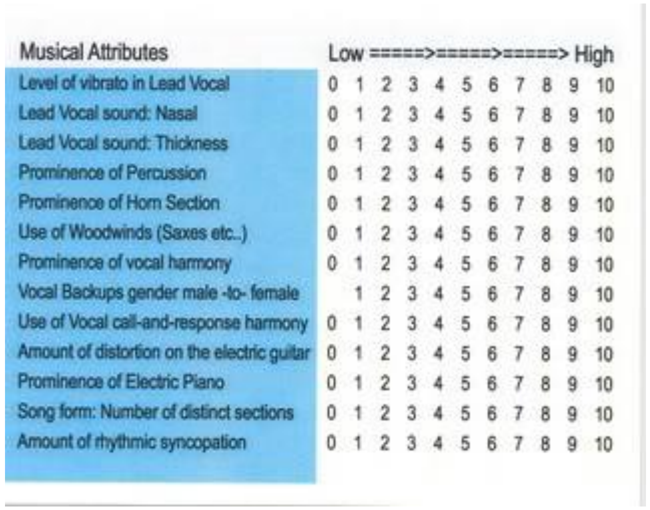
CF Issues

1. Data sparsity – not all users will review similar items so difficult to establish correlation
2. Cold-start – need initial data to make recommendations – a popular tool to combat this is LightFM*
3. Scalability – data grows as more users/items enter and current users review more items
4. Gray sheep – users who are different from groups / clusters
5. Shilling – manipulative reviews to promote or impede items
6. Lack of Diversity – popular items are recommend often with bias against the long-tail items

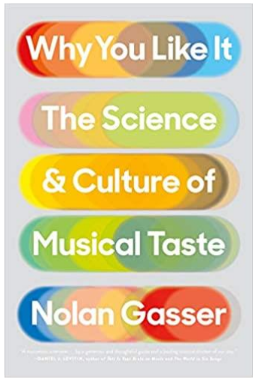
* https://www.eigentheories.com/blog/lightfm-vs-hybridsvd/#LightFM

Content-Based (CB) Recommendation

1. Content based recommendation – analytical description of the characteristics of content/item.
2. Match a user to the content characteristics as opposed to other users.
3. To figure out how songs are similar Pandora implemented The Music Genome Project.
4. They believe that songs are comprised of a series of characteristics. Similar to how the human genome describes a person, these characteristics describe a song.
5. The Music Genome Project applies values for each song in each of approximately 400 musical attributes.



Musical Attributes	Low =====>=====>=====> High										
Level of vibrato in Lead Vocal	0	1	2	3	4	5	6	7	8	9	10
Lead Vocal sound: Nasal	0	1	2	3	4	5	6	7	8	9	10
Lead Vocal sound: Thickness	0	1	2	3	4	5	6	7	8	9	10
Prominence of Percussion	0	1	2	3	4	5	6	7	8	9	10
Prominence of Horn Section	0	1	2	3	4	5	6	7	8	9	10
Use of Woodwinds (Saxes etc..)	0	1	2	3	4	5	6	7	8	9	10
Prominence of vocal harmony	0	1	2	3	4	5	6	7	8	9	10
Vocal Backups gender male -to- female		1	2	3	4	5	6	7	8	9	10
Use of Vocal call-and-response harmony	0	1	2	3	4	5	6	7	8	9	10
Amount of distortion on the electric guitar	0	1	2	3	4	5	6	7	8	9	10
Prominence of Electric Piano	0	1	2	3	4	5	6	7	8	9	10
Song form: Number of distinct sections	0	1	2	3	4	5	6	7	8	9	10
Amount of rhythmic syncopation	0	1	2	3	4	5	6	7	8	9	10



Why You Like It
The Science
& Culture of
Musical Taste
Nolan Gasser

Recommendation Metrics – Traditional

1. P @ k – Precision @ k – the number of correct recommendations out of k.
2. recall @ k – the number of correct recommendations that appear in k.

k	10	20	50	100	200	1000
P@k	0.400	0.450	0.380	0.230	0.115	0.023
recall@k	0.020	0.045	0.094	0.114	0.114	0.114

3. Precision decreases as k increases but recall increases.
4. By varying k, trade precision for recall.

S.Buttcher, C.L.A. Clarke, G.V.Cormack, Information Retrieval: Implementing and Evaluating Search Engines, The MIT Press, 2016.

