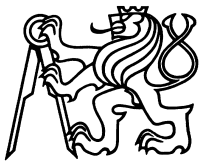
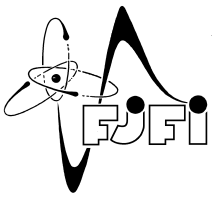


CZECH TECHNICAL UNIVERSITY IN PRAGUE
Faculty of Nuclear Sciences and Physical Engineering



Generative and discriminative models for set data

Generativní a diskriminativní modely pro množinová data

Research Project

Author: **Bc. Jakub Bureš**

Supervisor: **doc. Ing. Václav Šmídl, Ph.D.**

Language advisor: **Mgr.**

Academic year: 2020/2021

Acknowledgment:
I would like to thank ..... for (his/her expert guidance) and express my gratitude to ..... for (his/her language assistance).

Author's declaration:
I declare that this Bachelor's Degree Project is entirely my own work and I have listed all the used sources in the bibliography.

Prague, June 1, 2021 Jméno Autora

Název práce:

Název práce

Autor: Jméno Autora

Obor: Celý název oboru (nikoliv zkratka)

Zaměření: Celý název zaměření (Pokud obor neobsahuje zaměření, tuto řádku odstranit.)

Druh práce: Bakalářská práce

Vedoucí práce: prof. Ing. Jméno Školitele, DrSc., pracoviště školitele (název instituce, fakulty, katedry...)

Konzultant: doc. RNDr. Jméno Konzultanta, CSc., pracoviště konzultanta. Pouze pokud konzultant byl jmenován.

Abstrakt: Abstrakt max. na 10 řádků.

Klíčová slova: klíčová slova (nebo výrazy) seřazená podle abecedy a oddělená čárkou

Title:

Title of the Work

Author: Author's Name

Abstract: Max. 10 lines of English abstract text.

Key words: keywords in alphabetical order separated by commas

Contents

Introduction

Paragraphs of the Introduction...

Chapter 1

Multiple Instance Learning

In standard machine learning problems each sample is represented by a fixed vector $\mathbf{x}$ of values, nevertheless in multiple instance learning (MIL) we deal with samples which are represented by a set of vectors. These vectors are called *instances* and come from an instance space $\mathcal{X}$. Sets of these instances are called *bags* and come from bag space $\mathcal{B} = \mathcal{P}_F(\mathcal{X})$, where $\mathcal{P}_F(\mathcal{X})$ denotes all finite subsets of $\mathcal{X}$. With this in mind, we can easily write down any bag b as $b = \{\mathbf{x} \in \mathcal{X}\}_{\mathbf{x}\in b}$. Each bag can be arbitrarily large or empty thus the size of bag $|b| \in \mathbb{N}_0$. There may exist intrinsic labeling of instances, but we are only interested in labeling at the bag levels. Bag labels come from a finite set C and what we want in MIL is learning a predictor in the form $f : \mathcal{B}(\mathcal{X}) \to C$ which can also be rewritten in the form $f(\{\mathbf{x}\}_{\mathbf{x}\in b})$. We consider supervised setting, in which each sample of the dataset is attributed a label. We can denote available data by $\mathcal{D} = \{(b_i, y_i) \in \mathcal{B} \times C \mid i \in \{1, 2, \ldots, |\mathcal{D}|\}\}$, where $|\mathcal{D}|$ denotes size of $\mathcal{D}$.

1.1 Cross-Validation

Probably the simplest and most widely used method for estimating prediction error is *cross-validation*. This method directly estimates the expected extra-sample error

$$\mathrm{Err} = \mathbb{E}\left[L\left(Y, \hat{f}(X)\right)\right], \tag{1.1}$$

the average generalization error when the method $\hat{f}(X)$ is applied to an independent test sample form the joint distribution of X and Y- As mentioned earlier, we might hope that cross-validation estimates the conditional error, with the training set $\mathcal{T}$ held fixed. But cross-validation typically estimates well only the expected prediction error.

