RNNs for Recommendation & Personalization

Nick Pentreath
Principal Engineer

@MLnick

IBM CODE

About

@MLnick on Twitter & Github

Principal Engineer, IBM

CODAIT - Center for Open-Source Data & AI Technologies

Machine Learning & AI

Apache Spark committer & PMC

Author of *Machine Learning with Spark*

Various conferences & meetups

Center for Open Source Data and AI Technologies

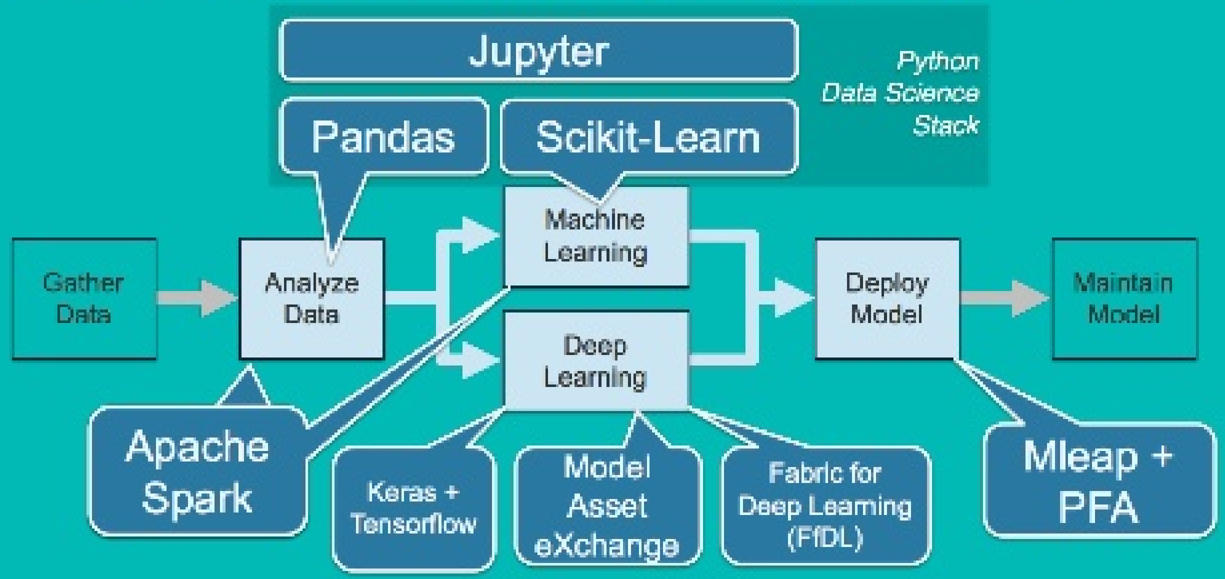
CODAIT aims to make AI solutions dramatically easier to create, deploy, and manage in the enterprise

Relaunch of the Spark Technology Center (STC) to reflect expanded mission

CODAIT

codait.org

Improving Enterprise AI Lifecycle in Open Source



Agenda

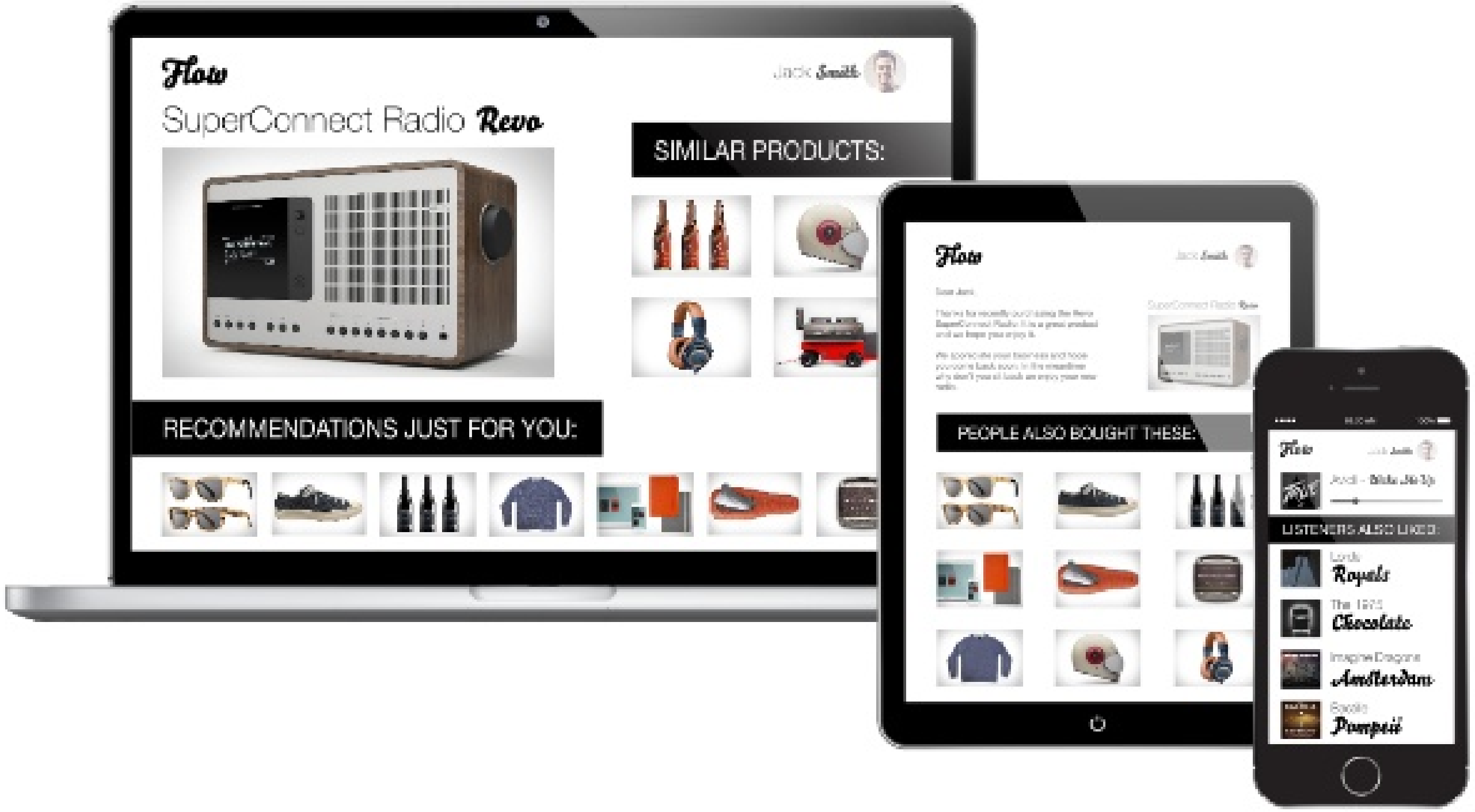
Recommender systems overview

Deep learning and RNNs

RNNs for recommendations

Challenges and future directions

Recommender Systems



Users and Items


```
{
  "user_id": "1",
  "name": "Joe Bloggs",
  "created_date": 1476884080,
  "updated_date": 1476946916,
  "last_active_date": 1476946962,
  "age": 32,
  "country": "GB",
  "city": "London",
  ...
}
```

```
{
  "item_id": "10",
  "name": "LOL Cats",
  "description": "catscatscats",
  "category": ["Cat Videos", "Humour", "Animals"],
  "tags": ["cat", "lol", "funny", "cats", "felines"],
  "created_date": 1476884080,
  "updated_date": 1476884080,
  "last_played_date": 1476946962,
  "likes": 100000,
  "author_id": "321",
  "author_name": "ilikecats",
  "channel_id": "CatVideoCentral",
  ...
}
```

Events

Implicit preference data

- Online – page view, click, app interaction
- Commerce – cart, purchase, return
- Media – preview, watch, listen

