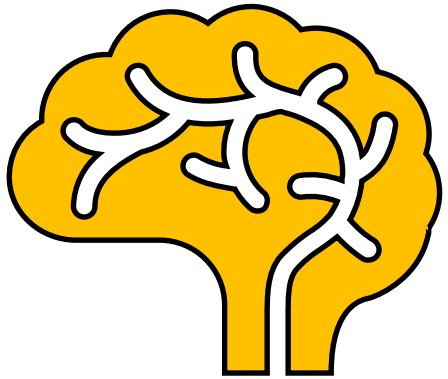
Improving TrustGAN: Making Deep Neural Networks More Trusthworthy



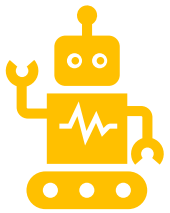
Student: Vivdici Ina

Coordonator: Ciortuz Liviu

AGENDA

- Introducere
- Contextul și relevanța sistemului
- Prezentarea sistemului
- Contribuțiile mele
- Concluzii și direcții viitoare

INTRODUCERE



De ce Învățare Automată?

Popularitate

Utilitate

Curiozitate

De ce TrustGAN [1]?

Siguranță

Necesitate

Actualitate

STATE OF THE ART

FOLOSIREA DROPOUT-ULUI [2]

Ce e dropout? [3]

- Probabilist excludem niște neuroni din rețea

Cum ajută la obținerea încrederii?

- **Modelarea a T forward passes** și folosirea **mediei pentru output** și a **varianței pentru confidence**

FOLOSIREA TCP IN LOC DE MCP [4]

Ce e TCP si MCP?

- MCP = maximum class probability
- TCP = true class probability

Cum e estimată TCP?

- Folosim 2 rețele: ConfidNet și ConvNet
- **ConvNet extrage features** importante
- **ConfidNet învață scorul certitudinii** folosind output-ul de la ConvNet

FOLOSIREA UNUI GAN [5]

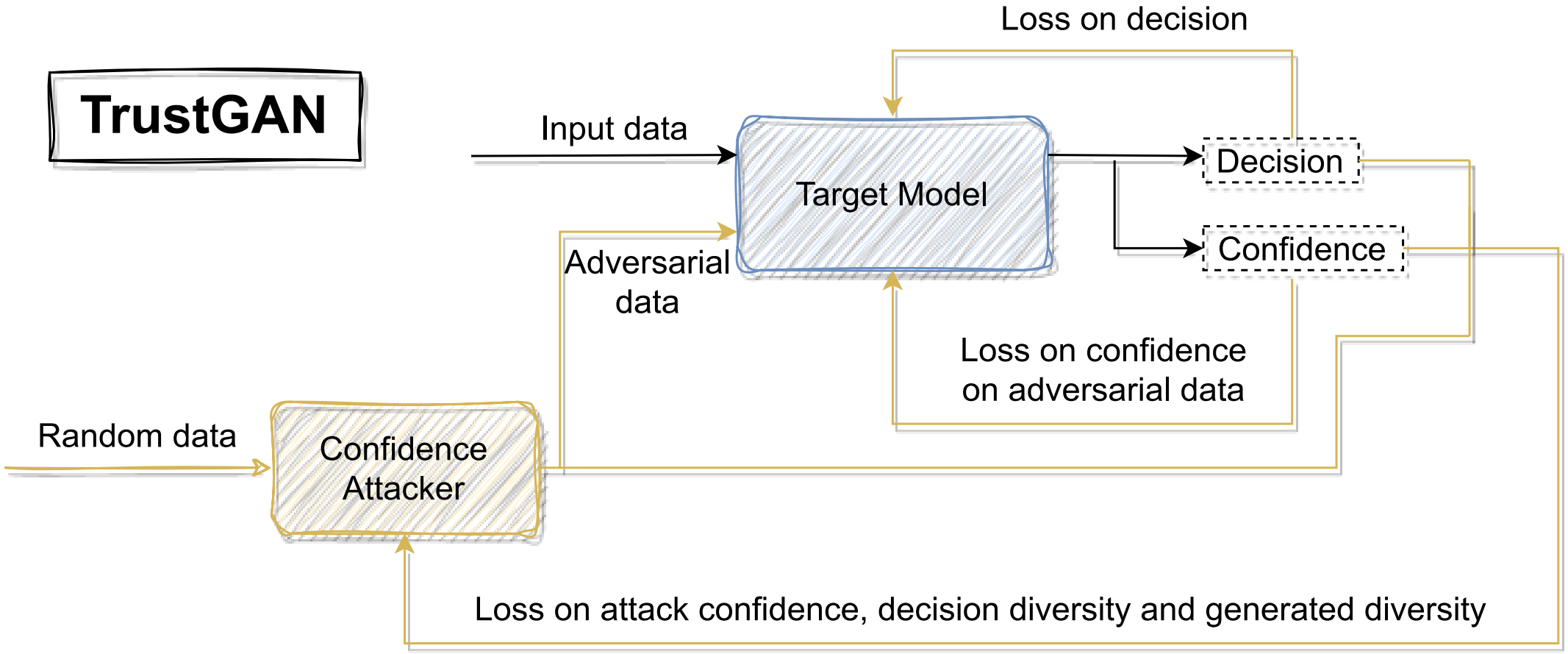
Ce e un GAN? [6]

