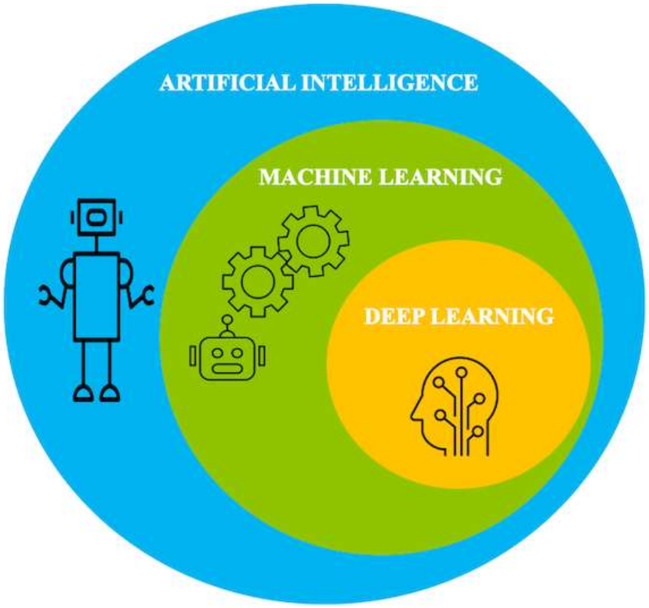
Graduate Certificate in Big Data Processing

Recommender Systems

Advanced Methods



Dr. Barry Shepherd
Institute of Systems Science
National University of Singapore
Email: barryshepherd@nus.edu.sg

Topics

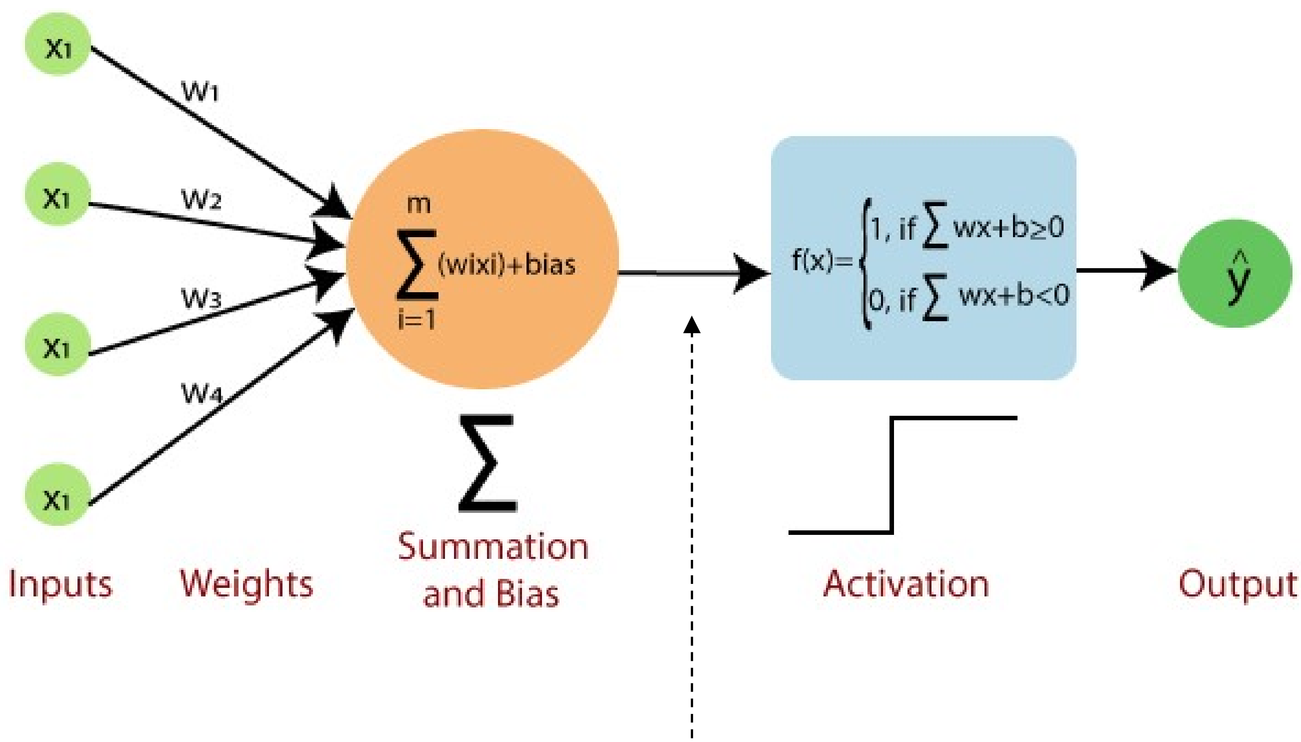
Deep Learning Methods are increasingly being used to help build recommendation systems. Examples are:

- Auto-Encoders
- Deep Collaborative Filtering
- Using NLP methods
- Recommender Systems utilising CNN's
- Graph-Based Methods

Survey of deep learning applied to recommendation systems:
https://arxiv.org/pdf/1707.07435.pdf

Neural Networks - 101

- A very simple Neural Network looks like this..



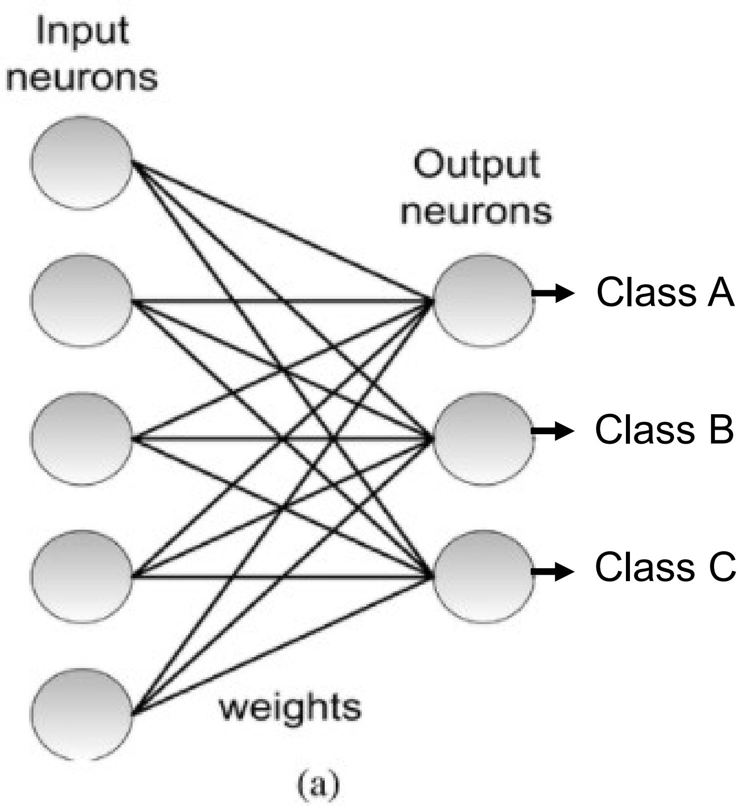
Using a simple step function (a hard threshold) as activation function causes the output to be binary (e.g. the network makes a True/False decision).

