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. Abstrakt max. na 10 řádků. Abstrakt max. na 10 řádků. Abstrakt max. na 10 řádků. Abstrakt max. na 10 řádků. Abstrakt max. na 10 řádků. Abstrakt max. na 10 řádků. Abstrakt max. na 10 řádků. Abstrakt max. na 10 řádků. Abstrakt max. na 10 řádků. Abstrakt max. na 10 řádků. Abstrakt max. na 10 řádků. Abstrakt max. na 10 řádků. Abstrakt max. na 10 řádků. Abstrakt max. na 10 řádků. Abstrakt max. na 10 řádků. Abstrakt max. na 10 řádků. Abstrakt max. na 10 řádků. Abstrakt max. na 10 řádků. Abstrakt max. na 10 řádků. Abstrakt max. na 10 řádků. Abstrakt max. na 10 řádků. Abstrakt max. na 10 řádků. Abstrakt max. na 10 řádků. Abstrakt max. na 10 řádků. Abstrakt max. na 10 řádků. Abstrakt max. na 10 řádků. Abstrakt max. na 10 řádků. 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. Max. 10 lines of English abstract text. Max. 10 lines of English abstract text. Max. 10 lines of English abstract text. Max. 10 lines of English abstract text. Max. 10 lines of English abstract text. Max. 10 lines of English abstract text. Max. 10 lines of English abstract text. Max. 10 lines of English abstract text. Max. 10 lines of English abstract text. Max. 10 lines of English abstract text. Max. 10 lines of English abstract text. Max. 10 lines of English abstract text. Max. 10 lines of English abstract text. Max. 10 lines of English abstract text. Max. 10 lines of English abstract text. Max. 10 lines of English abstract text. Max. 10 lines of English abstract text. Max. 10 lines of English abstract text. Max. 10 lines of English abstract text. Max. 10 lines of English abstract text. Max. 10 lines of English abstract text. Max.

10 lines of English abstract text. Max. 10 lines of English abstract text. Max. 10 lines of English abstract text.

Key words: keywords in alphabetical order separated by commas

Contents

Introduction

Paragraphs of the Introduction…

Chapter 1

Theoretical introduction

1.1 Terminology

At the beginning of this work and for avoiding confusion, it is appropriate to clarify the terminology.
The notation for random variables via uppercase letters, for example X, Y, Z, is very common and widely used. The realization of a random variable,also known as observed value, or simply observation, will be denoted by correspodning lowercase letters x, y. Most of the time, input variable will be denoted by the symbol x and output variable will be denoted by the symbol y. Bold symbols $\boldsymbol{X}$, $\boldsymbol{Y}, \boldsymbol{x}$, $\boldsymbol{y}$ will be used to distinguish vectors from scalars.

Remark 1.1 (Engineering notation). To simplify the notation, we will not distinguish

Usual goal (learning task) is to make a good prediction of the output $\boldsymbol{y}$, denoted by the symbol $\hat{\boldsymbol{y}}$, with given input $\boldsymbol{x}$. This prediction is obtained through learning a model $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = f(\boldsymbol{\theta}, \boldsymbol{x})$ that minimizes a loss function (also known as the error function) $\mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y})$, where $\boldsymbol{\theta}$ are the parameters of the model.
To contruct this prediction we need data, hence it is supposed that we have available set of observations $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, eventually, this may in fact be $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N}$, where $\boldsymbol{x}_i \in \mathbb{R}^D$ and $\boldsymbol{y}_i \in \mathbb{R}^C \quad \forall i = 1, \dots, N$, known as training data. Observations are considered to be independent and indentically distributed, often abbreviated as i.i.d.

1.2 Bayesian Inference

The Bayesian methodology is a well established approach to statistical inference and became very important technique in statistics and data analysis. As its name suggests, Bayesian statistics is based on application of Bayes' rule. In this chapter we briefly review basic concept of this approach.
Let the measured data be denoted by $\mathcal{D}$, defined according to previous section 1.1. A parametric probabilistic model of the data $\mathcal{D}$ is given by the probability density function $p(\mathcal{D}|\boldsymbol{\theta})$, where $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subset \mathbb{R}^s$ denotes parameters of the model. The main idea behind Bayesian theory is the

treatment of the uknown parameters $\boldsymbol{\theta}$ as a random variable. Bayes' rule is applied to infer model parameters $\boldsymbol{\theta}$, therefore

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\boldsymbol{\theta},\mathcal{D})}{p(\mathcal{D})} = \frac{p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int_{\Theta} p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}}. \tag{1.1}$$

Since $p(\mathcal{D})$ is just the normalization constant, Equation (1.1) is often simplified to

$$p(\boldsymbol{\theta}|\mathcal{D}) \propto p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta}). \tag{1.2}$$

Symbol $\propto$ means equal up to the normalization constant. The term $p(\boldsymbol{\theta}|\mathcal{D})$ is known as the *posterior* distribution, $p(\mathcal{D}|\boldsymbol{\theta})$ as the *observation model*, and $p(\boldsymbol{\theta})$ is called the *prior* distribution of the $\boldsymbol{\theta}$. Note that evaluation of the normalization constant can be computionally expensive, in higher dimension even intractable.

Popular choices for an optimal value of the point estimate are:

1. Maximum A posteriori estimate (MAP)

$$\widehat{\boldsymbol{\theta}}_{\mathrm{MAP}} = \operatorname*{argmax}_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\mathcal{D}) \tag{1.3}$$

   This method estimates $\boldsymbol{\theta}$ as the mode of the posterior distribution. It appears to be computionally attractive, as it is not necessary to evaluate the normalization constant.