- O rețea formată dintr-un discriminator și un generator
- **Discriminatorul** învață să distingă între exemple reale și false
- **Generatorul** învață să genereze exemple false astfel încât acestea să fie clasificate ca fiind reale

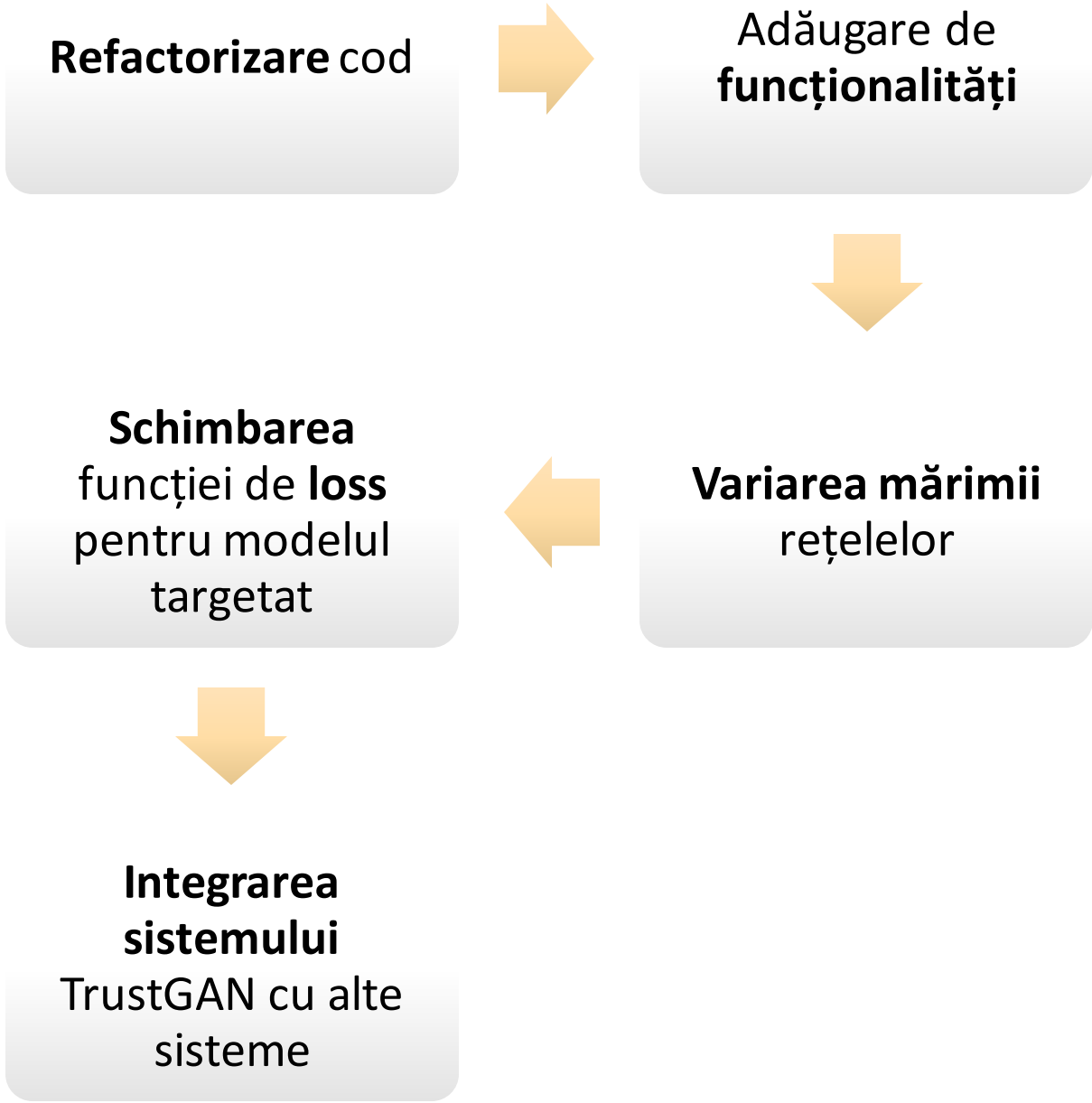
Care e ideea?

