# Recommender Systems || Lecture 7

Summer 2022

Dr. Karthik Mohan

Univ. of Washington, Seattle

July 21, 2022

# Today

1. SGD Review

# Today

1. SGD Review
2. Auto differentation

# Today

1. SGD Review
2. Auto differentation

# Today

1. SGD Review
2. Auto differentation
3. Auto Encoders for Recommendation Systems

# SGD in practice - mini-batch SGD!

## mini-batch SGD

# data pts.

Let $L(w) = \sum_{i=1}^{N} L_i(w)$ where $L_i$ is a function of only the *ith* data point $(x_i, y_i)$ and parameter $w$. Let $B$ be the number of batches and $k$ be the batch size.

1. **Initialize** $w = w_0$ (randomize) $(U^0, V^0)$

# SGD in practice - mini-batch SGD!

### mini-batch SGD

Let $L(w) = \sum_{i=1}^{N} L_i(w)$ where $L_i$ is a function of only the *ith* data point $(x_i, y_i)$ and parameter $w$. Let $B$ be the number of batches and $k$ be the batch size.

1. **Initialize** $w = w_0$ (randomize) Pick a batch of $k$ data points at random between 1 and N: $i_1, i_2, \ldots, i_k$!

   $\hookrightarrow$ k indices

# SGD in practice - mini-batch SGD!

## mini-batch SGD

Let $L(w) = \sum_{i=1}^{N} L_i(w)$ where $L_i$ is a function of only the *ith* data point $(x_i, y_i)$ and parameter $w$. Let $B$ be the number of batches and $k$ be the batch size.

1. **Initialize** $w = w_0$ (randomize) Pick a batch of $k$ data points at random between 1 and N: $i_1, i_2, \ldots, i_k$!
2. **Gradient Descent** $w^{k+1} \leftarrow w^k - lr * \underbrace{\sum_{j=1}^{k} \nabla_w L_{i_j}(w^k)}$

Assignment 2 part 2

Fill this in

# SGD in practice - mini-batch SGD!

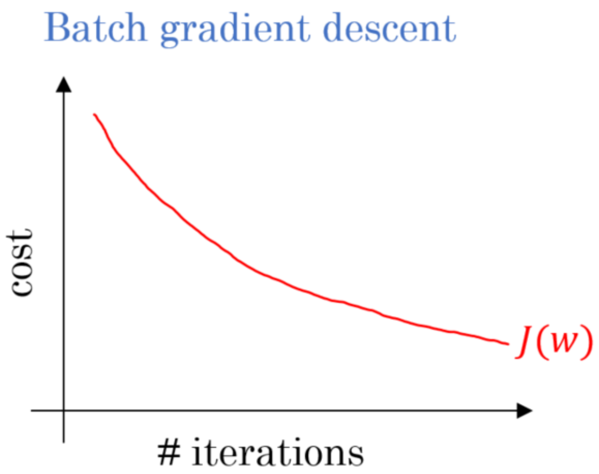
### mini-batch SGD

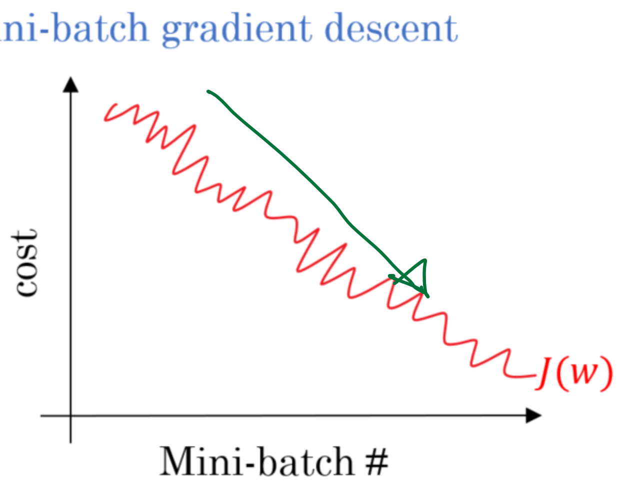
Let $L(w) = \sum_{i=1}^{N} L_i(w)$ where $L_i$ is a function of only the *ith* data point $(x_i, y_i)$ and parameter $w$. Let $B$ be the number of batches and $k$ be the batch size.

