Graph Neural Networks

CISC 7026: Introduction to Deep Learning

University of Macau

Agenda

1. Review
2. Quiz
3. Introduction to Prof. Li
4. Graph Neural Networks

GNNs

$$f(\boldsymbol{x}, \boldsymbol{\theta})_j = \boldsymbol{\theta}^\top \overline{\boldsymbol{x}}$$

Agenda

1. **Review**
2. Quiz
3. Introduction to Prof. Li
4. Graph Neural Networks

Review

Review

What happens on your computer when you watch Shrek?

Review

What happens on your computer when you watch Shrek?

Download $\boldsymbol{z} \in Z$ from the internet

Review

What happens on your computer when you watch Shrek?

Download $\boldsymbol{z} \in Z$ from the internet $\quad\quad Z \in \{0, 1\}^n$

Review

What happens on your computer when you watch Shrek?

Download $\boldsymbol{z} \in Z$ from the internet $\quad Z \in \{0, 1\}^n$

Information is no longer pixels, it is a string of bits

Review

What happens on your computer when you watch Shrek?

Download $\boldsymbol{z} \in Z$ from the internet $\quad Z \in \{0, 1\}^n$

Information is no longer pixels, it is a string of bits

We must **decode** $\boldsymbol{z}$ back into pixels

Review

What happens on your computer when you watch Shrek?

Download $\boldsymbol{z} \in Z$ from the internet $\quad Z \in \{0, 1\}^n$

Information is no longer pixels, it is a string of bits

We must **decode** $\boldsymbol{z}$ back into pixels

We need to undo (invert) the encoder f

$$f : X^t \mapsto Z$$

$$f^{-1} : Z \mapsto X^t$$

Review

What happens on your computer when you watch Shrek?

Download $\boldsymbol{z} \in Z$ from the internet $\quad Z \in \{0, 1\}^n$

Information is no longer pixels, it is a string of bits

We must **decode** $\boldsymbol{z}$ back into pixels

We need to undo (invert) the encoder f

$$f : X^t \mapsto Z$$

$$f^{-1} : Z \mapsto X^t$$

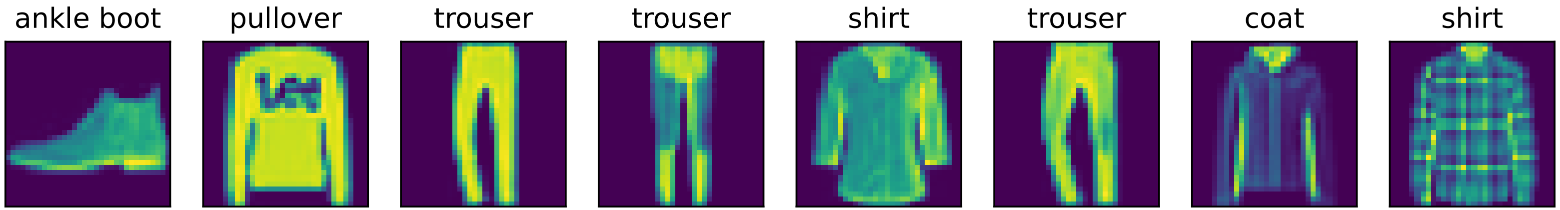
You CPU has a H.264 decoder built in to make this fast

