

COMP4211 - Machine Learning

Fall 2024, HKUST

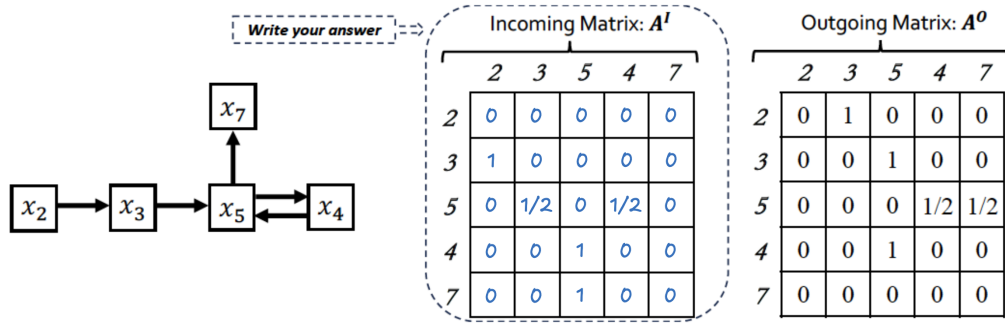
Problem Set

Name: Shawn Darren S. Chua

Student ID: 20946705

Problem 1.

(1)



- (2) $h^{l-1^\top} \in \mathbb{R}^d$ and $h^{l-1^\top} \mathbf{W}^I \in \mathbb{R}^d$, so $\mathbf{W}^I \in \mathbb{R}^{d \times d}$
 $h^{l-1^\top} \in \mathbb{R}^d$ and $h^{l-1^\top} \mathbf{W}^O \in \mathbb{R}^d$, so $\mathbf{W}^O \in \mathbb{R}^{d \times d}$
 Thus $\mathbf{W}^I, \mathbf{W}^O$ both have d^2 parameters.

- (3) $a_i^l \in \mathbb{R}^{2d \times 1}$ and $\mathbf{W}_z a_i^l \in \mathbb{R}^{d \times 1}$, so $\mathbf{W}_z \in \mathbb{R}^{d \times 2d}$
 $a_i^l \in \mathbb{R}^{2d \times 1}$ and $\mathbf{W}_r a_i^l \in \mathbb{R}^{d \times 1}$, so $\mathbf{W}_r \in \mathbb{R}^{d \times 2d}$
 $a_i^l \in \mathbb{R}^{2d \times 1}$ and $\mathbf{W}_h a_i^l \in \mathbb{R}^{d \times 1}$, so $\mathbf{W}_h \in \mathbb{R}^{d \times 2d}$
 Thus $\mathbf{W}_z, \mathbf{W}_r, \mathbf{W}_h$ all have $2d^2$ parameters.

$h_i^{l-1} \in \mathbb{R}^{d \times 1}$ and $\mathbf{U}_z h_i^{l-1} \in \mathbb{R}^{d \times 1}$, so $\mathbf{U}_z \in \mathbb{R}^{d \times d}$
 $h_i^{l-1} \in \mathbb{R}^{d \times 1}$ and $\mathbf{U}_r h_i^{l-1} \in \mathbb{R}^{d \times 1}$, so $\mathbf{U}_r \in \mathbb{R}^{d \times d}$
 $r_i^l \odot h_i^{l-1} \in \mathbb{R}^{d \times 1}$ and $\mathbf{U}_h (r_i^l \odot h_i^{l-1}) \in \mathbb{R}^{d \times 1}$, so $\mathbf{U}_h \in \mathbb{R}^{d \times d}$
 Thus $\mathbf{U}_z, \mathbf{U}_r, \mathbf{U}_h$ all have d^2 parameters.

(4) Experiments to demonstrate the effectiveness of CGNN

- I will train and test the CGNN model on real-world datasets such as MovieLens and Amazon product reviews and compare its performance against baseline recommendation models like matrix factorization and collaborative filtering.
- I will conduct ablation studies to assess various components of CGNN, so I'll be able to better understand its functionality.
- To evaluate the robustness of CGNN, I will test it with inputs that vary in session lengths and item diversity. For example, one sequence may consist of a few copies of only one or two items, while another sequence may contain entirely different items.

Methods to prevent overfitting in CGNN

- Introduce dropout layers or use regularization techniques.
- Randomly modify the input data, either by adding new items or modifying existing items, to increase robustness.

(5) $\mathbf{W}^I, \mathbf{W}^O$ have a total of $2d^2$ parameters.

$\mathbf{b}^I, \mathbf{b}^O \in \mathbb{R}^d$ have a total of $2d$ parameters.

$\mathbf{W}_z, \mathbf{W}_r, \mathbf{W}_h$ have a total of $6d^2$ parameters.

$\mathbf{U}_z, \mathbf{U}_r, \mathbf{U}_h$ have a total of $3d^2$ parameters.

$\mathbf{W}_{q1}, \mathbf{W}_{q2}$ have the same shapes as $\mathbf{W}_{k1}, \mathbf{W}_{k2}$, so they have a total of $6d^2$ parameters.

We have to learn 100-dimensional embeddings for all 2520 unique items, for 252000 parameters

Summing these, we get $2d^2 + 2d + 6d^2 + 3d^2 + 6d^2 + md = 17d^2 + 2d + md = \mathbf{422200}$ parameters.

Problem 2.

We have to learn 100-dimensional embeddings for all $1000+520 = 1520$ users, for 152000 parameters

$e^{(0)} \in \mathbb{R}^{100}, e^{(1)} \in \mathbb{R}^{80}, e^{(2)} \in \mathbb{R}^{40}, e^{(3)} \in \mathbb{R}^{20}$, so

$$\begin{aligned} \mathbf{W}_1^{(1)} &\in \mathbb{R}^{80 \times 100}, & \mathbf{W}_2^{(1)} &\in \mathbb{R}^{80 \times 100}, & \mathbf{W}_3^{(1)} &\in \mathbb{R}^{80 \times 200} \\ \mathbf{W}_1^{(2)} &\in \mathbb{R}^{40 \times 80}, & \mathbf{W}_2^{(2)} &\in \mathbb{R}^{40 \times 80}, & \mathbf{W}_3^{(2)} &\in \mathbb{R}^{40 \times 160} \\ \mathbf{W}_1^{(3)} &\in \mathbb{R}^{20 \times 40}, & \mathbf{W}_2^{(3)} &\in \mathbb{R}^{20 \times 40}, & \mathbf{W}_3^{(3)} &\in \mathbb{R}^{20 \times 80} \end{aligned}$$

Therefore, we have a total of $152000 + 80 \cdot 400 + 40 \cdot 320 + 20 \cdot 160 = 200000$ parameters.