Other activation functions support numeric outputs (e.g. sigmoid)

The output here is a dot product of the input values times the weights, plus a bias - similar to a linear regression model, except weights/coefficients are learned differently

Multi-layer Feed-Forward Networks (FNN)

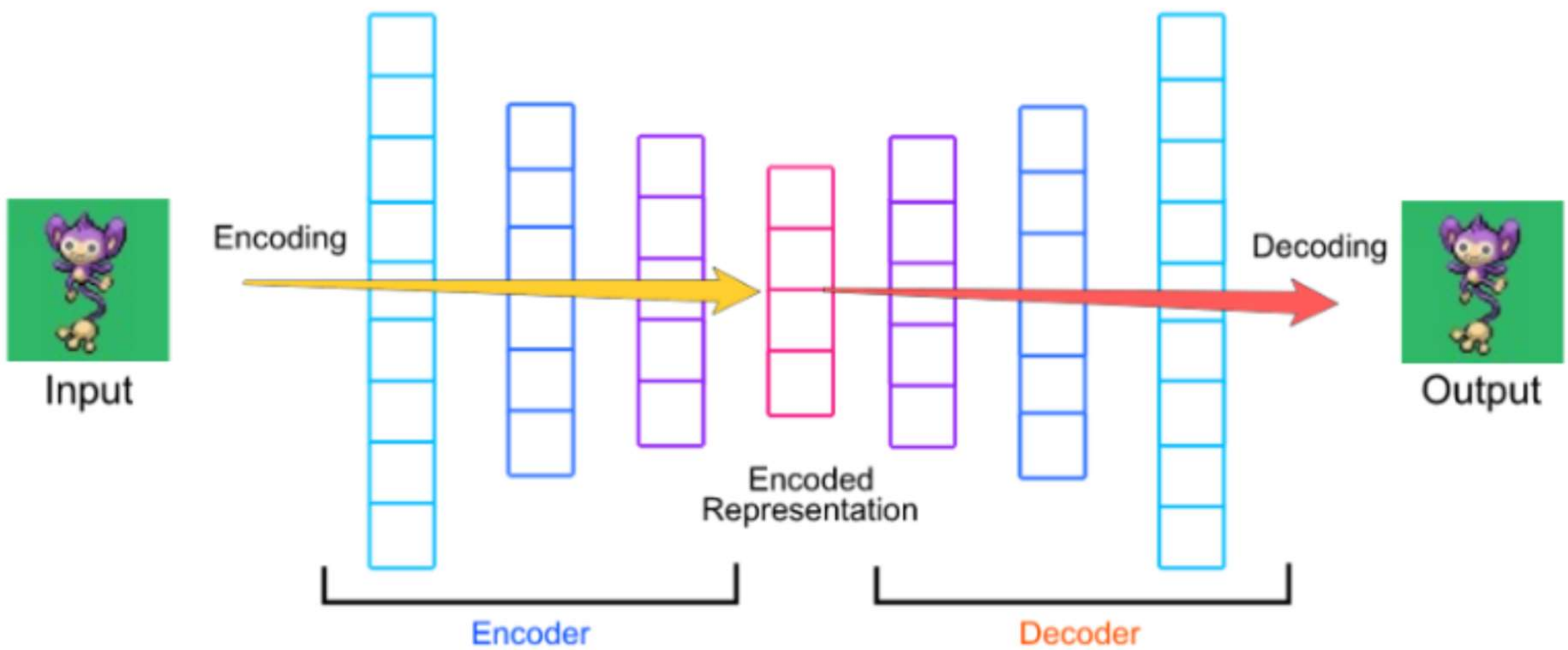
- Add multiple output nodes for classification problems
- Add hidden layers to learn non-linear functions and decision boundaries
- All information travels in one direction only, all nodes are fully connected



(a) Architecture of a single layer perceptron. The architecture consists of a layer on input neurons fully connected to a single layer of output neurons. (b) Extension to a multi-layer perceptron including more than one layer of trainable weights. In this example, the network includes 3 layers: input, hidden and output layer. Each connection between two neurons is given by a certain weight.

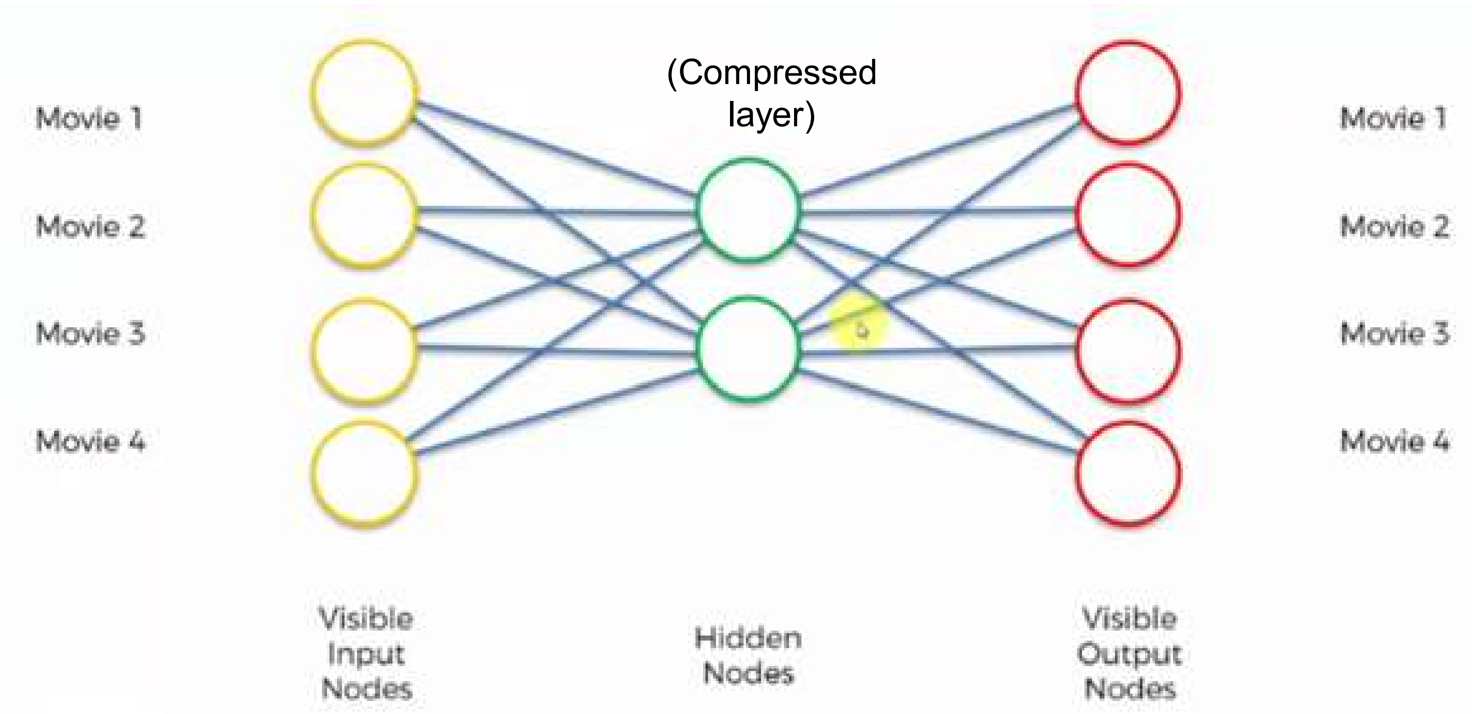
Auto-Encoders

- Has two parts, an encoder and a decoder, each can have many layers.
- Learns how to compress the input data while minimising the error at the output layer (unsupervised learning).
- The compressed layer is a shared layer. It is an encoding of the inputs (aka an ***embedding***), hence the encoder can be thought of as a feature learner/extractor
- The compressed layer is often used as an input to another NN
- One well-known application area is image compression