Explicit preference data

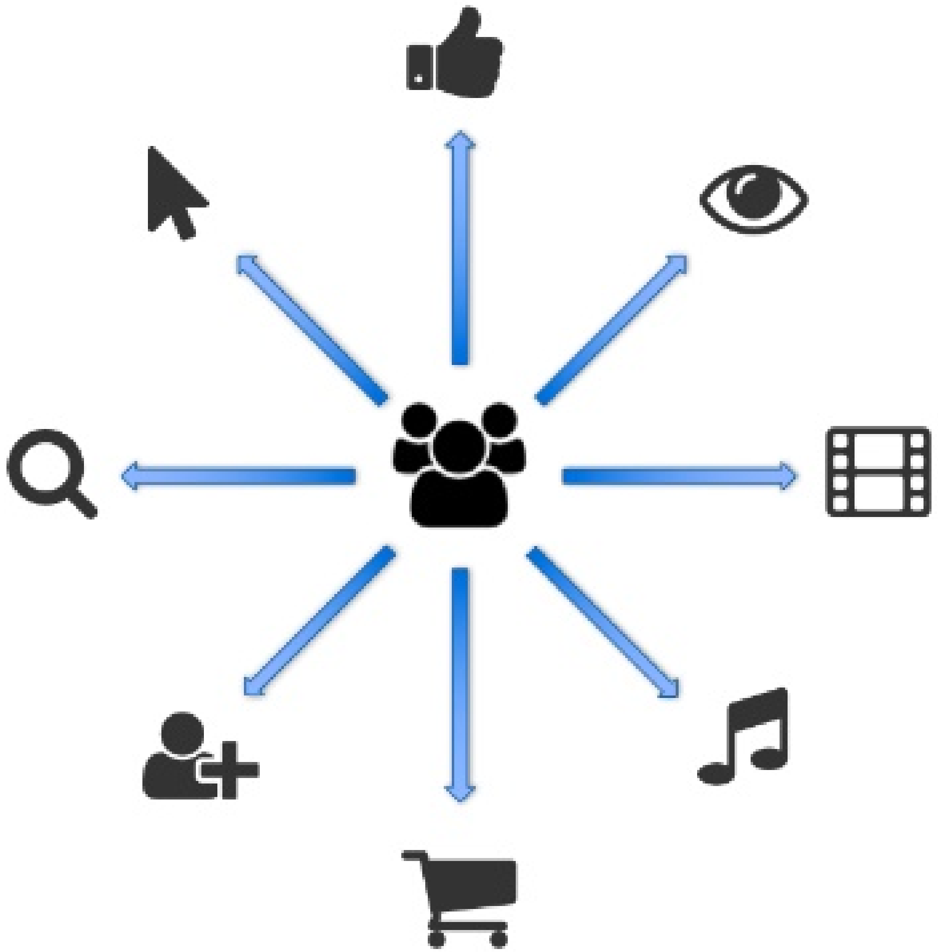
- Ratings, reviews

Intent

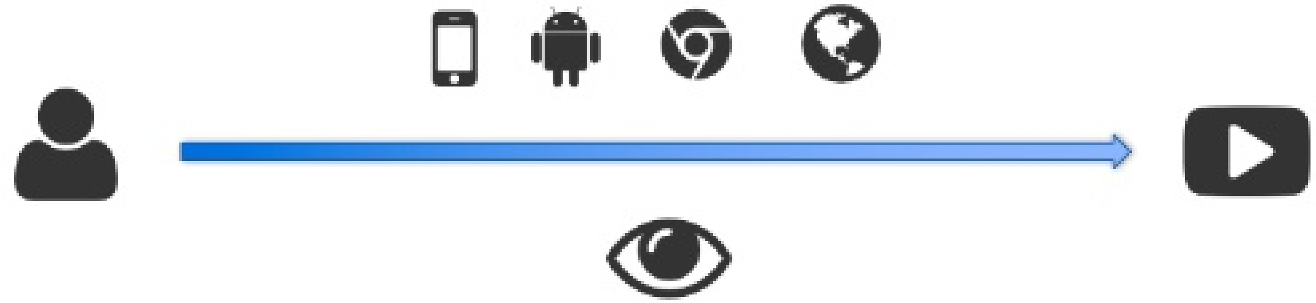
- Search query

Social

- Like, share, follow, unfollow, block



Context

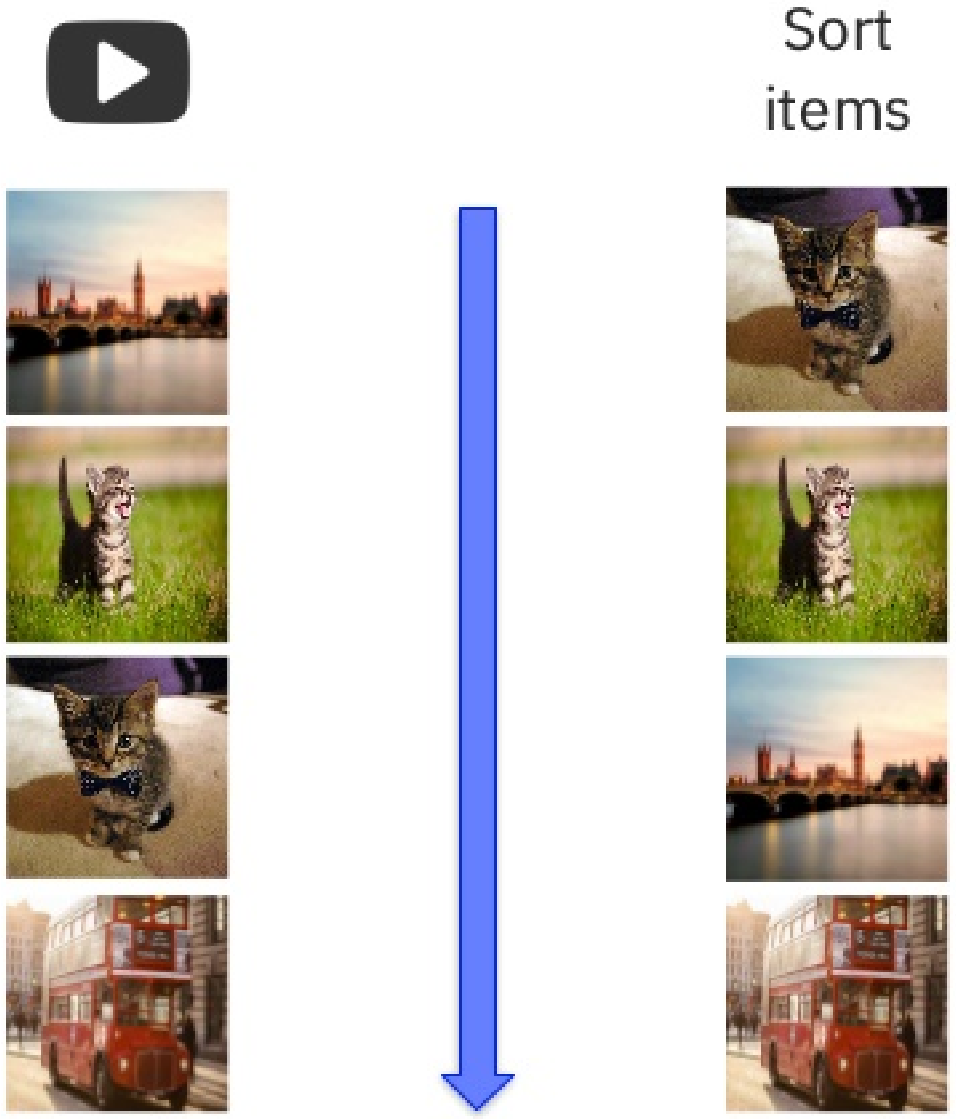



```
{
  "user_id": "1",
  "item_id": "10",
  "event_type": "page_view",
  "timestamp": 1476884080,
  "referrer": "http://codait.org",
  "ip": "123.12.12.12",
  "device_type": "Smartphone",
  "user_agent_os": "Android",
  "user_agent_type": "Mobile Browser",
  "user_agent_family": "Chrome Mobile",
  "geo":"51.5085, 0.0298"
  ...
}
```

Prediction

Prediction is ranking

– Given a user and context, rank the available items in order of likelihood that the user will interact with them



Matrix Factorization

The *de facto* standard model

– Represent user ratings as a user-item matrix

– Find two smaller matrices (called the *factor* matrices) that approximate the full matrix

– Minimize the reconstruction error (i.e. rating prediction / completion)

– Efficient, scalable algorithms

- Gradient Descent
- Alternating Least Squares (ALS)

– Prediction is simple

– Can handle implicit data

$$\begin{bmatrix} 3 & & & & 4 \\ & 1 & & & \\ & 5 & & 2 & \\ 1 & & & & 3 \\ & & 2 & 1 & \end{bmatrix} \Rightarrow \begin{bmatrix} -1.1 & 3.2 & 4.3 \\ 0.2 & 1.4 & 3.1 \\ 2.5 & 0.3 & 2.3 \\ 4.3 & -2.4 & 0.5 \\ 3.6 & 0.3 & 1.2 \end{bmatrix} \times \begin{bmatrix} 0.2 & 1.7 & 2.3 \\ 1.9 & 0.4 & 0.8 \\ 1.5 & -1.2 & 0.3 \\ -0.4 & 2.1 & 0.6 \\ 2.7 & 0.8 & 1.4 \end{bmatrix}$$

