Training deep Autoencoders for collaborative filtering

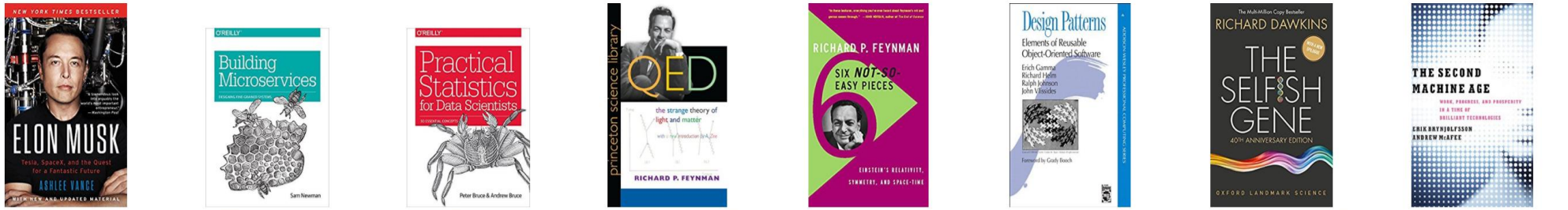
Oleksii Kuchaiev & Boris Ginsburg

NVIDIA

Motivation

Personalized recommendations

amazon for You



Elon Musk: Tesla, SpaceX, and the... 2,067 Kindle Price: $11.99

Building Microservices: Designing... 134 Kindle Price: $24.99

Practical Statistics for Data... 13 Kindle Price: $18.35

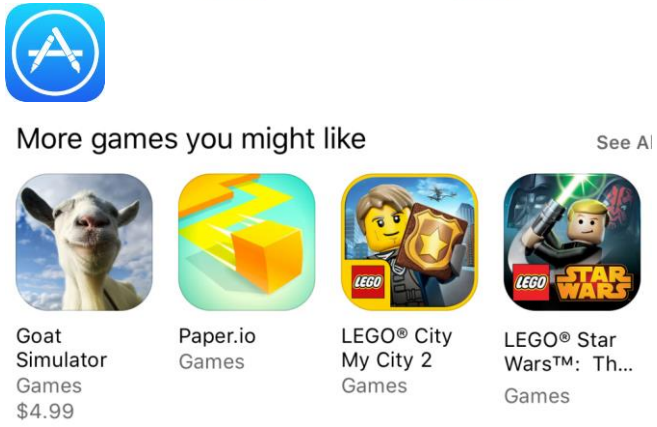
QED: The Strange Theory of Light... 211 Kindle Price: $9.48

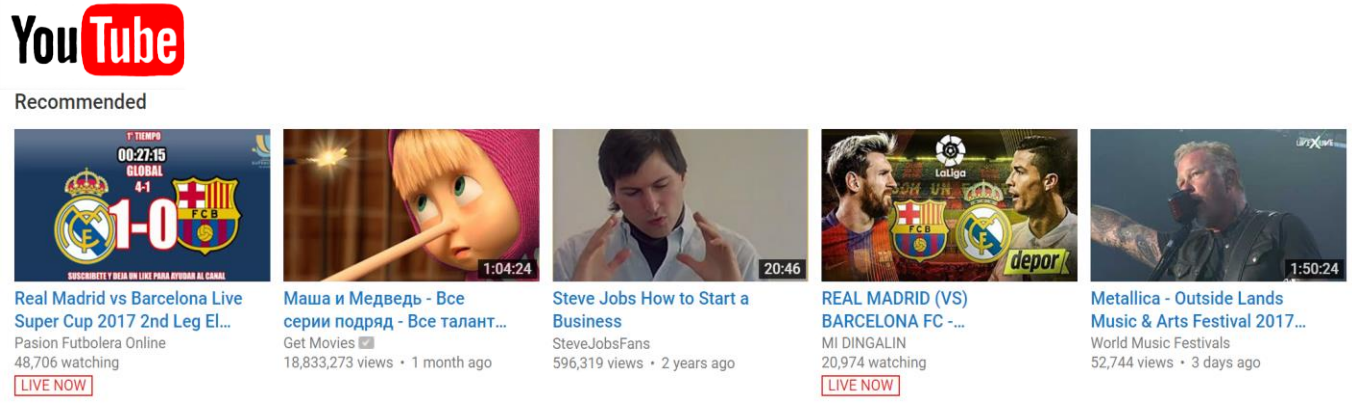
Six Not-So-Easy Pieces: Einstein's... 69 Kindle Price: $9.99

Design Patterns: Elements of... 481 Kindle Price: $35.61

The Selfish Gene: 40th Anniversary... 56 Kindle Price: $8.13

The Second Machine Age: Work... 573 Kindle Price: $8.98





Key points (spoiler alert)

1. Deep autoencoder for collaborative filtering
    1. Improves generalization
2. Right activation function (SELU, ELU, LeakyRELU) enables deep architectures
    1. No layer-wise pre-training, or skip connections
3. Heavy use of dropout
4. *Dense re-feeding* for faster and better training
5. Beats other models on time-split Netflix data (RMSE of 0.9099 vs 0.9224)
6. (PyTorch-based) https://github.com/NVIDIA/DeepRecommender

Oleksii Kuchaiev and Boris Ginsburg "Training Deep AutoEncoders for Collaborative Filtering", *arXiv preprint arXiv:1708.01715* (2017).