Auto-Encoders & Collaborative Filtering

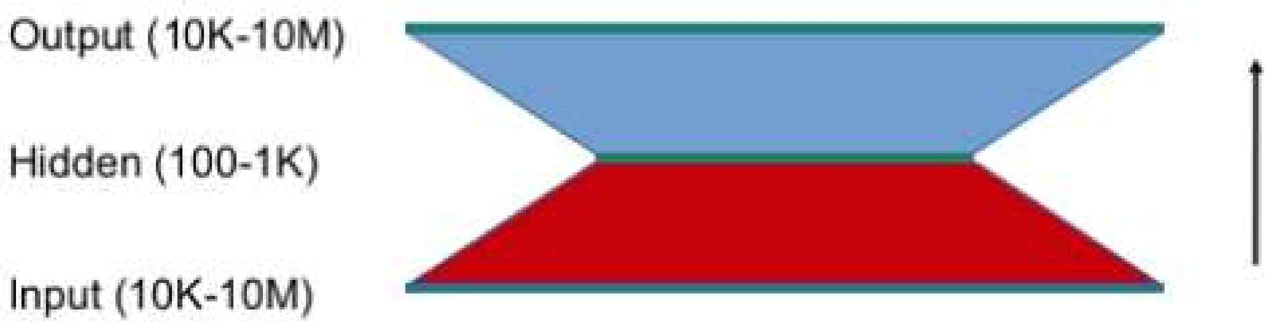
- Auto-encoders can be used to predict ratings for unseen items
- The hidden (compressed) layer can be though of as analogous to the latent features in MF approaches.



- After training, we can input a user with only a few known ratings and the output will contain the predicted ratings for that user for all of the movies

Recommendation Systems at Scale

- Amazon DSSTNE ~ Deep Scalable Sparse Tensor Neural Engine
- Based on Auto-Encoder neural network architecture



"We use DSSTNE to train neural networks and generate recommendations that power various personalized experiences on the retail website and Amazon devices."

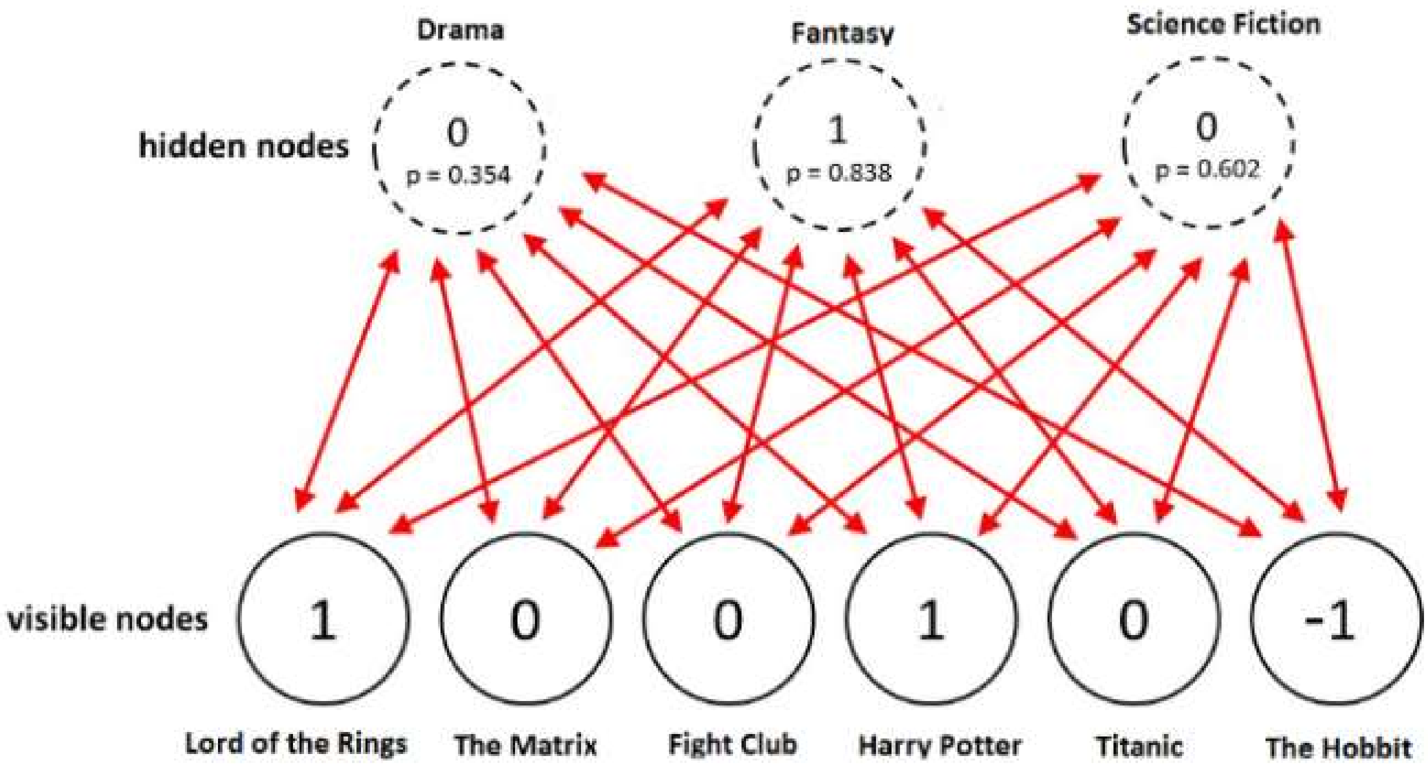
"Our models often have hundreds of thousands of nodes in the input and output layers. At this scale, we can easily reach trillions of weights for a fully-connected network, even if it is shallow. Therefore, our models often do not fit the memory of a single GPU"

In **model parallel training**, the model is distributed across *N* GPUs – the dataset is replicated to all GPU nodes. Contrast this with *data parallel* training where each GPU only trains on a subset of the data, then shares the weights with each other using synchronization techniques such as a parameter server.

https://aws.amazon.com/blogs/big-data/generating-recommendations-at-amazon-scale-with-apache-spark-and-amazon-dsstne/

RBM's and the Netflix Winning Solution

- Apart from Matrix Factorisation, the Netflix winner also used a ***Restricted Bolzman Machine***
- Similar to ***Auto-encoders*** but with only two layers and with the signal passing in both directions. Hence learning is a stochastic process (not deterministic) . Also it has a bias on both layers