2. Mean or expected value

$$\widehat{\boldsymbol{\theta}}_{\mathrm{B}} = \int_{\Theta} \boldsymbol{\theta}\, p(\boldsymbol{\theta}|\mathcal{D})\mathrm{d}\boldsymbol{\theta} = \mathbb{E}_{\boldsymbol{\theta}\sim p(\boldsymbol{\theta}|\mathcal{D})}\left[\boldsymbol{\theta}\right] \tag{1.4}$$

   Mean value, unlike MAP estimate, may be very expensive to compute because of the required integration. This may lead to further approximations such as EM algorithm [].

1.2.1 Choice of prior distribution

For the posterior computation, it is necessary to specify the prior distribution $p(\boldsymbol{\theta})$, unfortunatelly, this might not be easily determined. This can be achieved via knowledge of previous models, expert knowledge, their combination or even uncertainty about $\boldsymbol{\theta}$ can be viable option.

There is also many practical aspects of priors:

- Regularization - supplementing the data, if there is insufficient data or poorly defined model.

- Imposing various restrictions on the parameters $\boldsymbol{\theta}$ reflecting physical constraints. The choice of prior distribution with bounded support will result to posterior distribution with bounded support as well.

- Non–informative prior - if the data are informative enough to make a prediction, it is proposed to choose a prior with minimal impact on the posterior distribution, such as uniform distribution. However, typical choices of non–informative priors are so–called *Jeffreys priors* [].

1.2.2 Prediction

We are not usually interested in the value of $\widehat{\boldsymbol{\theta}}$ itself but rather, once the model is estimated, we are intersted in making prediction of output variable $\boldsymbol{y}^\star$ for the new input variable $\boldsymbol{x}^\star$. Note that the symbol $\mathcal{D}$ contains all of the previous given data $\boldsymbol{x}$ and $\boldsymbol{y}$. The posterior predictive distribution is then determined by distribution of the $\boldsymbol{y}^\star$, marginalized over the posterior

$$p(\boldsymbol{y}^\star|\boldsymbol{x}^\star,\mathcal{D}) = \int_{\boldsymbol{\Theta}} p(\boldsymbol{y}^\star|\boldsymbol{x}^\star,\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})\mathrm{d}\boldsymbol{\theta}. \tag{1.5}$$

When the distribution $p(\boldsymbol{\theta}|\mathcal{D})$ is not available, we have to approximate leveraging the Dirac delta function $\delta(x)$

$$p(\boldsymbol{y}^\star|\boldsymbol{x}^\star,\mathcal{D}) = \int_{\boldsymbol{\Theta}} p(\boldsymbol{y}^\star|\boldsymbol{x}^\star,\boldsymbol{\theta})\delta(\boldsymbol{\theta}-\widehat{\boldsymbol{\theta}})\mathrm{d}\boldsymbol{\theta} = p(\boldsymbol{y}^\star|\boldsymbol{x}^\star,\widehat{\boldsymbol{\theta}}), \tag{1.6}$$

causing an error. In typical MAP, this is known as *over–fitting*. Prediction error is then defined by a loss function measuring errors between $\boldsymbol{y}$ and $\widehat{\boldsymbol{y}}$

$$\mathrm{Err} = \mathcal{L}\left(\boldsymbol{y},\widehat{\boldsymbol{y}}\right), \tag{1.7}$$

where $\widehat{\boldsymbol{y}} = \widehat{f}_{\boldsymbol{\theta}}\left(\boldsymbol{x}\right) = f(\widehat{\boldsymbol{\theta}},\boldsymbol{x})$. As an example, we can mention couple of typically used loss functions

$$\mathcal{L}\left(\boldsymbol{y},\widehat{\boldsymbol{y}}\right) = \begin{cases} \left\|\boldsymbol{y}-\widehat{\boldsymbol{y}}\right\|_2^2 & \text{squared error} \\ \left\|\boldsymbol{y}-\widehat{\boldsymbol{y}}\right\| & \text{absolute error} \end{cases}. \tag{1.8}$$

We shall also discuss the problem of the model complexity. Consider a polynomial regression problem, where the model is defined by

$$f_{\boldsymbol{\theta}}(x) = \sum_{i=0}^{s-1}\theta_i x^i. \tag{1.9}$$

Here, model complexity is very intuitive as it is just the order of the polynomial, $s-1$. Smaller orders of the polynomial (may) give rather poor fits to the data in contrast to much higher order polynomial giving an excellent fit. Such polynomial passes exactly through each datapoint, however, oscilates wildly and gives poor prediction for new input variable $x^\star$.

To obtain some quantitative insight into dependance of the generalization performance on model complexity, consider separate test set of data (testing data) used to asses the performance of the model. In general, prediction error evaluated on the training data for increasing model complexity approaches zero. On the other hand, prediction error evaluated on the testing data for increasing model complexity is (from certain point) increasing as well. The typical scenario is illustrated in Figure 1.1.

1.3 Cross-Validation

The simplest and most widely used method for estimating prediction error of the prediction model $\widehat{f}_{\boldsymbol{\theta}}$ is called *cross-validation*. It is used for direct estimating of the expected extra-sample