- Ideea este de a **învăța modelul să distingă între datele care fac parte din distribuția dorită** și cele care nu fac parte din ea

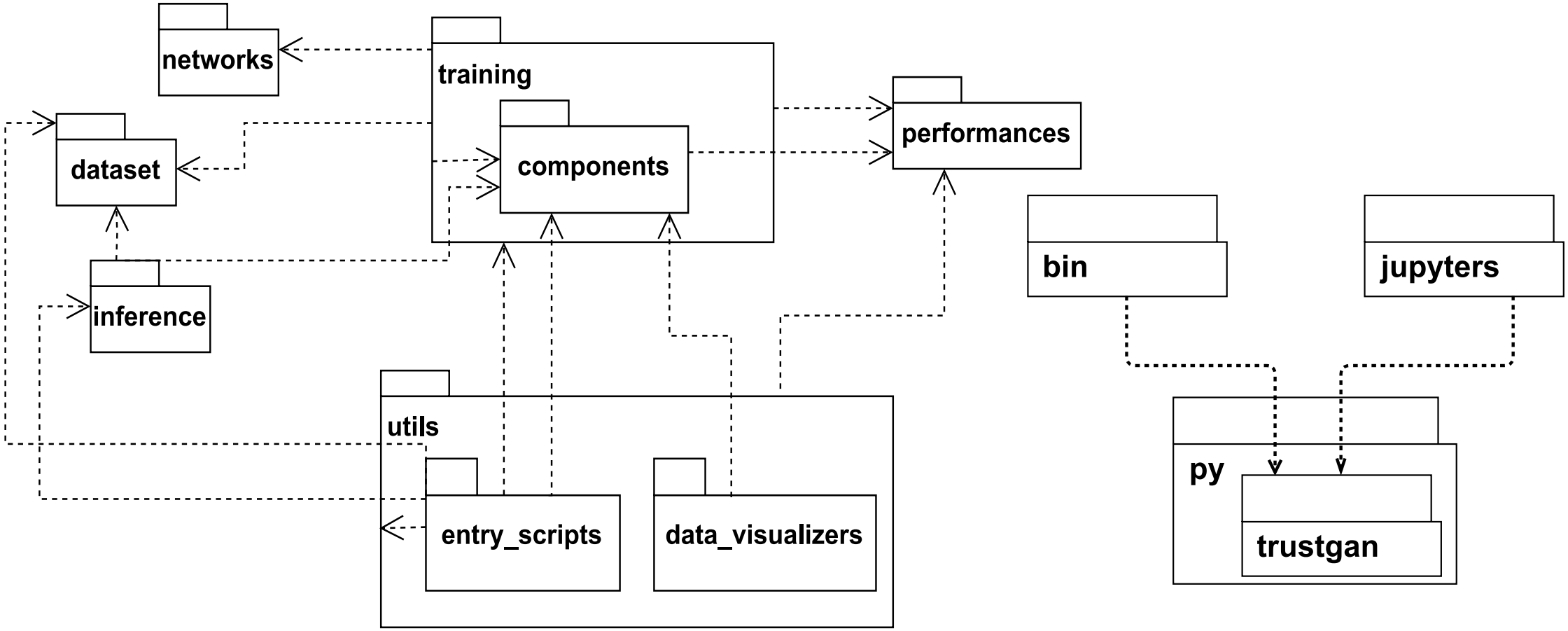
MODUL DE LUCRU A SISTEMULUI TRUSTGAN



Contribuții personale



REFACTORIZARE COD



COD REFACTORIZAT

COD ORIGINAL

ADĂUGARE DE FUNCȚIONALITĂȚI

Cross validare **k-fold**

Reprezentare grafică în **Tensorboard**

Memorare și reprezentare grafică a **timpului de execuție și memoriei GPU** folosită

Noi funcții de **loss**

Noi parametri configurabili pentru script-ul de antrenare

Script-ul pentru **inferență**

Primirea **parametrilor pentru construirea** unei instanțe de rețea neuronală ca **parametru** pentru script

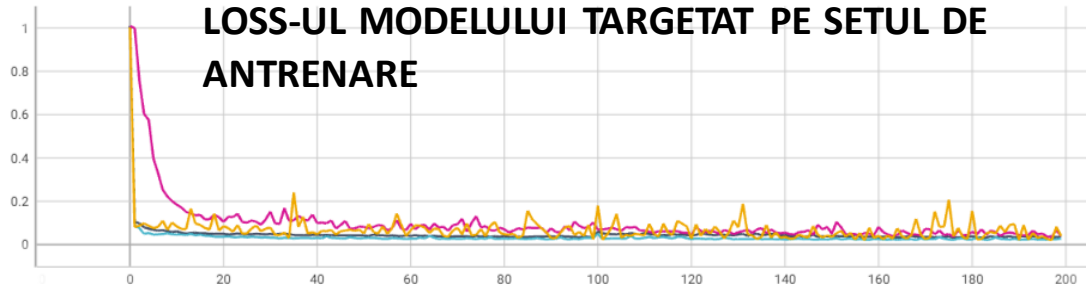
NOTITE ASUPRA VARIERII MARIMII SI LOSS-URILOR

- Toate experimentele au fost executate folosind **Kaggle Code**, pe **GPU**
- S-a folosit setul de date **MNIST**
- Au fost executate cate **200 epoci** pentru fiecare model
- Au fost folosit **5-fold cross-validation** pentru fiecare
- Reteaua tinta **nu a fost antrenata preventiv**
- **Validarea** a fost efectuata odata **la 25 epoci**