1. **Initialize** $w = w_0$ (randomize) Pick a batch of $k$ data points at random between 1 and N: $i_1, i_2, \ldots, i_k$!
2. **Gradient Descent** $w^{k+1} \leftarrow w^k - lr * \sum_{j=1}^{k} \nabla_w L_{i_j}(w^k)$
3. **Iterate** Repeat step 2 and 3 until $w$ converges, i.e.

$$\underline{\|w^{k+1} - w^k\| / \|w^k\| \leq 10^{-3}}$$

# GD vs Mini-batch convergence behavior



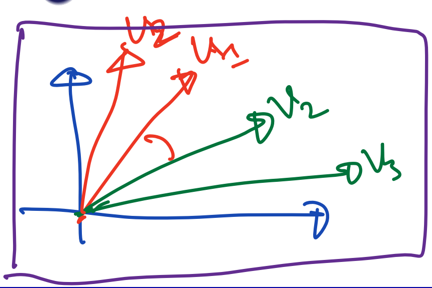
# GD vs mini-batch SGD

| Factor | GD | Mini-batch SGD |
|---|---|---|
| Data | All per iteration | Mini-batch (usually 128 or 256) |
| Randomness | Deterministic | Stochastic |
| Error reduction | Monotonic | Stochastic |
| Computation | High | Low |
| Memory big data | Intractable | Tractable |
| Convergence | Low relative error | Few "passes" on data |
| Local Minima traps | Yes | No |

# ICE #1

The correct loss function based on user factors (V) and movie factors (U) for this problem is given by:

1. $L(X; U, V) = (u_1^T v_1 - 1)^2 + (u_1^T v_3)^2 + (u_2^T v_2 - 1)^2 + (u_2^T v_4)^2 + (u_3^T v_1)^2 + (u_3^T v_2 - 1)^2 + (u_4^T v_1)^2 + (u_4^T v_2)^2 + (u_4^T V_3 - 1)^2$
2. $L(X; U, V) = (v_1^T u_1 - 1)^2 + (v_1^T u_3)^2 + (v_2^T u_2 - 1)^2 + (v_2^T u_4)^2 + (v_3^T u_1)^2 + (v_3^T u_2 - 1)^2 + (v_4^T u_1)^2 + (v_4^T u_2)^2 + (v_4^T u_3 - 1)^2$
3. None of the above
4. Both are valid




Users

Movies

| | Tom | Henry | Anaida | Shakir |
|---|---|---|---|---|
| Sun | 1 | ? | 0 | ? |
| Moon | ? | 1 | ? | 0 |
| Stars | 0 | 1 | ? | ? |
| Galaxies | 0 | 0 | 1 | ? |

# Automatic Differentiation in Torch

Notebook Example

# Programming 2, Part 2

Notebook Walkthrough

Project Proposal (1 pager)

# Auto Encoders for Recommendation Systems

$$\underbrace{L(X}_{\text{Data}};\underbrace{U,V)}_{\text{parameters}} = \frac{1}{N}\sum_{(i,j)\in O}\big(X_{ij} - \underbrace{U_i^T V_j}_{\text{Bilinear fn.}}\big)^2$$

1. Natural extension of matrix factorization
2. Matrix Factorization methods are "bi-linear" (linear in $U$, $V$) and linear models have limitations in learning power
3. Auto Encoders can have non-linearity