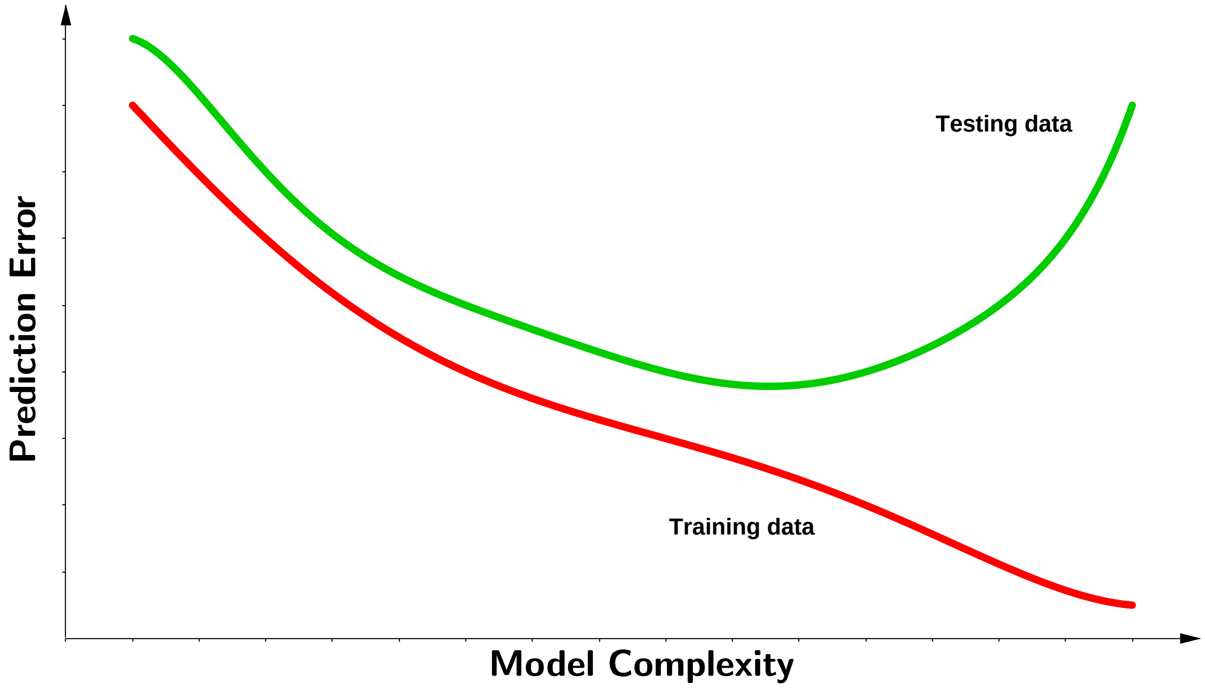


Figure 1.1: Evaluation of prediction error as a function of model complexity.

error

$$\widehat{\mathrm{Err}} = \mathbb{E}\left[\mathcal{L}\left(\boldsymbol{y},\hat{\boldsymbol{y}}\right)\right], \tag{1.10}$$

the measure how accurately is the model able to predict output values for previously unseen data - independent test sample.

1.3.1 K-Fold Cross-Validation

In an ideal case, if we have sufficient number of data, we can set aside a test set and use it to assess the performance of our prediction model. Since data are often scarce, this is usually not possible. Very elegant solution to this problem is via K-fold cross-validation. It uses part of the available data for fitting the model, and a different part for testing. We split the data into K roughly equal-sized parts, for example, when $K = 5$, the scenario is shown in Figure 1.2.

train	train	test	train	train

Figure 1.2: Splitting the data into $K = 5$ roughly equal-sized parts.

For the j^{th} part (third in Figure 1.2), we train the model to the other $K - 1$ parts of the data, and calculate the prediction error of the fitted model when predicting the j^{th} part of the data. We repeat this process for $j = \{1, 2, \ldots, K\}$ and combine the K estimates of prediction error.
Let $\gamma : \{1, \ldots, N\} \to \{1, \ldots, K\}$ be an indexing function that indicates the partition to which observation i is allocated by the randomization. Symbol $\hat{f}_{\boldsymbol{\theta}}^{-j}(\boldsymbol{x})$ denotes the fitted model, computed with the j^{th}th part of the data removed. Then the cross-validation estimate of prediction error is defined by

$$\text{CV}\left(\hat{f}_{\boldsymbol{\theta}}\right) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}\left(\boldsymbol{y}_i, \hat{f}_{\boldsymbol{\theta}}^{-\gamma(i)}(\boldsymbol{x}_i)\right). \tag{1.11}$$

Typical choices of K are 5 or 10 and even case $K = N$ that is known as *leave-one-out* cross–validation. Generally, there is not an universal way of choosing K, since it strongly depends on the available number of data.
The biggest problem of this method is a fact that it is computionally very expensive. For extremely complicated and complex models that are trained for hours or days, is cross–validation inconvenient approach of estimating the prediction error.

Chapter 2

Hybrid Generative and Discriminative models

In this chapter, we review basics of discriminative modeling that was proposed in []. Given a data distribution in the form of probability density $p(\boldsymbol{x})$ and a label distribution with probability density $p(y|\boldsymbol{x})$ containing C categories, thus variable y is now cathegorical taking on C possible values. A classification problem is typically solved using a parametric function $f_{\boldsymbol{\theta}} : \mathbb{R}^D \to \mathbb{R}^C$, where $\boldsymbol{\theta}$ denotes parameters of the model. This function maps each data point $\boldsymbol{x} \in \mathbb{R}^D$ to C real-valued numbers known as logits. They are used to parametrize a categorical distribution using the function

$$q_{\boldsymbol{\theta}}\left(y|\boldsymbol{x}\right) = \frac{\exp\left(f_{\boldsymbol{\theta}}\left(\boldsymbol{x}\right)\left[y\right]\right)}{\sum_{i=1}^{C}\exp\left(f_{\boldsymbol{\theta}}\left(\boldsymbol{x}\right)\left[y_i\right]\right)}, \tag{2.1}$$

which is known as the Softmax function. Note that the convention $f_{\boldsymbol{\theta}}\left(\boldsymbol{x}\right)\left[y\right]$ means the y^{th} element of the $f_\theta\left(\boldsymbol{x}\right)$. For learning $f_{\boldsymbol{\theta}}$ is usually minimized cross-entropy loss function

$$\min_{\boldsymbol{\theta}} -\mathbb{E}_{(x,y)\sim P_{\text{data}}}\left[\log q_{\boldsymbol{\theta}}\left(y|\boldsymbol{x}\right)\right]. \tag{2.2}$$

