Recommender Systems & Embeddings

Jin Lu – UMDearborn

Outline

- Embeddings

Outline

- Embeddings
- Dropout Regularization

Outline

- Embeddings
- Dropout Regularization
- Recommender Systems

Embeddings

Embeddings

Symbolic variable

- Text: characters, words, bigrams...
- Recommender Systems: item ids, user ids
- Any categorical descriptor: tags, movie genres, visited URLs, skills on a resume, product categories...

Embeddings

Symbolic variable

- Text: characters, words, bigrams...
- Recommender Systems: item ids, user ids
- Any categorical descriptor: tags, movie genres, visited URLs, skills on a resume, product categories...

Embeddings

Symbolic variable

- Text: characters, words, bigrams...
- Recommender Systems: item ids, user ids
- Any categorical descriptor: tags, movie genres, visited URLs, skills on a resume, product categories...

Embeddings

Symbolic variable
- Text: characters, words, bigrams...
- Recommender Systems: item ids, user ids
- Any categorical descriptor: tags, movie genres, visited URLs, skills on a resume, product categories...

Notation:

Symbol s in vocabulary V

One-hot representation

$$onehot(\text{'salad'}) = [0,0,1,\ldots,0] \in \{0,1\}^{|V|}$$

One-hot representation

$$onehot(\text{'salad'}) = [0, 0, 1, ..., 0] \in \{0, 1\}^{|V|}$$

- Sparse, discrete, large dimension $|V|$
- Each axis has a meaning
- Symbols are equidistant from each other:

euclidean distance = $\sqrt{2}$

Embedding

$embedding(\text{'salad'}) = [3.28, -0.45, \ldots, 7.11] \in R^d$

Embedding

$$embedding(\text{'salad'}) = [3.28, -0.45, \ldots, 7.11] \in R^d$$

- Continuous and dense
- Can represent a huge vocabulary in low dimension, typically: $d \in \{16, 32, \ldots, 4096\}$
- Embedding metric can capture semantic distance
- Axis have no meaning *a priori*

Embedding

$$embedding(\text{'salad'}) = [3.28, -0.45, \ldots, 7.11] \in R^d$$

•Continuous and dense

•Can represent a huge vocabulary in low dimension, typically: $d \in \{16, 32, \ldots, 4096\}$

•Embedding metric can capture semantic distance

•Axis have no meaning *a priori*

Neural Networks compute transformations on continuous vectors

Implementation with Keras

Size of vocabulary $n = |V|$, size of embedding d

```
# input: batch of integers
Embedding(output_dim=d, input_dim=n, input_length=1)
# output: batch of float vectors
```

Implementation with Keras

Size of vocabulary $n = |V|$, size of embedding d

```
# input: batch of integers
Embedding(output_dim=d, input_dim=n, input_length=1)
# output: batch of float vectors
```

- Equivalent to one-hot encoding multiplied by a weight matrix $\mathbf{W} \in R^{n \times d}$:

$$embedding(x) = onehot(x).\mathbf{W}$$

Implementation with Keras

Size of vocabulary $n = |V|$, size of embedding d

```
# input: batch of integers
Embedding(output_dim=d, input_dim=n, input_length=1)
# output: batch of float vectors
```

- Equivalent to one-hot encoding multiplied by a weight matrix $\mathbf{W} \in R^{n \times d}$:

$$embedding(x) = onehot(x).\mathbf{W}$$

- **W** is typically **randomly initialized**, then **tuned by backprop**

Implementation with Keras

Size of vocabulary $n = |V|$, size of embedding d

```
# input: batch of integers
Embedding(output_dim=d, input_dim=n, input_length=1)
# output: batch of float vectors
```

- Equivalent to one-hot encoding multiplied by a weight matrix $\mathbf{W} \in R^{n \times d}$:

$$embedding(x) = onehot(x).\mathbf{W}$$

- $\mathbf{W}$ is typically **randomly initialized**, then **tuned by backprop**

- $\mathbf{W}$ are trainable parameters of the model.

Distance and similarity in Embedding space

Euclidean distance

$$d(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|_2$$

- Simple with good properties
- Dependent on norm (embeddings usually unconstrained)

Distance and similarity in Embedding space

Euclidean distance

$$\boldsymbol{d}(\boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|_2$$

- Simple with good properties
- Dependent on norm (embeddings usually unconstrained)

Cosine similarity

$$\boldsymbol{cosine}(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{\|\boldsymbol{x}\| \cdot \|\boldsymbol{y}\|}$$

- Angle between points, regardless of norm
- $\boldsymbol{cosine}(\boldsymbol{x}, \boldsymbol{y}) \in [-\mathbf{1}, \mathbf{1}]$
- Expected cosine similarity of random pairs of vectors is 0

Distance and similarity in Embedding space

If x and y both have unit norms:

$$\|\boldsymbol{x}-\boldsymbol{y}\|_{\boldsymbol{2}}^{\boldsymbol{2}}=\boldsymbol{2}(\boldsymbol{1}-\boldsymbol{cosine}(\boldsymbol{x},\boldsymbol{y}))$$

Distance and similarity in Embedding space

If x and y both have unit norms:

$$\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 = \mathbf{2}(\mathbf{1} - \boldsymbol{cosine}(\boldsymbol{x},\boldsymbol{y}))$$

or alternatively:

$$\boldsymbol{cos}(\boldsymbol{x},\boldsymbol{y}) = \mathbf{1} - \frac{\|\boldsymbol{x}-\boldsymbol{y}\|_2^2}{\mathbf{2}}$$

Distance and similarity in Embedding space

If x and y both have unit norms:

$$\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 = \mathbf{2}(\mathbf{1}-\boldsymbol{cosine}(\boldsymbol{x},\boldsymbol{y}))$$

or alternatively:

$$\boldsymbol{cos}(\boldsymbol{x},\boldsymbol{y}) = \mathbf{1}-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|_2^2}{\mathbf{2}}$$

Alternatively, dot product (unnormalized) is used in practice as a pseudo similarity

Visualizing Embeddings

- Visualizing requires a projection in 2 or 3 dimensions
- Objective: visualize which embedded symbols are similar

Visualizing Embeddings

- Visualizing requires a projection in 2 or 3 dimensions
- Objective: visualize which embedded symbols are similar

PCA

- Limited by linear projection, embeddings usually have complex high dimensional structure

Visualizing Embeddings

- Visualizing requires a projection in 2 or 3 dimensions
- Objective: visualize which embedded symbols are similar

PCA

•Limited by linear projection, embeddings usually have complex high dimensional structure

t-SNE

Visualizing data using t-SNE, L van der Maaten, G Hinton, *The Journal of Machine Learning Research*, 2008

t-Distributed Stochastic Neighbor Embedding

- Unsupervised, low-dimension, non-linear projection
- Optimized to preserve relative distances between nearest neighbors
- Global layout is not necessarily meaningful