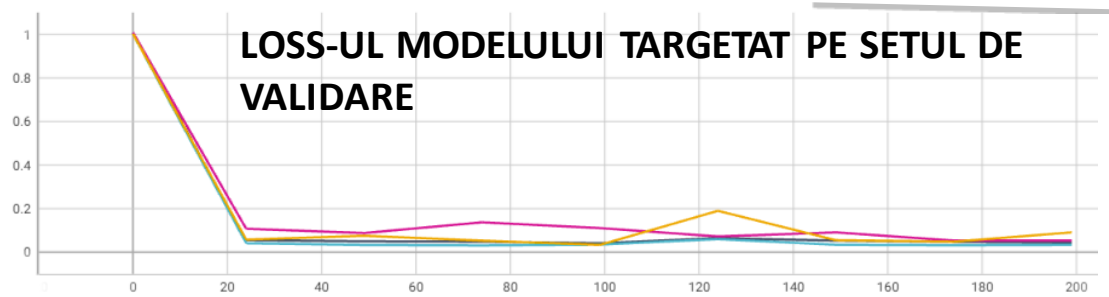
VARIAREA MĂRIMII REȚELELOR: COMPARAȚIE DINTRE LOSS-URILE MODELELOR

gan = 2, net = 4
gan = 3, net = 4
gan = 4, net = 5
gan = 5, net = 7

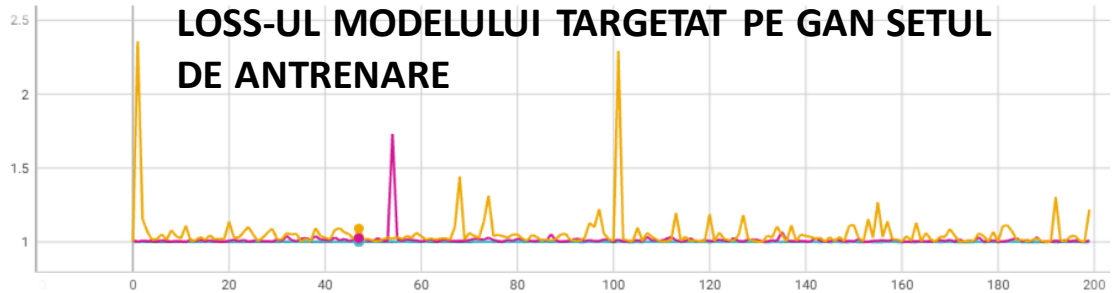
LOSS-UL MODELULUI TARGETAT PE SETUL DE ANTRENARE



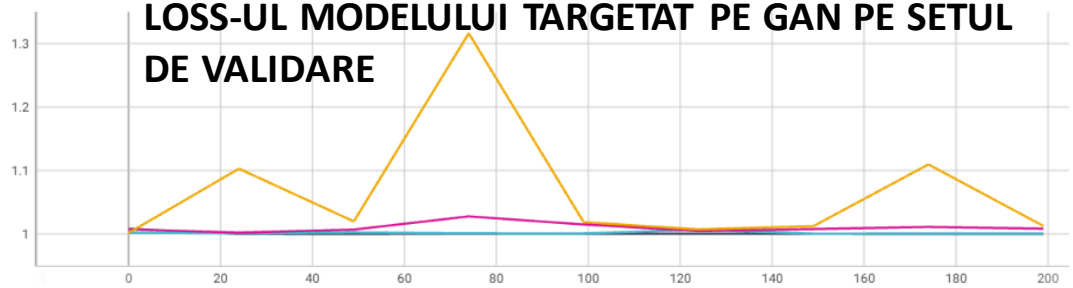
LOSS-UL MODELULUI TARGETAT PE SETUL DE VALIDARE



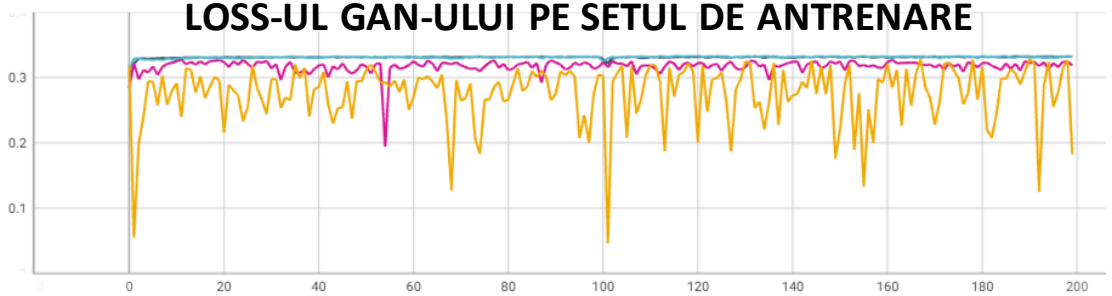
LOSS-UL MODELULUI TARGETAT PE GAN SETUL DE ANTRENARE