# Model 1: AutoRec

Loss :-

$\frac{1}{|U|}\sum_{u}(x^u - \hat{x}^u)^2$

MSE → mean squared Error

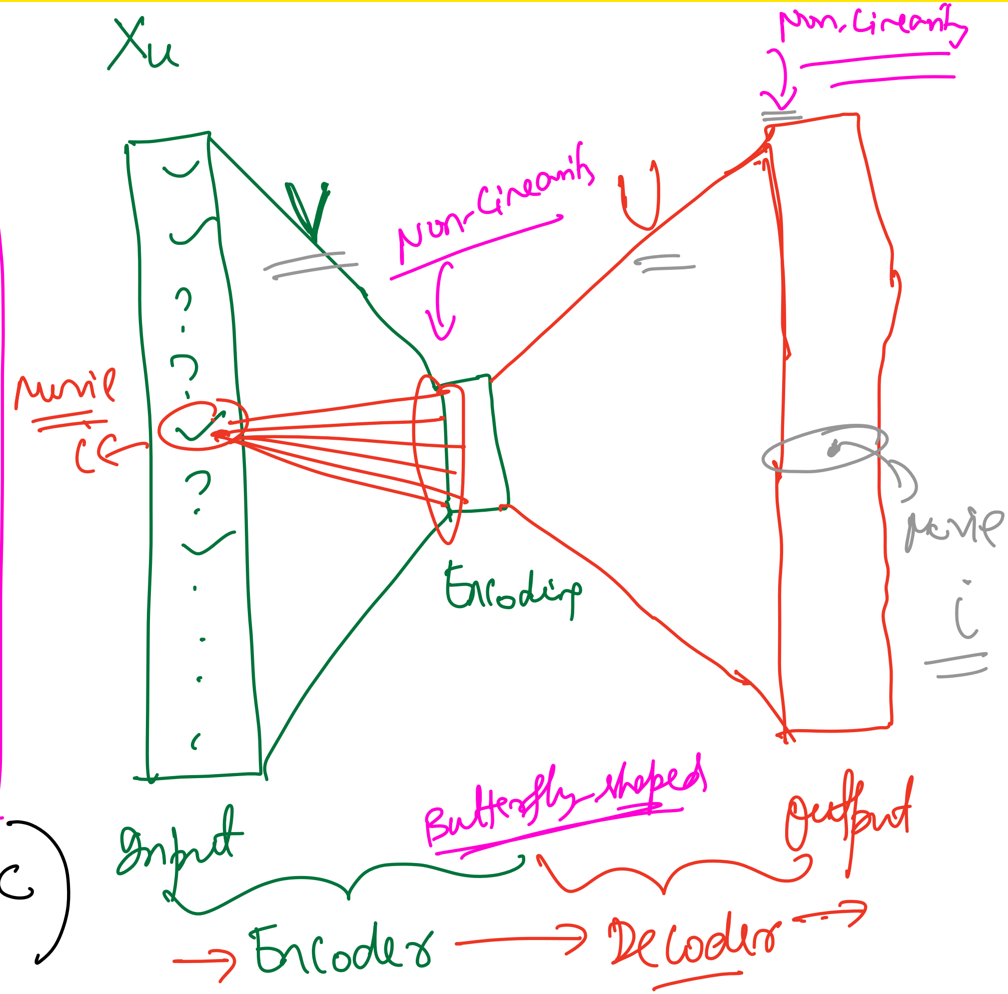
Reference: AutoRec

AutoEncoders for RecSys

Estimated $x^u$

Non-Linearity

$$\hat{x}^u = g\left(U f(V x^u + b) + c\right)$$



# Model 1: AutoRec

| $f(\cdot)$ | $g(\cdot)$ | RMSE |
|---|---|---|
| Identity | Identity | 0.872 |
| Sigmoid | Identity | 0.852 |
| Identity | Sigmoid | **0.831** |
| Sigmoid | Sigmoid | 0.836 |