Rationale for this objective comes from minimizing the Kullback-Leibler divergence with a target distribution $p(y|\boldsymbol{x})$ [**?**]. Kullback-Leibler divergence for two distributions Ψ and Π with corresponding probablity density functions ψ and π is defined as

$$D_{\text{KL}}\left(\Psi||\Pi\right) = \mathbb{E}_{x\sim\Psi}\left[\log\frac{\psi(x)}{\pi(x)}\right], \tag{2.3}$$

which can be further rewritten in the form

$$\mathbb{E}_{x\sim\Psi}\left[\log\frac{\psi(x)}{\pi_{\boldsymbol{\theta}}(x)}\right] = \mathbb{E}_{x\sim\Psi}\left[\log\psi(x)\right] - \mathbb{E}_{x\sim\Psi}\left[\log\pi_{\boldsymbol{\theta}}(x)\right], \tag{2.4}$$

where subscript $\boldsymbol{\theta}$ emphasizes that $\pi_{\boldsymbol{\theta}}(x)$ is our approximative density we get to control. Finally, by minimizing with respect to $\pi_{\boldsymbol{\theta}}(x)$ we obtain

$$\min_{\theta} D_{\text{KL}}\left(\Psi||\Pi\right) = \min_{\theta} -\mathbb{E}_{x\sim\Psi}\left[\log\pi_\theta\left(x\right)\right] \tag{2.5}$$

2.1 Energy-Based Models

Energy-based models (EBM) was introduced here [] and []. EBM assume that probability densities $p(\boldsymbol{x})$ for $\boldsymbol{x} \in \mathbb{R}^D$ can be expressed in the form

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{\exp\left(-E_{\boldsymbol{\theta}}(\boldsymbol{x})\right)}{Z(\boldsymbol{\theta})}, \tag{2.6}$$

where function $E_{\boldsymbol{\theta}} : \mathbb{R}^D \to \mathbb{R}$ is called Energy and maps each datapoint $\boldsymbol{x}$ to a scalar. The denominator $Z(\boldsymbol{\theta})$ is a normalization constant (also known as partition function), thus

$$Z(\boldsymbol{\theta}) = \sum_{\boldsymbol{x}} \exp\left(-E_{\boldsymbol{\theta}}(\boldsymbol{x})\right), \tag{2.7}$$

where the summation is over all datapoints $\boldsymbol{x}$ available. The sum turns into integral for a continuous $\boldsymbol{x}$. Very imporant observations made authors in [?], where they show that classifiers in supervised learning are secretly energy-based models () on $p(\boldsymbol{x}, y)$ and can be expressed as

$$p(\boldsymbol{x}, y) = \frac{\exp\left(f_{\boldsymbol{\theta}}(\boldsymbol{x})[y]\right)}{Z(\boldsymbol{\theta})}. \tag{2.8}$$

It is obvious that $f_{\boldsymbol{\theta}}(\boldsymbol{x})[y] = -E_{\boldsymbol{\theta}}(\boldsymbol{x}, y)$. It could come in handy to have only density model of datapoints $p(\boldsymbol{x})$ without labels. This could be achieved by marginalizing $p(\boldsymbol{x}, y)$ over y

$$p(\boldsymbol{x}) = \frac{\sum_y \exp\left(f_{\boldsymbol{\theta}}(\boldsymbol{x})[y]\right)}{Z(\boldsymbol{\theta})}, \tag{2.9}$$

where energy is given by $E_{\boldsymbol{\theta}}(\boldsymbol{x}) = -\log \sum_y \exp\left(f_{\boldsymbol{\theta}}(\boldsymbol{x})[y]\right)$. Very useful property appears when computing $p(y|\boldsymbol{x})$. We can take advatage of the definition of conditional distribution $p(y|\boldsymbol{x}) = \frac{p(x,y)}{p(x)}$, yielding

$$p_{\boldsymbol{\theta}}(y|\boldsymbol{x}) = \frac{\exp\left(f_{\boldsymbol{\theta}}(\boldsymbol{x})[y]\right)}{\sum_{y'} \exp\left(f_{\boldsymbol{\theta}}(\boldsymbol{x})[y']\right)}. \tag{2.10}$$

Note that the normalization constant $Z(\boldsymbol{\theta})$ canceled out and we ended up with the same function, which was introduced in (2.1). The obvious continuation is that we analogously derive term $p(\boldsymbol{x}|y) = \frac{p(x,y)}{p(y)}$. The denomitator is once again given by marginalizing $p(\boldsymbol{x}, y)$, but this time over $\boldsymbol{x}$.

2.2 Contrastive learning

To briefly introduce contrastive learning [?] let's just mention that it is very usual to optimize an objective, which can be written in the form as follows

$$\min_{\theta} -\mathbb{E}_{p_{\text{data}}(x)}\left[\log \frac{\exp\left(h_\theta(\boldsymbol{x}) \cdot h_\theta(\boldsymbol{x}')\right)}{\sum_{i=1}^{K} \exp\left(h_\theta(\boldsymbol{x}) \cdot h_\theta(\boldsymbol{x}_i)\right)}\right]. \tag{2.11}$$

Function $h_\theta : \mathbb{R}^D \rightarrow \mathbb{R}^H$ maps each data point to a representation space with dimension H,while $\boldsymbol{x}$ and $\boldsymbol{x}'$ are two different augmented views of the same data point. Note that the inner product between two vectors can replaced with any distance metric, for example Euclidean distance.
What this objective does is basically that it tries to maximally distinguish an input $\boldsymbol{x}_i$ from alternative inputs $\boldsymbol{x}'_i$.