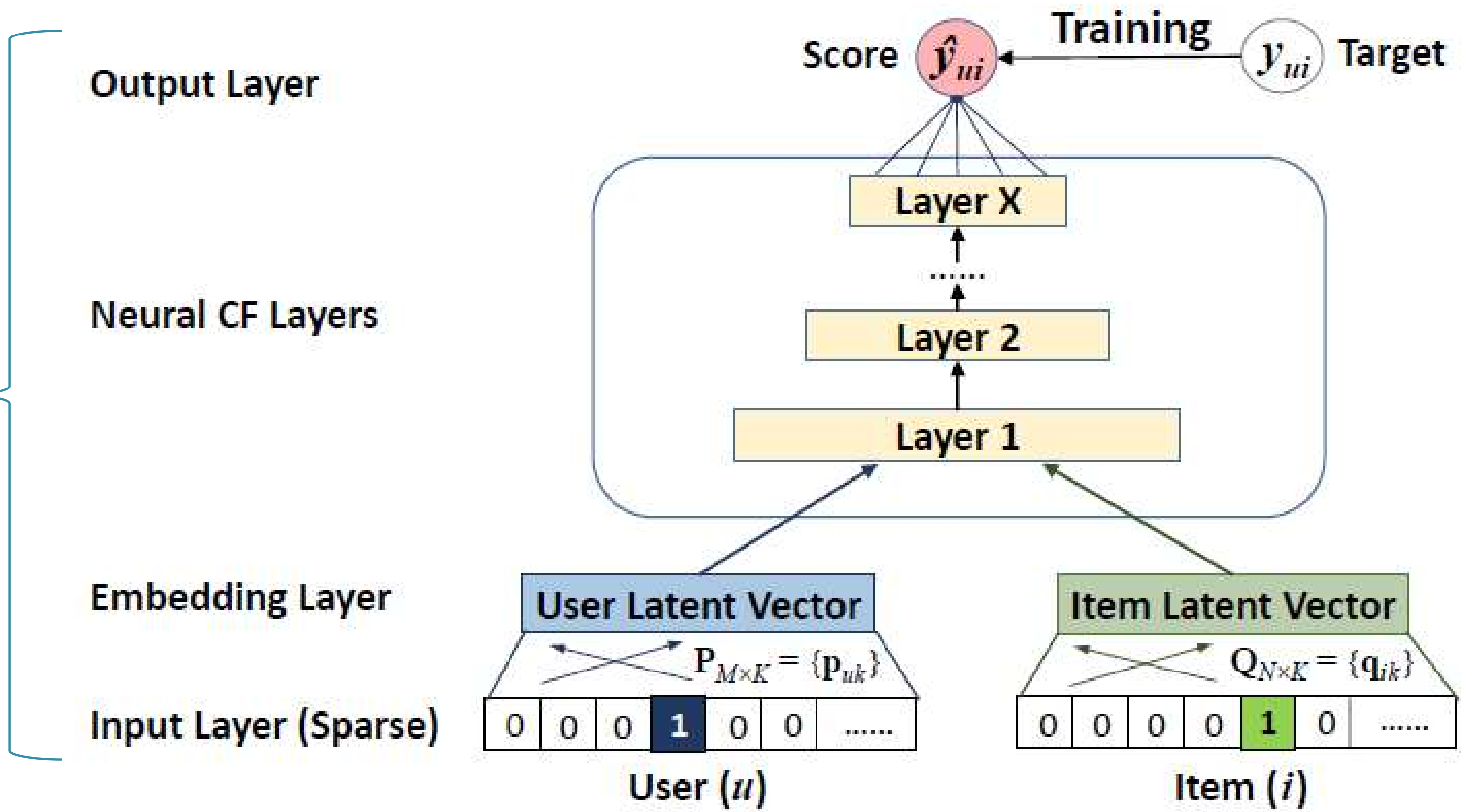
Latent features

Neural Collaborative Filtering (NCF)

- The two inputs (user & item) are fed into ***embedding*** layers. Each embedding layer learns to map its input into a lower dimensional space (analogous to the latent feature space in MF)
- The two ***embeddings*** feed into a feedforward neural network (FNN) with a single output that outputs a score (e.g. the predicted rating).

When using MF, ratings are predicted by taking the **dot product** of the user latent features and the item latent features. The dot product is a *linear function* only.

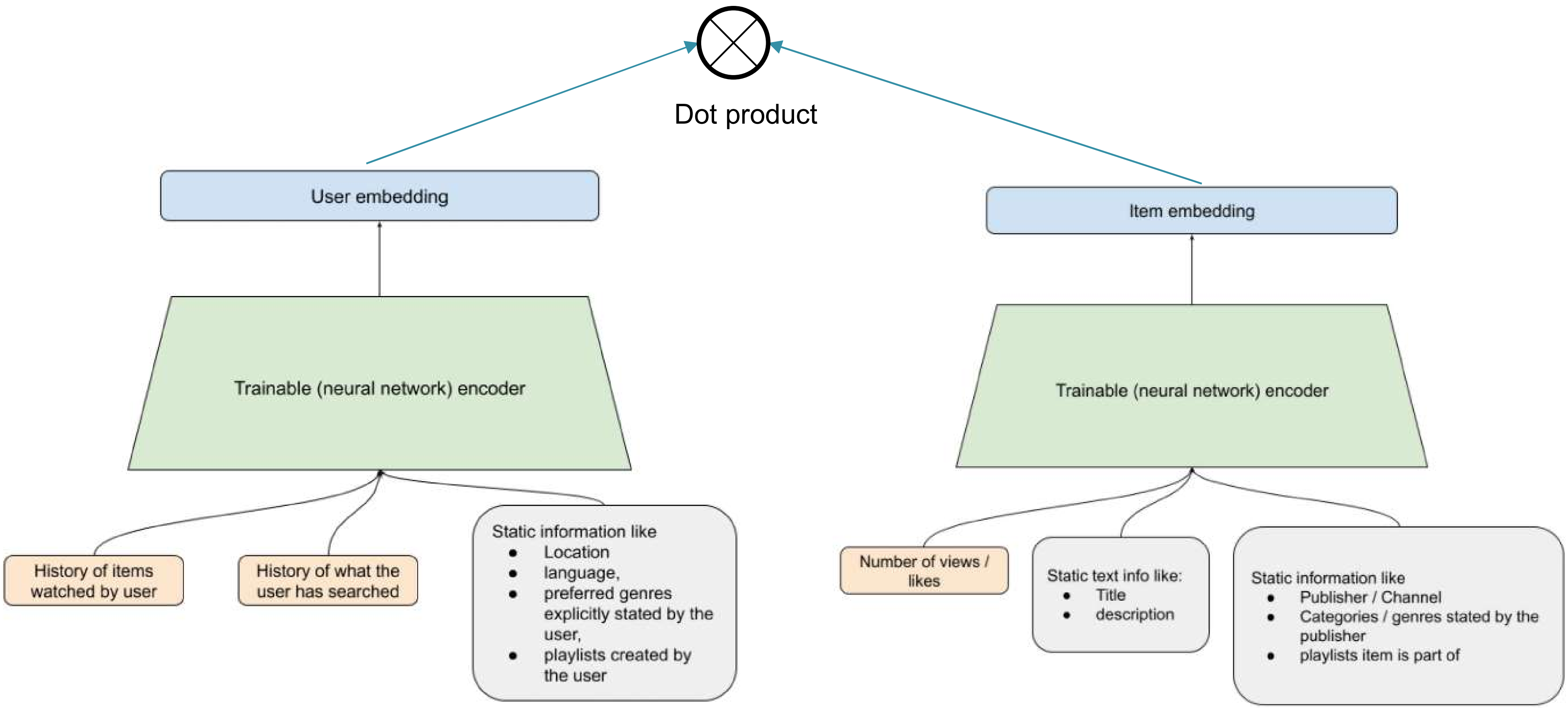
In NCF, the dot product is replaced with a FNN that can learn an *arbitrary function*.



