

**RĪGAS TEHNISKĀ UNIVERSITĀTE**  
Datorzinātnes un informācijas tehnoloģijas fakultāte  
Lietišķo datorsistēmu institūts  
Mākslīgā intelekta un sistēmu inženierijas katedra

**Oļegs Korsaks**  
Bakalaura studiju programmas „Datorsistēmas”  
students, stud. apl. nr. 051RDB146

**SALIDZINOŠĀ ANALĪZE DATU  
KOPU FORMĀTIEM PYTORCH  
ATTĒLU KLASIFIKĀCIJAS  
UZDEVUMIEM**

**Atskaite par bakalaura darbu**

Zinātniskais vadītājs  
Mg.sc.ing, Pētnieks  
**ĒVALDS URTĀNS**

Rīga 2022

# Table of Contents

1. Ievads.....	3
1.1. Dziļā māšīnmācīšanās.....	3
Pamata arhitektūras.....	3
Linārie slāņi.....	3
Aktivizācijas funkcijas.....	3
Kļūdas funkcijas.....	3
MAE.....	3
MSE.....	3
CCE.....	4
BCE.....	4
Atpakaļizplatīšanās algoritms.....	4
Apmācāmo parametru optizācijas algoritmi.....	4
Metrikas.....	4
1.2. Attēlu klasifikācija.....	5
1.3. PyTorch vide.....	5
2. Metodoloģija.....	5
2.1. Datu ielādes metodes.....	5
2.1.1. Failu sistēma.....	5
2.1.2. Linux mmap.....	5
2.1.3. nVidia mmap.....	5
2.1.4. HDF5.....	5
2.1.5. PostgreSQL.....	5
2.2. Datu kopas.....	5
2.2.1. CIFAR10.....	5
2.2.2. Tiny ImageNet.....	5
2.3. Apmācības protokols.....	5
3. Rezultāti.....	6
4. Tālākie pētījumi.....	6
5. Secinājumi.....	6

# 1. Ievads

## 1.1. Dziļā māšīnmācīšanās

### Pamata arhitektūras

Makslīgais neironu tīkls sastāv no viena ieejas slāņa, viena vai dažiem slēptiem slāņiem, kā arī viena izejas slāņa. Katrs slānis satur vienu vai vairākus neironus. Tie ir saistīti ar blakusslāņu neironiem. Eksistē tīkli, kuru neironi tiek pilnīgi saistīti ar visiem neironiem blakusslāņos, bet var būt saistīti tikai daļēji. Katrai saitei ir savs svars un nobīde. Tie ir apmācamie parametri. Apmācīšanas procesa mērķis ir atrast tādus svarus un nobīdes visām saitēm, ar kuriem tīkls veidotu pareizus rezultātus izejas slānī.

### *Linārie slāņi*

### *Aktivizācijas funkcijas*

Softmax (priekš klasifikācijas)

### Kļūdas funkcijas

Kļūdas funkcija palīdz noteikt cik tālu tekošā prognozēta vērtība ir no patiesas. Un ja to pielietot visiem datu eksemplāriem – ar to var noteikt, cik labi tekošais modelis var prognozēt rezultātus kopumā. Ideālā gadījumā kļūdai jābūt vienāgai nullei, gan apmācības datu eksemplāriem, gan pārbaudes. Tātad apmācība cenšas kļūdu samazināt.

### **MAE**

Mean absolute error vai vidēja absolūta kļūda:

$$L_{MAE} = \frac{1}{N} \cdot \sum_{i=0}^N |h_{\theta}(x_i) - y_i|$$

Tā ir vidēja absolūta starpība starp pareizas un prognozētas vērtībām. Kļūda pieaug lineāri un tiek pielietota regresijas uzdevumiem, kuru rezultāts ir viena vērtība.

### **MSE**

Mean squared error vai vidēja kvadrātiskā kļūda:

$$L_{MSE} = \frac{1}{N} \cdot \sum_{i=0}^N (h_{\theta}(x_i) - y_i)^2$$

Tā ir vidēja starpība starp pareizas un prognozētas vērtībām, kas tiek pacelta kvadrātā. Kvadrāts palīdz izvairīties no negatīvām vērtībām. Kā arī kļūdas vērtība pieaug straujāk, salīdzinot ar MAE. Tiek pielietota regresijas uzdevumiem.