Review

Task: Compress images for your clothing website to save on costs

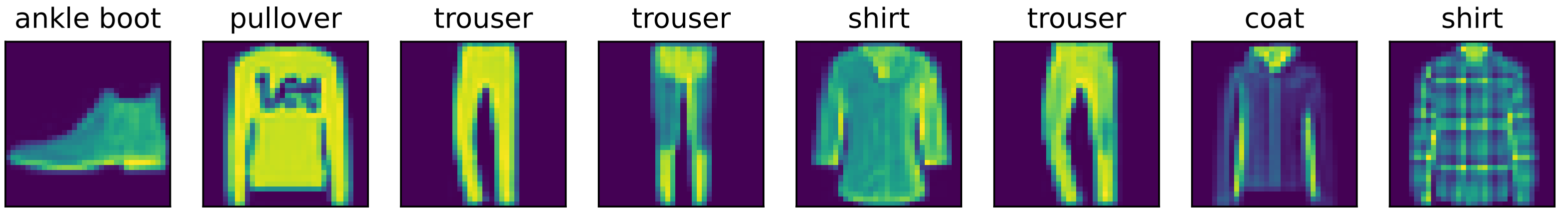
Review

Task: Compress images for your clothing website to save on costs



Review

Task: Compress images for your clothing website to save on costs

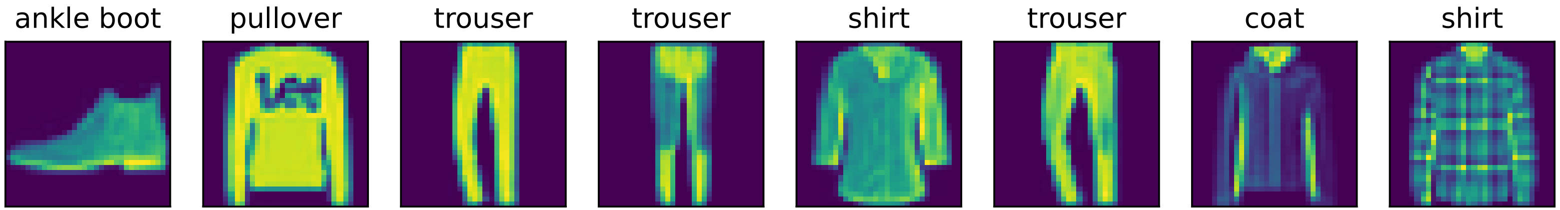


$$X \in [0, 1]^{d_x}$$

$$Z \in \mathbb{R}^{d_z}$$

Review

Task: Compress images for your clothing website to save on costs



$$X \in [0, 1]^{d_x}$$

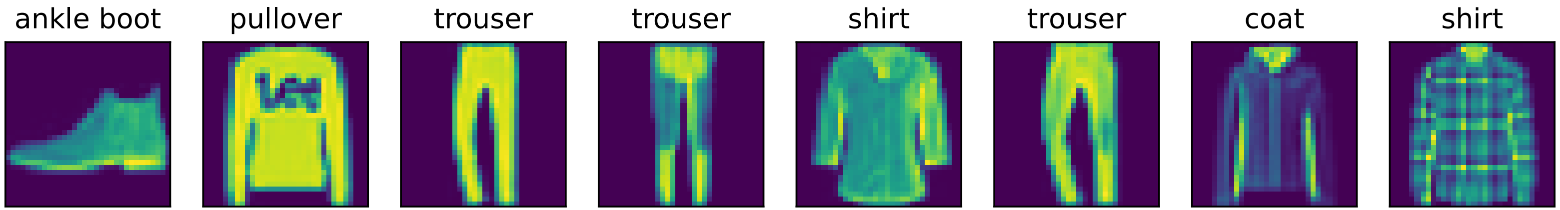
$$Z \in \mathbb{R}^{d_z}$$

$$d_x : 28 \times 28$$

$$d_z : 4$$

Review

Task: Compress images for your clothing website to save on costs



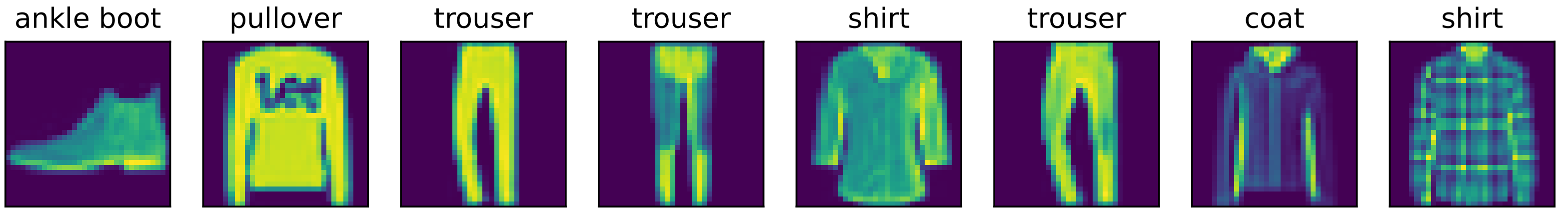
$$X \in [0, 1]^{d_x} \qquad Z \in \mathbb{R}^{d_z}$$

