MAC5921 – Deep Learning

Aula 15 – 17/10/2023

Wrapping up?

Nina S. T. Hirata

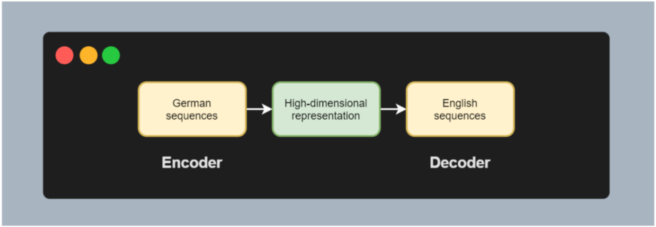
Primeira parte desta aula
embalado pelos LLMs discutidos na aula anterior

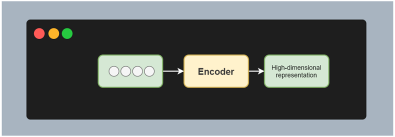
Paradigmas de treinamento Multi-GPU (by Diogo)

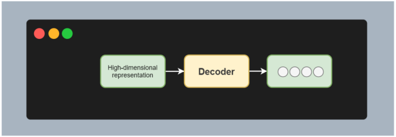
Sobre a perspectiva da realização do ajuste fino de um modelo de linguagem gigante utilizando GPUs com pouca memória

Problemas Seq2seq – sequence to sequence

Atualmente, usa-se encoder-decoder models







Transformer (Vaswani, https://arxiv.org/abs/1706.03762) é seq2seq

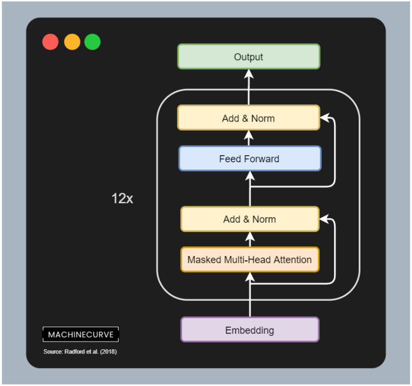
Foi proposto para o problema de tradução

Depois, evoluiu para os chamados *Language models*

Autoregressive models – Natural language generation

use all previous predictions for generating the next one, in a cyclical fashion

Caso do GPT – inspirado no lado *decoder* do *transformer*



Autoencoding models – representation learning

learn an encoded representation of the inputs by corrupting inputs and generating the original variants

Caso do BERT – inspirado no lado *encoder* do *transformer*

Diferença entre **Autoregressive models** e **Autoencoding models**

está basicamente na forma como são treinados

Abordagens gerativas × discriminativas

$$\boxed{P(\mathbf{y} \mid \mathbf{x})} = \frac{\boxed{P(\mathbf{x} \mid \mathbf{y})\, P(\mathbf{y})}}{P(\mathbf{x})} = \frac{P(\mathbf{x}, \mathbf{y})}{P(\mathbf{x})}$$

Discriminativa   Gerativa

Discriminativa: aprende $P(\mathbf{y} \mid \mathbf{x})$

Generativa: aprende $P(\mathbf{x}, \mathbf{y})$

Generative Models – *quick intro* de alguns métodos

Gaussian Mixture Models (GMMs)
Hidden Markov Models (HMMs)
Recurrent Neural Networks (RNNs)

Variational Autoencoders (VAEs) https://arxiv.org/abs/1312.6114

Generative Adversarial Network (GAN), https://arxiv.org/abs/1406.2661

Flow-based models

Diffusion models

GPT (Generative Pre-trained Transformer) models

Variational Autoencoders (VAEs)

Usam arquitetura *encoder-decoder* para mapear o dado de entrada para um espaço latente, para em seguida reconstrui-la na saída

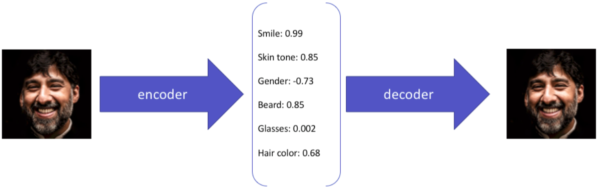
O treinamento envolve otimização de parâmetros do modelo de forma a minimizar uma função de perda que envolve **erro de reconstrução** e **regularização da distribuição no espaço latente**

https://jaan.io/what-is-variational-autoencoder-vae-tutorial/

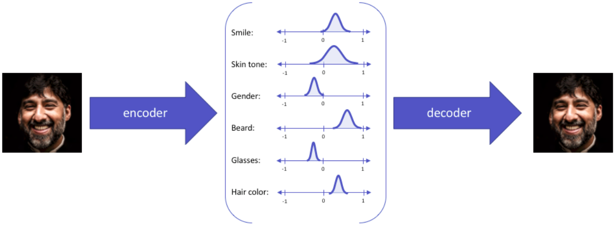
The *loss function* of the variational autoencoder is the negative log-likelihood with a regularizer. Because there are no global representations that are shared by all datapoints, we can decompose the loss function into only terms that depend on a single datapoint l_i. The total loss is then $\sum_{i=1}^{N} l_i$ for N total datapoints. The loss function l_i for datapoint x_i is:

$$l_i(\theta, \phi) = -\mathbb{E}_{z \sim q_\theta(z|x_i)}[\log p_\phi(x_i \mid z)] + \mathbb{KL}(q_\theta(z \mid x_i) \mid\mid p(z))$$

x entrada, z representação latente



Latent attributes



Latent attributes

https://www.v7labs.com/blog/autoencoders-guide

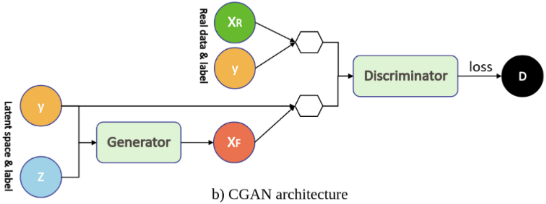
GAN (https://arxiv.org/abs/1406.2661)

a) GAN architecture

https://mc.ai/a-tutorial-on-conditional-generative-adversarial-nets-keras-implementation/

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}[\log(1 - D(\mathbf{z}))]$$

Conditional GAN (GAN) (https://arxiv.org/pdf/1411.1784.pdf)

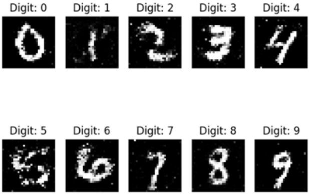


b) CGAN architecture

https://mc.ai/a-tutorial-on-conditional-generative-adversarial-nets-keras-implementation/

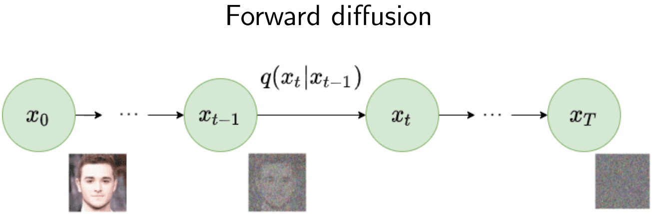
$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[\log D(\mathbf{x}|\mathbf{y})] + \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})}[\log(1 - D(\mathbf{z}|\mathbf{y}))]$$

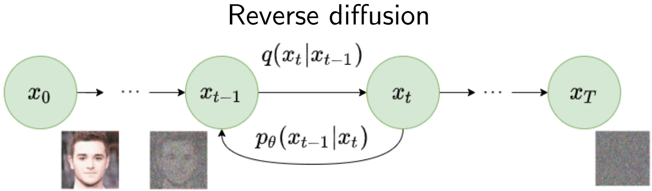
(a) GAN

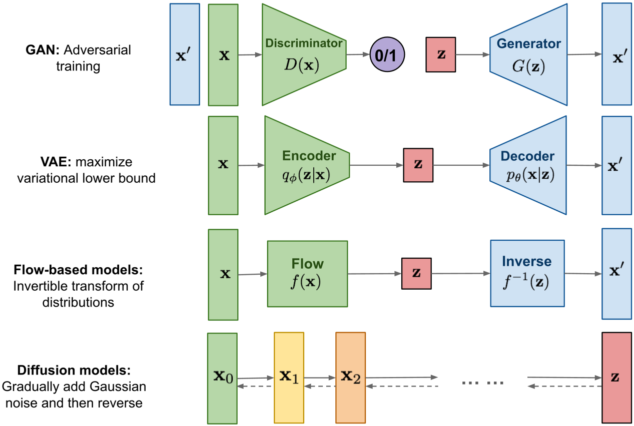

Digit: 0
Digit: 1
Digit: 2
Digit: 3
Digit: 4
Digit: 5
Digit: 6
Digit: 7
Digit: 8
Digit: 9

(b) CGAN

Diffusion models







https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

Modelos gerativos

Detalhes (talvez) em alguma das próximas aulas?

Trazendo outra perspectiva do que vimos até agora

Tipos de redes neurais que vimos

em termos de estrutura dos dados

1. **Redes neurais convencionais** (*fully connected*)
Dados tabulares

2. **Redes neurais convolucionais** (CNN) – adjacência local
Relação espacial/posicional: pixels vizinhos em imagens, elementos adjacentes em uma sequência

3. **Redes neurais recorrentes** (RNN) – recorrência
Relação de dependência temporal/sequencial

4. **Transformers** – mecanismos de attention
Atenção global, contexto

5. **Graph Neural Networks** – mecanismos de *message passing*
Relações não-regulares: redes complexas / redes sociais (grafos)

Graph Neural Networks

Understanding Convolutions on Graphs

https://distill.pub/2021/understanding-gnns/

A Gentle Introduction to Graph Neural Networks

https://distill.pub/2021/gnn-intro/

Bastante provável que este tópico não seja coberto neste curso

Próxima aula?