Autoencoders & collaborative filtering

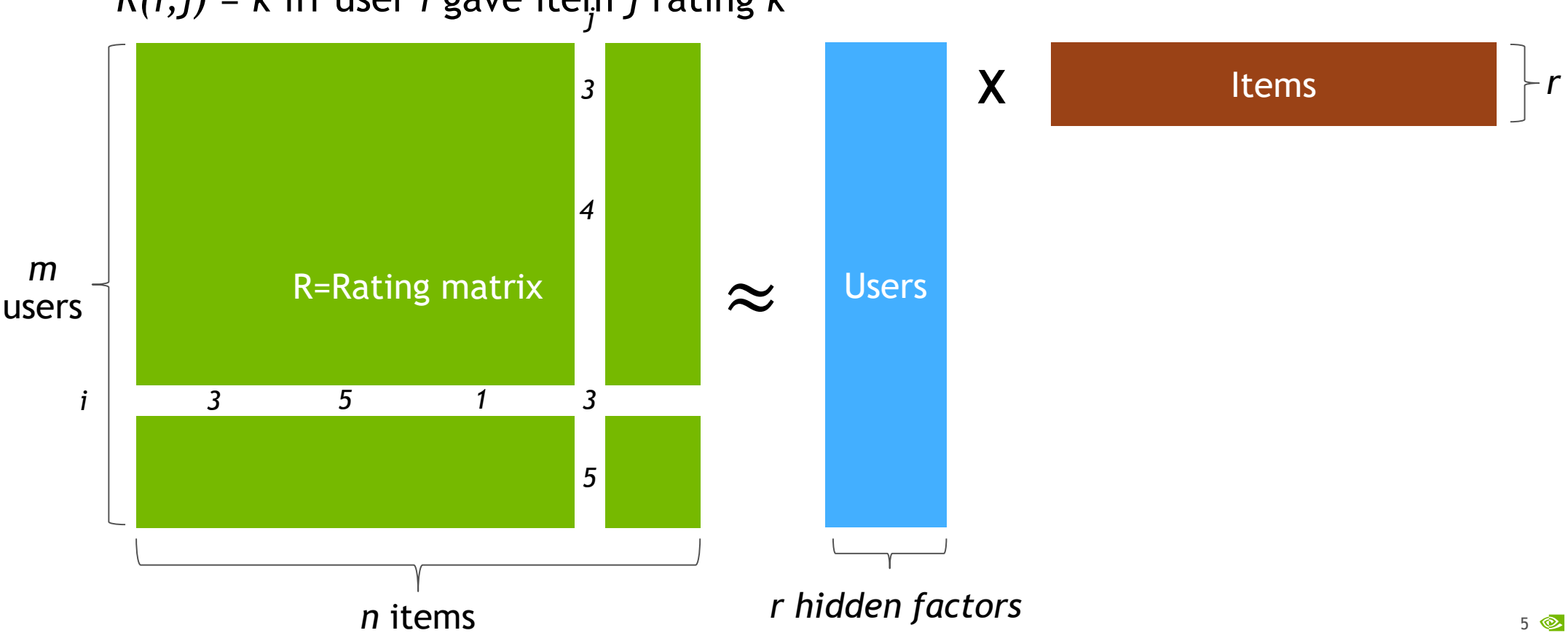
Effects of the activation types
Overfitting the data
Going deeper
Dropout
Dense re-feeding
Conclusions

Oleksii Kuchaiev and Boris Ginsburg "Training Deep AutoEncoders for Collaborative Filtering“, *arXiv preprint arXiv:1708.01715* (2017).

Collaborative filtering

Rating prediction

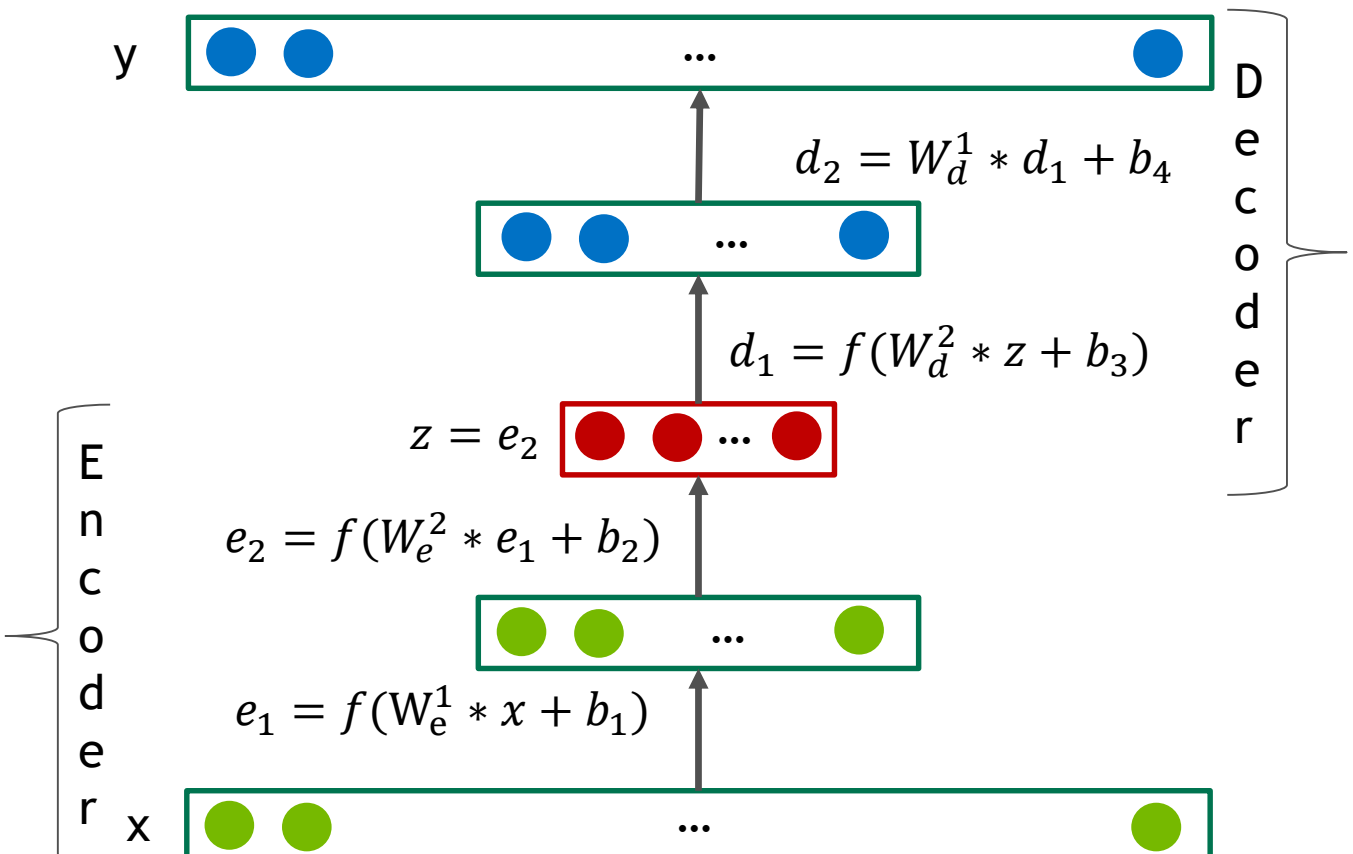
$R(i,j) = k$ iff user i gave item j rating k