Neural Collaborative Filtering, see https://www.comp.nus.edu.sg/~xiangnan/papers/ncf.pdf

Adding "Side" Information

- CF using Matrix Factorisation does not use side information – information other than user ratings of items. Using a NN to learn embeddings allows side information to be utilised.
- Below system uses a dot product, but a FNN could also be used (as in previous slide)
- This architecture is often called a two-tower architecture



https://www.linkedin.com/pulse/personalized-recommendations-iv-two-tower-models-gaurav-chakravorty

Google Play Store Recommendations

- Comprises a retrieval system (that retrieves relevant items from a huge inventory given a query) and a ranking system (that ranks the retrieved items, similar to click models).
- Below shows the retrieval part:

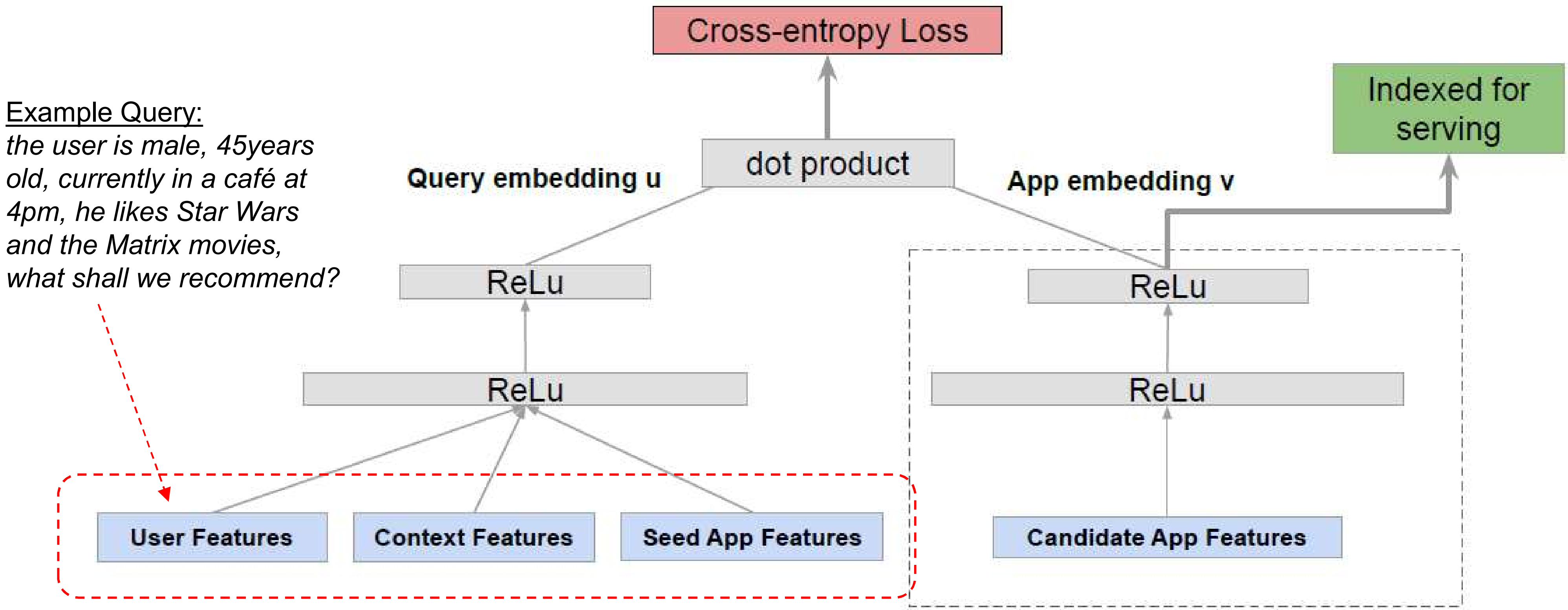
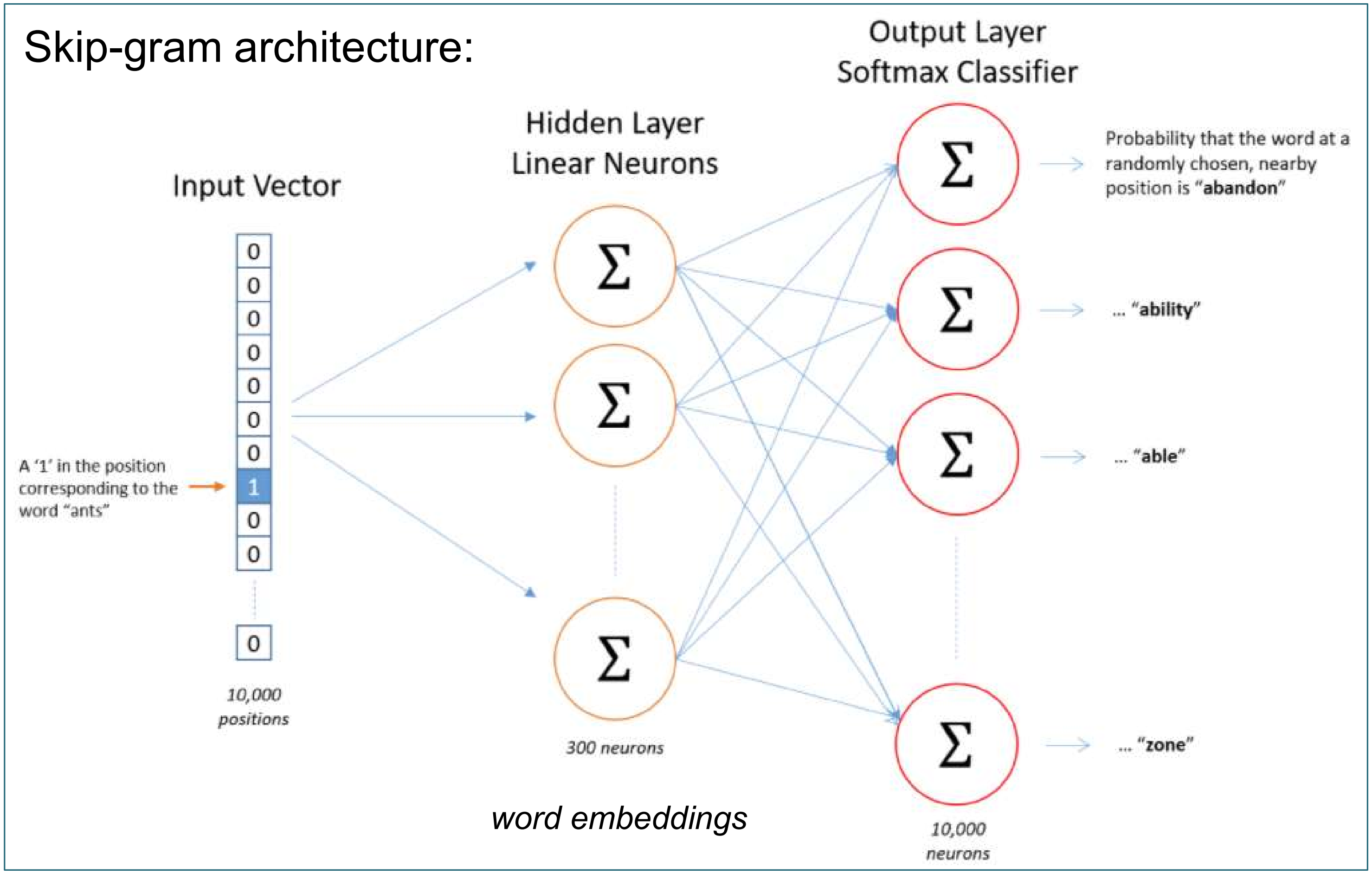


Figure 5: Two-tower model architecture for Google Play app recommendation.

https://research.google/pubs/pub50257/

Leveraging NLP methods: Prod2Vec

- NLP methods exist for mapping words into a vector space such that similar words are near each other. The word vectors (embeddings) represent how a word interacts with other words in a corpus
- **Word2Vec** (Google) trains words against other words that neighbour them in the input corpus. Skip-Gram is one architecture. Can be used to predict the next word in a sentence (language model)



Prod2Vec: Treats products as word tokens and uses a browsing session (clickstream) to learn product embeddings. A sequence of viewed products is analogous to a sentence in word2vec