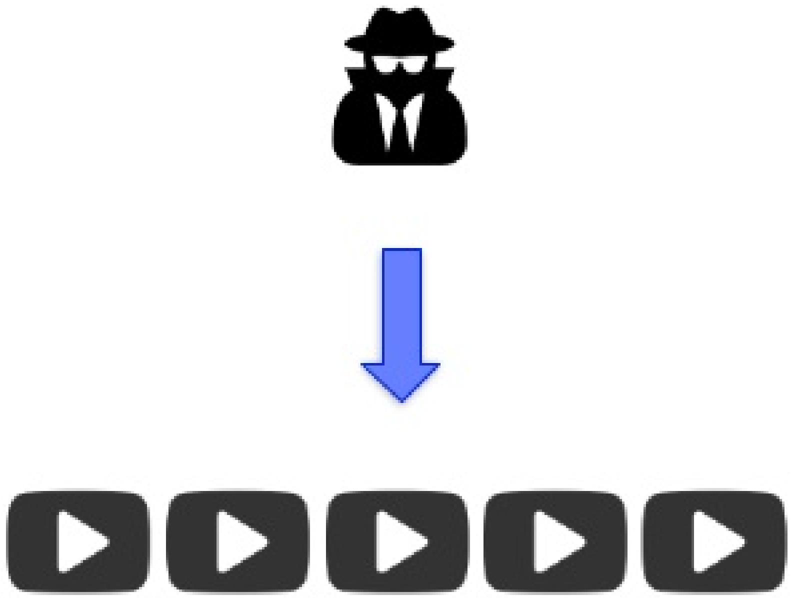
Cold Start

New items

- No historical interaction data
- Typically use baselines (e.g. populariy) or item content

New (or unknown) users

- Previously unseen or anonymous users have no user profile or historical interactions
- Have context data (but possibly very limited)
- Cannot directly use collaborative filtering models
  - Item-similarity for current item
  - Represent session as aggregation of items
  - Contextual models can incorporate short-term history



Deep Learning and Recurrent Neural Networks

Source: Wikipedia

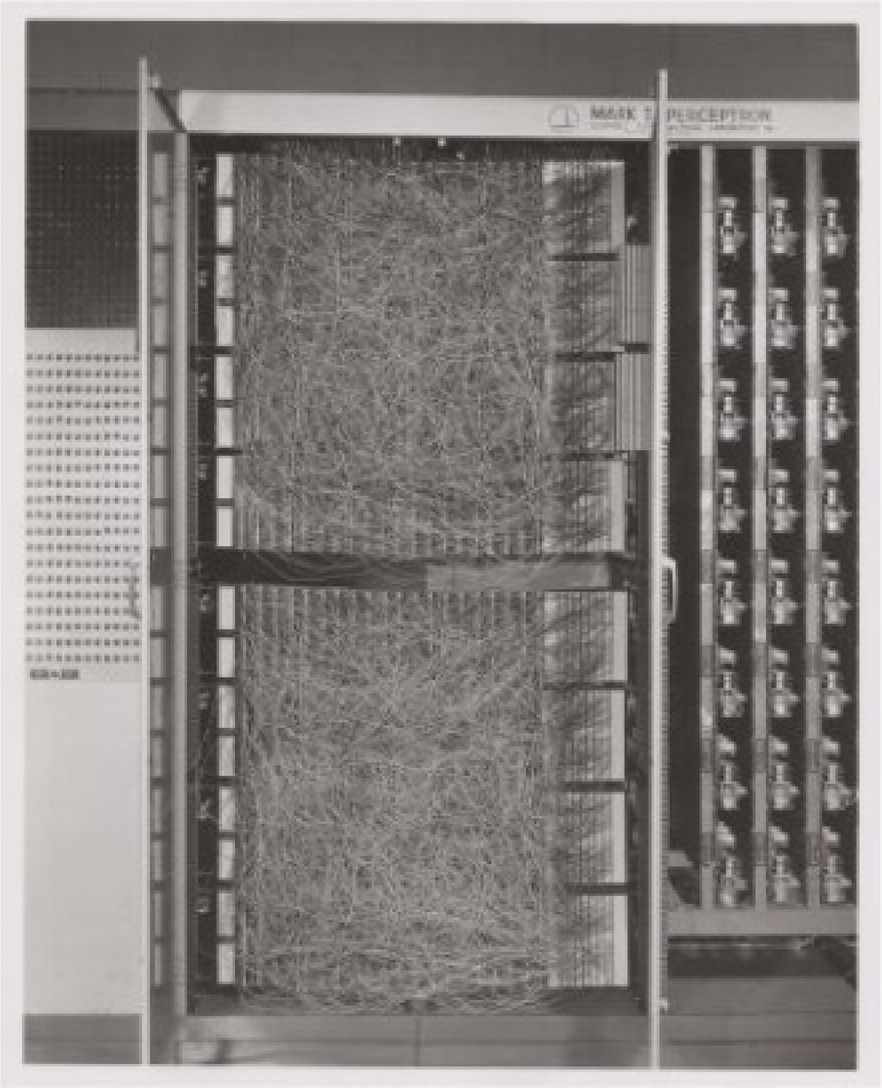
Overview

Original theory from 1940s; computer models originated around 1960s; fell out of favor in 1980s/90s

Recent resurgence due to

- Bigger (and better) data; standard datasets (e.g. ImageNet)
- Better hardware (GPUs)
- Improvements to algorithms, architectures and optimization

Leading to new state-of-the-art results in computer vision (images and video); speech/text; language translation and more



Source: Stanford CS231n

Modern Neural Networks

Deep (multi-layer) networks

Computer vision

- Convolution neural networks (CNNs)
- Image classification, object detection, segmentation

Sequences and time-series

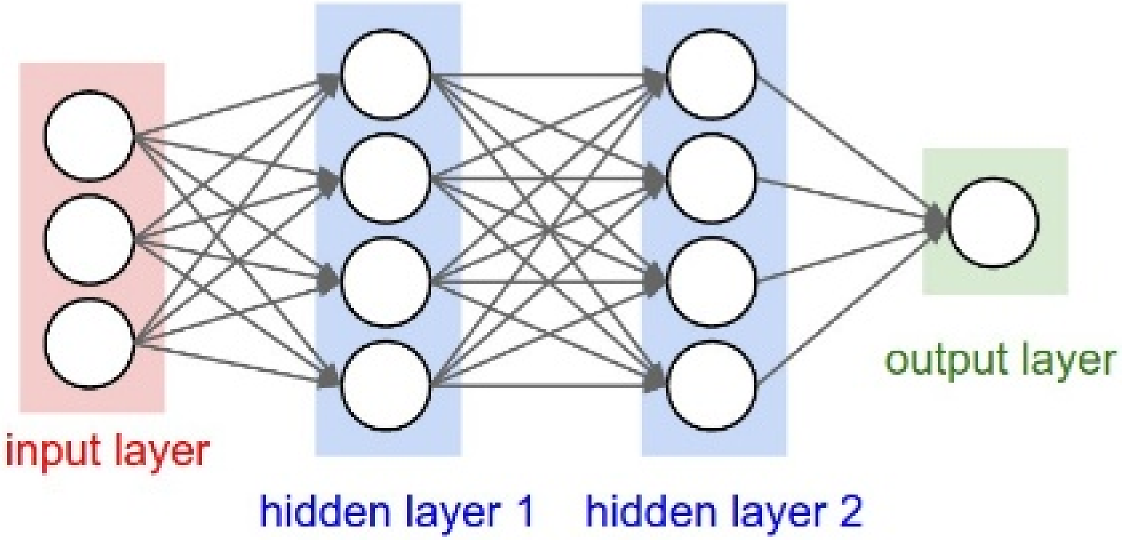
- Recurrent neural networks (RNNs)
- Machine translation, text generation
- LSTMs, GRUs