Some of the most popular approaches – **Alternative Least Squares (ALS)**

Autoencoder

Deep learning tool of choice for dimensionality reduction



z = encoder(x), encoding
y = decoder(z), reconstruction of x
y = decoder(encoder(x))

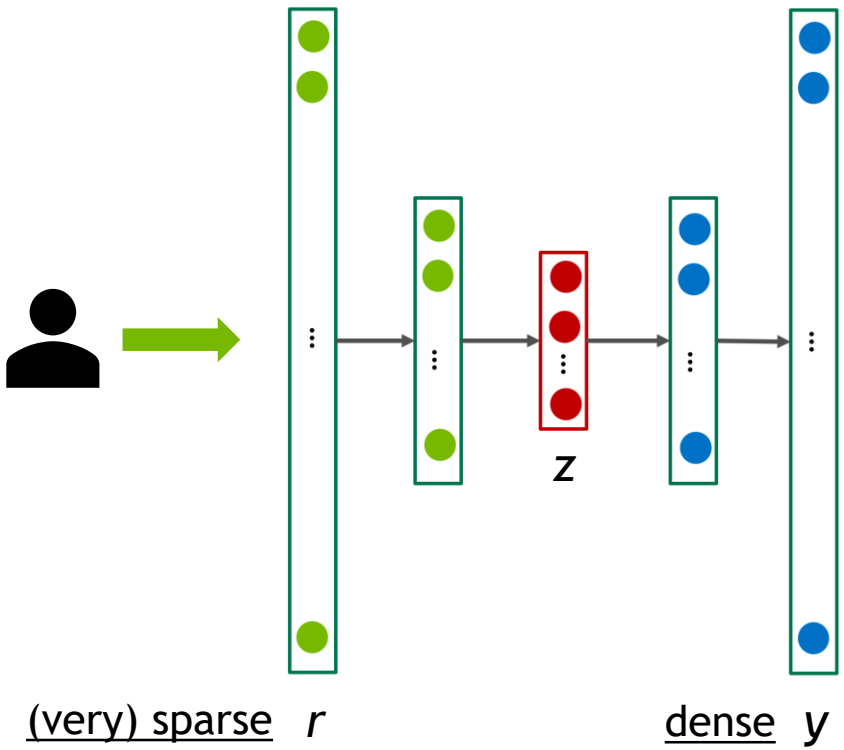
Autoencoder can be thought of as generalization of PCA

"Constrained" if decoder weights are transpose of encoder
"De-noising" if noise is a added to *x*.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), 504-507.

AutoEncoders for recommendations

User (item) based



Masked Mean Squared Error

$$MMSE = \frac{m_i * (r_i - y_i)^2}{\sum_{i=0}^{i=n} m_i}$$

$$m_i = 1 \text{ if } r_i \neq 0 \text{ else } m_i = 0.$$

Sedhain, Suvash, et al. "Autorec: Autoencoders meet collaborative filtering." Proceedings of the 24th International Conference on World Wide Web. ACM, 2015.

Dataset

Netflix prize training data set

Time split to predict *future* ratings

	Full	3 months	6 months	1 year
Training	12/99-11/05	09/05-11/05	06/05-11/05	06/04-05/05
Users	477,412	311,315	390,795	345,855
Ratings	98,074,901	13,675,402	29,179,009	41,451,832
Testing	12/05	12/05	12/05	06/05
Users	173,482	160,906	169,541	197,951
Ratings	2,250,481	2,082,559	2,175,535	3,888,684

Wu, Chao-Yuan, et al. "Recurrent recommender networks." Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. ACM, 2017.

Benchmark

Netflix prize training data set

$$RMSE = \sqrt{\frac{\sum_{r_i \neq 0}(r_i - y_i)^2}{\sum_{r_i \neq 0} 1}}$$

	PMF	I-AR	U-AR	T-SVD	**RRN**
IMDb	2.3913	2.0521	2.0290	2.0037	**1.9703**
Netflix 6 months	0.9584	0.9778	0.9836	0.9589	**0.9427**
Netflix full	0.9252	0.9364	0.9647	0.9275	**0.9224**

PMF: Mnih, Andriy, and Ruslan R. Salakhutdinov. "Probabilistic matrix factorization." Advances in neural information processing systems. 2008.

I-AR, U-AR: Sedhain, Suvash, et al. "Autorec: Autoencoders meet collaborative filtering." Proceedings of the 24th International Conference on World Wide Web. ACM, 2015.

RRN: Wu, Chao-Yuan, et al. "Recurrent recommender networks." Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. ACM, 2017.