Recommendation Metrics – Traditional

Average precision - indication of effectiveness across the full range of recall values and defined as

$$AP = \frac{1}{|Relevant|} \sum_{i=1}^{k} relevant(i) * P@i$$

Where

$relevant(i) = 1$ if recommendation item at rank i is correct and 0 if not.

$Relevant$ is the set of relevant/correct recommendations from universe of items.

Note that AP is an approximation of the area under a precision-recall curve.

Recommendation Metrics – Non-Traditional

1. Normalized Discounted Cumulative Gain (nDCG) measures graded relevance of recommendations by comparing the relevance values of a recommendation set with the "ideal" result.

2. A gain value needs to be assigned to each relevance level. For example, suppose song recommendations are ranked on a 4-point scale as:

    1. Highly relevant: 10
    2. Relevant: 5
    3. Marginally relevant: 1
    4. Nonrelevant: 0

3. Suppose the first 6 songs are 1: relevant, 2: highly relevant, 3:nonrelevant, 4: relevant, 5: marginally relevant, and 6: highly relevant with gain vector G = {5, 10, 0, 5, 1, 10}.

4. Calculate cumulative gain vector CG where the value of element k is the sum from 1 to k elements in G, so CG = {5, 15, 15, 20, 21, 31}.

Recommendation Metrics – Non-Traditional

5. Discount lower ranked recommendations as

$$DCG[k] = \sum_{i=1}^{k} \frac{G[i]}{log_2(1+i)}$$

so DCG = {5.0, 11.3, 11.3, 13.5, 13.8, 17.4}.

5. Normalize the DCG against an "ideal" gain vector that maximizes cumulative gains at all rankings. Assume the item universe contains 2 highly relevant, 2 relevant and 2 marginally relevant items. Then the ideal G = {10, 10, 5, 5, 1, 1}.