$$d_x : 28 \times 28 \qquad d_z : 4$$

$$f : X \times \Theta \mapsto Z \qquad f^{-1} : Z \times \Theta \mapsto X$$

Review

Task: Compress images for your clothing website to save on costs



$$X \in [0, 1]^{d_x} \qquad Z \in \mathbb{R}^{d_z}$$

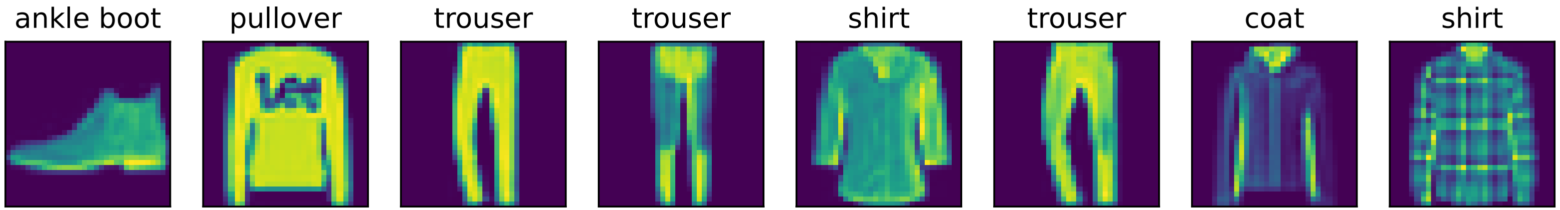
$$d_x : 28 \times 28 \qquad d_z : 4$$

$$f : X \times \Theta \mapsto Z \qquad f^{-1} : Z \times \Theta \mapsto X$$

What is the structure of f, f^{-1}?

Review

Task: Compress images for your clothing website to save on costs



$$X \in [0, 1]^{d_x}$$

$$Z \in \mathbb{R}^{d_z}$$

$$d_x : 28 \times 28$$

$$d_z : 4$$

$$f : X \times \Theta \mapsto Z$$

$$f^{-1} : Z \times \Theta \mapsto X$$

What is the structure of f, f^{-1}?

How do we find $\boldsymbol{\theta}$?

Review

$$\boldsymbol{z} = f(\boldsymbol{x}, \boldsymbol{\theta}_e) = \sigma\left(\boldsymbol{\theta}_e^\top \overline{\boldsymbol{x}}\right)$$

$$\boldsymbol{x} = f^{-1}(\boldsymbol{z}, \boldsymbol{\theta}_d) = \sigma\left(\boldsymbol{\theta}_d^\top \overline{\boldsymbol{z}}\right)$$

Review

$$\boldsymbol{z} = f(\boldsymbol{x}, \boldsymbol{\theta}_e) = \sigma\left(\boldsymbol{\theta}_e^\top \overline{\boldsymbol{x}}\right)$$

$$\boldsymbol{x} = f^{-1}(\boldsymbol{z}, \boldsymbol{\theta}_d) = \sigma\left(\boldsymbol{\theta}_d^\top \overline{\boldsymbol{z}}\right)$$

What if we plug $\boldsymbol{z}$ into the second equation?

Review

Let us try another way

$$\boldsymbol{z} = f(\boldsymbol{x}, \boldsymbol{\theta}_e) = \sigma(\boldsymbol{\theta}_e^\top \overline{\boldsymbol{x}})$$

$$\boldsymbol{x} = f^{-1}(\boldsymbol{z}, \boldsymbol{\theta}_d) = \sigma(\boldsymbol{\theta}_d^\top \overline{\boldsymbol{z}})$$

What if we plug $\boldsymbol{z}$ into the second equation?

$$\boldsymbol{x} = f^{-1}(f(\boldsymbol{x}, \boldsymbol{\theta}_e), \boldsymbol{\theta}_d) = \sigma(\boldsymbol{\theta}_d^\top \boldsymbol{\sigma}(\boldsymbol{\theta}_e^\top \overline{\boldsymbol{x}}))$$

Review

$$\boldsymbol{x} = f^{-1}(f(\boldsymbol{x}, \boldsymbol{\theta}_e), \boldsymbol{\theta}_d) = \sigma\left(\boldsymbol{\theta}_d^\top \boldsymbol{\sigma}\left(\boldsymbol{\theta}_{\boldsymbol{e}}^\top \overline{\boldsymbol{x}}\right)\right)$$

Review

$$\boldsymbol{x} = f^{-1}(f(\boldsymbol{x}, \boldsymbol{\theta}_e), \boldsymbol{\theta}_d) = \sigma\left(\boldsymbol{\theta}_d^\top \boldsymbol{\sigma}\left(\boldsymbol{\theta}_{\boldsymbol{e}}^\top \overline{\boldsymbol{x}}\right)\right)$$

More generally, f, f^{-1} may be any neural network

Review

$$\boldsymbol{x} = f^{-1}(f(\boldsymbol{x}, \boldsymbol{\theta}_e), \boldsymbol{\theta}_d) = \sigma\left(\boldsymbol{\theta}_d^\top \boldsymbol{\sigma}\left(\boldsymbol{\theta}_{\boldsymbol{e}}^\top \overline{\boldsymbol{x}}\right)\right)$$