Autoencoders & collaborative filtering

Effects of the activation types

Overfitting the data

Going deeper

Dropout

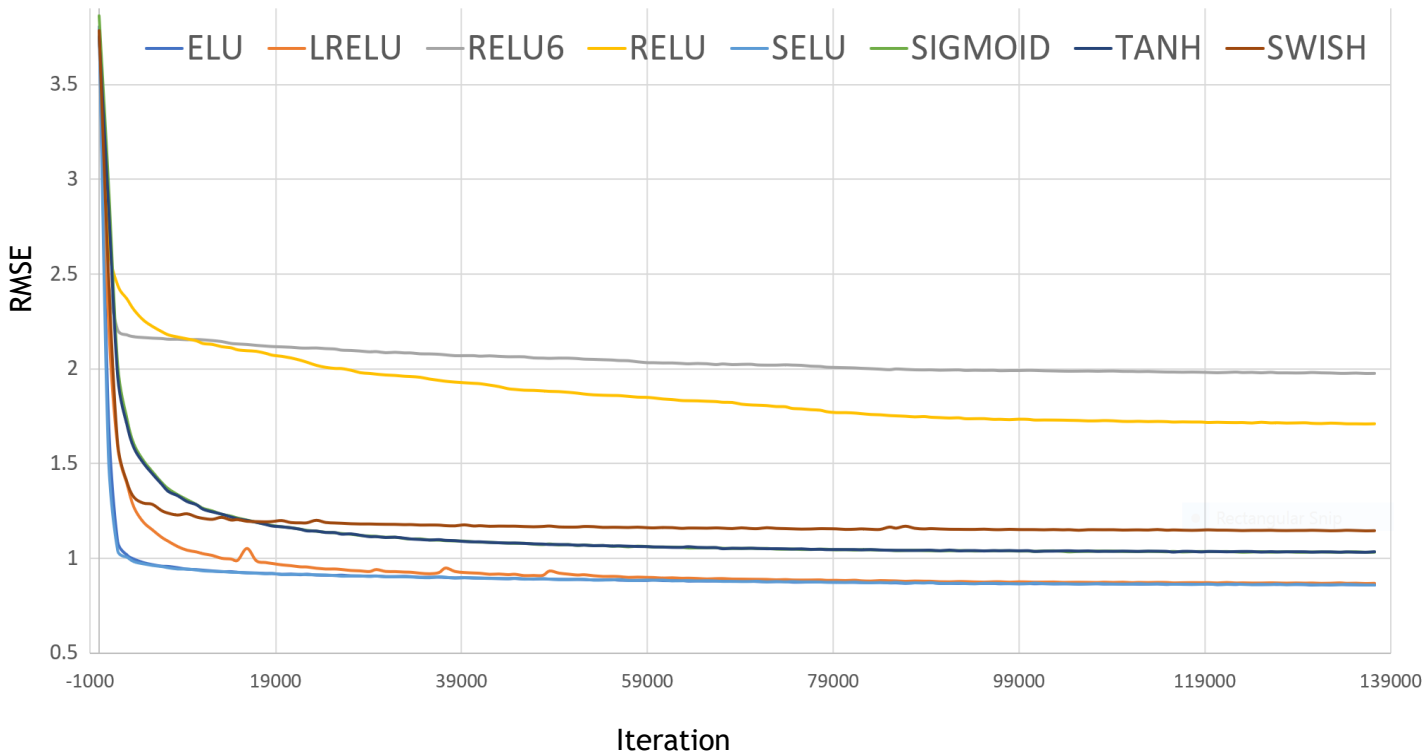
Dense re-feeding

Conclusions

Oleksii Kuchaiev and Boris Ginsburg "Training Deep AutoEncoders for Collaborative Filtering“, *arXiv preprint arXiv:1708.01715* (2017).

Activation function matters

- We found that on this task ELU, SELU and LRELU perform much better than SIGMOID, RELU, RELU6, TANH and SWISH



Apparently important:

a) non-zero negative part

b) Unbounded positive part

Training RMSE per mini-batch. All lines correspond to 4-layers autoencoder (2 layer encoder and 2 layer decoder) with hidden unit dimensions of 128. Different line colors correspond to different activation functions.

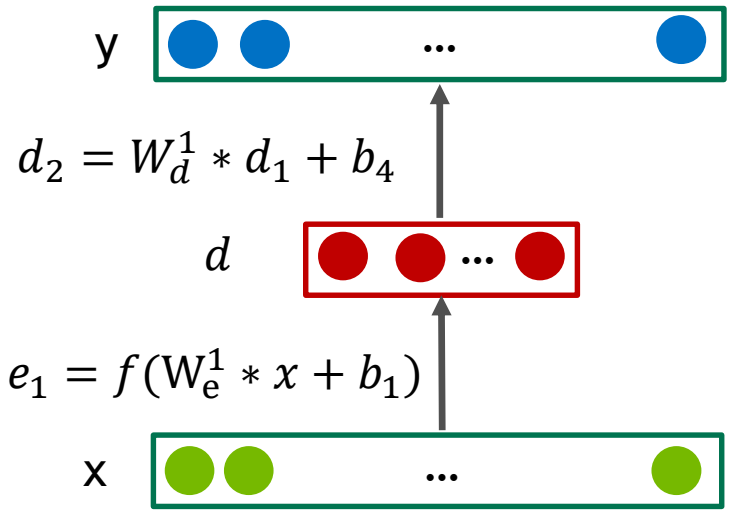
Autoencoders & collaborative filtering
Effects of the activation types
Overfitting the data
Going deeper
Dropout
Dense re-feeding
Conclusions