2.3 Hybrid Dicriminative and Generative Models, HDGM

In this section will be discussed an approach, how to combine both of the already mentioned models. Authors of the arcticle [?] proposed a solution, however the rationale of this objective originates from [?], where authors show that hybrid models can outperform purely generative or purely discriminative counterparts.
The primary goal is to train a model that can classify $\boldsymbol{x}$ to classes y. In addition, learned models should be capable of out-of-distribution detection and could generate. To achieve these goals, a hybrid model consists of a discriminative conditional and a generative conditional by maximizing the sum of both conditional log-likelihoods

$$\min_{\boldsymbol{\theta}} -\mathbb{E}_{p_{\text{data}}(\boldsymbol{x},y)} \left[\log q_{\boldsymbol{\theta}}\left(y|\boldsymbol{x}\right) + \log q_{\boldsymbol{\theta}}\left(\boldsymbol{x}|y\right)\right], \tag{2.12}$$

where the first term

$$q_{\boldsymbol{\theta}}\left(y|\boldsymbol{x}\right) = \frac{\exp\left(f_{\boldsymbol{\theta}}\left(\boldsymbol{x}\right)[y]\right)}{\sum_{i=1}^{C}\exp\left(f_{\boldsymbol{\theta}}\left(\boldsymbol{x}\right)[y_i]\right)} \tag{2.13}$$

is a standard Softmax neural net classifier (as mentioned in Equation (2.1)) and the second term

$$q_{\boldsymbol{\theta}}\left(\boldsymbol{x}|y\right) = \frac{\exp\left(f_{\boldsymbol{\theta}}\left(\boldsymbol{x}\right)[y]\right)}{\sum_{j=1}^{N}\exp\left(f_{\boldsymbol{\theta}}\left(\boldsymbol{x}_j\right)[y]\right)}. \tag{2.14}$$

This objective has issues with the unknown partition function $\sum_{j=1}^{N}\exp\left(f_{\boldsymbol{\theta}}\left(\boldsymbol{x}_j\right)[y]\right)$, which is often intractable, specifically if the number of datapoints is very high. To overcome this obstacle, it is proposed a approximation with a contrastive loss (2.11)

$$\mathbb{E}_{p_{\text{data}}(\boldsymbol{x},y)}\left[\log q_{\boldsymbol{\theta}}\left(\boldsymbol{x}|y\right)\right] \tag{2.15}$$

$$= \mathbb{E}_{p_{\text{data}}(\boldsymbol{x},y)}\left[\log\frac{\exp\left(f_{\boldsymbol{\theta}}\left(\boldsymbol{x}\right)[y]\right)}{\sum_{\boldsymbol{x}}\exp\left(f_{\boldsymbol{\theta}}\left(\boldsymbol{x}\right)[y]\right)}\right] \tag{2.16}$$

$$\approx \mathbb{E}_{p_{\text{data}}(x,y)}\left[\log\frac{\exp\left(f_{\theta}\left(x\right)[y]\right)}{\sum_{i=1}^{K}\exp\left(f_{\theta}\left(x_i\right)[y]\right)}\right], \tag{2.17}$$

where K denotes the number of normalization samples and it has to be sufficiently large - becoming exact in the limit of summing over all $x \in \mathcal{X}$. Now it is possible to substitute this approximation to Equation (2.12), which gives a hybrid combination of supervised learning and constrastive learning in the form

$$\min_{\theta} -\mathbb{E}_{p_{\text{data}}(x,y)} \left[\alpha \log q_{\theta}(y|x) + (1-\alpha) \log q_{\theta}(x|y)\right] \tag{2.18}$$

$$= \min_{\theta} -\mathbb{E}_{p_{\text{data}}(x,y)} \left[\alpha \log \frac{\exp\left(f_{\theta}(x)[y]\right)}{\sum_{y'} \exp\left(f_{\theta}(x)[y']\right)} + (1-\alpha) \log \frac{\exp\left(f_{\theta}(x)[y]\right)}{\sum_{i=1}^{K} \exp\left(f_{\theta}(x_i)[y]\right)}\right]. \tag{2.19}$$

Parameter α is a weight between $[0, 1]$. It is obvious that in the case of $\alpha = 1$, the objective reduces to the standard cross entropy loss (2.2), while $\alpha = 0$, obejctive is reduced to a case called an end-to-end supervised version of contrastive learning. The choice of parameter α is a decision of the designer, however authors in [?] evaluated many possible variants in experiments and found out that the choice of $\alpha = 0.5$ yields to the highest performance on a classification accuracy.

2.4 Toy problem - Polynomial Regression

We would like to test hybrid discriminative and generative approach on simple example before we head into more difficult cases. The goal is to train a model of the form (2.18) that was derived in previous section
Assume data $\{x, y\}_{i=1}^{N}$, where $x_i, y_i \in \mathbb{R}$, therefore this is only 2-dimensional problem. According to energy–based models, we know that for joint distribution it holds

$$p(x, y) = \frac{\exp\left(f_{\theta}(x)[y]\right)}{Z(\boldsymbol{\theta})}, \tag{2.20}$$

where the model model is given by

$$f_{\boldsymbol{\theta}}(x)[y] = -E_{\boldsymbol{\theta}}(x, y). \tag{2.21}$$

At this point, we would like to transform our problem to the polynomial regression. We need to be aware of the discriminative term in the Equation (2.12), because we do not want to clasify, but we would like to find the best fit to the given data. For this reason, we replace that with the typical regression loss

$$S = S(\boldsymbol{\theta}) = \sum_{k=1}^{N} \left(y_k - \sum_{i=0}^{s-1} \theta_i x_k^i\right)^2. \tag{2.22}$$