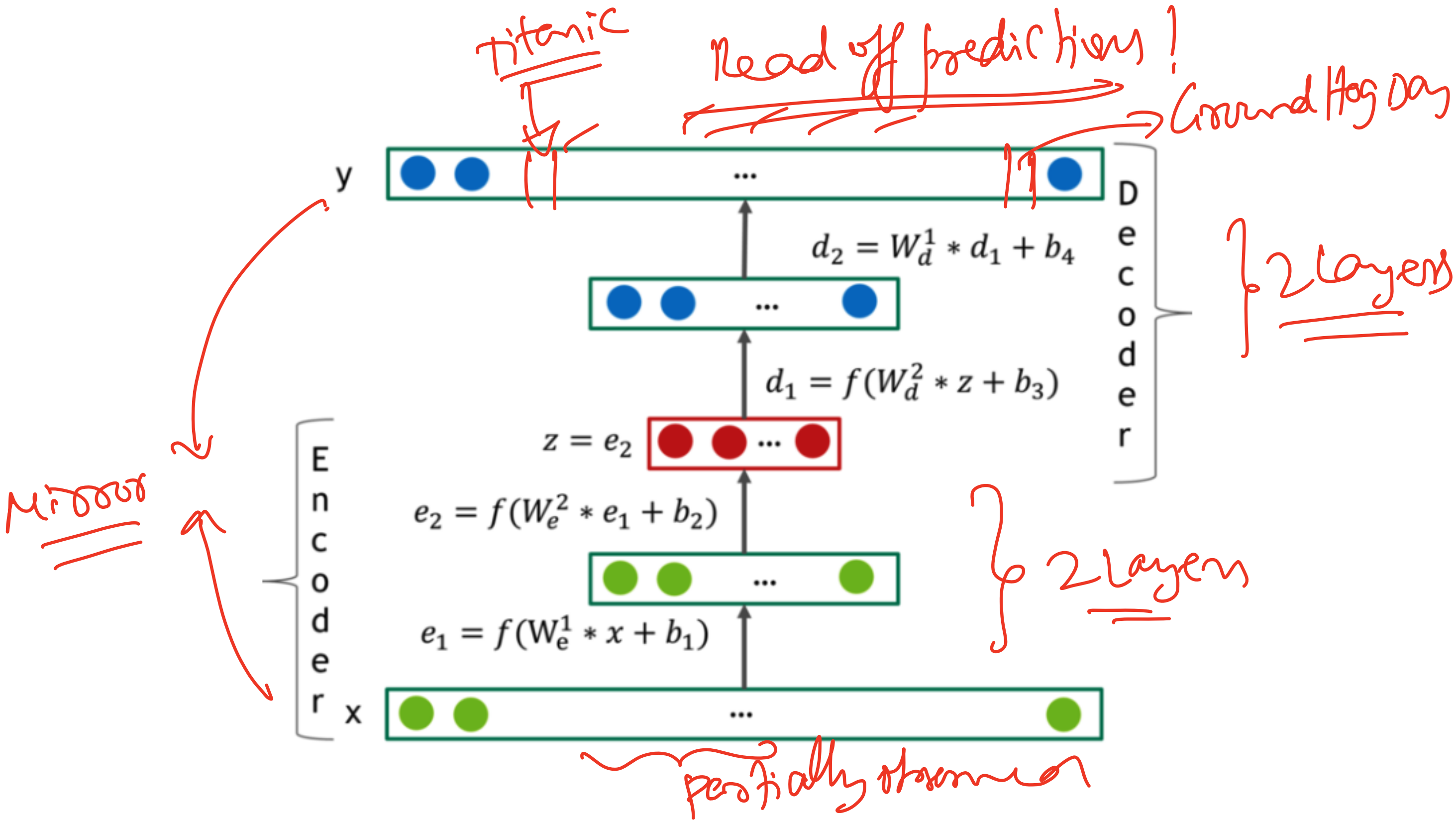
movielens

Netflix Prize Dataset

| | ML-1M | ML-10M | Netflix |
|---|---|---|---|
| BiasedMF | 0.845 | 0.803 | 0.844 |
| I-RBM | 0.854 | 0.825 | - |
| U-RBM | 0.881 | 0.823 | 0.845 |
| LLORMA | 0.833 | **0.782** | 0.834 |
| I-AutoRec | **0.831** | **0.782** | **0.823** |