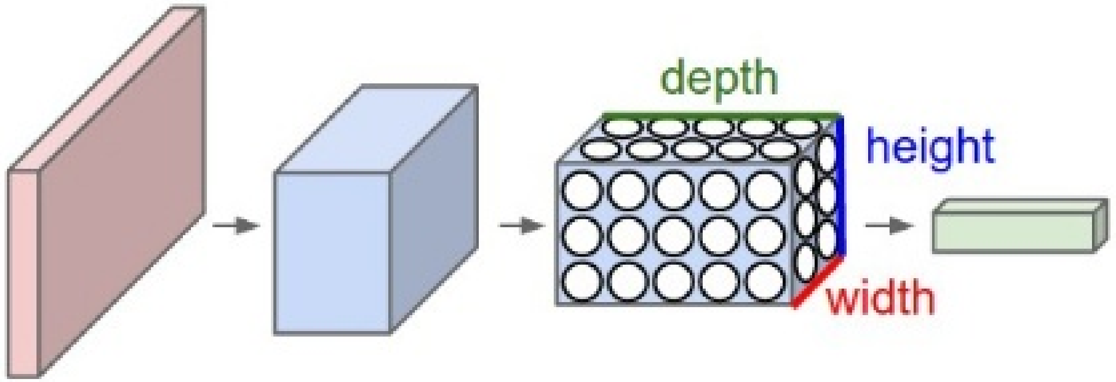
Embeddings

- Text, categorical features

Deep learning frameworks

- Flexibility, computation graphs, auto-differentiation, GPUs



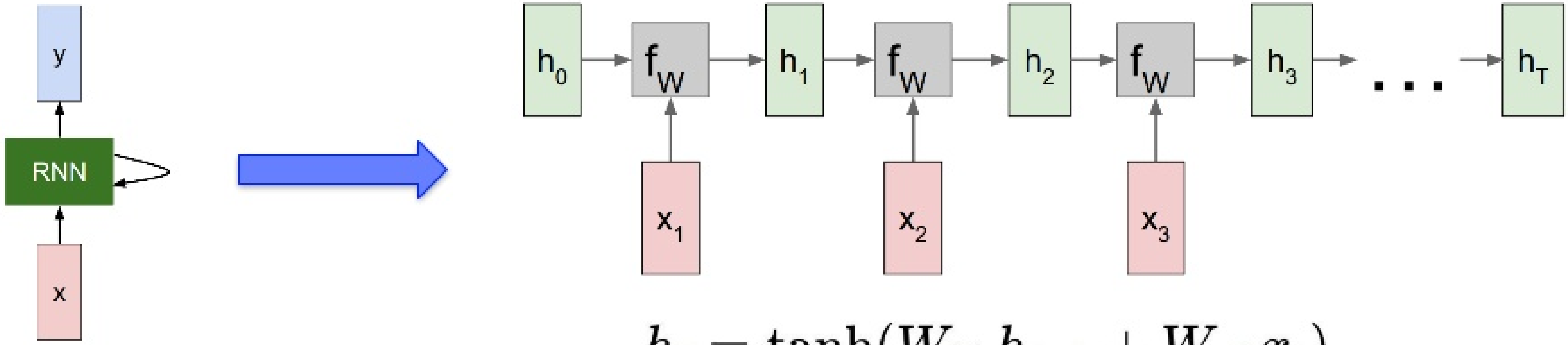


Source: Stanford CS231n

Recurrent Neural Networks

Neural Network on Sequences …

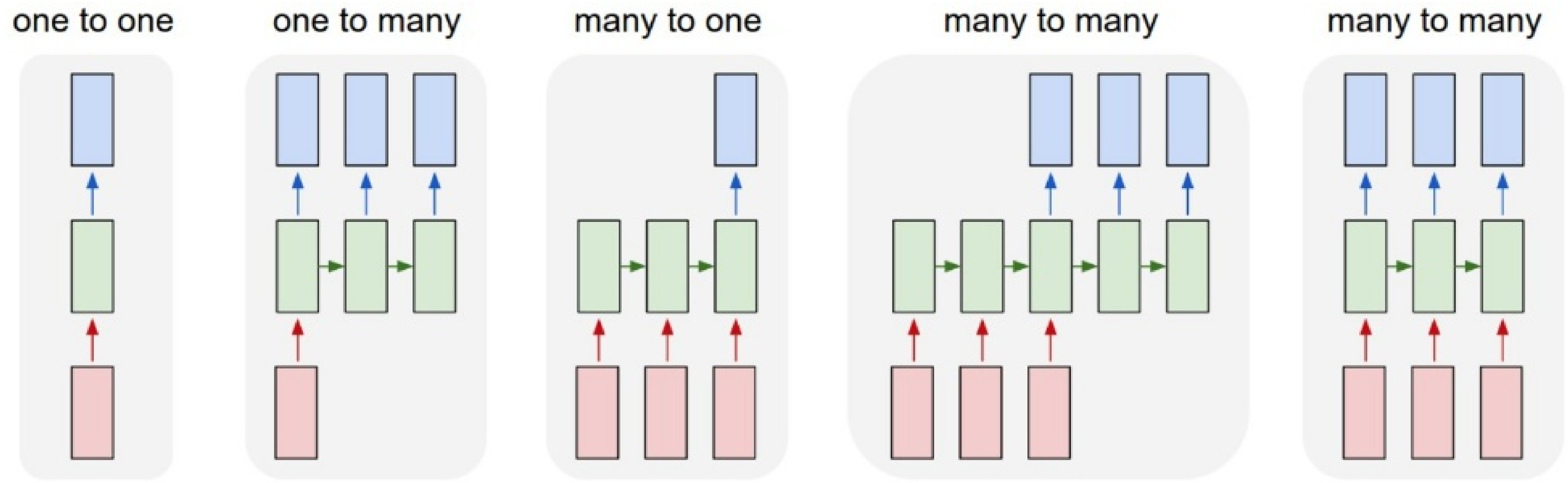
- … sequence of neural network (layers)
- Hidden layers (state) dependent on previous state as well as current input
- “memory” of what came before
- Share weights across all time steps
- Training using backpropagation through time (BPTT)



$$h_t = \tanh(W_{hh} h_{t-1} + W_{xh} x_t)$$

Source: Andrej Karpathy

Recurrent Neural Networks



Source: Stanford CS231n

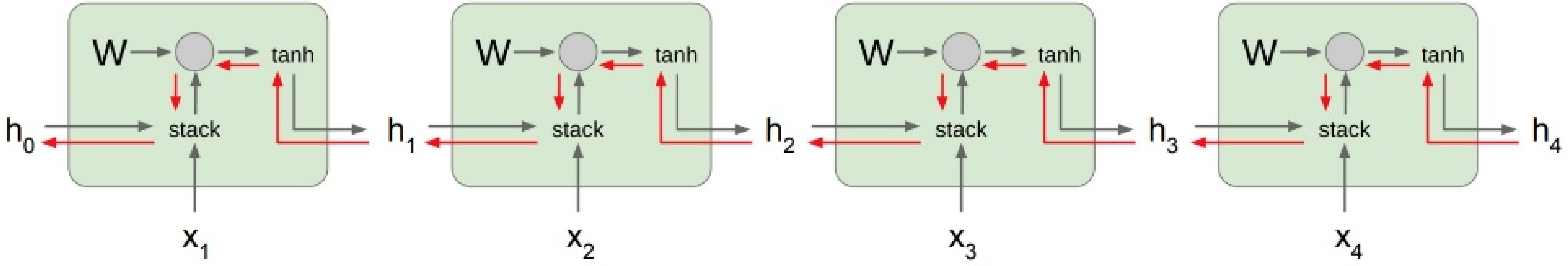
Recurrent Neural Networks

Issues

- Exploding gradients - clip / scale gradients
- Vanishing gradients

Solutions

- Truncated BPTT
- Restrict sequence length
- Cannot encode very long term memory

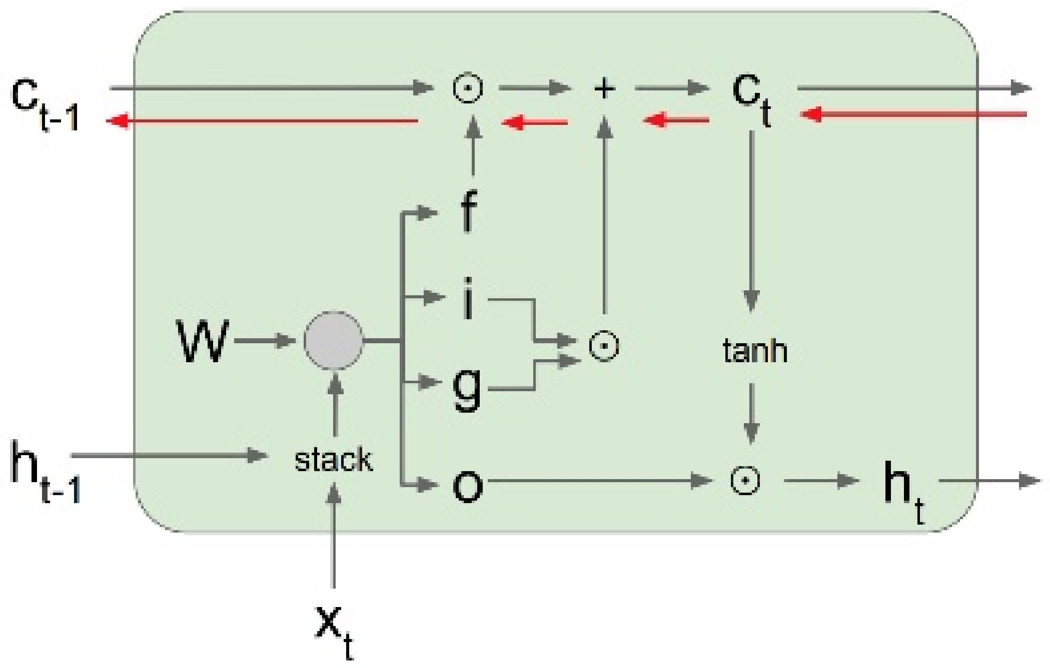


Source: Stanford CS231n; Hochreiter et al.