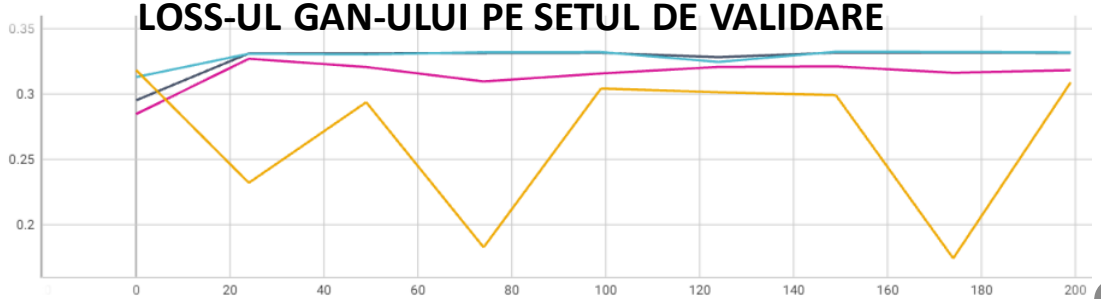
LOSS-UL MODELULUI TARGETAT PE GAN PE SETUL DE VALIDARE



LOSS-UL GAN-ULUI PE SETUL DE ANTRENARE



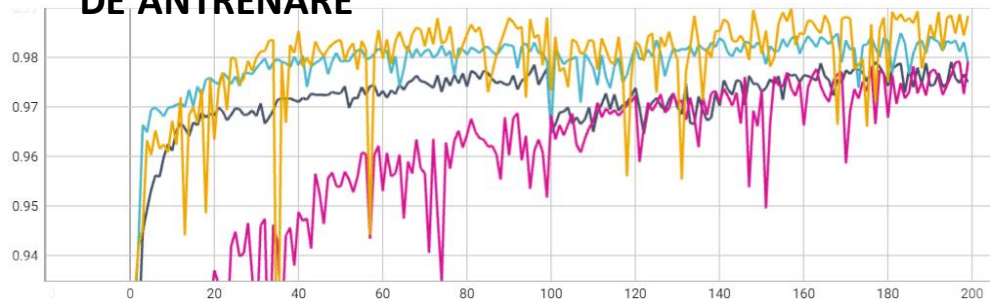
LOSS-UL GAN-ULUI PE SETUL DE VALIDARE



VARIAREA MĂRIMII REȚELELOR: COMPARAREA ACURATEȚII MODELULUI TARGETAT SI UTILIZĂRII RESURSELOR

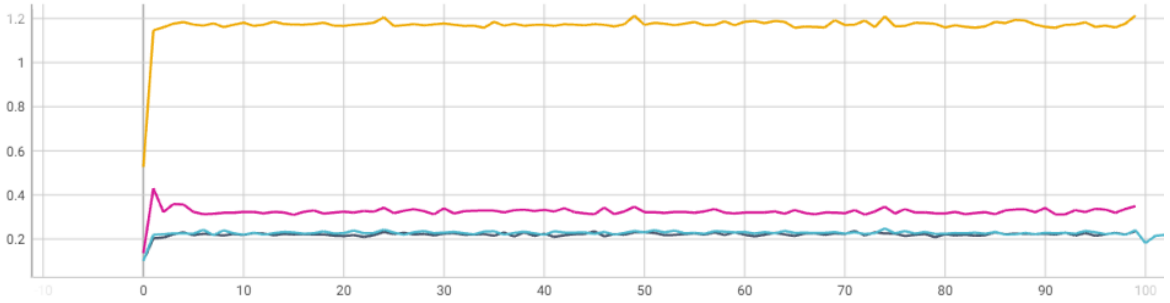
gan = 2, net = 4
gan = 3, net = 4
gan = 4, net = 5
gan = 5, net = 7