More generally, f, f^{-1} may be any neural network

$$\boldsymbol{x} = f^{-1}(f(\boldsymbol{x}, \boldsymbol{\theta}_e), \boldsymbol{\theta}_d)$$

Review

$$\boldsymbol{x} = f^{-1}(f(\boldsymbol{x}, \boldsymbol{\theta}_e), \boldsymbol{\theta}_d) = \sigma\left(\boldsymbol{\theta}_d^\top \boldsymbol{\sigma}(\boldsymbol{\theta}_{\boldsymbol{e}}^\top \overline{\boldsymbol{x}})\right)$$

More generally, f, f^{-1} may be any neural network

$$\boldsymbol{x} = f^{-1}(f(\boldsymbol{x}, \boldsymbol{\theta}_e), \boldsymbol{\theta}_d)$$

Turn this into a loss function using the square error

Review

$$\boldsymbol{x} = f^{-1}(f(\boldsymbol{x}, \boldsymbol{\theta}_e), \boldsymbol{\theta}_d) = \sigma\left(\boldsymbol{\theta}_d^\top \boldsymbol{\sigma}\left(\boldsymbol{\theta}_{\boldsymbol{e}}^\top \overline{\boldsymbol{x}}\right)\right)$$

More generally, f, f^{-1} may be any neural network

$$\boldsymbol{x} = f^{-1}(f(\boldsymbol{x}, \boldsymbol{\theta}_e), \boldsymbol{\theta}_d)$$

Turn this into a loss function using the square error

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\theta}) = \sum_{j=1}^{d_x} \left(x_j - f^{-1}(f(\boldsymbol{x}, \boldsymbol{\theta}_e), \boldsymbol{\theta}_d)_j\right)^2$$

Review

$$\boldsymbol{x} = f^{-1}(f(\boldsymbol{x}, \boldsymbol{\theta}_e), \boldsymbol{\theta}_d) = \sigma\left(\boldsymbol{\theta}_d^\top \boldsymbol{\sigma}\left(\boldsymbol{\theta}_{\boldsymbol{e}}^\top \overline{\boldsymbol{x}}\right)\right)$$

More generally, f, f^{-1} may be any neural network

$$\boldsymbol{x} = f^{-1}(f(\boldsymbol{x}, \boldsymbol{\theta}_e), \boldsymbol{\theta}_d)$$

Turn this into a loss function using the square error

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\theta}) = \sum_{j=1}^{d_x} \left(x_j - f^{-1}(f(\boldsymbol{x}, \boldsymbol{\theta}_e), \boldsymbol{\theta}_d)_j\right)^2$$

Forces the networks to compress and reconstruct $\boldsymbol{x}$

Review

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\theta}) = \sum_{j=1}^{d_x} \left(x_j - f^{-1}(f(\boldsymbol{x}, \boldsymbol{\theta}_e), \boldsymbol{\theta}_d)_j \right)^2$$

Review

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\theta}) = \sum_{j=1}^{d_x} \left(x_j - f^{-1}(f(\boldsymbol{x}, \boldsymbol{\theta}_e), \boldsymbol{\theta}_d)_j \right)^2$$

Define over the entire dataset

$$\mathcal{L}(\boldsymbol{X}, \boldsymbol{\theta}) = \sum_{i=1}^{n} \sum_{j=1}^{d_x} \left(x_{[i],j} - f^{-1}\left(f\left(\boldsymbol{x}_{[i]}, \boldsymbol{\theta}_e\right), \boldsymbol{\theta}_d\right)_j \right)^2$$

Review

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\theta}) = \sum_{j=1}^{d_x} \left(x_j - f^{-1}(f(\boldsymbol{x}, \boldsymbol{\theta}_e), \boldsymbol{\theta}_d)_j \right)^2$$

Define over the entire dataset

$$\mathcal{L}(\boldsymbol{X}, \boldsymbol{\theta}) = \sum_{i=1}^{n} \sum_{j=1}^{d_x} \left(x_{[i],j} - f^{-1}\left(f\left(\boldsymbol{x}_{[i]}, \boldsymbol{\theta}_e\right), \boldsymbol{\theta}_d\right)_j \right)^2$$

We call this the **reconstruction loss**

Review

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\theta}) = \sum_{j=1}^{d_x} \left(x_j - f^{-1}(f(\boldsymbol{x}, \boldsymbol{\theta}_e), \boldsymbol{\theta}_d)_j \right)^2$$

Define over the entire dataset

$$\mathcal{L}(\boldsymbol{X}, \boldsymbol{\theta}) = \sum_{i=1}^{n} \sum_{j=1}^{d_x} \left(x_{[i],j} - f^{-1}\left(f\left(\boldsymbol{x}_{[i]}, \boldsymbol{\theta}_e\right), \boldsymbol{\theta}_d\right)_j \right)^2$$

We call this the **reconstruction loss**

It is an unsupervised loss because we only provide $\boldsymbol{X}$ and not $\boldsymbol{Y}$!