Recurrent Neural Networks

Long Short Term Memory (LSTM)

- Replace simple RNN layer (activation) with a LSTM cell
- Cell has 3 gates - Input (i), Forget (f), Output (o)
- Activation (g)
- Backpropagation depends only on elementwise operations (no matrix operations over *W*)



$$c_t = f \odot c_{t-1} + i \odot g$$

$$h_t = o \odot \tanh(c_t)$$

Gated Recurrent Unit (GRU)

- Effectively a simplified version of LSTM
- 2 gates instead of 3 - input and forget gate is combined into an update gate. No output gate

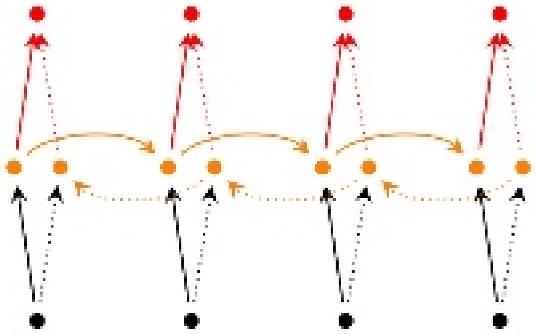
GRU has fewer parameters, LSTM may be more expressive

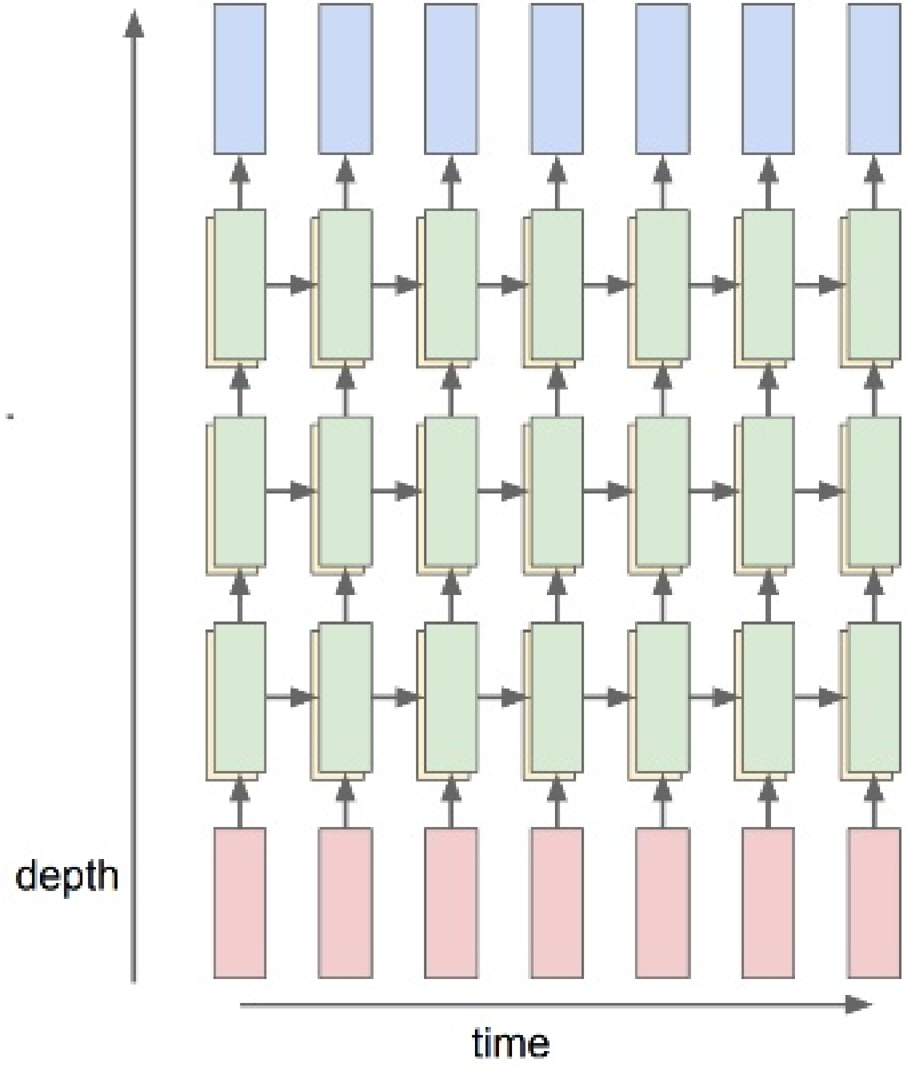
Source: Stanford CS231n; Denny Britz

Recurrent Neural Networks

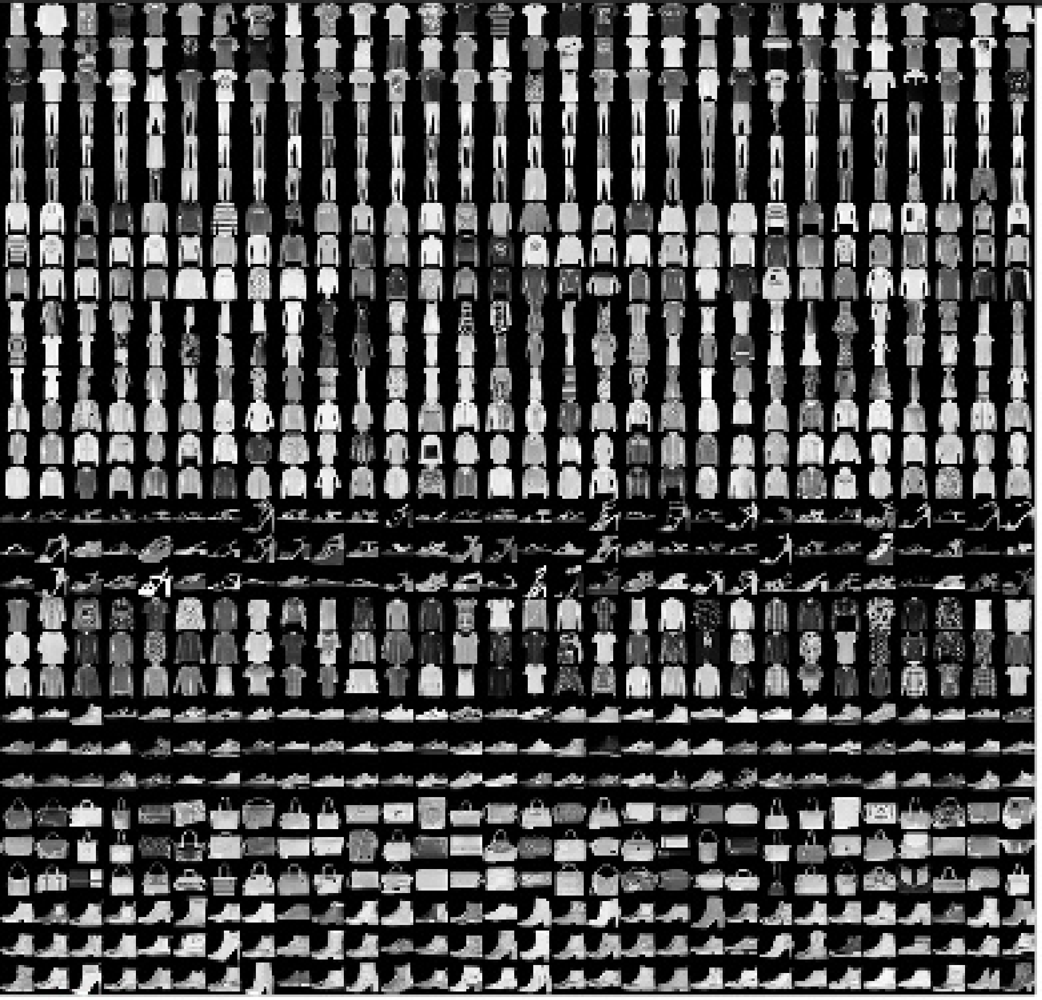