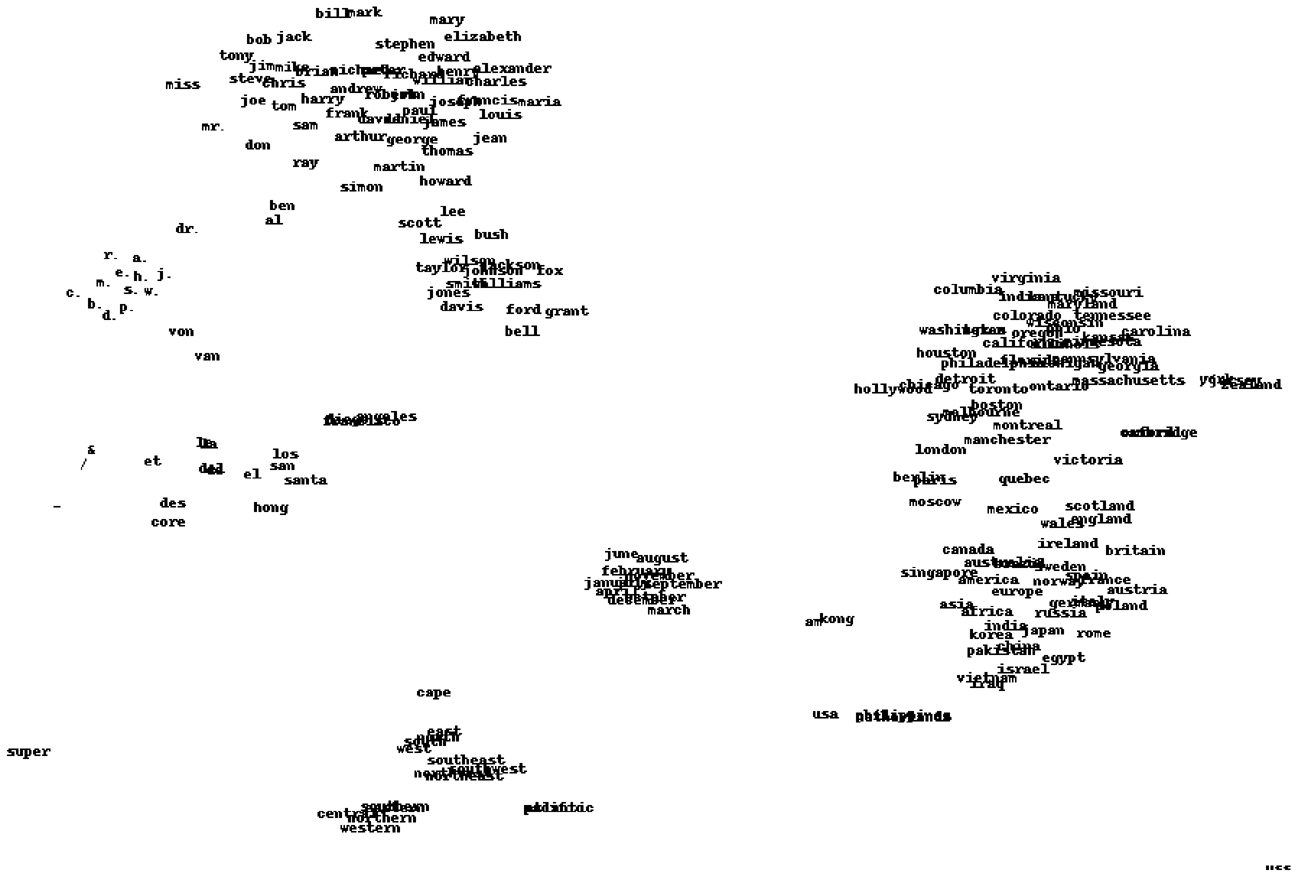
t-Distributed Stochastic Neighbor Embedding

- Unsupervised, low-dimension, non-linear projection
- Optimized to preserve relative distances between nearest neighbors
- Global layout is not necessarily meaningful

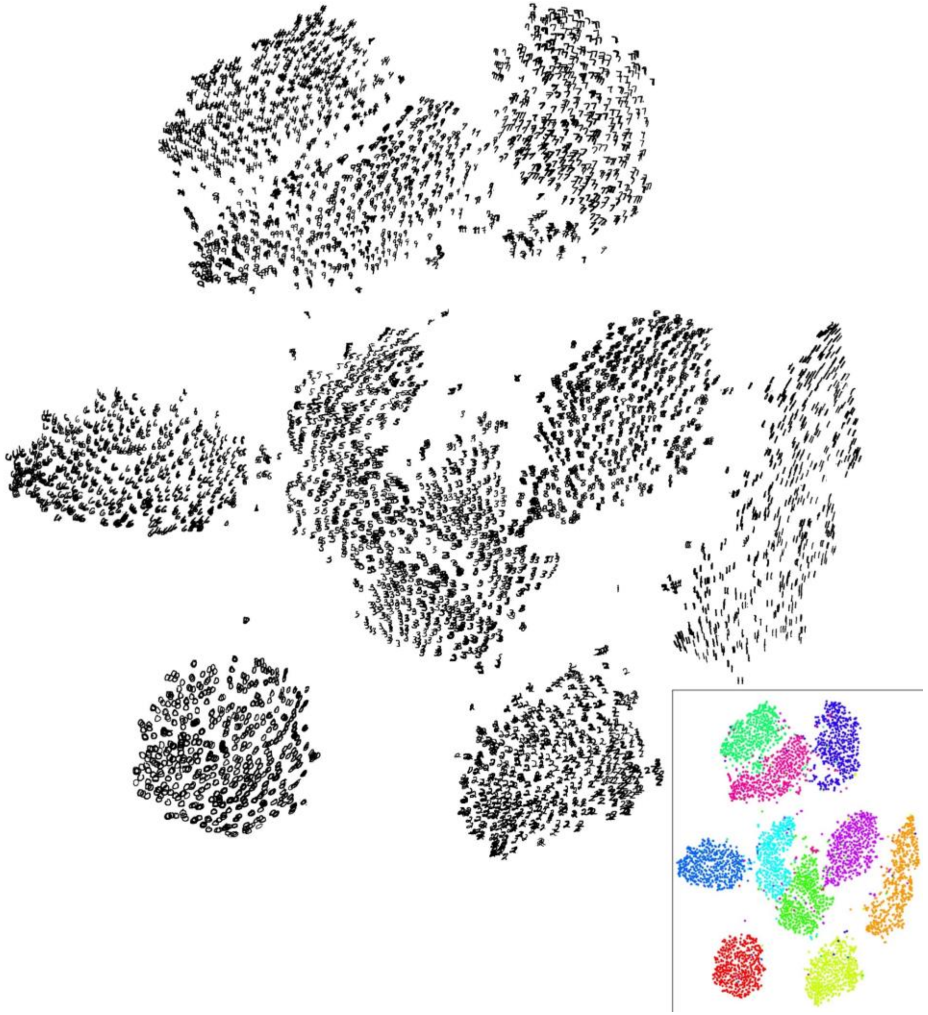
t-SNE projection is non deterministic (depends on initialization)

