**Slovenská technická univerzita v Bratislave**

**Fakulta informatiky a informačných technológii**

Backpropagation algoritmus

**Zadanie č. 3a**

Contents

[Popis zadania 3](#_Toc183801815)

[Reprezentácia údajov a konfigurácie 3](#_Toc183801816)

[Postup riešenia 4](#_Toc183801817)

[Porovnávanie parametrov 7](#_Toc183801818)

[Testovanie 7](#_Toc183801819)

[**Zhodnotenie** 13](#_Toc183801820)

# Popis zadania

Úlohou bolo vytvoríte neurónovú sieť na klasifikáciu ručne písaných číslic z dátového súboru MNIST. Zobrať súbor údajov zo 60 000

# Reprezentácia údajov a konfigurácie

Na trénovacie a testovacie dáta použijeme údaje z databázy MNIST.

Veľkosť dávky (batch) je zaznamenaná v konfiguračnom súbore, ktorý vyzerá nasledovne:

[Settings]

load\_model = False

optimizer(sgd, sgd\_momentum, adam) = adam

epoch\_count = 20

# Postup riešenia

Na zatáčku programu bolo importovane knižnica pre náhodne generovanie, matematická Tato cast Obsahuje všetky potrebné knižnice pre Torch a MNIST klasifikátor. Okrem toho aj knižnicu pre načrtnutie štatistických grafov, knižnicu na sledovanie času, cesty k súborom, farebné výstupy a konfiguračný súbor.

* import torch
* import torchvision
* import torchvision.transforms as transforms
* from torch.utils.data import DataLoader

Všetky potrebné knižnice je potrebne nainštalovať použitím príkazu “pip install <názov knižnice>“.

Po implementácii knižníc bolo zadefinovaných niekoľko najdôležitejších funkcií, takých ako:

"""Parameters for the configuration file"""  
load\_model = None  
auto\_save\_model = None  
optimizer = None  
epoch\_count = 0  
train\_batch\_size = 0  
test\_batch\_size = 0  
show\_confusion\_matrix = False  
learning\_rate = 0  
momentum = 0

Ďalej sú určené začiatočné parametre, ak konfiguračný súbor neexistuje, s následným načítaním konfiguračných parametrov zo súboru.

Ďalej nasleduje trieda MLP modelu:

class MLP(nn.Module):  
 def \_\_init\_\_(self):  
 super(MLP, self).\_\_init\_\_()  
 self.fc1 = nn.Linear(784, 256)  
 self.fc2 = nn.Linear(256, 64)

# Porovnávanie parametrov

Počas testovanie prišli sme k takým výsledným charakteristikám trénovaniach metód:

# Testovanie

Počas testovania používal a uviedol rôzne trénovanie metódy(sgd, sgd\_momentum, adam), počet vrstiev a ich veľkosti, rôzne aktivačné funkcie(ReLU (Rectified Linear Unit), Sigmoid, Tanh), rýchlosť učenia(learning rate) a veľkosť dávky.

Prvý test som začal zo vstupnými parametrami, chcel som porovnať rôzne trénovanie metódy:

optimizer(sgd, sgd\_momentum, adam) = adam

epoch\_count = 200

train\_batch\_size = 64

test\_batch\_size = 64

learning\_rate = 0.001

Vstupná vrstva:

* Vstup: 784 neurónov
* Výstup: 256 neurónov

Posledné 3 epochy(trénovanie a testovanie):

Najlepšia uspesnost: 98.5

# Najlepšie nájdene parametre

# **Zhodnotenie**

Vytvorená neurónová sieť na klasifikáciu ručne písaných číslic z dátového súboru MNIST je určený na trénovanie a testovanie vytvoreného systému s rôznymi parametrami, ako napríklad: optimalizačný metód, vhodné aktivačné funkcie, rýchlosť učenia a veľkosť dávky (batch) pre optimalizačný algoritmus. Taktiež treba bolo cestou pokusov a omylov navrhnúť počet vrstiev a ich veľkosti pre vytvorenú model. Otestovať, porovnať výsledky testovania a vyhodnotiť úspešnosť nerehneného modelu.

Výtvory, natrénovaný model je schopný klasifikovať číslice s percentom úspešnosti 98 a vyššie. Závisí od poctu epoch a nastavení modelu na trénovaní. Najlepší výsledok bol opísaný v testovacej časti a dokázal schopnosť siete riešiť podobne problémy na klasifikáciu číslic.

Taktiež program umožňuje konfigurovať konfiguráciu pred behom programu. Čo výrazne zrýchľuje prístup k jednotným špecifikáciám a prispôsobeným nástrojom pre určite potreby klienta. Ako napríklad používateľ môže si zvoliť spôsob učenia site, aká najviac mu vyhovuje.

Dosť dôležitou úlohou bolo správne vybrať a nastrojiť parametre pre učenia. Boli porovnané rôzne spôsoby konfigurácie, a navrhnutá najlepšia konfigurácia, ktorú sa dala vytvoriť.