6. The ideal DCG = {10.0, 16.3, 18.8, 21.0, 21.3, 21.7}.

7. The normalized DCG is

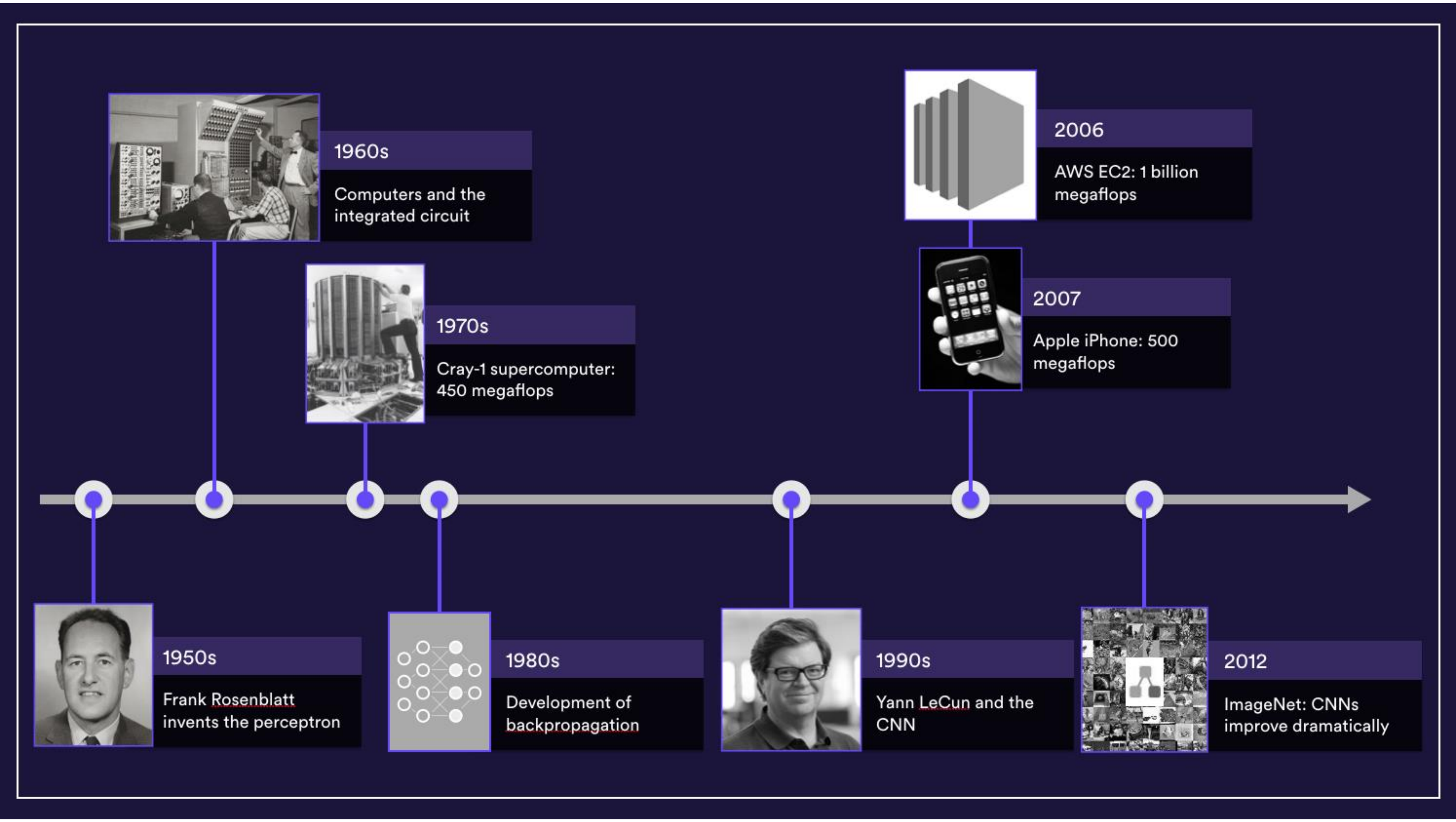
$$nDCG[k] = \frac{DCG[k]}{ideal\ DCG[k]}$$

so nDCG = {0.50, 0.69, 0.60, 0.64, 0.65, 0.80}.

Artificial Neural Network

1. A neural network can be seen as simple processing unit that is massively parallel, capable to store knowledge and apply this knowledge to make predictions.
2. A neural network mimics the brain in a way the network acquires knowledge from its environment through a learning process.
3. In the learning process, the synaptic weights of the network are modified in an orderly fashion to attain the desired objective.
4. In 1950, the neuro-psychologist Karl Lashley's thesis was published in which he described the brain as a distributed system.
5. As a result, these networks perform very well in areas like speech, audio and image recognition where the inputs / signals are inherently nonlinear.
6. Artificial neural networks have the ability to learn from supplied data, known as adaptive learning, while the ability of a neural network to create its own organization or representation of information is known as self-organisation.

History of Neural Network / Deep Learning



1950s
Frank Rosenblatt invents the perceptron

1960s
Computers and the integrated circuit

1970s
Cray-1 supercomputer: 450 megaflops

1980s
Development of backpropagation

1990s
Yann LeCun and the CNN

2006
AWS EC2: 1 billion megaflops

2007
Apple iPhone: 500 megaflops

2012
ImageNet: CNNs improve dramatically

https://www.vidora.com/ml-in-business/the-future-of-machine-learning/

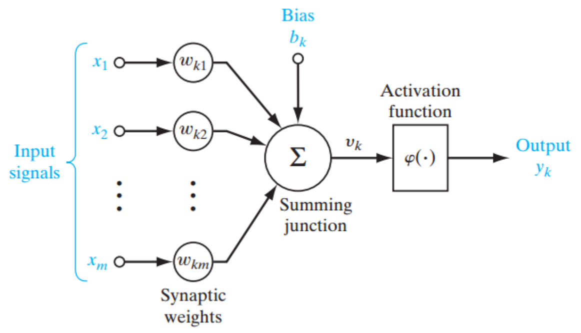
Single Layer Perceptron (SLP)

