Deep Learning Exam
WS 2023/24

Dr. Ozan Özdenizci

4th March 2024

First of all, I would like you to know that these notes were written 10 minutes after the exam finished, so they are as fresh as possible.

1 Architecture

Here are two network models: Model A in Fig. 1 and Model B in Fig. 2

1.1 a

Give an example when you would choose Model A over Model B. Write a loss function.

1.2 b

Give an example when would you choose Model B over Model A. Also specify the loss function.

2 Neural Networks

2.1 a

What is the Momentum and how do we apply it for the gradient descent, please give the formula. What will change if we apply Nesterov Momentum?

2.2 b

What is adversarial training?(question was unclear if they mentioned the regularization technique or DDPM, but I suppose it was about regularization)

3 RNN

3.1 a

There is a RNN network in Figure 3 with only one output value. Give the supervised problem where we can apply it.

3.2 b

Why exactly are we using Back Propagation Through Time(BPTT) in RNNs, please explain the problem with exploding/vanishing gradients and how is it resolved.

4 Deep generative networks

4.1 a

Please explain the simplified versions of algorithms in Figure 4for training DDPM, especially the 5th raw of Algorithm 1 and 4th raw of Algorithm 2.

4.2 b

How to make arithmetic operations in GANs

4.3 c

Explain generative pretrain transformer(GPT)

5 Practical part

There is a function:

$y = \mathrm{ax}^2 + bx + c$

The loss function is

$\mathrm{L} = \sum_i e^{(t_i - y(x_i))^2}$

Please give the update rule for parameters a, b, and c for the stochastic gradient descent with batch size 1.

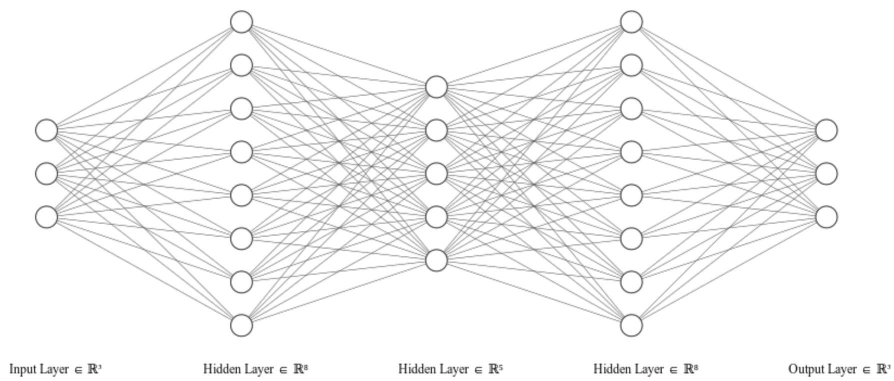

Figure 1: Model A

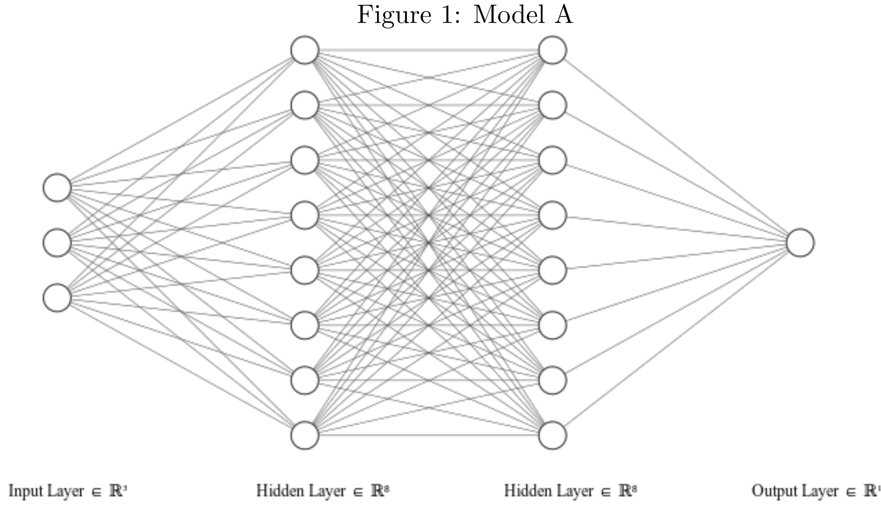

Figure 2: Model B

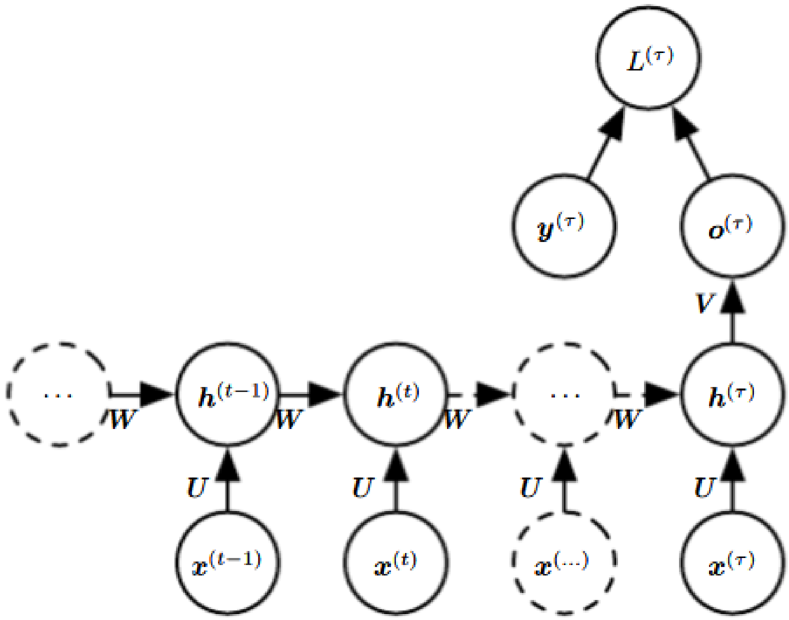

Figure 3: RNN

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \mathrm{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

Figure 4: DDMP algorithms