Since a joint probability distribution can be break down into parts via chain-rule

$$p(x, y) = p(y, x) = p(y|x) \cdot p(x), \tag{2.23}$$

from polynomial regression we can obtain conditional probability density

$$p(y|x, \boldsymbol{\theta}) = \mathcal{N}\left(\sum_{i=0}^{s-1} \theta_i x^i, \sigma^2\right). \tag{2.24}$$

In this case, we also need to determine prior distribution on x. To keep this example simple, let the distribution takes the form

$$p(x|\tau,\sigma) = \mathcal{N}\left(0,\sigma^2\tau^2\right), \tag{2.25}$$

where σ is known parameter and the choice of parameter τ is based on the fact that we would like to have a non–informative prior, thus τ should be adequately high. It results to

$$p(x,y) = \mathcal{N}\left(0,\sigma^2\tau^2\right)\cdot\mathcal{N}\left(\sum_{i=0}^{s-1}\theta_i x^i,\sigma^2\right) = \frac{1}{2\pi\sigma\tau}\exp\left(-\frac{\left(y-\sum_{i=0}^{s-1}\theta_i x^i\right)^2}{2\sigma^2}-\frac{x^2}{2\sigma^2\tau^2}\right), \tag{2.26}$$

whereas our desirable model is given by

$$f_\theta(x)[y] = -\frac{\left(y-\sum_{i=0}^{s-1}\theta_i x^i\right)^2}{2\sigma^2}-\frac{x^2}{2\sigma^2\tau^2}. \tag{2.27}$$

We can now substite (2.27) and (2.22) into Equation (2.12), yielding

$$\min_{\boldsymbol{\theta}}\left\{\alpha\, S\left(\boldsymbol{\theta}\right) - \mathbb{E}_{p_{\text{data}}(x,y)}\left[(1-\alpha)\log q_\theta\left(x|y\right)\right]\right\} = \tag{2.28}$$

$$\min_{\boldsymbol{\theta}}\left\{\alpha\, S\left(\boldsymbol{\theta}\right) - \mathbb{E}_{p_{\text{data}}(x,y)}\left[(1-\alpha)\log\frac{\exp\left(f_\theta\left(x\right)\left[y\right]\right)}{\sum_{i=1}^{N}\exp\left(f_\theta\left(x_i\right)\left[y\right]\right)}\right]\right\} = \tag{2.29}$$

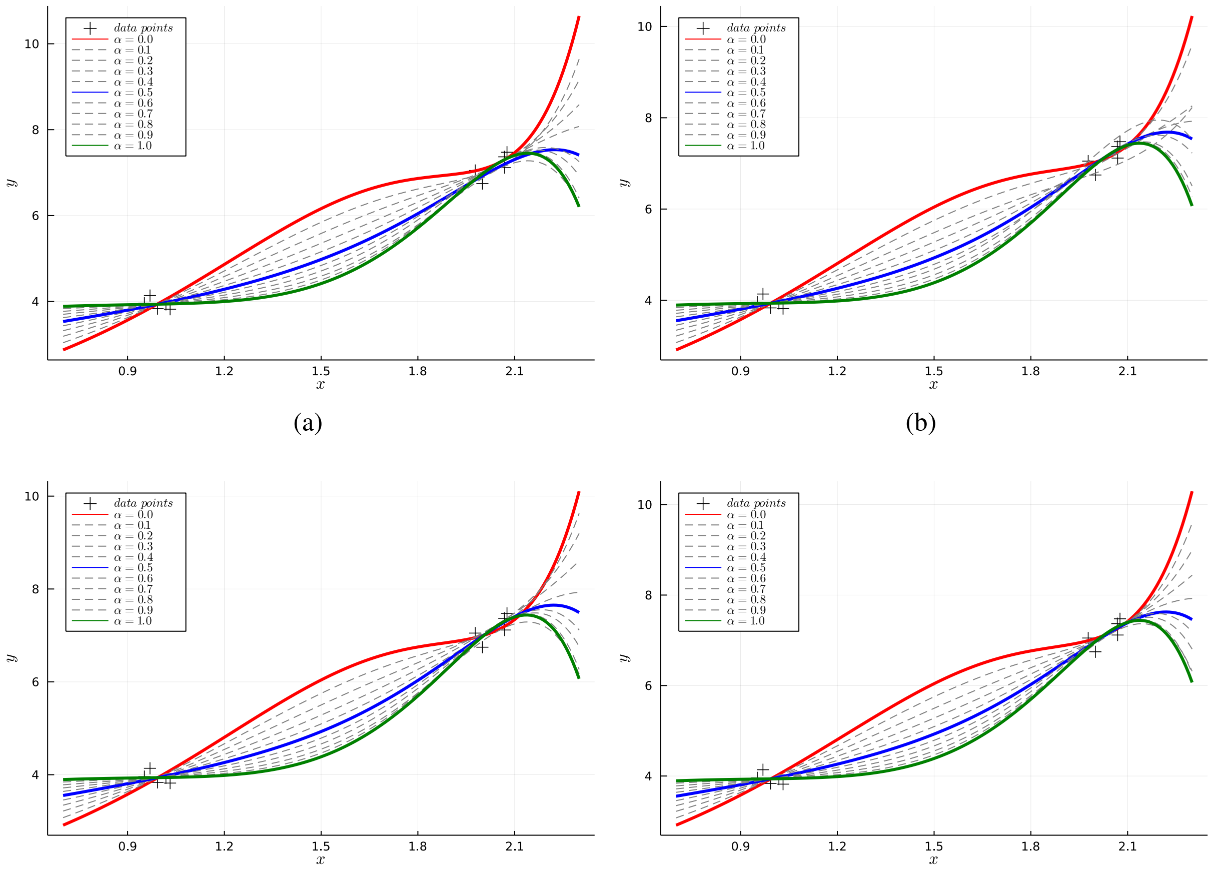
$$\min_{\boldsymbol{\theta}}\left\{\alpha\, S\left(\boldsymbol{\theta}\right) - \mathbb{E}_{p_{\text{data}}(x,y)}\left[(1-\alpha)\log\frac{\exp\left(-\frac{\left(y-\sum_{i=0}^{s-1}\theta_i x^i\right)^2}{2\sigma^2}-\frac{x^2}{2\sigma^2\tau^2}\right)}{\sum_{k=1}^{N}\exp\left(-\frac{\left(y-\sum_{i=0}^{s-1}\theta_i x_k^i\right)^2}{2\sigma^2}-\frac{x_k^2}{2\sigma^2\tau^2}\right)}\right]\right\}. \tag{2.30}$$