- Critical parameter: perplexity, usually set to 20, 30
- See http://distill.pub/2016/misread-tsne/

Example word vectors



Visualizing Mnist



Dropout Regularization

Regularization

- Size of the embeddings

Regularization

- Size of the embeddings
- Depth of the network

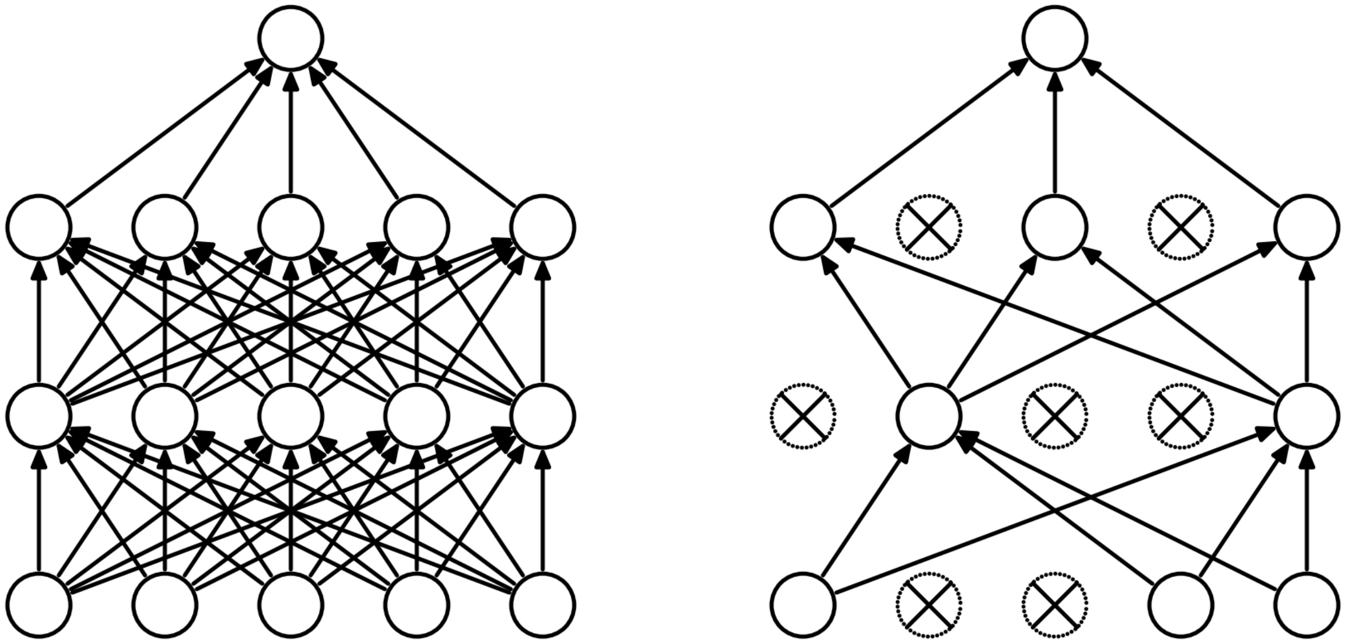
Regularization

- Size of the embeddings
- Depth of the network
- L_2 penalty on embeddings

Regularization

- Size of the embeddings
- Depth of the network
- L_2 penalty on embeddings
- Dropout
    - Randomly set activations to 0 with probability p
    - Bernoulli mask sampled for a forward pass / backward pass pair
    - Typically only enabled at training time

Dropout



(a) Standard Neural Net

(b) After applying dropout.

Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Srivastava et al., *Journal of Machine Learning Research* 2014

Dropout

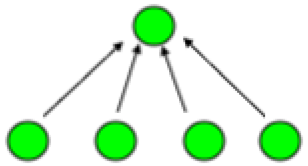
Interpretation

- Reduces the network dependency to individual neurons
- More redundant representation of data

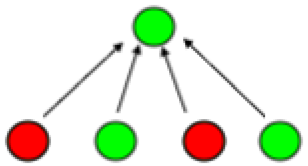
Ensemble interpretation

- Equivalent to training a large ensemble of shared-parameters, binary-masked models
- Each model is only trained on a single data point

Dropout



(Image a)



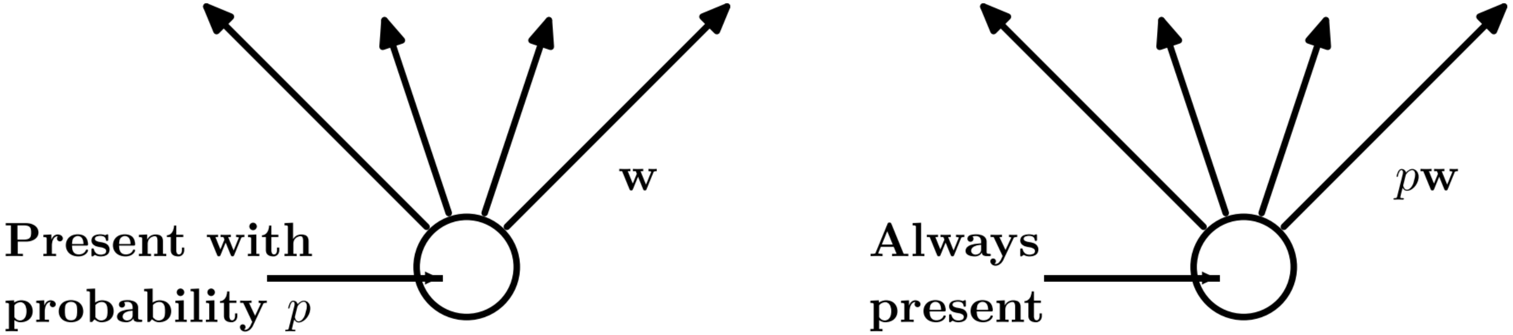
(Image b)

The weight on each unit will initially be ¼ = 0.25.

If we apply dropout with $p = 0.5$ to this layer, it could end up looking like image b. Since only two units are considered, they will each have an initial weight of ½ = 0.5. But we don't want these weights to be fixed at this high number during testing.

Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Srivastava et al., *Journal of Machine Learning Research* 2014

Dropout

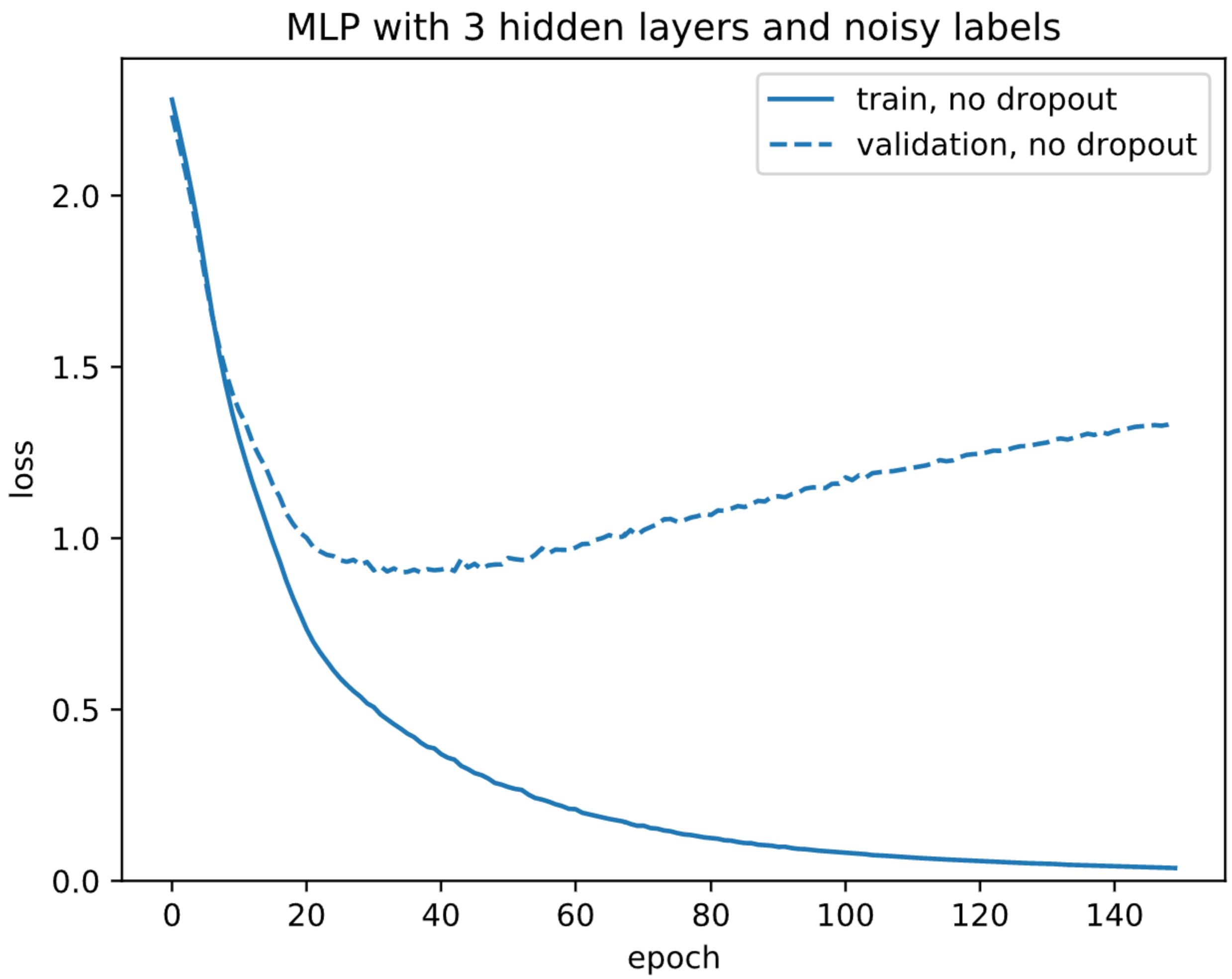


(a) At training time (b) At test time

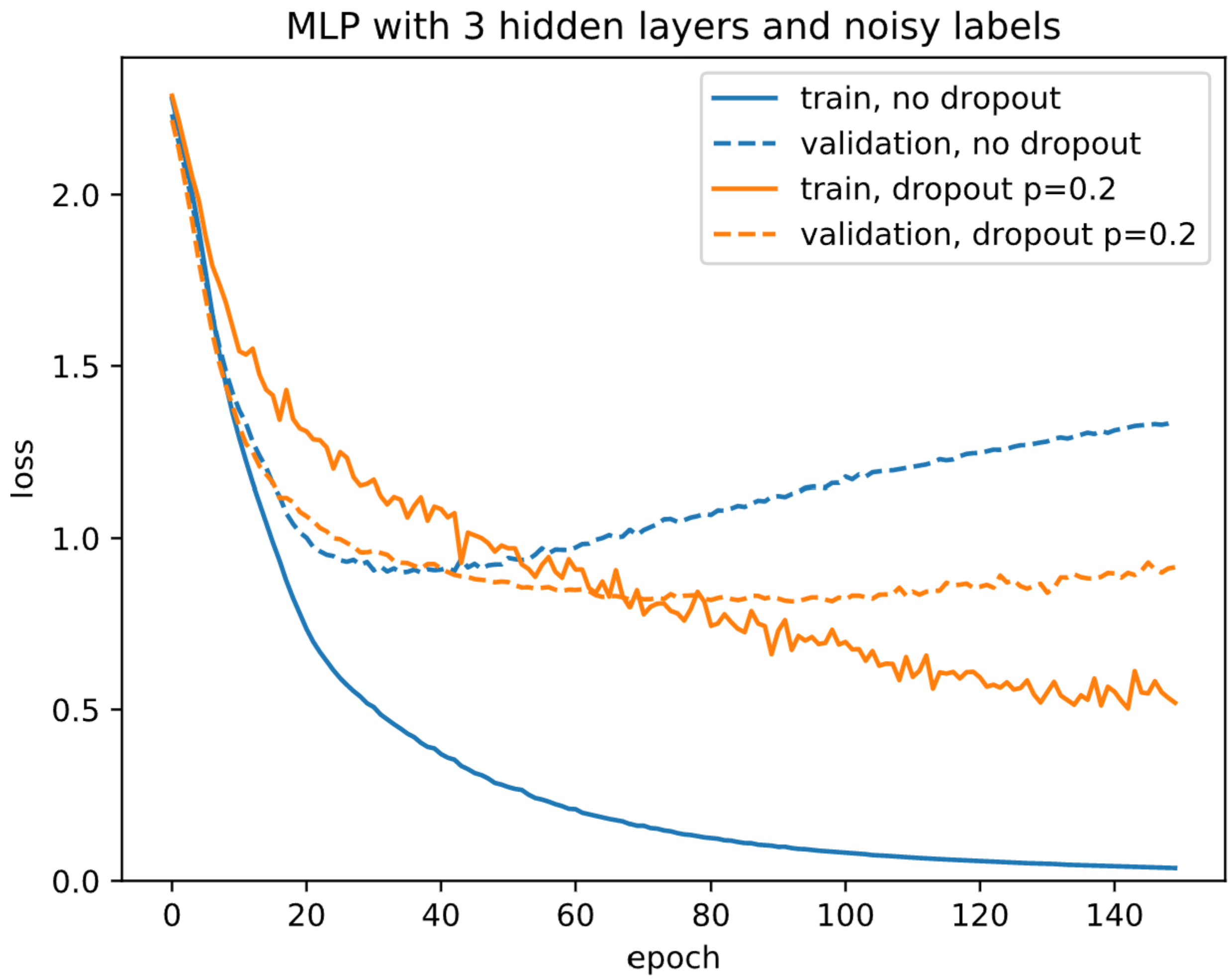
At test time, multiply weights by p to keep same level of activation

Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Srivastava et al., *Journal of Machine Learning Research* 2014

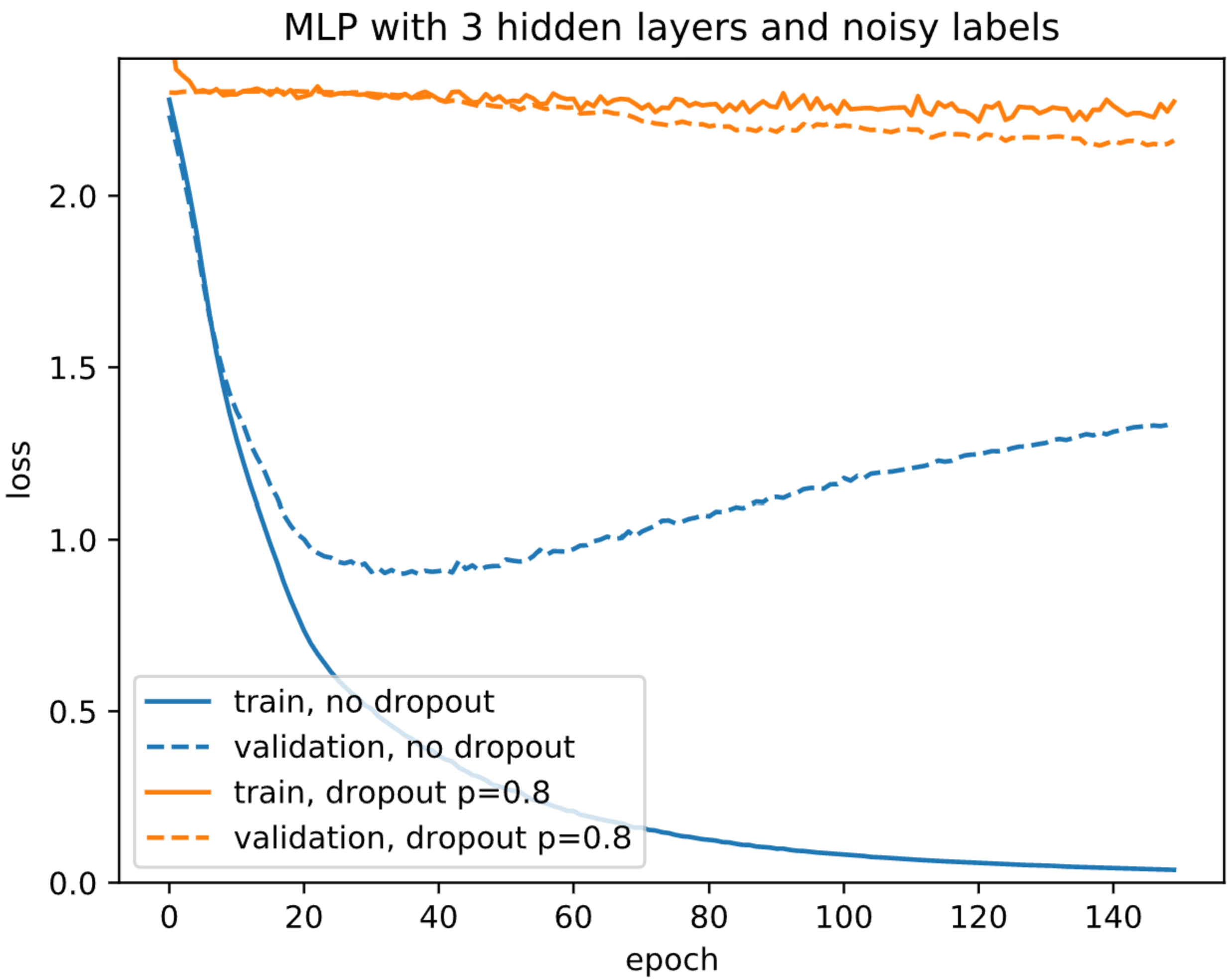
Overfitting Noise



Overfitting Noise



Overfitting Noise



Implementation with Keras

```
model = Sequential()
model.add(Dense(hidden_size, input_shape, 
activation='relu'))
model.add(Dropout(p=0.5))
model.add(Dense(hidden_size, activation='relu'))
model.add(Dropout(p=0.5))
model.add(Dense(output_size, activation='softmax'))
```

Recommendation Systems

Recommendation Systems

Recommend contents and products

Movies on Netflix and YouTube, weekly playlist and related Artists on Spotify, books on Amazon, related apps on app stores, "Who to Follow" on twitter...

Recommendation Systems

Recommend contents and products

Movies on Netflix and YouTube, weekly playlist and related Artists on Spotify, books on Amazon, related apps on app stores, "Who to Follow" on twitter...

Prioritized social media status updates

Recommendation Systems

Recommend contents and products

Movies on Netflix and YouTube, weekly playlist and related Artists on Spotify, books on Amazon, related apps on app stores, "Who to Follow" on twitter...