Reference: AutoRec

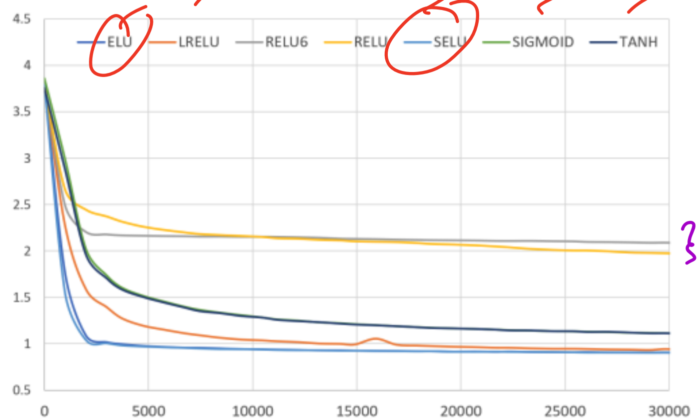
# Model 2: Deep Rec




Training Deep Auto Encoders for collaborative filtering

# Model 2: Deep Rec

RMSE

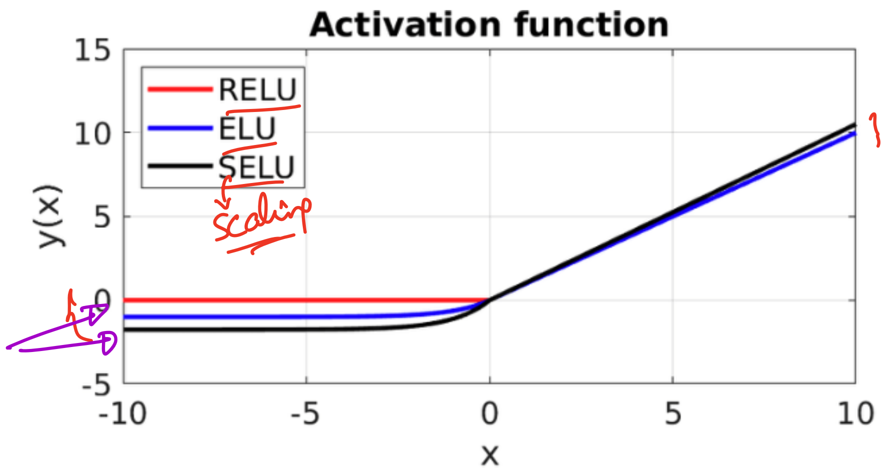



→ RMSE = 2!!

Mini Batches

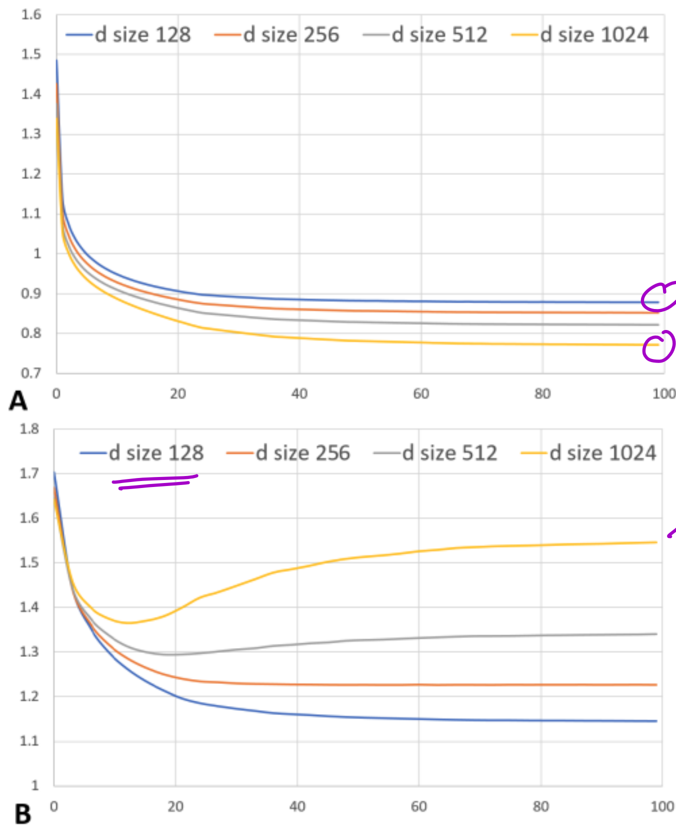
Training Deep Auto Encoders for collaborative filtering