Finally, we simplify the generative term $\log q_\theta\left(x|y\right)$ into

$$\log q_\theta\left(x|y\right) = \left(-\frac{\left(y-\sum_{i=0}^{s-1}\theta_i x^i\right)^2}{2\sigma^2}-\frac{x^2}{2\sigma^2\tau^2}\right) - \log\sum_{k=1}^{N}\exp\left(-\frac{\left(y-\sum_{i=0}^{s-1}\theta_i x_k^i\right)^2}{2\sigma^2}-\frac{x_k^2}{2\sigma^2\tau^2}\right) \tag{2.31}$$

2.4.1 Results

We would to test the sensitivity of this approach on the unknown parameter τ and order of the polynomial $s-1$.



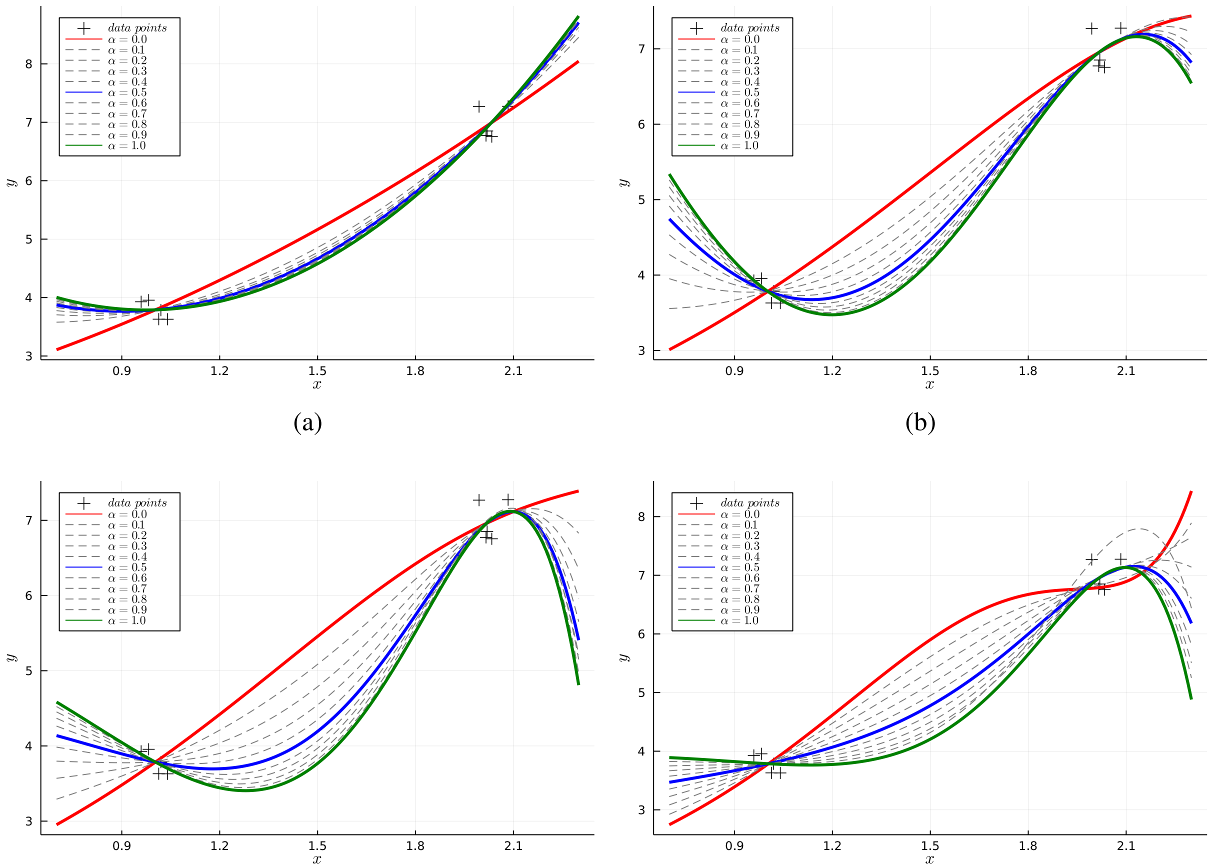
(a)

(b)

(c)

(d)

Figure 2.1: Sensitity to unknown parameter τ.



(a)

(b)

(c)

(d)

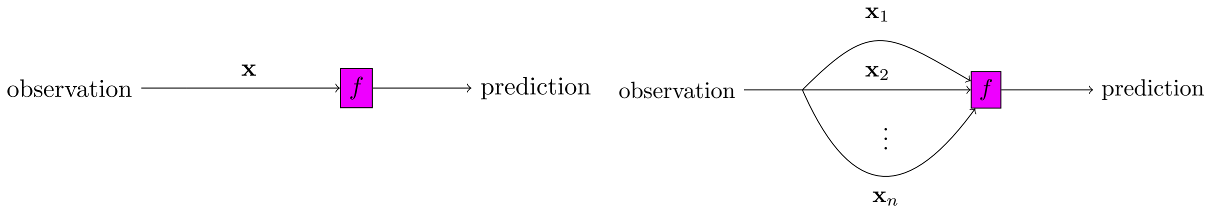
Figure 2.2: Sensitity to choice of the order of polynomial $s - 1$.