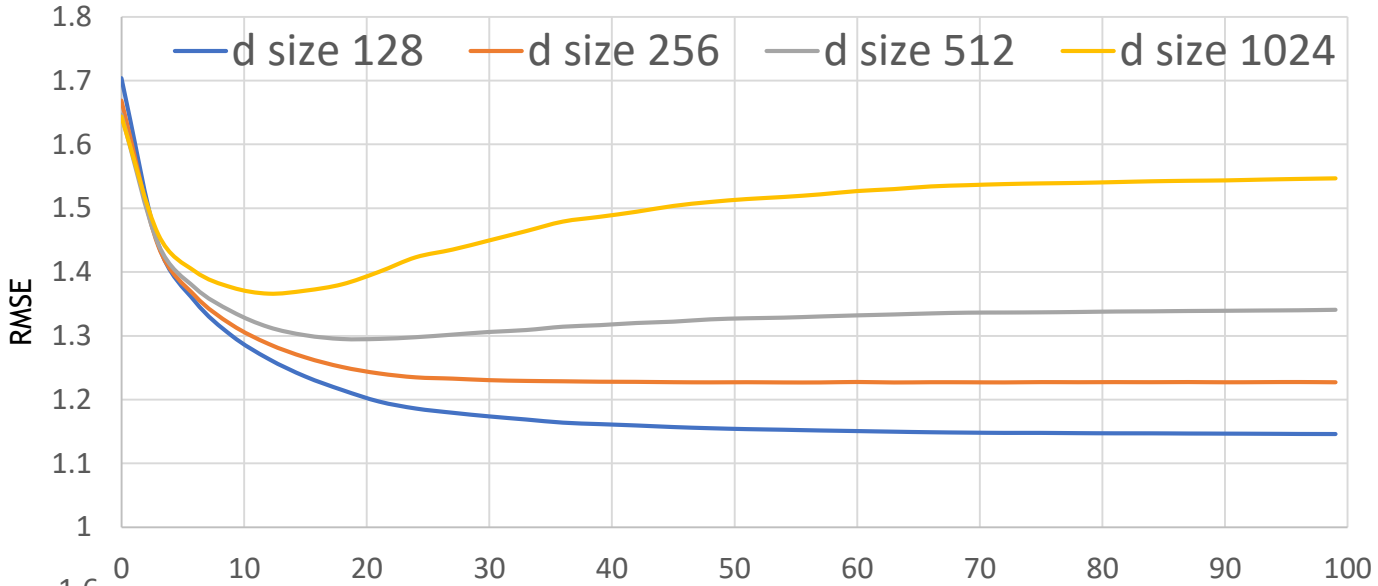
Oleksii Kuchaiev and Boris Ginsburg "Training Deep AutoEncoders for Collaborative Filtering“, *arXiv preprint arXiv:1708.01715* (2017).

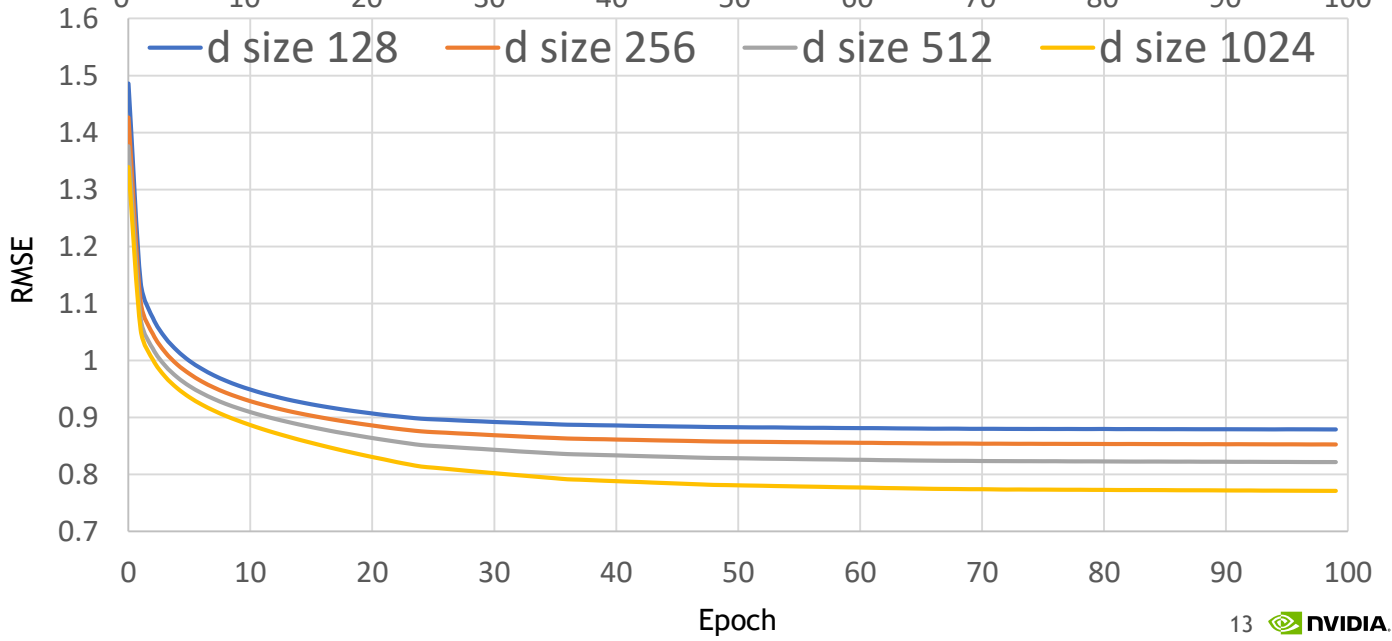
Overfit your data

Wide layers

generalize poorly







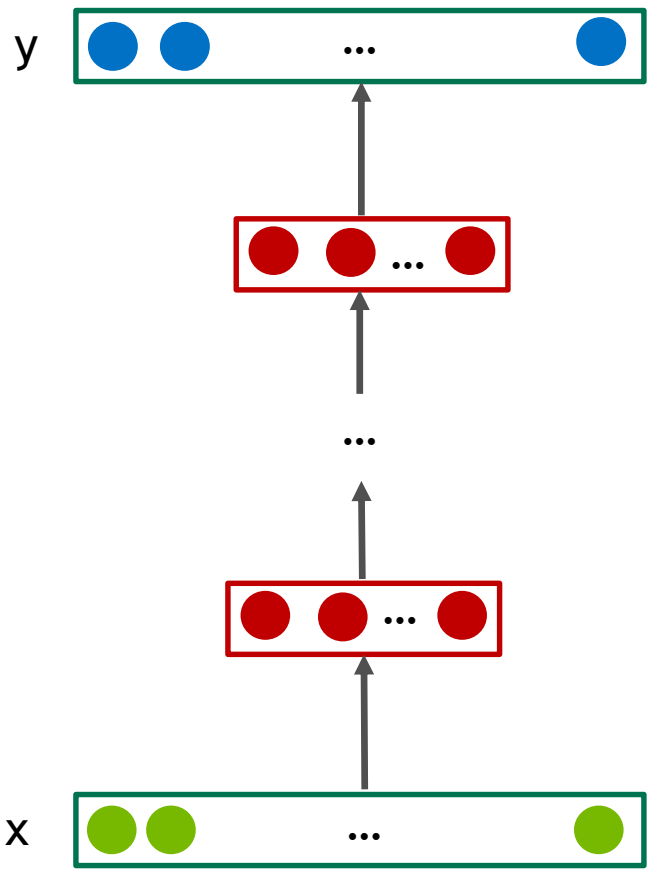
Evaluation RMSE > 1.1 on Netflix full

Autoencoders & collaborative filtering
Effects of the activation types
Overfitting the data
Going deeper
Dropout
Dense re-feeding
Conclusions

Oleksii Kuchaiev and Boris Ginsburg "Training Deep AutoEncoders for Collaborative Filtering“, *arXiv preprint arXiv:1708.01715* (2017).