# Model 2: Deep Rec




Training Deep Auto Encoders for collaborative filtering

# Model 2: Deep Rec




1 d=128

2 → Good training ✓

→ Bad Eval Error ✗

Training Deep Auto Encoders for collaborative filtering

# Model 2: Deep Rec

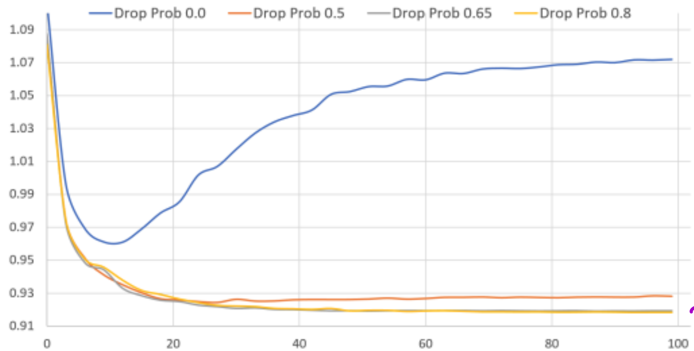
| Number of layers | Evaluation RMSE | params |
|---|---|---|
| 2 | 1.146 | 4,566,504 |
| 4 | 0.9615 | 4,599,528 |
| 6 | 0.9378 | 4,632,552 |
| 8 | 0.9364 | 4,665,576 |
| 10 | 0.9340 | 4,698,600 |
| 12 | 0.9328 | 4,731,624 |

Point of Diminishing Returns

5Mn parameters

Training Deep Auto Encoders for collaborative filtering

# Model 2: Deep Rec




→ No dropout!

→ Even Lower RMSE with high dropouts

**Figure 4: Effects of dropout. Y-axis: evaluation RMSE, X-axis: epoch number. Model with no dropout (Drop Prob 0.0) clearly over-fits. Model with drop probability of 0.5 over-fits as well (but much slowly). Models with drop probabilities of 0.65 and 0.8 result in RMSEs of 0.9192 and 0.9183 correspondingly.**

Training Deep Auto Encoders for collaborative filtering

# Model 2: Deep Rec

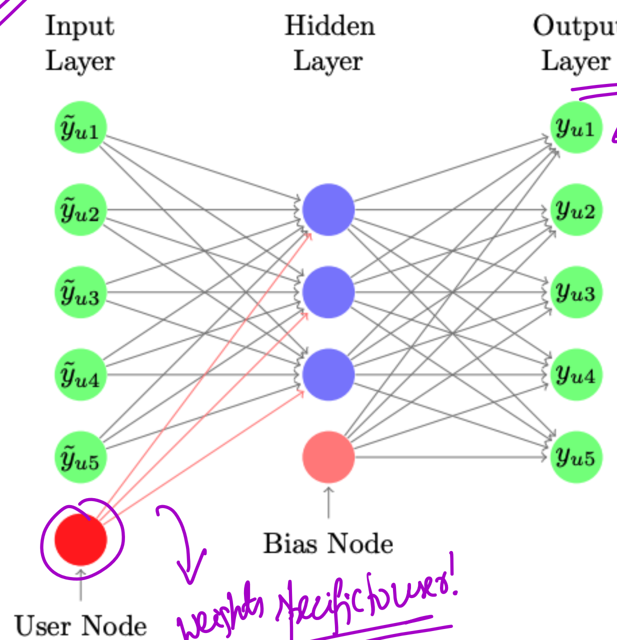
| DataSet | I-AR | U-AR | RRN | DeepRec |
|---|---|---|---|---|
| Netflix 3 months | 0.9778 | 0.9836 | 0.9427 | **0.9373** |
| Netfix Full | 0.9364 | 0.9647 | 0.9224 | **0.9099** |

Training Deep Auto Encoders for collaborative filtering

# Model 3: Collaborative De-noising AuotEncoder

Corruption of input

(Partial Injection of noise)




weights specific to users!

Noise forces model to be more Robust!!!

Precision & Recall@N

CDAE

## Summarize

1. MF model - Bi-Linear
2. Bi-linear models Limitations (RMSE limitations)
3. Auto-Encoders work well for RecSys
4. Non-Linear & they can be deep
5. Attain SOTA on many benchmark datasets
6. Scalability issues! (E.g. output layer becomes too big).

DSSM | Siamese Networks } In next lecture!
(NLP)