Chapter 3

Multiple Instance Learning

3.1 Fundamentals

In standard machine learning problems (ML) each sample is represented by a fixed vector $\boldsymbol{x}$ of observations, however in multiple instance learning (MIL) it is dealt with samples which are represented by a set of vectors. The term multiple instance learning originates from [Dietrich] and in [simon], authors proposed following nomenclature for MIL, which will be gladly used in our work.
These vectors are called *instances* and come from an instance space $\mathcal{X}$, for example $\mathbb{R}^n$. 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 as $b = \{\boldsymbol{x} \in \mathcal{X}\}_{\boldsymbol{x} \in b}$. Each bag b can be arbitrarily large or empty thus the size of bag is defined in the form $|b| \in \mathbb{N}_0$. 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 $f(\{\boldsymbol{x}\}_{\boldsymbol{x} \in b})$. In contrast to ML, where a predictor is learned in the form $f : \mathbb{R}^n \to C$. We consider supervised setting, in which each sample of the dataset is attributed a label. We can denote available data by notation $\mathcal{D} = \left\{ (b_i, y_i) \in \mathcal{B} \times C \mid i \in \left\{1, 2, \ldots, |\mathcal{D}|\right\}\right\}$, where $|\mathcal{D}|$ apparently denotes the size of $\mathcal{D}$.



(a) Standard machine learning

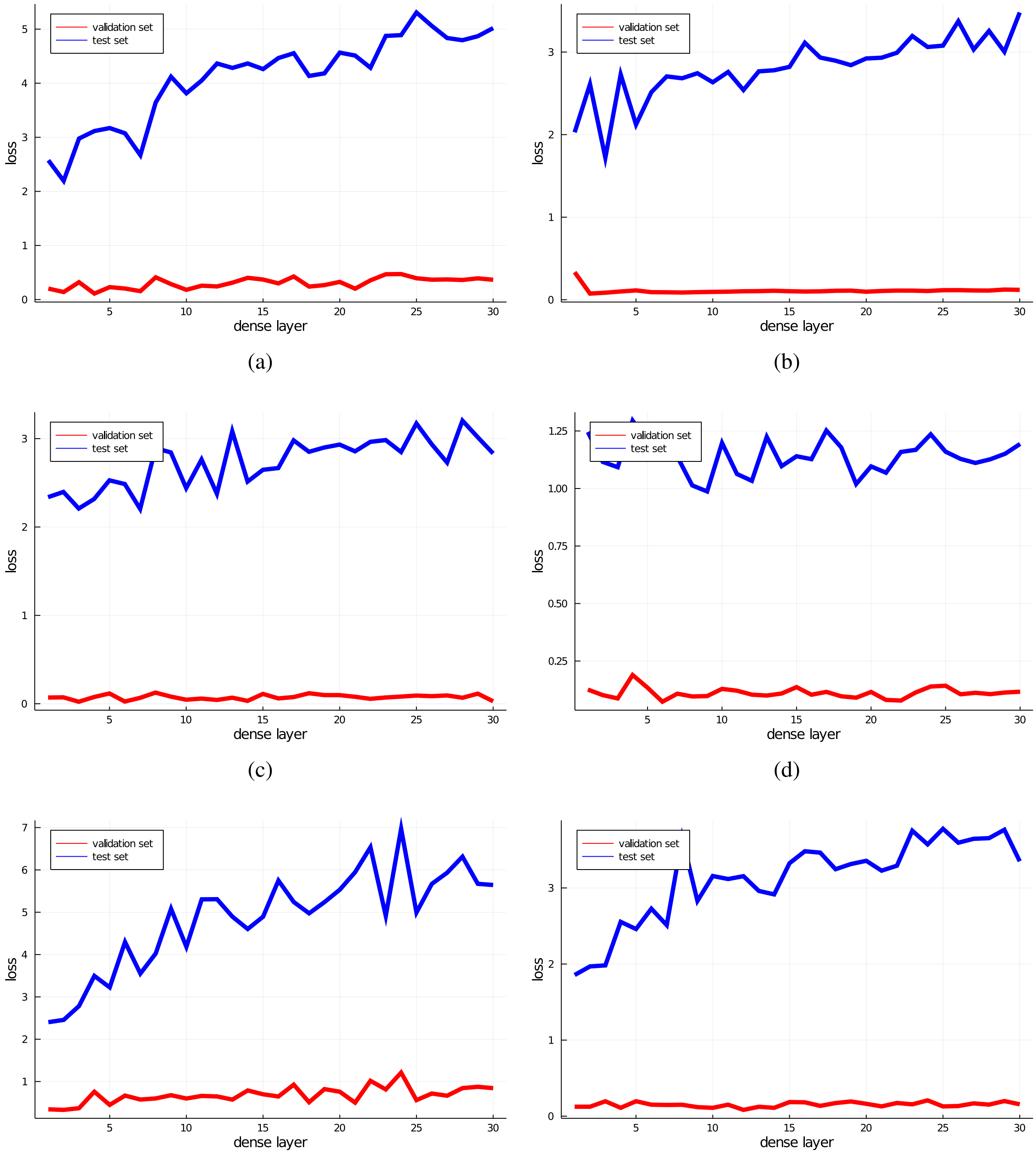
(b) Multiple instance learning

Figure 3.1: The difference between standard ML and MIL [?]. Standard ML is special case of MIL with $|b| = 1$.

3.2 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 3.2, the loss evaluated on different sets varies therefore it is likely to make mistakes when choosing our hyperparameter if we do not have enough input data.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 3.2: Evaluation of loss function with the use of validation and test set on different MILL datasets.

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.