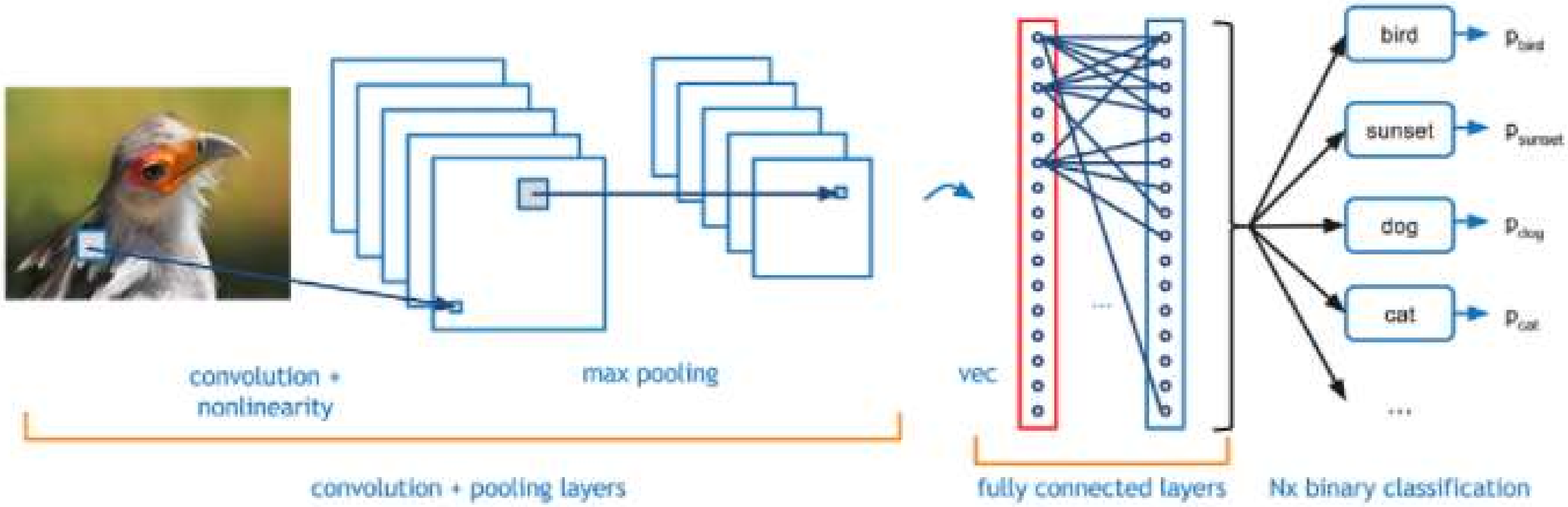
After training, supply a product as input to get likelihoods of other products being nearby.

Or use the vectors (embeddings) to generate a product-product similarity matrix for content-based filtering

https://www.coveo.com/blog/what-is-prod2vec/

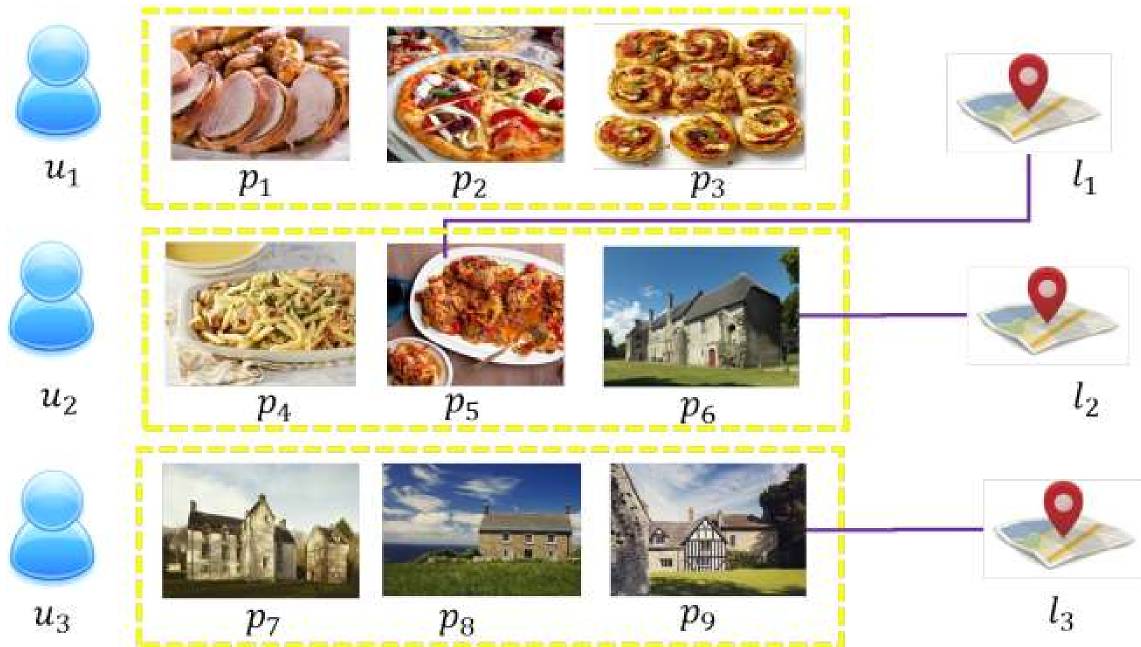
Convolutional Neural Networks (CNN)

- CNN is a feed-forward NN with convolution layers and pooling operations
- Widely used for computer vision, a key advantage is its ability to (automatically) extract features from images
- For recommender systems, a convolution layer is often added to help process and extract features from non-text inputs, e.g. images, audio, music, video, …



Visual 'Likes': Handling Product Images

- E.g. Point of Interest Recommendation
- With location-based social networks such as Yelp and Instagram, users can upload photos of Points of Interest. These reflect user interests and also provide informative descriptions about locations. E.g. a user who posts architecture photos is more likely to visit famous landmarks; a user who posts images about food is more likely to visit restaurants.



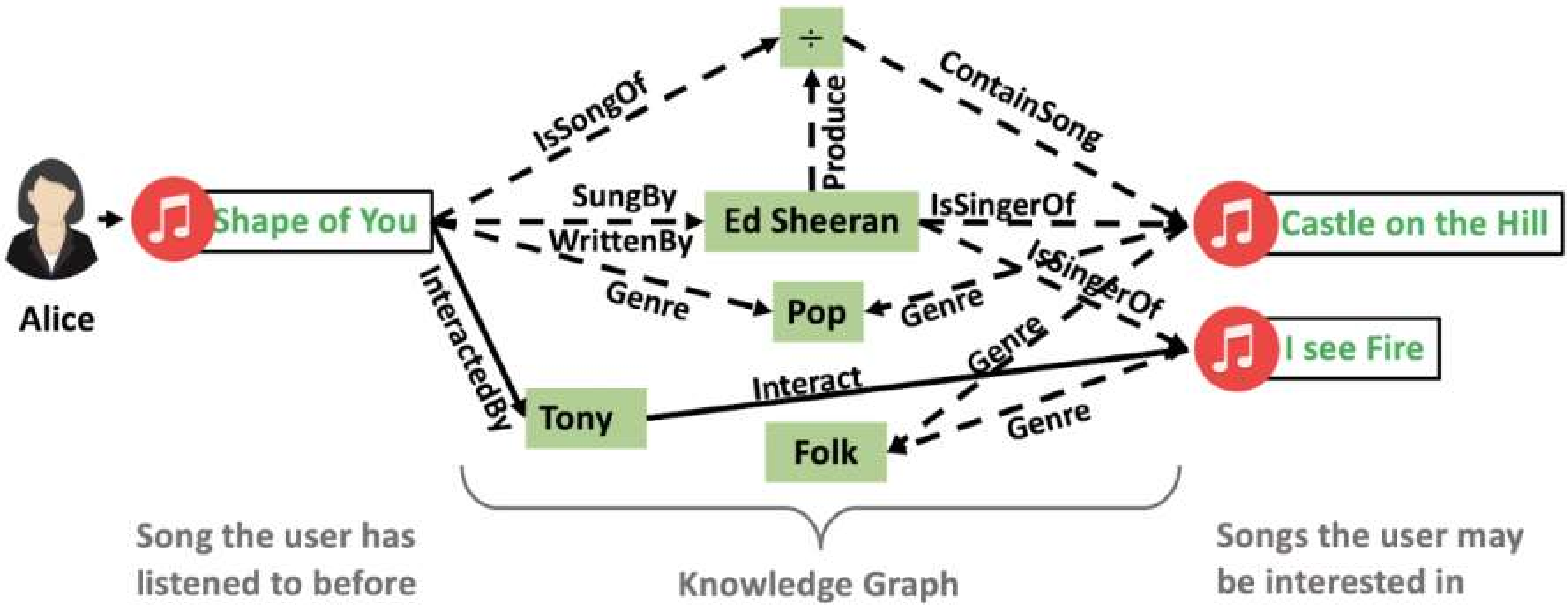
Example of Images Posted by Users.