ACURATEȚEA MODELULUI TARGETAT PE SETUL DE ANTRENARE

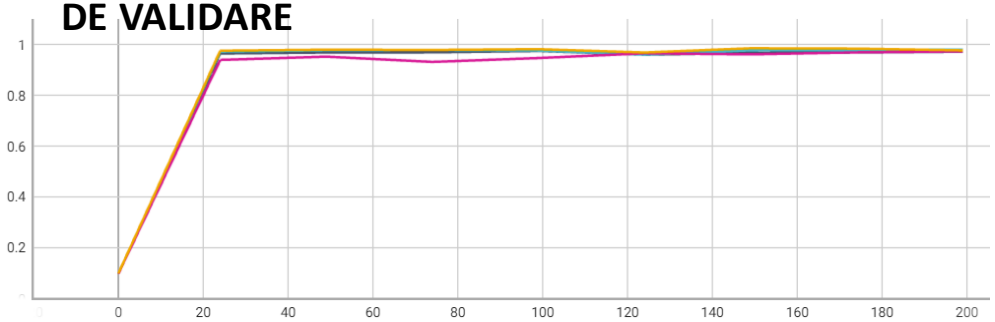


TIMPUL DE EXECUȚIE A UNEI EPOCI ÎN MINUTE

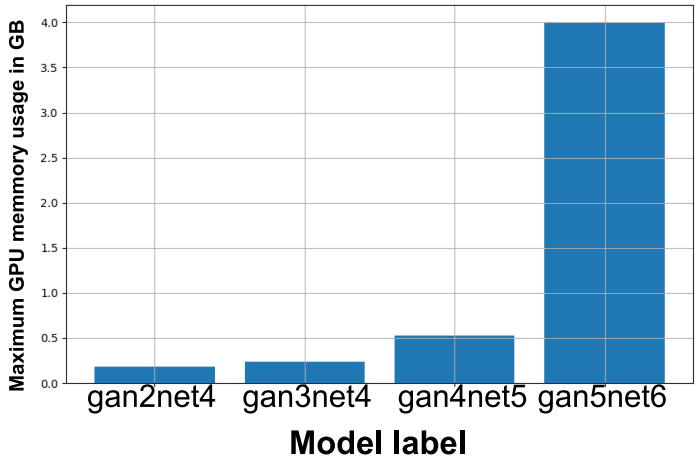
6/1
3/1



ACURATEȚEA MODELULUI TARGETAT PE SETUL DE VALIDARE



MAXIMUL MEMORIEI GPU FOLOSITE

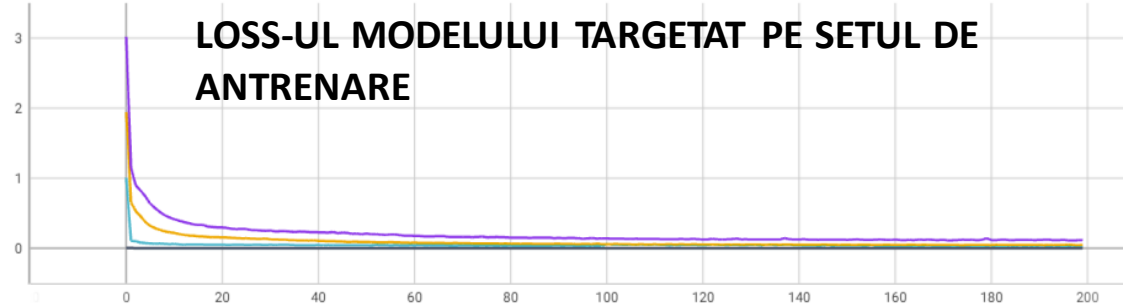


16/1
8/1

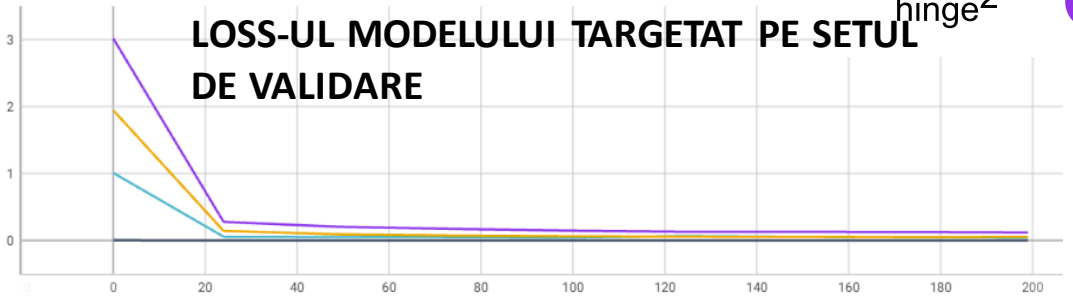
VARIAREA FUNCTIEI DE LOSS A MODELULUI TARGETAT: COMPARATIE DINTRE LOSS-URILE MODELELOR