Variants

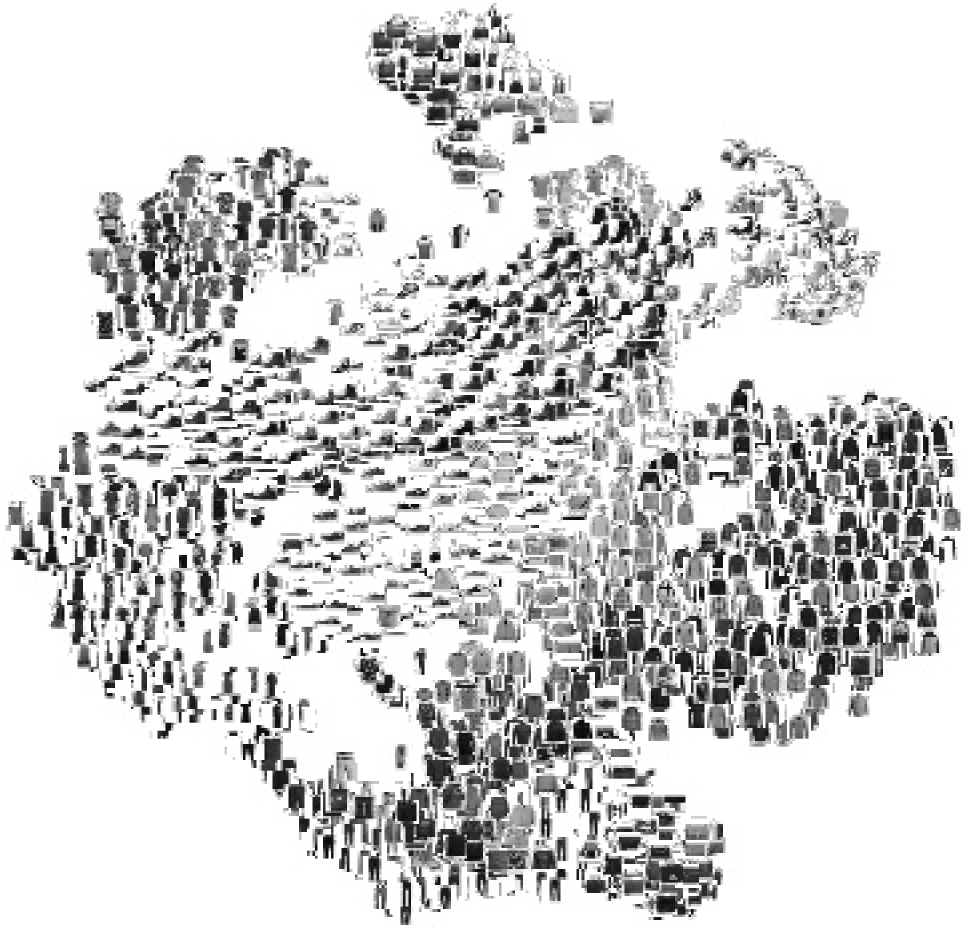
- Multi-layer (deep) RNNs
- Bi-directional
- Deep bi-directional
- Attention

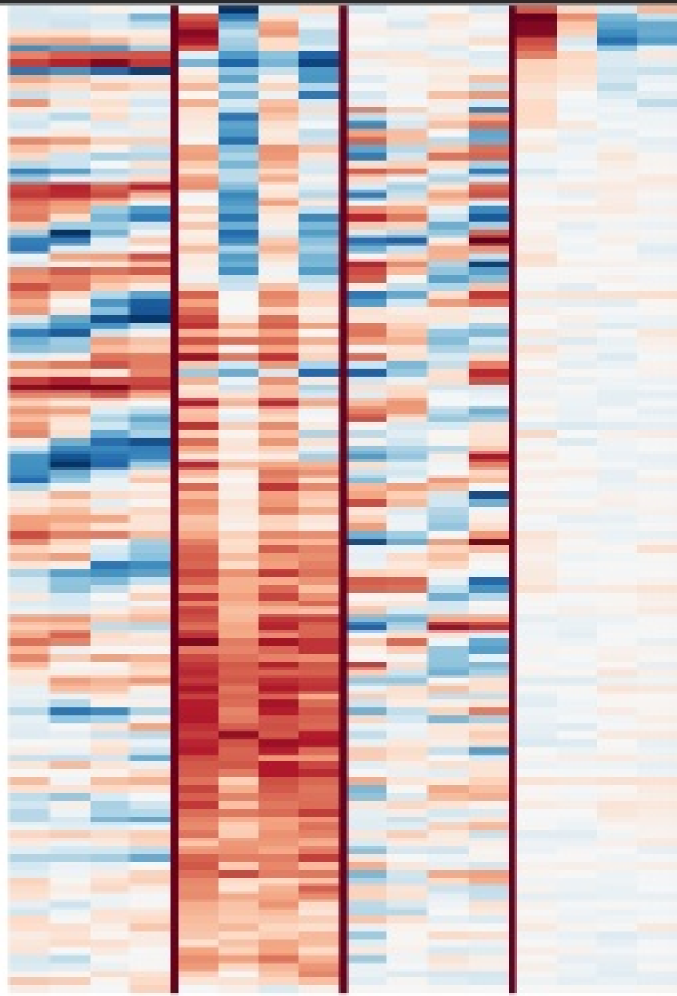




RNNs for Recommendations





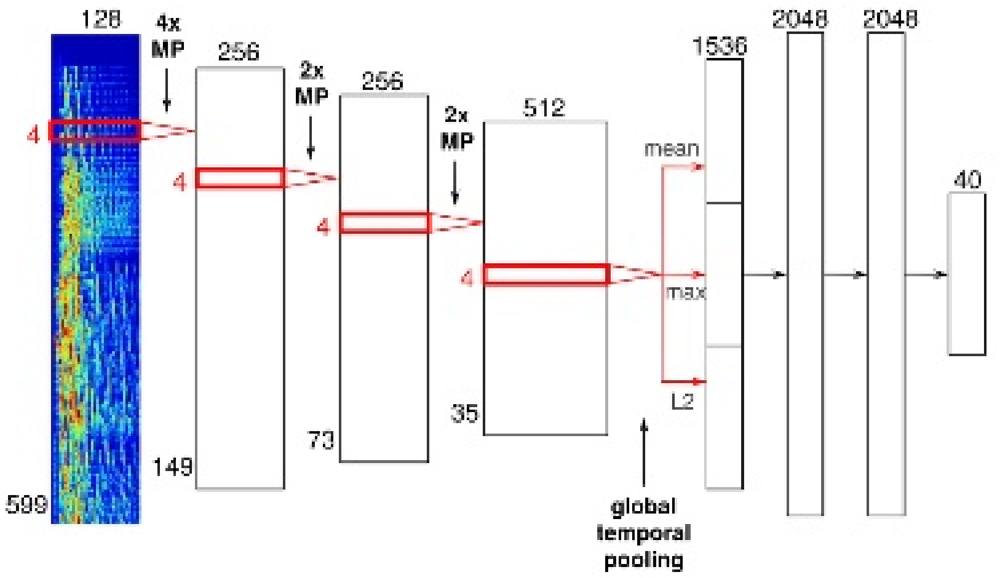


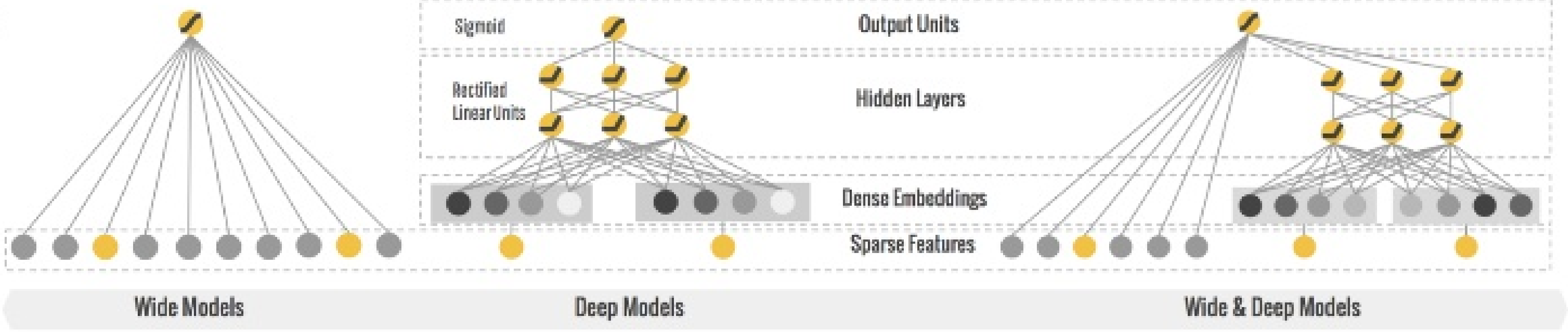
Source: Spotify / Sander Dieleman
Google Research

Deep Learning for Recommenders Overview

Most approaches have focused on combining

- Performance of collaborative filtering models (especially matrix factorization)
  - Embeddings with appropriate loss = MF
- Power of deep learning for feature extraction
  - CNNs for image content, audio, etc.
  - Embeddings for categorical features
  - Linear models for interactions
  - RNNs for text