K-Fold Cross-Validation

Ideally, if we had enough data, we would set aside a validation set and use it to assess the performance of our prediction model. Since data are often scarce, this is usually not possible. To finesse the problem, K-fold cross-validation uses part of the available data to fit the model, and a different part to test it. We split the data into K roughly equal-sized parts; for example, when $K = 5$, the scenario could look like this

train	train	validation	train	train

For the kth part (third above), we fit the model to the other $K - 1$ parts of the data, and calculate the prediction error of the fitted model when predicting the kth part of the data. We do this for $k = \{1, 2, \ldots, K\}$ and combine the K estimates of prediction error.
Here are more details. Let $\kappa : \{1, \ldots, N\} \to \{1, \ldots, K\}$ be an indexing function that indicates the partition to which observation i is allocated by the randomization. Denote by $\hat{f}^{-k}(x)$ the fitted function, computed with the kth part of the data removed. Then the cross-validation estimate of prediction error is

$$\mathrm{CV}\left(\hat{f}\right) = \frac{1}{N}\sum_{i=1}^{N} L\left(y_i, \hat{f}^{-\kappa(i)}(x_i)\right). \tag{1.2}$$

Typical choices of K are 5 or 10 and even case $K = N$, which is known as *leave-one-out* cross-validation.

1.2 Experiment with mill datasets

Suppose we have two classes 0 and 1 (known as binary classification), which means that bags are labeled either as 0 or 1. What happens, if we have many more bags, for example, labeled as class 1? This situation is very common in anomaly detection, where known anomalies are quite rare.
Let's assume train set is composed of 80% bags labeled as 1 and 5% bags labeled as 0, all randomly chosen. Test set is composed of 20% bags labeled as 1 and 95% labeled as 0, in other words it is complement of train set. Validation set is very similar to train set in terms of ratios, it contains 20% bags labeled as 1 and 2% bags labeled as 0. Train set is used to train our model, after that we evaluate loss function of the model with help of validation a test set, where number of dense layers is our hyperparameter. The purpose of this simulation is to find number of dense layer in which the loss is minimal and compare these 2 values. This experiment was performed 5 times then results were averaged, totally on 6 different datasets.

As we can see in Figure 1.1, the loss evaluated on different sets varies therefore it is likely to make mistakes when choosing our hyperparameter if we don't have enough input data.

1.3 Experiment with Point Cloud Mnist

We applied previous method on Point Cloud Mnist 2D (PCM) dataset. However, we needed to make a little adjustment to this dataset, because we perform the binary classification. Originally, PCM has tottaly 10 labels (labels are numbers 0-9), so we separated arbitrarily 8 of them to obtain just 2 classes. In addition, we randomly split data into bags.
The advantage of PCM is that we can split the data into bags in many ways, e.g. 0&1 , 2&3, 5&7,... we can even take 0&1 as first class and, for example, 2&3 as a second class.