Deeper models

Generalize better



Number of layers	Evaluation RMSE	params
2	1.146	4,566,504
4	0.9615	4,599,528
6	0.9378	4,632,552
8	0.9364	4,665,576
10	0.9340	4,698,600
12	0.9328	4,731,624

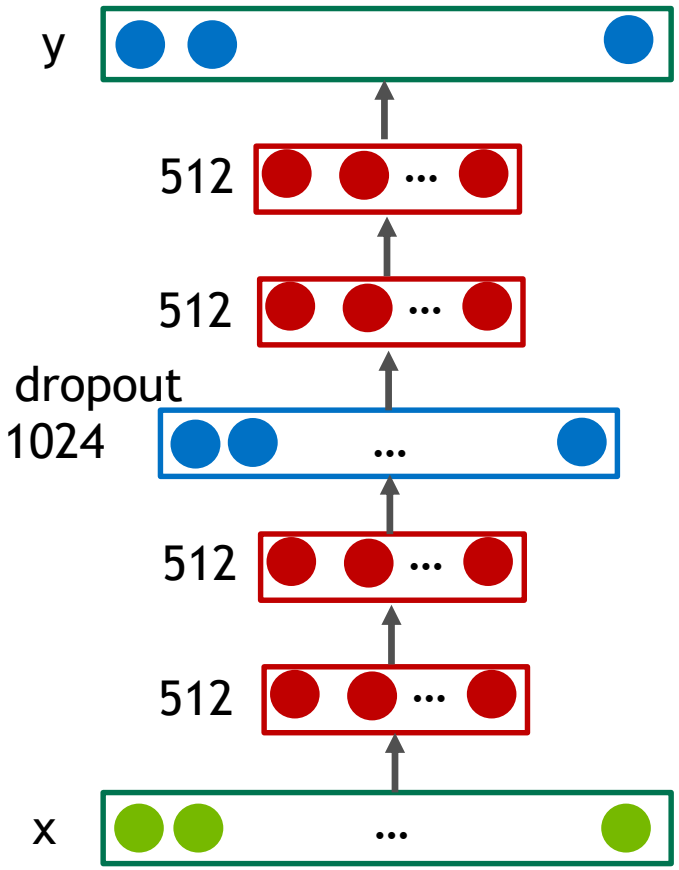
No layer-wise pre-training **necessary**!

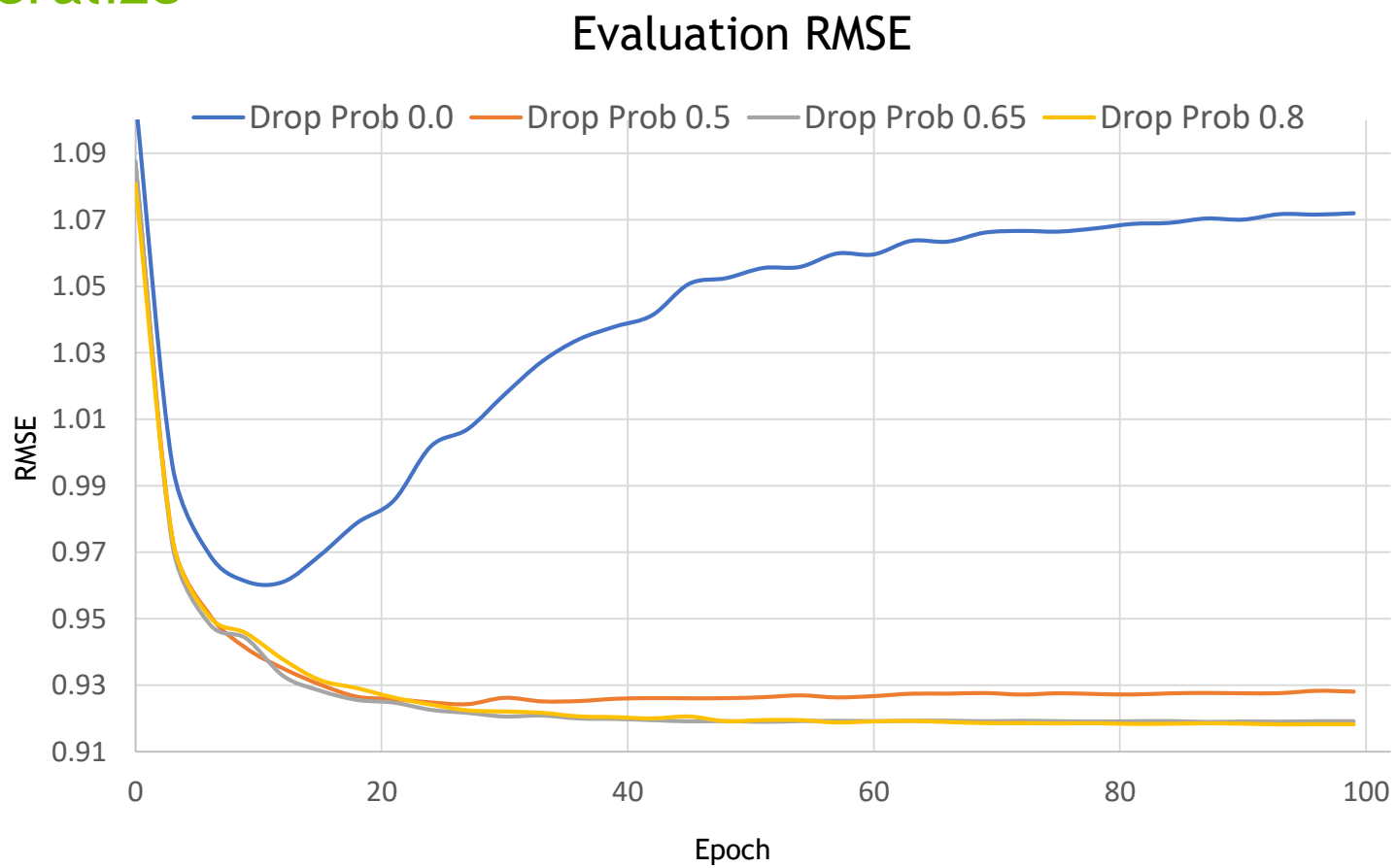
Autoencoders & collaborative filtering
Effects of the activation types
Overfitting the data
Going deeper
Dropout
Dense re-feeding
Conclusions