Review

We can use autoencoders for more than compression

Review

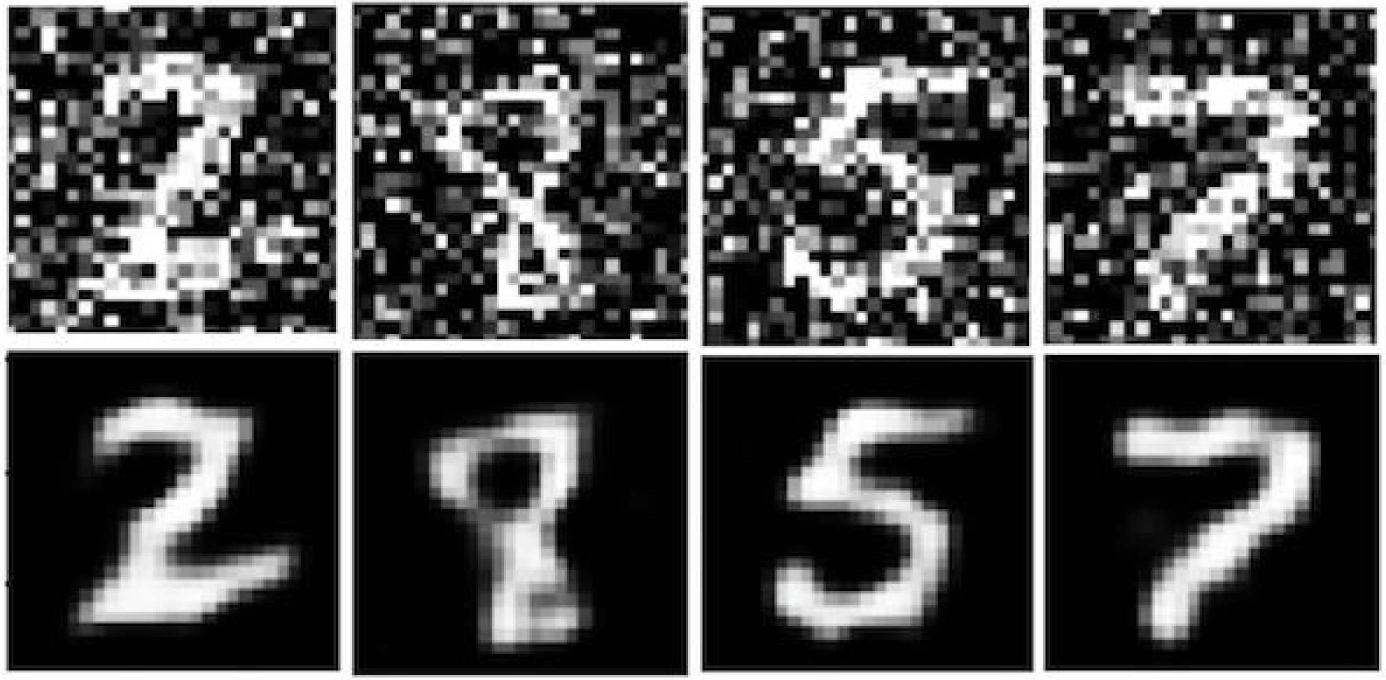
We can use autoencoders for more than compression

We can make **denoising autoencoders** that remove noise

Review

We can use autoencoders for more than compression

We can make **denoising autoencoders** that remove noise



Review

Generate some noise $\quad \varepsilon \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$

Review

Generate some noise	$\boldsymbol{\varepsilon} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$
Add noise to the image	$\boldsymbol{x} + \boldsymbol{\varepsilon}$

Review

Generate some noise $\boldsymbol{\varepsilon} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$