**CCE**

**BCE**

## Atpakaļizplatīšanās algoritms

Šis algoritms cenšas mainīt tīkla svarus un nobīdes tā, lai kļūda būtu 0.

$$\theta_0 := \theta_0 - \alpha \cdot \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$

$$b := b - \alpha \cdot \frac{\partial}{\partial b} J(b, W)$$

$$\theta_1 := \theta_1 - \alpha \cdot \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$

$$W := W - \alpha \cdot \frac{\partial}{\partial W} J(b, W)$$

Pieņemsim, ka modelis ir:

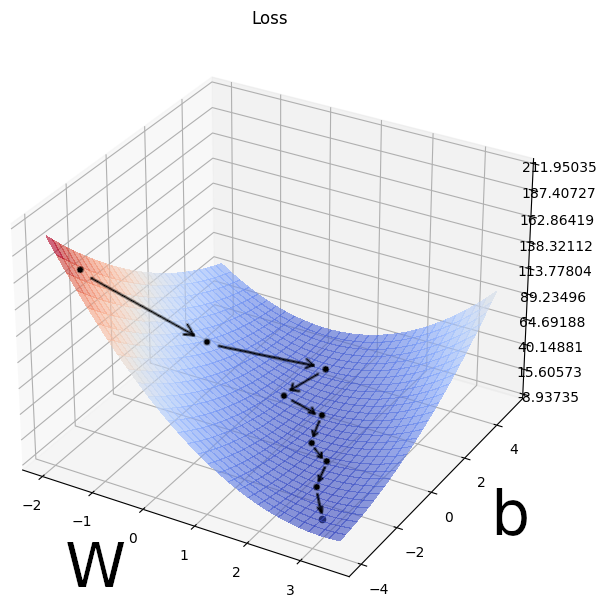
$$y' = M(x) = \text{Linear}(W_1, b_1, W_2, b_2, x) = \text{Linear}(W_2, b_2, \text{ReLU}(\text{Linear}(W_1, b_1, x)))$$

$$\text{Linear}(W_i, b_i, x_i) = W_i \cdot x_i + b_i$$

$$\text{ReLU}(x_i) = \begin{cases} x_i, & x_i \geq 0 \\ 0, & x_i < 0 \end{cases}$$

Kur, piemēram,  $J(\theta_0, \theta_1) = L_{MAE} = \frac{1}{N} \cdot \sum_{i=0}^N (h_{\theta}(x_i) - y_i)$  ir MAE kļūdas funkcija un  $\alpha$  ir apmācības

koeficients, kurš noteic, cik strauji svars  $W$  un nobīde  $b$  tiek mainīti. Kļūdas funkcijas atvasinājums noteic vai parametru ir jāpalielina, vai jāsamazina un uz kādu lielumu.



# Apmācāmo parametru optizācijas algoritmi

SGD

## Metrikas

novērtēt cik labs rezultāts Accuracy, F1

## 1.2. Attēlu klasifikācija

ConvNets

ResNet / DenseNet

## 1.3. PyTorch vide

Modeļu implementāciju <> viendājumi

Datu ielādes process (DataSet)

## 2. Metodoloģija

### 2.1. Datu ielādes metodes

#### 2.1.1. Failu sistēma

#### 2.1.2. Linux mmap

#### 2.1.3. nVidia mmap

#### 2.1.4. HDF5

#### 2.1.5. PostgreSQL

### 2.2. Datu kopas

#### 2.2.1. CIFAR10

#### 2.2.2. Tiny ImageNet

.. vel kādas

^ Atrast HiRes datu kopas priekš klasifikācijas

^ Atrast dažādu izmēru attēlu datu kopas

^ 3 dažādu izmēru / konfigurāciju datu kopas

## **2.3. Apmācības protokols**

\* Jāizdomā metrikas kā noteikt ietekmi datu ielādes metodēm

\* Vidējais epocha ātrums sekundēs

## **3. Rezultāti**

Datu ielādes metodes PRET datu kopām

## **4. Tālākie pētījumi**

## **5. Secinājumi**