A CNN (Convolutional Neural Network) is used to extract features from the uploaded images, these are then used as inputs by a probabilistic matrix factorisation engine

http://www.public.asu.edu/~skai2/files/WWW_suhang.pdf

Recommendation as a Graph Problem

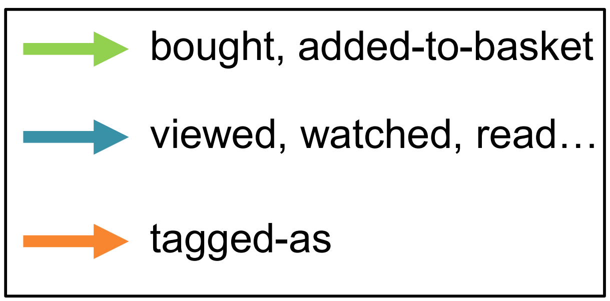
- Graphs can more easily express relationships between entities. E.g. between users, products and product creators (restaurants, studios, authors, musicians, artists etc).
- Using graphs, the connectivity between users and items can be discovered as paths, which provide rich and complementary information to user-item interactions when making recommendations

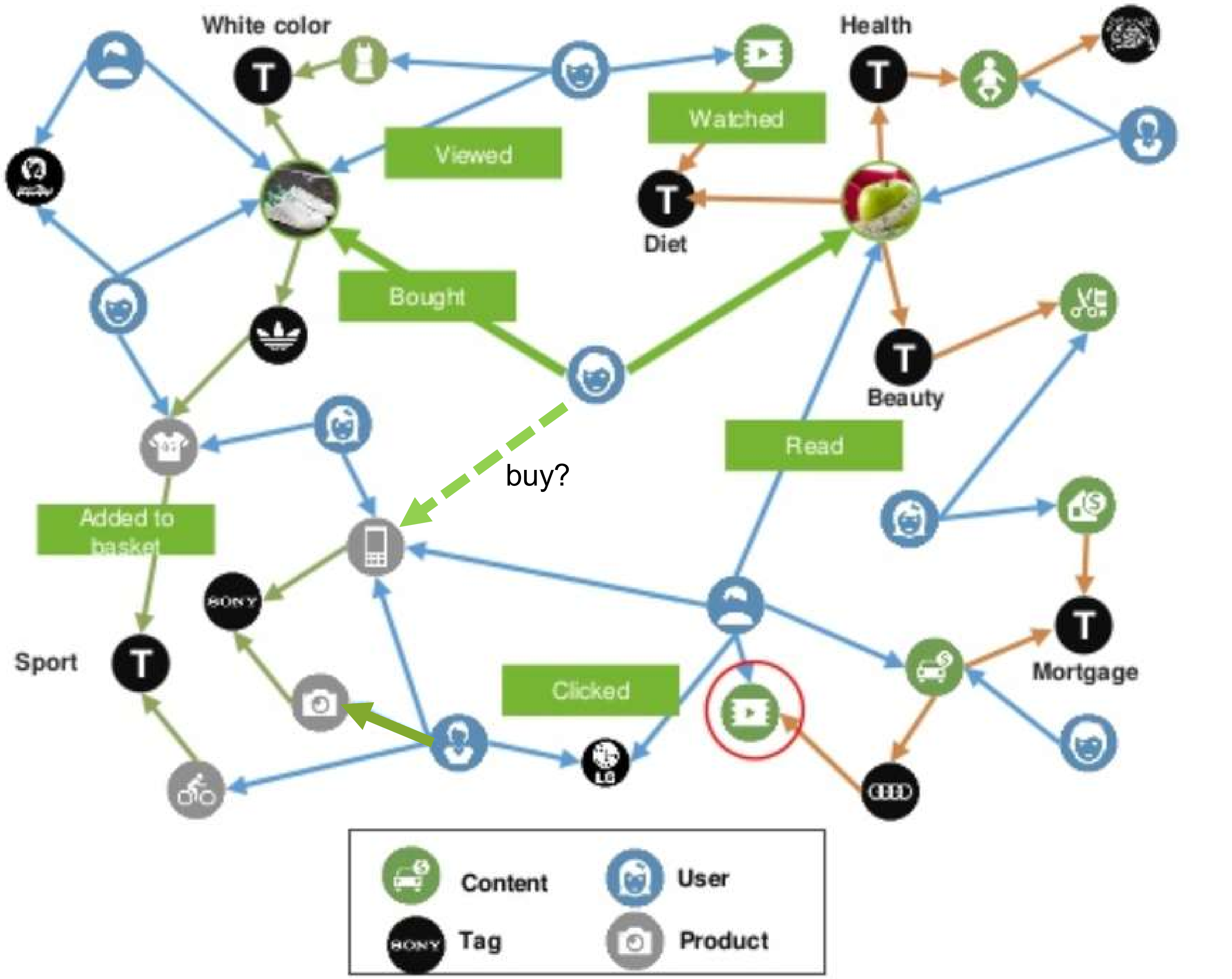


https://tech.ebayinc.com/research/explainable-reasoning-over-knowledge-graphs-for-recommendation/

Recommendation using Link Prediction

- Is a link likely to exist between a user node and an item node?
- For example, user interactions with an e-retail site





Link Prediction: Predictive Modelling Approach

- Generate training examples by considering all possible links ~ (a,b) pairs in the graph
- Model inputs = node properties and graph topology (relationship) features
- Model output = T/F does a link exist

node1					node2					Topology ??				
sex	**age**	**income**	**status**	**…**	**sex**	**age**	**income**	**status**	**…**					**link**
M	45	12000	single	…	M	23	5000	married	…					Y
M	45	12000	single		F	51	7200	married						N

Model inputs

Model output

- Instead of explicitly creating/selecting the features (model inputs) , we can use ML to learn a lower dimensional vector representation of the graph nodes and topology – these representations are called embeddings

Example: Predicting Road Links

neo4j docs

Node embeddings

> *This chapter provides explanations and examples for the node embedding algorithms in the Neo4j Graph Data Science library.*

Node embedding algorithms compute low-dimensional vector representations of nodes in a graph. These vectors, also called embeddings, can be used for machine learning. The Neo4j Graph Data Science library contains the following node embedding algorithms:

- Production-quality
  - FastRP
- Beta
  - GraphSAGE
- Alpha
  - Node2Vec

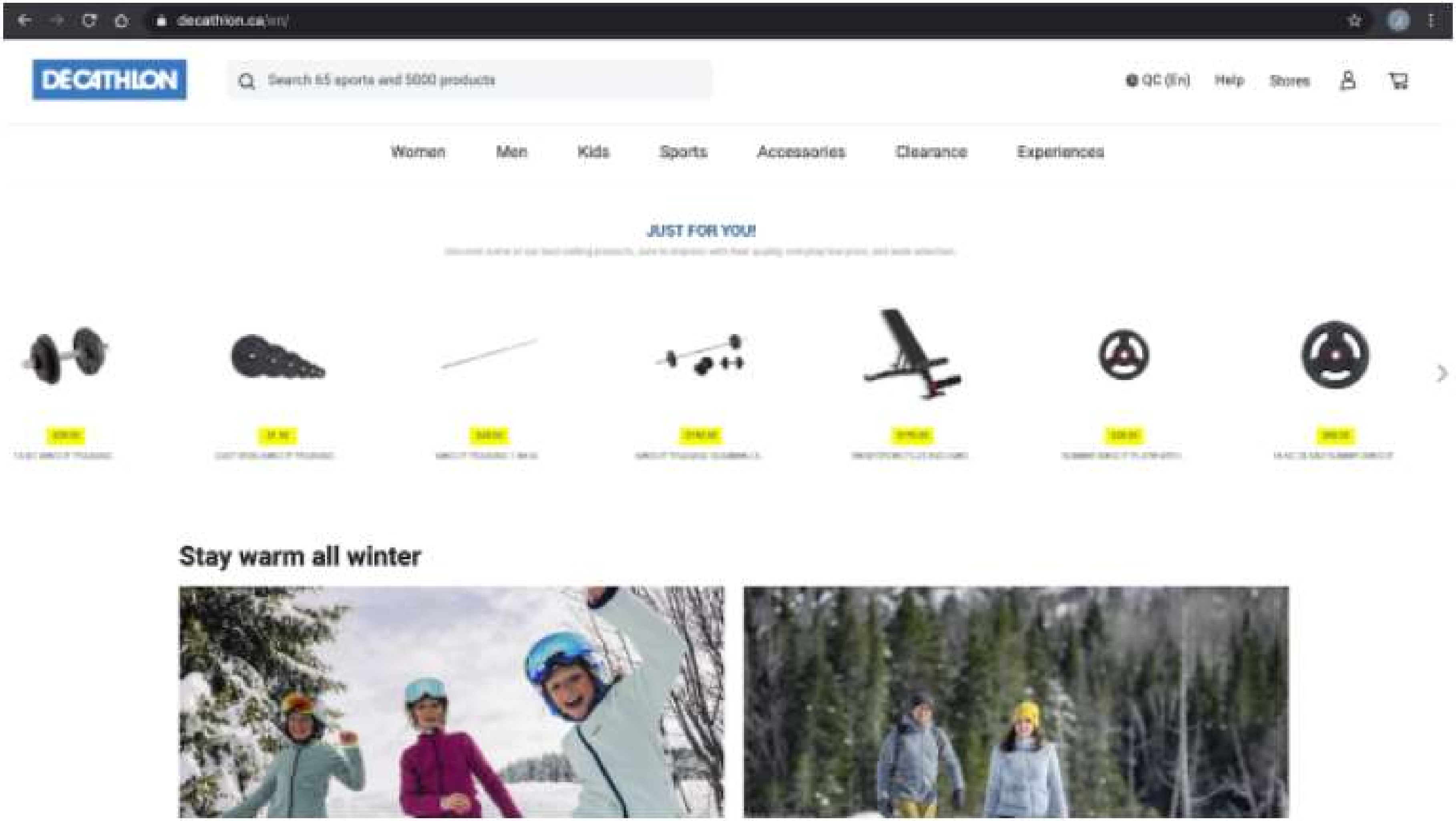
https://neo4j.com/developer/graph-data-science/link-prediction/

Road	Origin cntry	Origin place	Destn cntry	Destn place	Dist.	Water crossing
E01	GB	Larne	GB	Belfast	36	FALSE
E01	GB	Belfast	IRL	Dublin	165	FALSE
E01	IRL	Dublin	IRL	Wexford	140	FALSE
E01	IRL	Wexford	IRL	Rosslare	19	FALSE

place	embedding
"Larne"	[2.1327764987945557, 1.0994384288787842, 0.49434158205986023, 2.040088176727295, 1.0031780004501343, 0.873113751411438, -2.413508176803589, -2.3154311180114746, 0.9832170009613037, -2.1525325775146484]
"Belfast"	[2.593827962875366, 0.7630942463874817, 0.7039147019386292, 2.3797214031219482, 0.7776350378990173, 0.5327662825584412, -2.3597097396850586, -1.9403077363967896, 1.2757759094238281, -1.681670904159546]
"Dublin"	[1.9023665189743042, 0.795767068862915, 0.722218930721283, 2.71242094039917, 0.21453168988227844, 0.3935716152191162, -2.1960527896881104, -2.6851491928100586, 1.0708848237991333, -0.6451507806777954]
"Wexford"	[2.0736780166625977, 1.1650514602661133, 0.4161956012248993, 2.8500888347625732, -0.12804043292999268, 0.355782151222229, -2.719728946685791, -2.6983509063720703, 0.5993242859840393, 0.265157550573349]
"Rosslare"	[1.8750609159469604, 1.2445515394210815, 0.630532443523407, 2.4588329792022705, 0.1135958656668663, 0.007978626526892185, -2.7186481952667236, -2.418004035949707, 0.3507269024848938, 0.9638977646827698]

road	Node1 (origin) embedding					Node2 (destination) embedding					Link
E01											**Y**
E07											Y
---											N

Decathlon Recommender Case-Study*



Example of item recommendations for a user interested in weightlifting. (Reconstructed, not actual screenshot of website)

**A research project conducted with Decathlon Canada:*
https://medium.com/decathlontechnology/building-a-recommender-system-using-graph-neural-networks-2ee5fc4e706d

Decathlon Data

- The nodes are *users*, *items* and *sports* *
- The edges are *click, purchase, favorite, related and belonging* *
- Example edge (link) data:

customer_id	item_id	timestamp	click	purchase
imbvblxwvtiywunh	3384934262863770	2018-01-01	0	1
eqeefiaxvjxsyfxk	5460080147661416	2018-01-02	1	0
...	...	...	...	...
mkfqblfqwgzivfga	8625784510427076	2019-12-30	1	0
cnspkotxubxnxtzk	5150255386059428	2019-12-31	0	1

User-item interactions. (Data was modified to protect confidentiality)

* Typically, GNN recommender systems use bipartite graphs, with only user and item nodes. We added sports as nodes and multiple edge types. Compared to a simple bipartite graph, this complex graph significantly enhances the model's performance; the model has a richer source of information to learn from.

Decathlon Data

- Example Node data:

item_id	male	female	junior	label	family	department	universe	linked sports
3384934262863770	0	1	0	Jacket 920 Hood W Greyneps	Woman Pant Jacket Sweat	Woman Apparel	Pilates, Soft Gym	Musculation
...	...	...	...	...	...	...	...	...
8625784510427076	0	0	0	Bar 0,35m	Free Weights And Equipment	Bodybuilding	Cross-training,Bodybuilding	Musculation

Item features. (Data was modified to protect confidentiality)

customer_id	declared sports
imbvblxwvtiywunh	baseball; boxing
xbnkvygodmlyfete	
...	...
mkfqblfqwgzivfga	soccer
cnspkotxubxnxtzk	capoeira; beachtennis

User features. (Data was modified to protect confidentiality)