Add noise to the image $\boldsymbol{x} + \boldsymbol{\varepsilon}$

Original loss $\mathcal{L}(\boldsymbol{X}, \boldsymbol{\theta}) = \sum_{i=1}^{n} \sum_{j=1}^{d_x} \left(x_{[i],j} - f^{-1}\left(f\left(\boldsymbol{x}_{[i]}, \boldsymbol{\theta}_e\right), \boldsymbol{\theta}_d\right)_j \right)^2$

Review

Generate some noise $\boldsymbol{\varepsilon} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$

Add noise to the image $\boldsymbol{x} + \boldsymbol{\varepsilon}$

Original loss $$\mathcal{L}(\boldsymbol{X}, \boldsymbol{\theta}) = \sum_{i=1}^{n} \sum_{j=1}^{d_x} \left(x_{[i],j} - f^{-1}\left(f\left(\boldsymbol{x}_{[i]}, \boldsymbol{\theta}_e \right), \boldsymbol{\theta}_d \right)_j \right)^2$$

Denoising loss $$\mathcal{L}(\boldsymbol{X}, \boldsymbol{\theta}) = \sum_{i=1}^{n} \sum_{j=1}^{d_x} \left(x_{[i],j} - f^{-1}\left(f\left(\boldsymbol{x}_{[i]} + \boldsymbol{\varepsilon}, \boldsymbol{\theta}_e \right), \boldsymbol{\theta}_d \right)_j \right)^2$$

Review

Generate some noise $\boldsymbol{\varepsilon} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$

Add noise to the image $\boldsymbol{x} + \boldsymbol{\varepsilon}$

Original loss $\mathcal{L}(\boldsymbol{X}, \boldsymbol{\theta}) = \sum_{i=1}^{n} \sum_{j=1}^{d_x} \left(x_{[i],j} - f^{-1}\left(f\left(\boldsymbol{x}_{[i]}, \boldsymbol{\theta}_e\right), \boldsymbol{\theta}_d\right)_j \right)^2$

Denoising loss $\mathcal{L}(\boldsymbol{X}, \boldsymbol{\theta}) = \sum_{i=1}^{n} \sum_{j=1}^{d_x} \left(x_{[i],j} - f^{-1}\left(f\left(\boldsymbol{x}_{[i]} + \boldsymbol{\varepsilon}, \boldsymbol{\theta}_e\right), \boldsymbol{\theta}_d\right)_j \right)^2$

Autoencoder will learn to remove noise when reconstructing image

Review

Then, we discussed variational autoencoders

Review

Then, we discussed variational autoencoders

However, to save time we will review these and write code next time

Agenda

1. **Review**
2. Quiz
3. Introduction to Prof. Li
4. Graph Neural Networks

Agenda

1. Review
2. **Quiz**
3. Introduction to Prof. Li
4. Graph Neural Networks

Quiz

All phones and laptops away or face down

Quiz

All phones and laptops away or face down

No cheating, you will get 0 and I tell Dean

Quiz

All phones and laptops away or face down

No cheating, you will get 0 and I tell Dean

I will hand out the quizzes face down

Quiz

All phones and laptops away or face down

No cheating, you will get 0 and I tell Dean

I will hand out the quizzes face down

Turn them over when I say start

Quiz

All phones and laptops away or face down

No cheating, you will get 0 and I tell Dean

I will hand out the quizzes face down

Turn them over when I say start

You will have 20 minutes

Quiz

All phones and laptops away or face down

No cheating, you will get 0 and I tell Dean

I will hand out the quizzes face down

Turn them over when I say start

You will have 20 minutes

Good luck!

Agenda

1. Review
2. **Quiz**
3. Introduction to Prof. Li
4. Graph Neural Networks

Agenda

1. Review
2. Quiz
3. **Introduction to Prof. Li**
4. Graph Neural Networks

Introduction to Prof. Li

Prof. Qingbiao Li did his PhD at Cambridge and a postdoc at Oxford

Introduction to Prof. Li

Prof. Qingbiao Li did his PhD at Cambridge and a postdoc at Oxford

He has two Master's degrees, one from University of Edinburgh and one from Imperial College London

Introduction to Prof. Li

Prof. Qingbiao Li did his PhD at Cambridge and a postdoc at Oxford

He has two Master's degrees, one from University of Edinburgh and one from Imperial College London

His interests are in deep learning for medicine and multirobot system

Introduction to Prof. Li

Prof. Qingbiao Li did his PhD at Cambridge and a postdoc at Oxford

He has two Master's degrees, one from University of Edinburgh and one from Imperial College London

His interests are in deep learning for medicine and multirobot system

Today, he will teach you about **Graph Neural Networks**