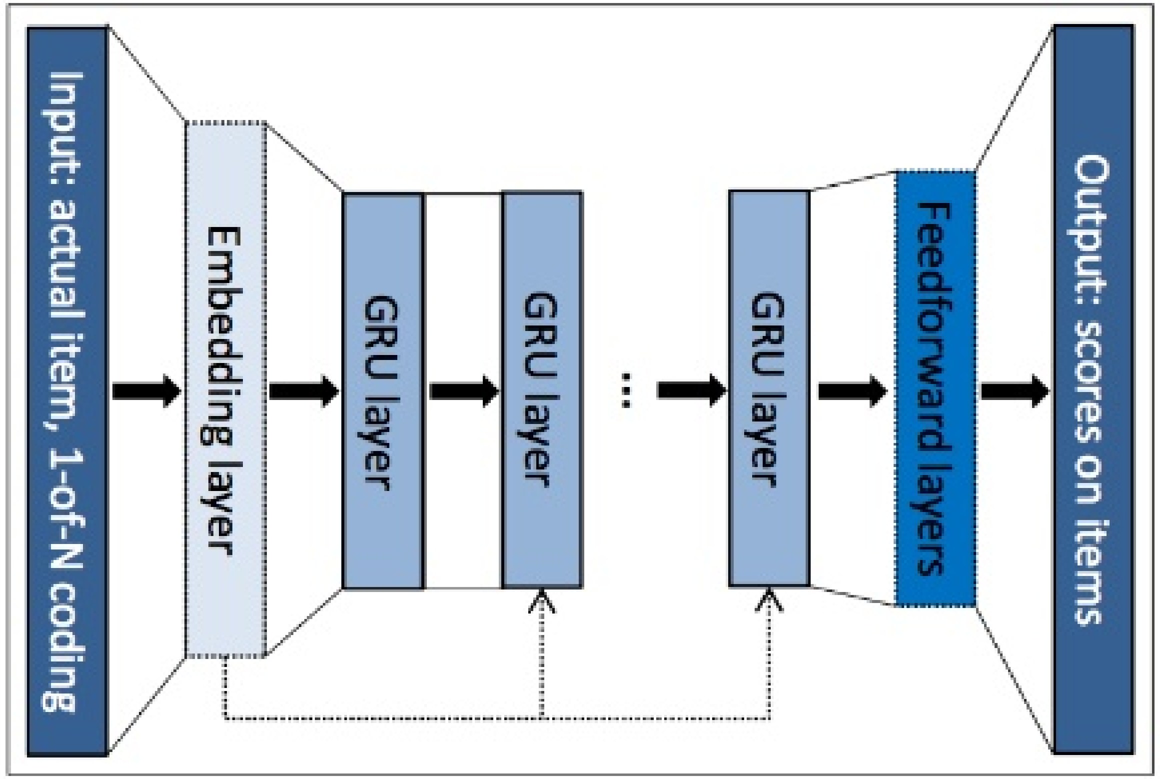


Source: Hidasi, Karatzoglou, Baltrunas, Tikk

Session-based recommendation

Apply the advances in sequence modeling from deep learning

- RNN architectures trained on the sequence of user events in a session (e.g. products viewed, purchased) to predict next item in session
- Adjustments for domain
  - Item encoding (1-of-N, weighted average)
  - Parallel mini-batch processing
  - Ranking losses – BPR , TOP1
  - Negative item sampling per mini-batch
- Report 20-30% accuracy gain over baselines

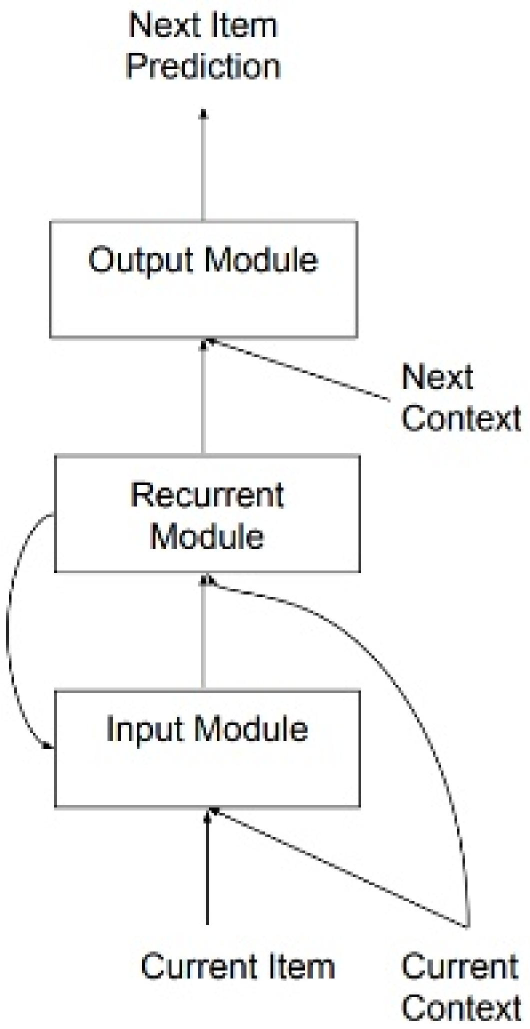


Source: Smirnova, Vasile

Contextual Session-based models

Add contextual data to the RNN architecture

- Context included time, time since last event, event type
- Combine context data with input / output layer
- Also combine context with the RNN layers
- About 3-6% improvement (in Recall@10 metric) over simple RNN baseline
- Importantly, model is even better at predicting sales (vs view, add to cart events) and at predicting new / fresh items (vs items the user has already seen)

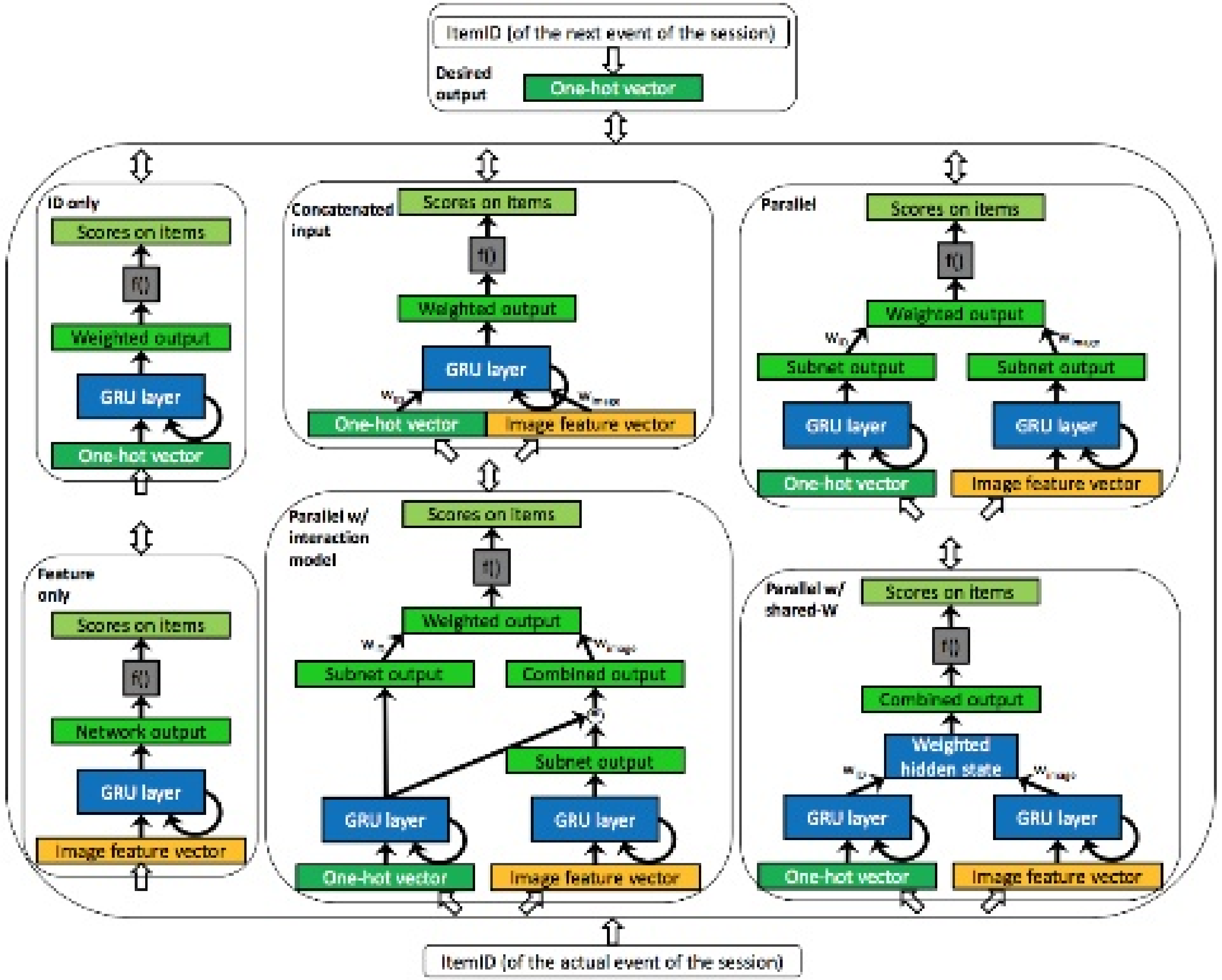


Source: Hidasi, Quadrana, Karatzoglou, Tikk

Content and Session-based models

Add content data to the RNN architecture

- Parallel RNN (p-RNN)
- Follows trend in combining DL architectures for content feature extraction with CF models for interaction data
  - CNN for image data
  - BOW for text (alternatives are Word2Vec-style models and RNN language models)
- Some training tricks
  - Alternating – keep one subnet fixed, train other
  - Residual – subnets trained on residual error
  - Interleaved – alternating training per mini-batch