1. The simplest type of perceptron has a single layer of weights connecting the inputs and output.
2. In this way, it can be considered the simplest kind of feed-forward network.
3. In a feed forward network, the information always moves in one direction; it never goes backwards.
4. A Single Layer Perceptron (SLP) represents the m weights that are seen as a set of synapses or connecting links between one layer and another layer within the network. This parameter indicates how important each feature (x_j) is.
5. Below is the adder function of features of the input multiplied by their respective synaptic connection:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j$$

6. The bias b_k, acts as an affine transformation to the output of the adder function u_k giving v_k , the induced local field as: $v_k = u_k + b_k$

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review, 65*(6), 386–408.



SLP Learning Rule

$$w_{i,j}^{(next\ step)} = w_{i,j} + \eta(y_j - \hat{y}_j)x_i$$

$w_{i,j}$ is the connection weight between i^{th} input neuron and j^{th} the output neuron

x_i is the i^{th} input value of the current training instance

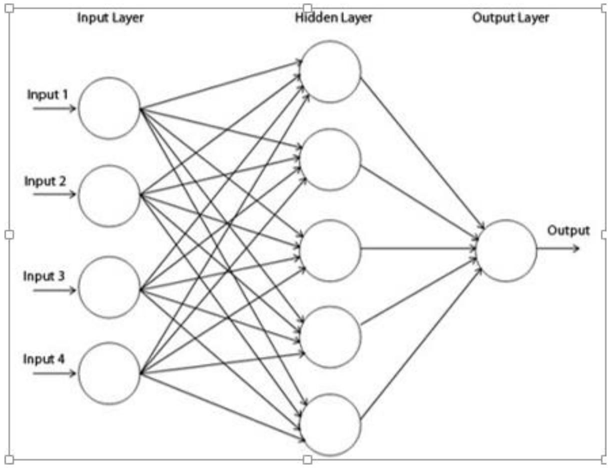
$\hat{y}_j$ is the output of the j^{th} output neuron for the current training instance

y_j is the target/actual output of the j^{th} output neuron for the current training instance

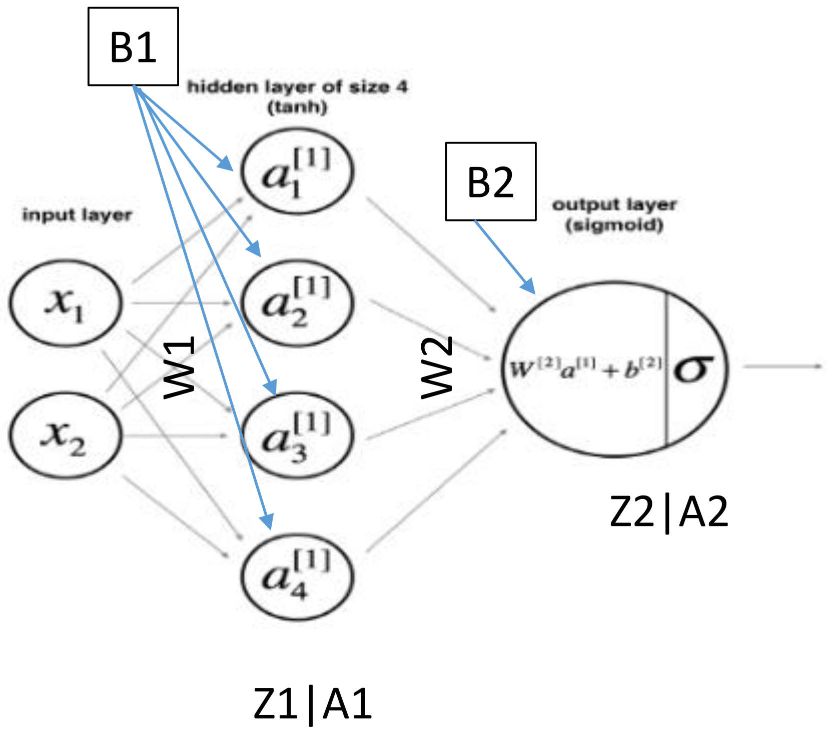
η is the learning rate

Multilayer Perceptron (MLP)