labels	difference
0&1	0.81
2&3	2.07
4&5	2.05
6&7	1.99
8&9	1.91

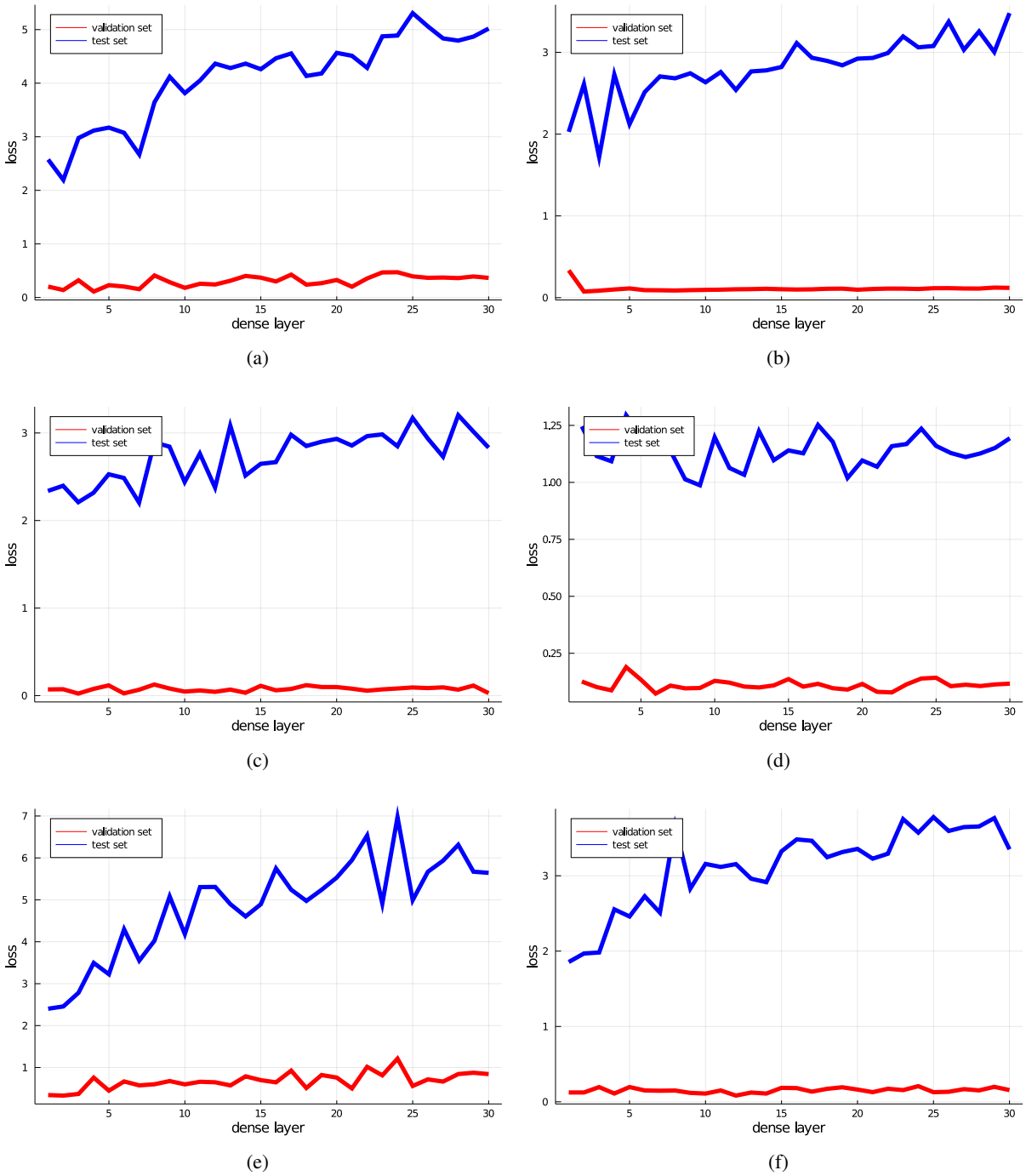


Figure 1.1: Evaluation of loss function with the use of validation and test set on different MILL datasets.