Prioritized social media status updates

Personalized search engine results

Recommendation Systems

Recommend contents and products

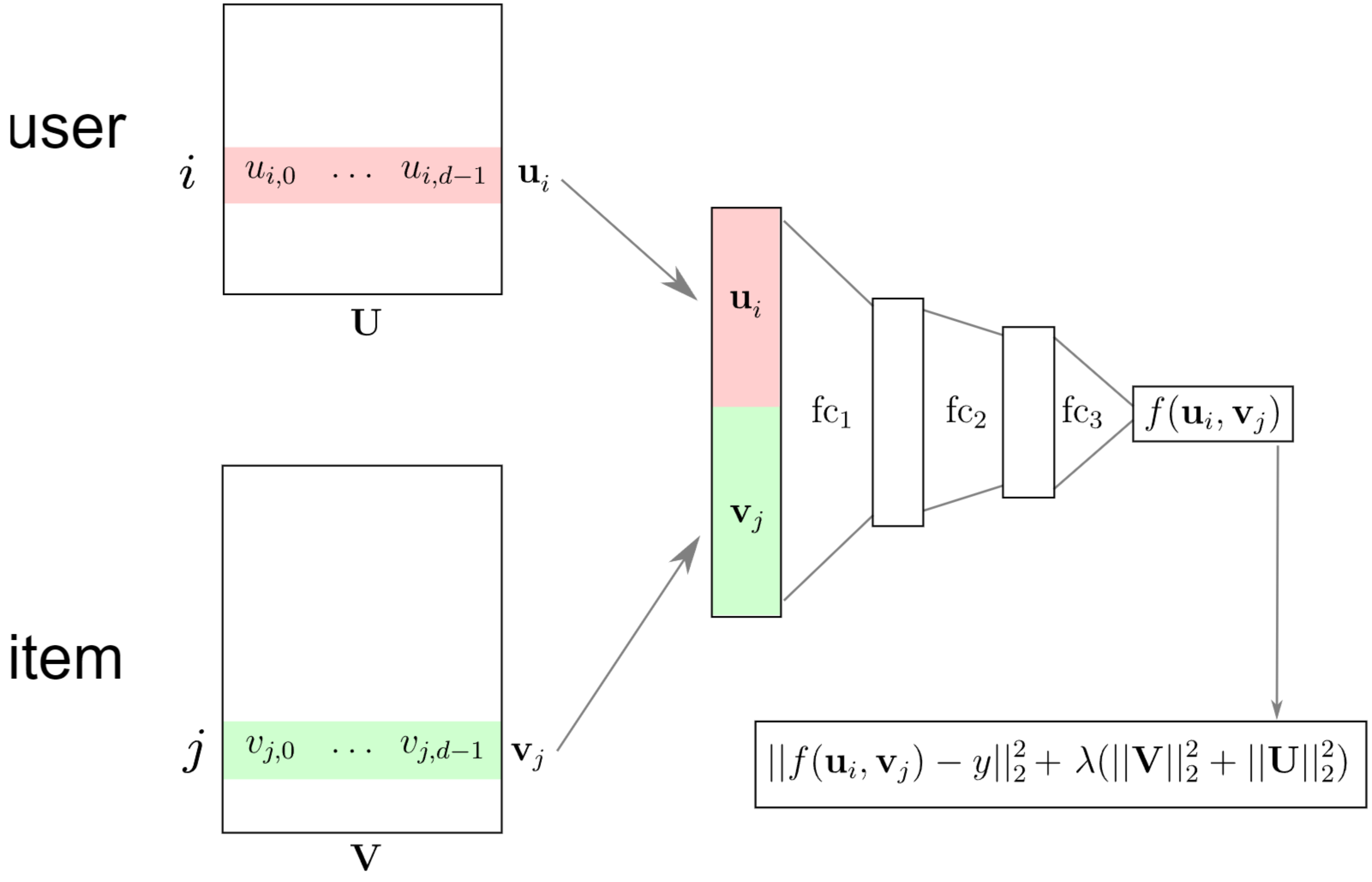
Movies on Netflix and YouTube, weekly playlist and related Artists on Spotify, books on Amazon, related apps on app stores, "Who to Follow" on twitter...