Oleksii Kuchaiev and Boris Ginsburg "Training Deep AutoEncoders for Collaborative Filtering“, *arXiv preprint arXiv:1708.01715* (2017).

Dropout

Helps wider models generalize





Autoencoders & collaborative filtering
Effects of the activation types
Overfitting the data
Going deeper
Dropout
Dense re-feeding
Conclusions

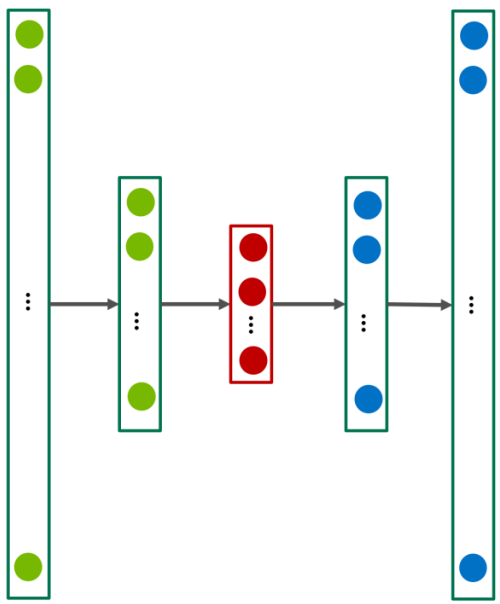
Oleksii Kuchaiev and Boris Ginsburg "Training Deep AutoEncoders for Collaborative Filtering“, *arXiv preprint arXiv:1708.01715* (2017).

Dense re-feeding

Note that *x* is *sparse* but *f(x)* is *dense*
For *x*, most of the loss is masked

$$MMSE = \frac{m_i * (r_i - y_i)^2}{\sum_{i=0}^{i=n} m_i}$$

f(x)=



Intuition: idealized scenario

Imagine **perfect *f***

$$\forall x_i \neq 0\colon f(x)_i = x_i$$

If user later rates <u>new</u> item *k* with rating *r, then:*

$$f(x)_k = r$$

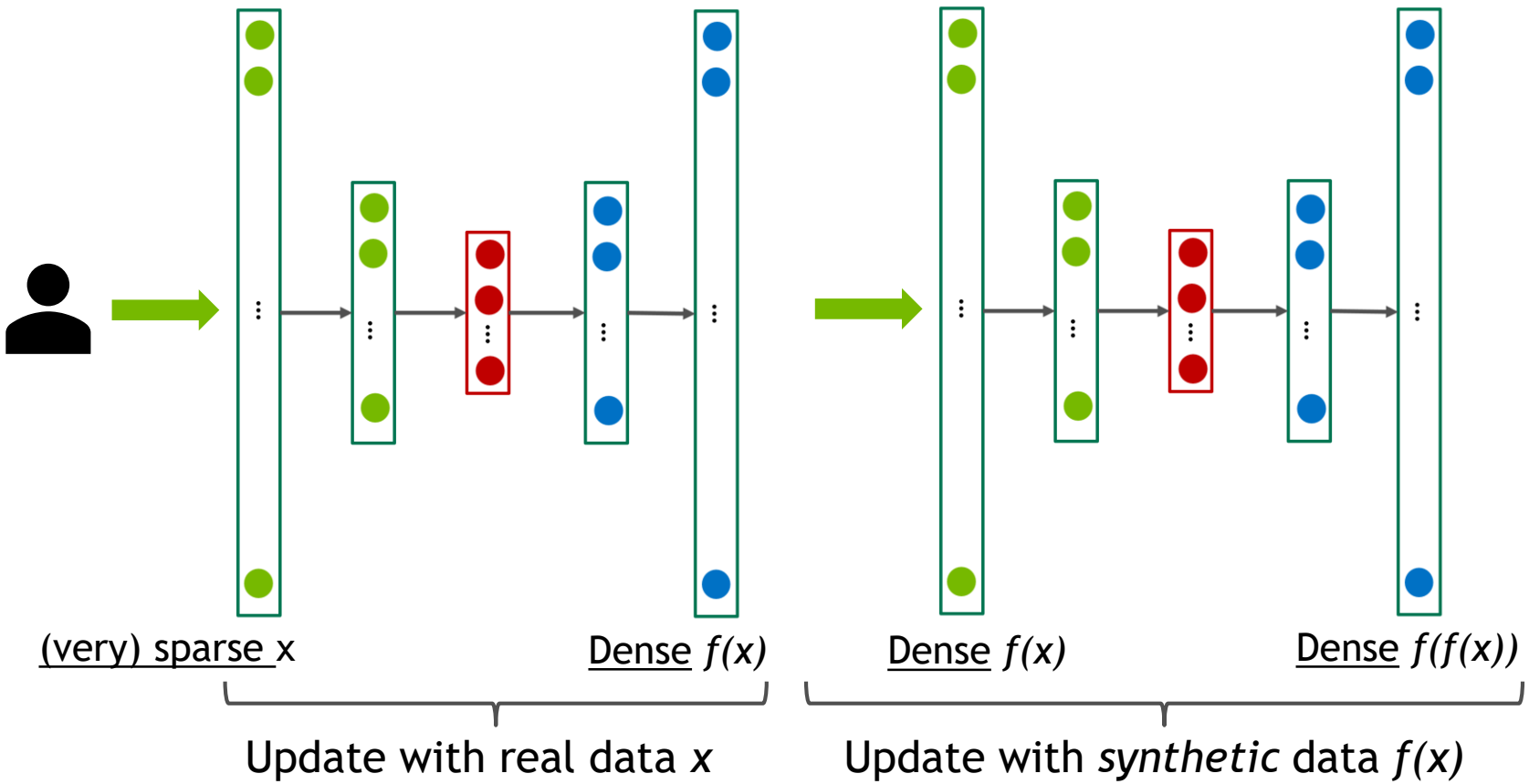
By induction:

$$f\big(f(x)\big) = f(x)$$

Thus, *f(x)* should be a ***fixed point*** of *f* for every valid *x*

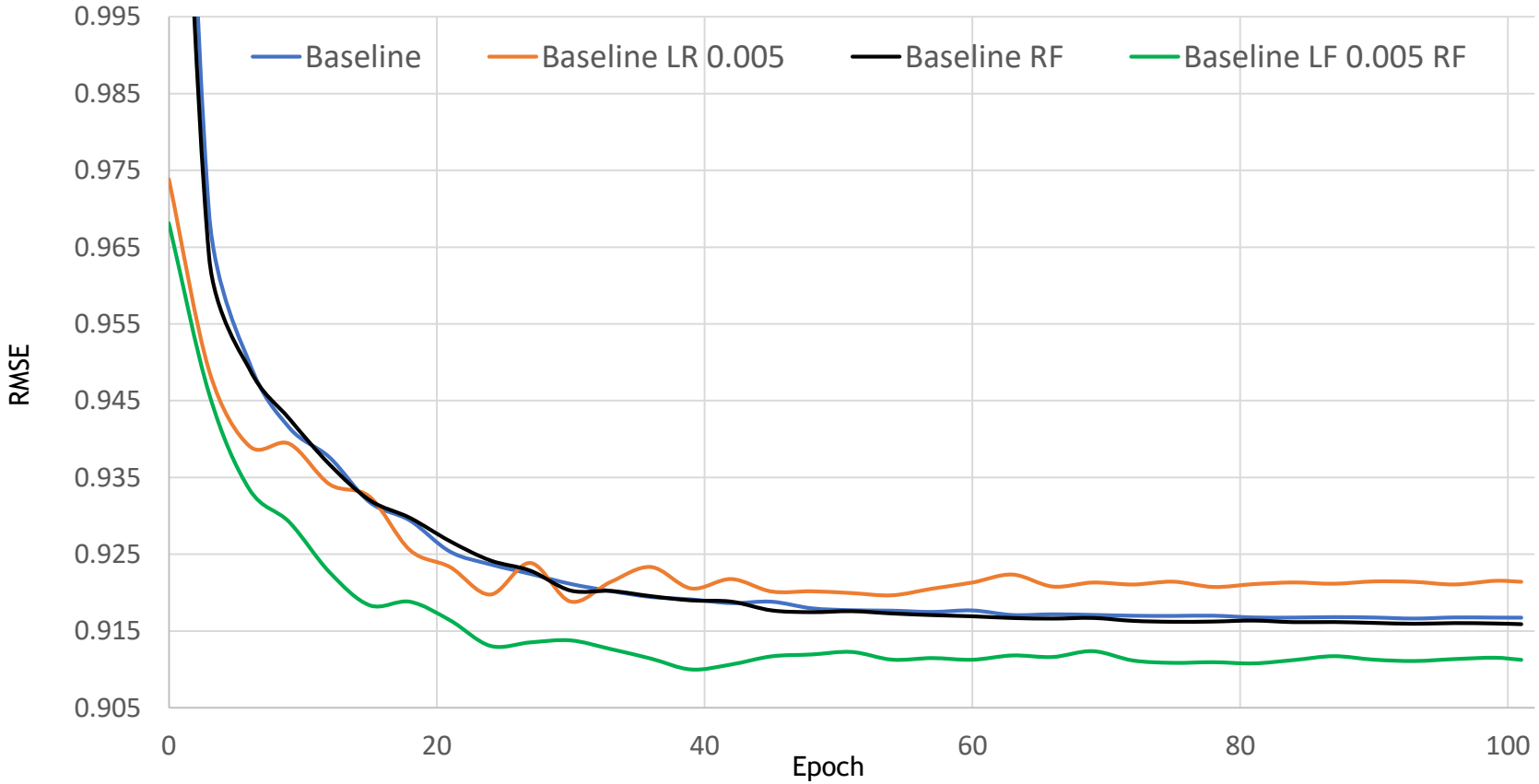
Dense re-feeding

Attempt to enforce fixed point constraint



Dense re-feeding

Together with bigger LR improves generalization



Results

Netflix time split data

DataSet	I-AR	U-AR	RRN	DeepRec
Netflix 3 months	0.9778	0.9836	0.9427	**0.9373**
Netfix Full	0.9364	0.9647	0.9224	**0.9099**

I-AR, U-AR: Sedhain, Suvash, et al. "Autorec: Autoencoders meet collaborative filtering." Proceedings of the 24th International Conference on World Wide Web. ACM, 2015.

RRN: Wu, Chao-Yuan, et al. "Recurrent recommender networks." Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. ACM, 2017.

DeepRec is our 6 layer model

Conclusions

1. Autoencoders can replace ALS and be competitive with other methods
2. Deeper models generalize better
    1. No layer-wise pre-training is necessary
3. Right activation function enables deep architectures
    1. Negative parts are important
    2. Unbounded positive part
4. Heavy use of dropout is needed for wider models
5. Dense *re-feeding* further improves generalization

Oleksii Kuchaiev and Boris Ginsburg "Training Deep Autoencoders for Collaborative Filtering“, *arXiv preprint arXiv:1708.01715* (2017).

Code, docs and tutorial:
https://github.com/NVIDIA/DeepRecommender