1. Moving onwards, multi-layer perceptron, also known as feed-forward neural networks, consists of a sequence of layers each fully connected to the next one.
2. A multilayer perceptron (MLP) has one or more hidden layers along with the input and output layers, each layer contains several neurons that interconnect with each other by weight links.
3. The number of neurons in the input layer will be the number of attributes in the dataset, neurons in the output layer will be the number of classes given in the dataset.
4. The figure shows a multilayer perceptron with a hidden layer and each node is connected to all nodes in previous layer.
5. To make the architecture deep, we introduce multiple hidden layers => Deep Learning.



NN using numpy



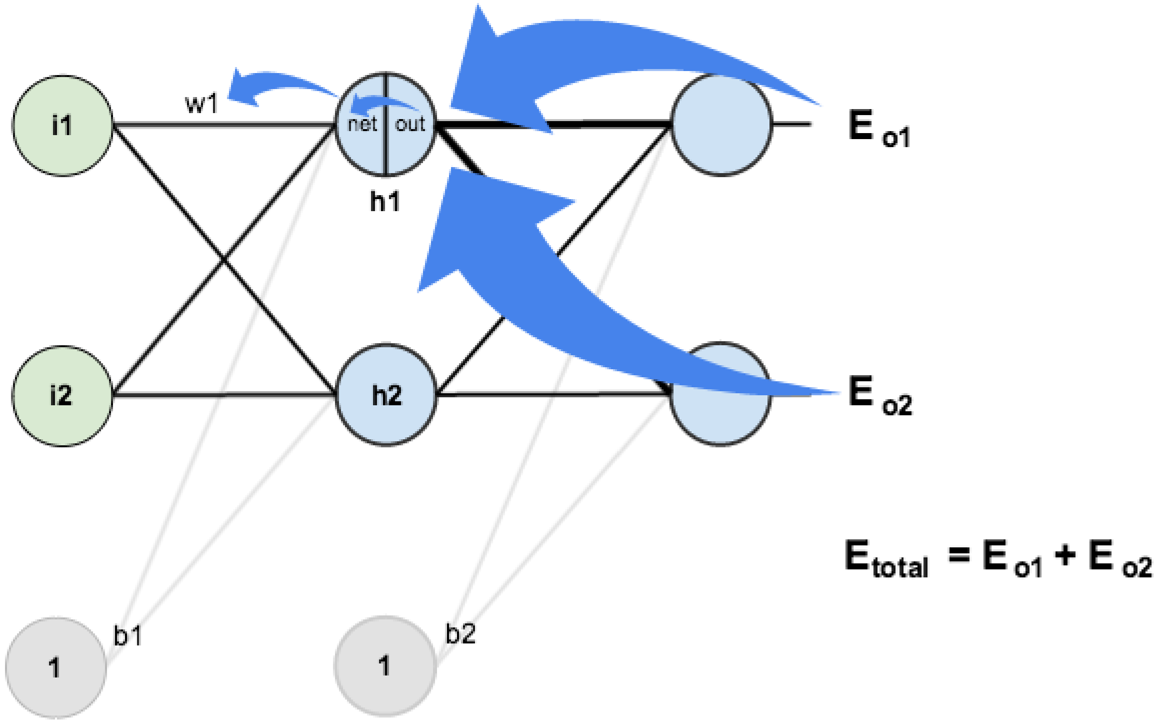
https://towardsdatascience.com/neural-net-from-scratch-using-numpy-71a31f6e3675

Back Propagation Algorithm

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

$$\downarrow$$

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$



https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/

Textbook Chapters and Assignment

- Materials covered available in book:
  - DL: Chapter 6
  - HML: Chapter 10
  - DLP: Chapters 2 – 3
- Book:
  - https://www.researchgate.net/publication/321162382_Artificial_Neural_Nets_For_Kids
- Code:
  - https://github.com/ageron/handson-ml2
  - https://github.com/NicolasHug/Surprise
  - https://github.com/lppier/Recommender_Systems