Prioritized social media status updates

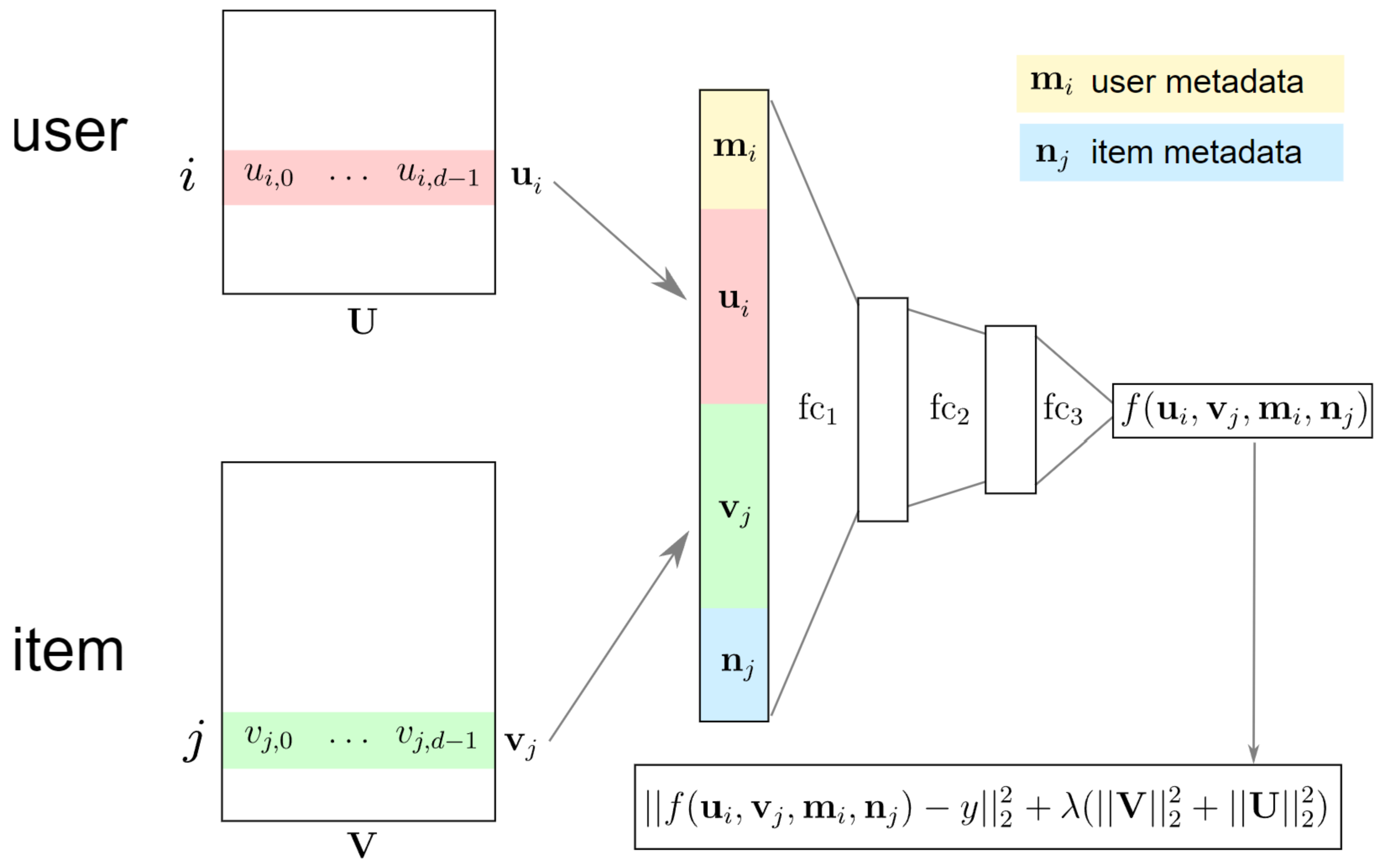
Personalized search engine results

Personalized ads

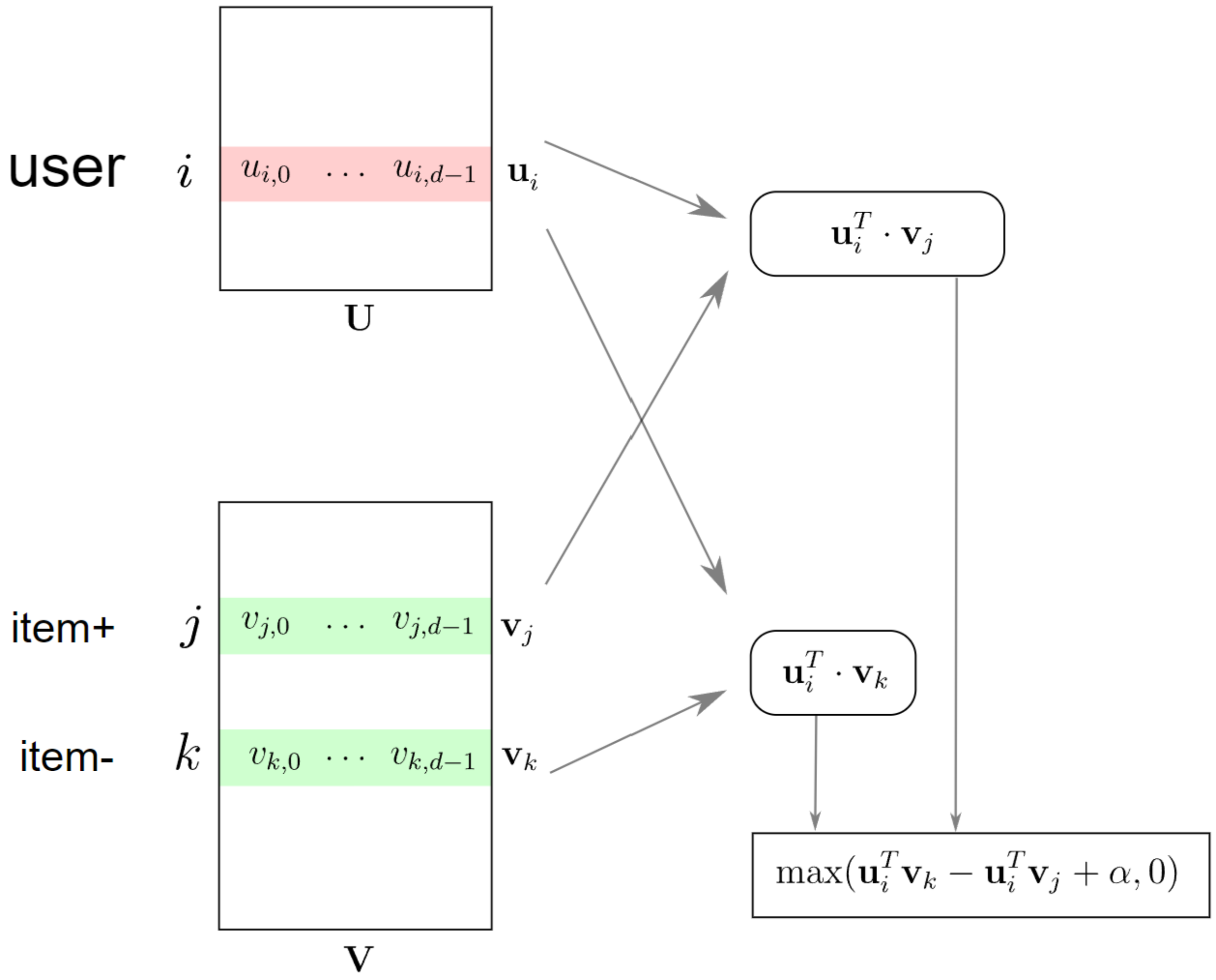
Deep RecSys Architecture



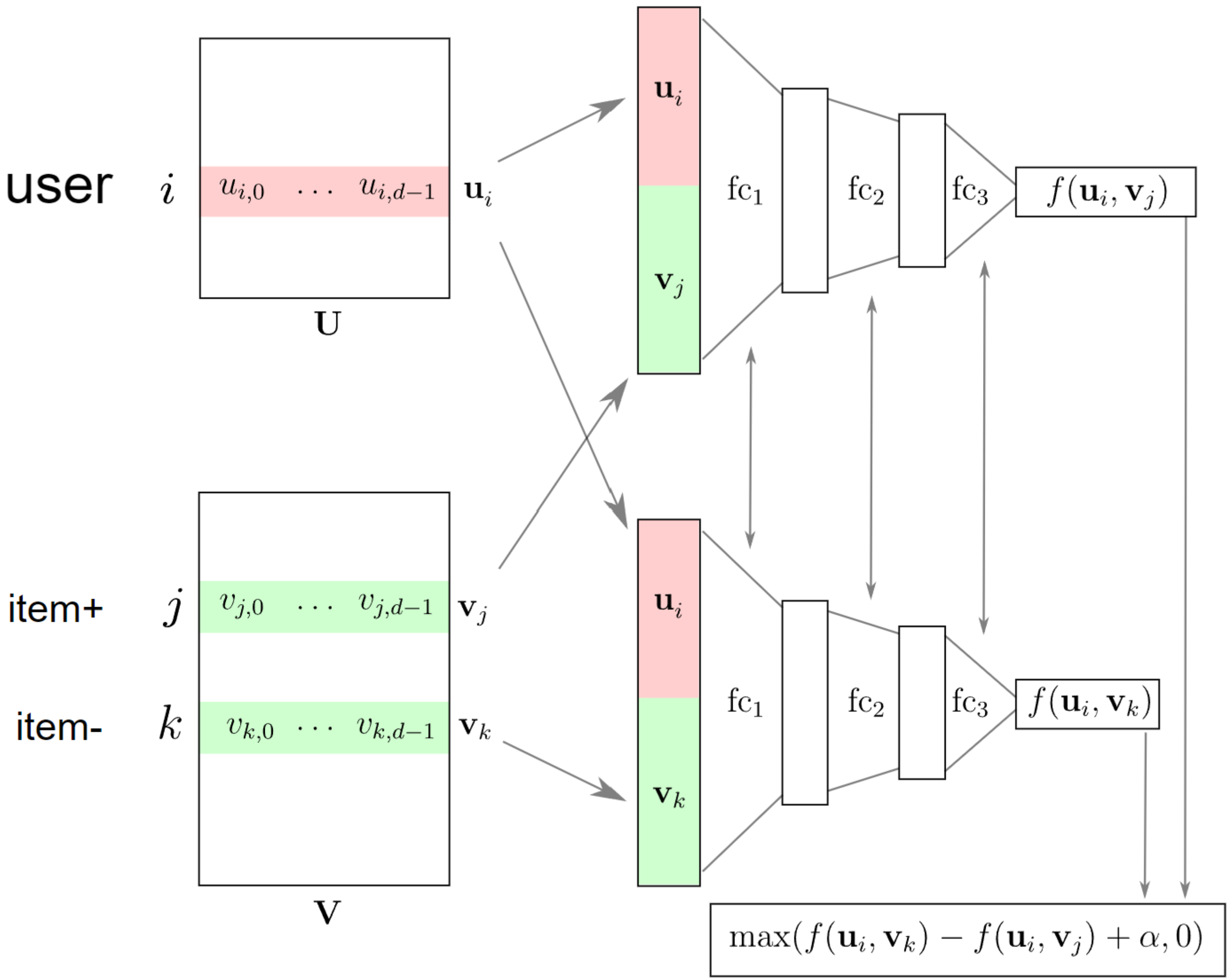
Deep RecSys with metadata



Implicit Feedback: Triplet loss



Deep Triplet Networks



Training a Triplet Model

- Gather a set of positive pairs user i and item j

- While model has not converged:

Training a Triplet Model

- Gather a set of positive pairs user i and item j
- While model has not converged:
  - Shuffle the set of pairs (i, j)

Training a Triplet Model

- Gather a set of positive pairs user i and item j

- While model has not converged:

  - Shuffle the set of pairs (i, j)
  - For each (I,j):
    - Sample item k uniformly at random

Training a Triplet Model

- Gather a set of positive pairs user i and item j
- While model has not converged:
  - Shuffle the set of pairs (i, j)
  - For each (I,j):
    - Sample item k uniformly at random
    - Call item k a negative item for user I

Training a Triplet Model

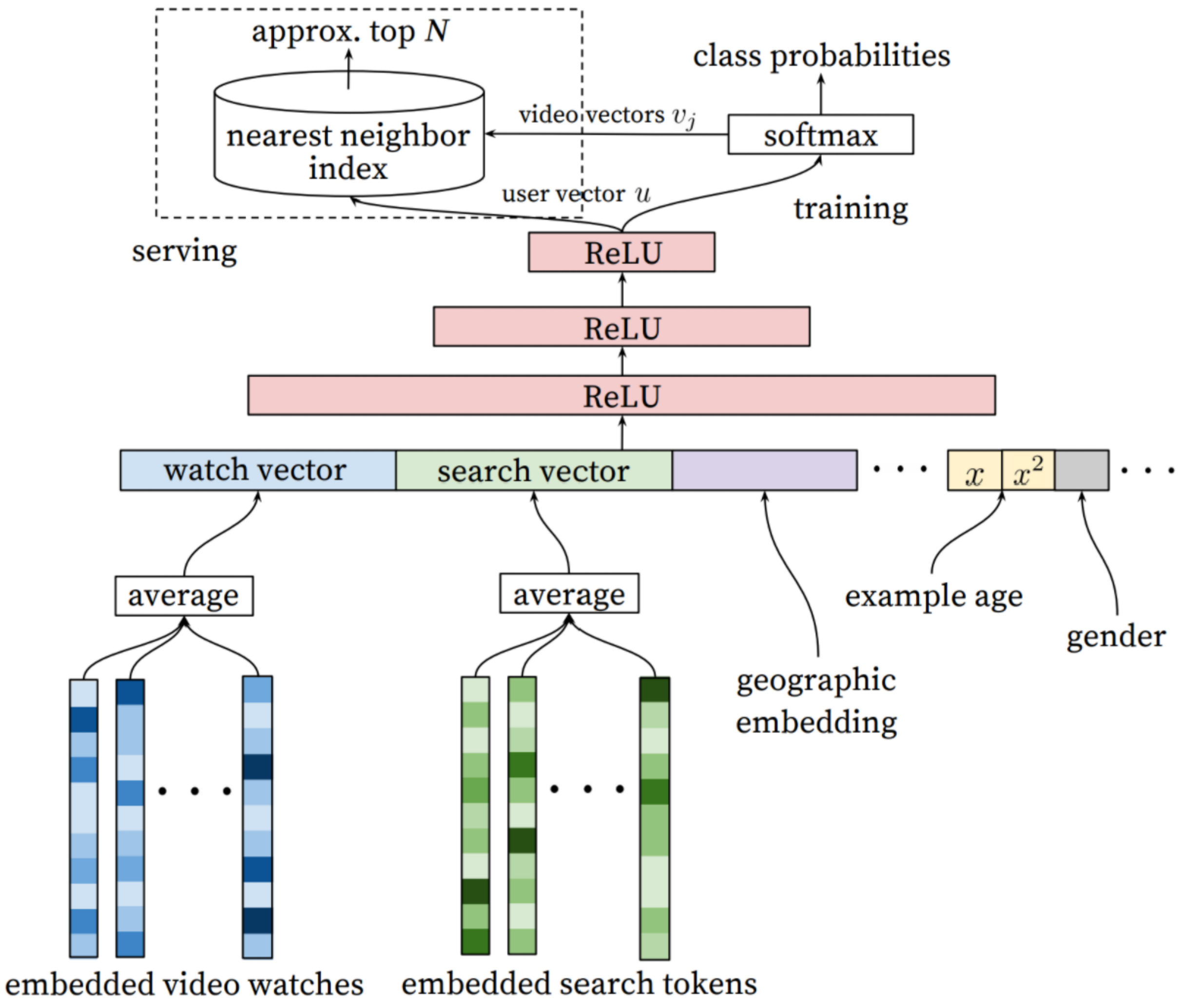
- Gather a set of positive pairs user i and item j

- While model has not converged:

  - Shuffle the set of pairs (i, j)
  - For each (I,j):
    - Sample item k uniformly at random
    - Call item k a negative item for user I
    - Train model on triplet (I, j, k)

Training a Triplet Model

- Gather a set of positive pairs user i and item j
- While model has not converged:
  - Shuffle the set of pairs (i, j)
  - For each (I,j):
    - Sample item k uniformly at random
    - Call item k a negative item for user I
    - Train model on triplet (I, j, k)



Deep Neural Networks for YouTube Recommendations https://research.google.com/pubs/pub45530.html

Ethical Considerations of Recommender Systems

Ethical Considerations of Recommender Systems

Amplification of existing discriminatory and unfair behaviors /

- Example: gender bias in ad clicks (fashion / jol
- Using the firstname as a predictive feature

Facebook Under Fire For Alleged Gender Discrimination In Job Advertisements

- 96% of the people shown the ad for mechanic jobs were men.
- 95% of those shown the ad for preschool nurse jobs were women.
- 75% of those shown the ad for pilot jobs were men.
- 77% of those shown the ad for psychologist jobs were women.

Amplification of the filter bubble and opinion polarization

- Personalization can amplify "people only follow people they agree with"
- Optimizing for "engagement" promotes content that cause strong emotional reaction (and turns normal users into *haters*?)
- RecSys can exploit weaknesses of some users, lead to addiction
- Addicted users clicks over-represented in future training data

Call to action

Designing Ethical Recommender Systems

- Wise modeling choices (e.g. use of "firstname" as feature)
- Conduct internal audits to detect fairness issues: SHAP, Integrated Gradients
- Learning representations that enforce fairness?

Call to action

Designing Ethical Recommender Systems

- Wise modeling choices (e.g. use of "firstname" as feature)
- Conduct internal audits to detect fairness issues: SHAP, Integrated Gradients
- Learning representations that enforce fairness?

Transparency

- Educate decision makers and the general public
- How to allow users to assess fairness by themselves?
- How to allow for independent audits while respecting the privacy of users?

Call to action

Designing Ethical Recommender Systems

- Wise modeling choices (e.g. use of "firstname" as feature)
- Conduct internal audits to detect fairness issues: SHAP, Integrated Gradients
- Learning representations that enforce fairness?

Transparency

- Educate decision makers and the general public
- How to allow users to assess fairness by themselves?
- How to allow for independent audits while respecting the privacy of users?

Active Area of Research