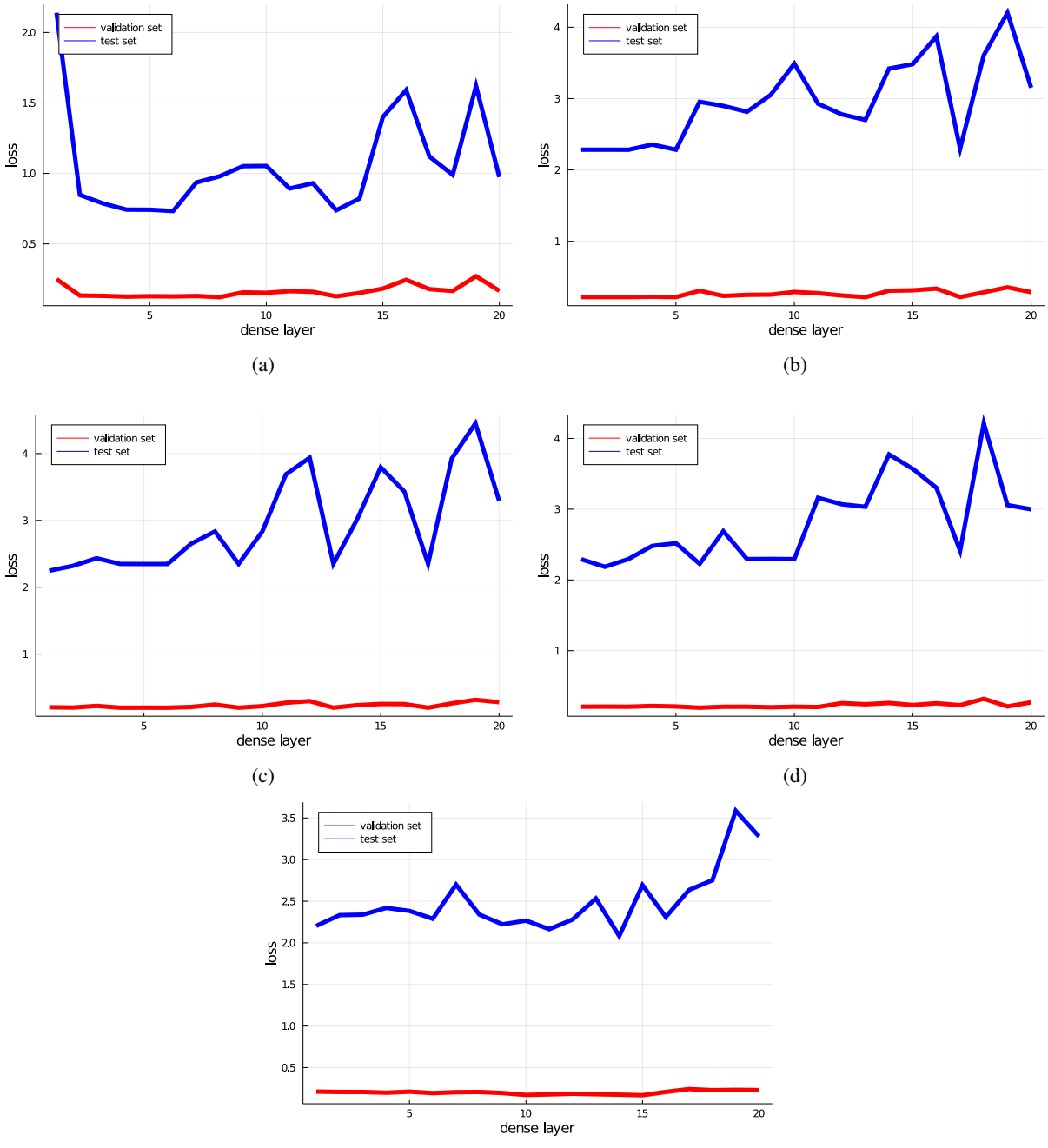


Figure 1.2: Evaluation of loss function with the use of validation and test set on Point Cloud Mnist

Conclusion

Text of the conclusion...

Bibliography

[1] S. Allen, J. W. Cahn: *A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening*. Acta Metall., 27:1084-1095, 1979.

[2] G. Ballabio et al.: *High Performance Systems User Guide*. High Performance Systems Department, CINECA, Bologna, 2005. `www.cineca.it`

[3] J. Becker, T. Preusser, M. Rumpf: *PDE methods in flow simulation post processing*. Computing and Visualization in Science, 3(3):159-167, 2000.