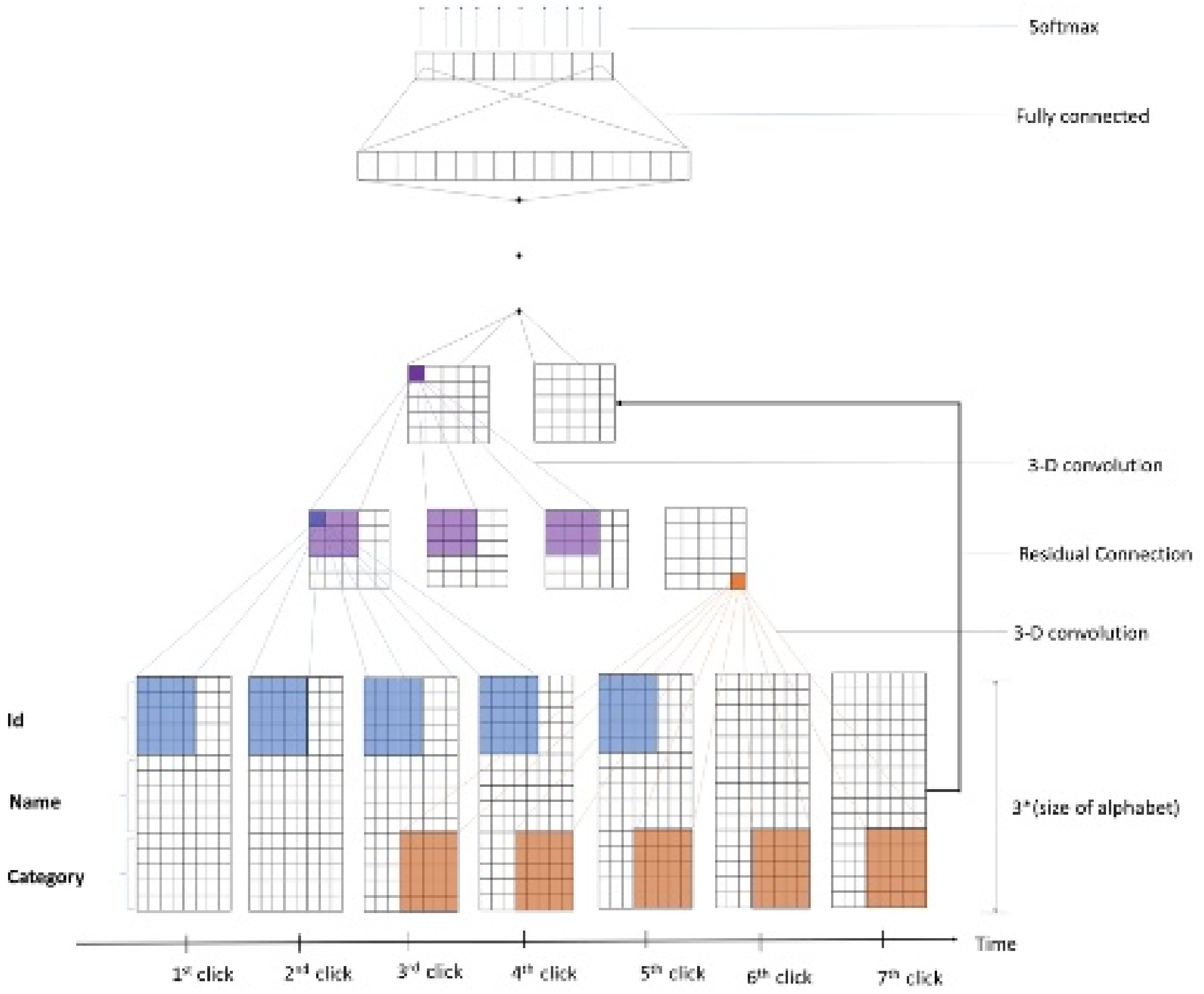


Source: Tuan, Phuong

3D CNNs for Session-based Recommendation

As we've seen in text / NLP, CNNs can also be effective in modeling sequences

- 3D convolutional models have been applied in video classification
- Potentially faster to train, easier to understand
- Use character-level encoding of IDs and item features (name, description, categories)
  - Compact representation
  - No embedding layer
- "ResNet" style architecture
- Show improvement over p-RNN



Challenges

Challenges particular to recommendation models

- Data size and dimensionality (input & output)
  - Sampling
- Extreme sparsity
  - Embeddings & compressed representations
- Wide variety of specialized settings
- Combining session, content, context and preference data
- Model serving is difficult – ranking, large number of items, computationally expensive
- Metrics – model accuracy and its relation to real-world outcomes and behaviors
- Need for standard, open, large-scale, datasets that have time and session data and are content- and context-rich
  - RecSys 15 Challenge – YouChoose dataset
- Evaluation – watch you baselines!
  - When Recurrent Neural Networks meet the Neighborhood for Session-Based Recommendation

Future Directions

Most recent and future directions in research & industry

- Improved RNNs
  - Cross-session models (e.g. Hierarchical RNN)
  - Further research on contextual models, as well as content and metadata
  - Attention models:
    - Attentive Neural Architecture for Music Recommendation
    - Neural Attentive Session-based Recommendation
- Combine sequence and historical models (long- and short-term user profiles)
- Domain-specific applications
  - Contextualized Location Sequence Recommender
- RecGAN (yes, GANs **and** RNNS!)
- Applications at scale
  - Dimensionality reduction, compressed encodings

Summary

DL for recommendation is just getting started (again)

- Huge increase in interest, research papers. Already many new models and approaches
- DL approaches have generally yielded incremental % gains
  - But that can translate to significant $$$
  - More pronounced in session-based
- Cold start scenarios benefit from multi-modal nature of DL models and explicit modeling of sequences
- Flexibility of DL frameworks helps a lot
- Benefits from advances in DL for images, video, NLP etc.
- Open-source libraries appearing (e.g. Spotlight)
- Check out DLRS workshops & tutorials @ RecSys 2016 / 2017, and upcoming in Oct, 2018
- RecSys challenges

Thank you!

codait.org

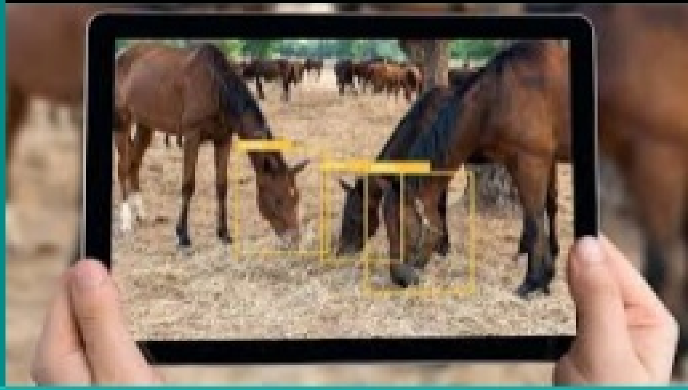
twitter.com/MLnick

github.com/MLnick

developer.ibm.com

FfDL

MAX



Sign up for IBM Cloud and try Watson Studio!

https://ibm.biz/BdYbTY

https://datascience.ibm.com/

Links & References

[Wikipedia: Perceptron]

[Stanford CS231n Convolutional Neural Networks for Visual Recognition]

[Stanford CS231n – RNN Slides]

[Recurrent Neural Networks Tutorial]

[The Unreasonable Effectiveness of Recurrent Neural Networks]

[Understanding LSTM Networks]

[Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation]

[Long short-term memory]

[Attention and Augmented Recurrent Neural Networks]

Links & References

Deep Content-based Music Recommendation

Google's Wide and Deep Learning Model

Deep Learning for Recommender Systems Workshops @ RecSys

Deep Learning for Recommender Systems Tutorial @ RecSys 2017

Session-based Recommendations with Recurrent Neural Networks

Recurrent Neural Networks with Top-k Gains for Session-based Recommendations

Sequential User-based Recurrent Neural Network Recommendations

Links & References

[Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks]

[Parallel Recurrent Neural Network Architectures for Feature-rich Session-based Recommendations]

[Contextual Sequence Modeling for Recommendation with Recurrent Neural Networks]

[When Recurrent Neural Networks meet the Neighborhood for Session-Based Recommendation]

[3D Convolutional Networks for Session-based Recommendation with Content Features]

[Spotlight: Recommendation models in PyTorch]

[RecSys 2015 Challenge – YouChoose Dataset]