D_{cs}
log
hinge
$hinge^3$
$hinge^2$

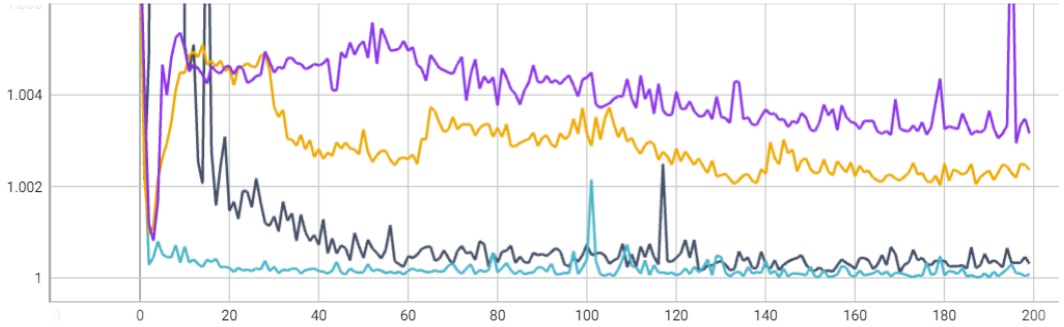
LOSS-UL MODELULUI TARGETAT PE SETUL DE ANTRENARE



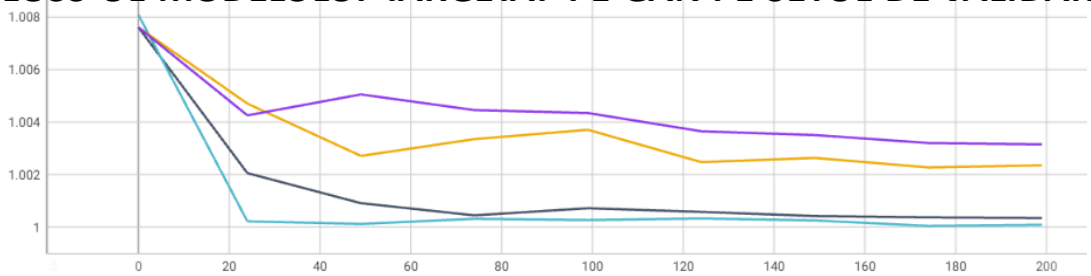
LOSS-UL MODELULUI TARGETAT PE SETUL DE VALIDARE



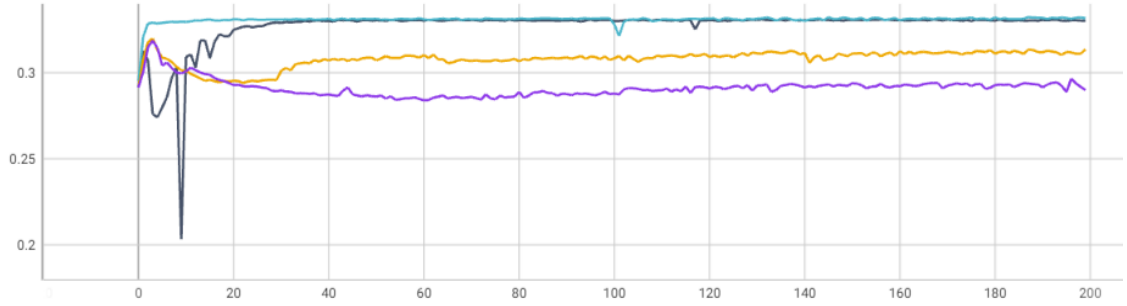
LOSS-UL MODELULUI TARGETAT PE GAN SETUL DE ANTRENARE



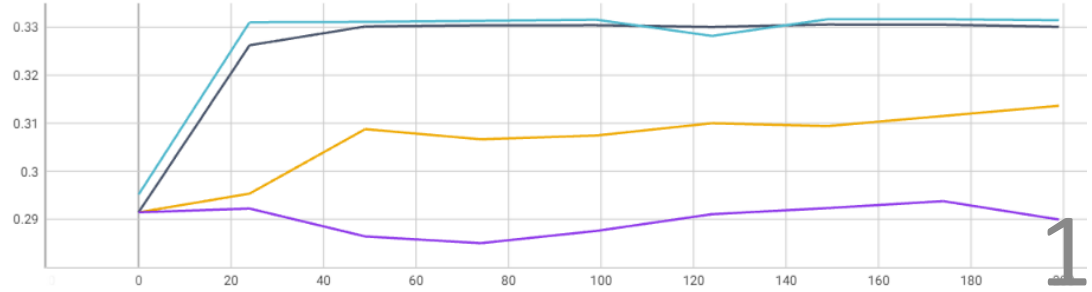
LOSS-UL MODELULUI TARGETAT PE GAN PE SETUL DE VALIDARE



LOSS-UL GAN-ULUI PE SETUL DE ANTRENARE



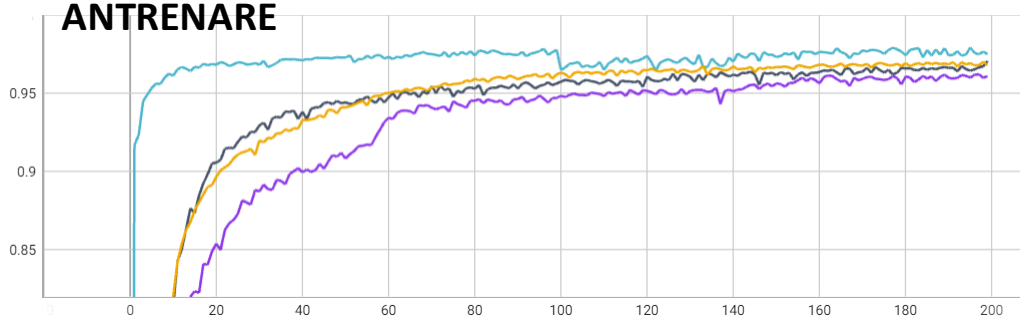
LOSS-UL GAN-ULUI PE SETUL DE VALIDARE



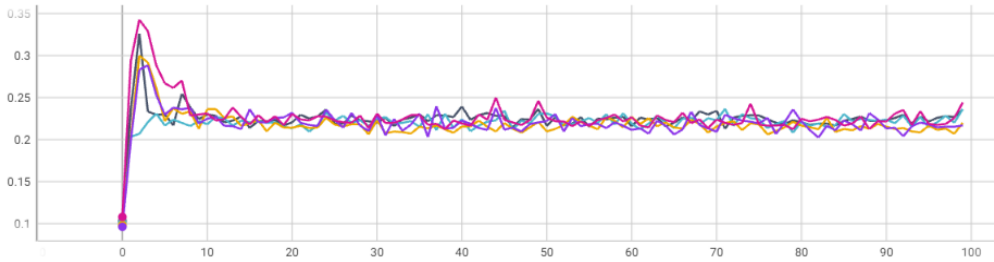
VARIAREA MĂRIMII REȚELELOR: COMPARAREA ACURATEȚII MODELULUI TARGETAT SI UTILIZĂRII RESURSELOR

D_{cs}
log
hinge
$hinge^3$
$hinge^2$

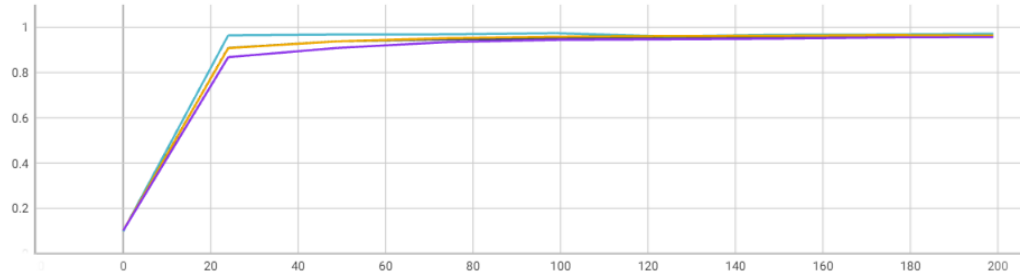
ACURATEȚEA MODELULUI TARGETAT PE SETUL DE ANTRENARE



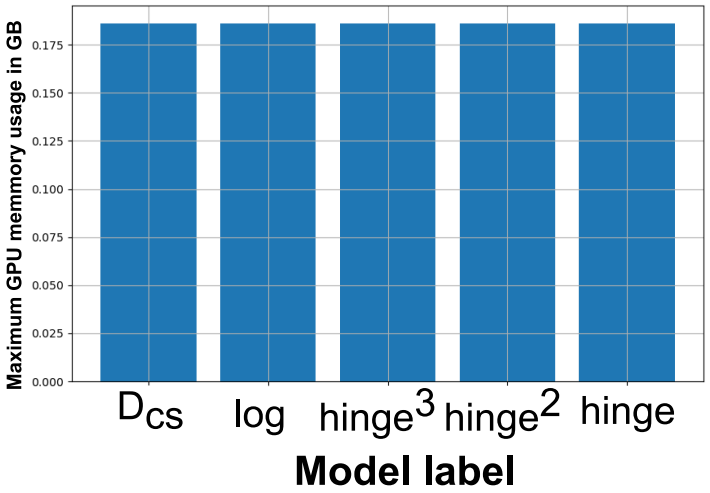
TIMPUL DE EXECUȚIE A UNEI EPOCI ÎN MINUTE



ACURATEȚEA MODELULUI TARGETAT PE SETUL DE VALIDARE



MAXIMUL MEMORIEI GPU FOLOSITE



FLEXIBILIZAREA SI INTEGRAREA SISTEMULUI CU UN ALT SISTEM

CONCLUZII SI DIRECTII VIITOARE

Concluzii:

- **Micsorarea marimii** atacatorului certitudinii si a modelului tinta **este utila** daca dorim o utilizare de resurse substantial redusa cu pretul unor performante putin mai mici
- Chiar daca, **cross entropia** s-a dovedit a fi **cel mai bun loss**, asta **nu inseamna ca Cauchy-Schwartz nu ar deveni mai bun** odata rulate inca 200 epoci
- **Refactorizarea sistemului** l-a apropiat de a fi **mai usor de inteles** si folosit
- **Adaugarea functionalitatilor a imbunatit sistemul** si l-a ajutat sa fie mai usor de folosit in diverse contexte
- **TrustGAN** nu e un sistem creat doar pentru MNIST, ci **poate fi folosit si in bioinformatica**

Directii viitoare:

- Efectuarea experimentelor prezentate de mine cu o **retea tinta pre-antrenata**
- **Compararea TrustGAN** cu alte sisteme
- **Imbunatatirea si flexibilizarea template**-ului elaborat
- Efectuarea a **inca 200 epoci** folosind functia de **loss Cauchy-Schwartz**
- **Modificarea marimii unui batch [7]**

VA MULTUMESC PENTRU ATENTIE

Intrebari?

BIBLIOGRAFIE

[1] H. du Mas des Bourboux, “Trustgan: Training safe and trustworthy deep learning models through generative adversarial networks,” 2022, accessed on: 28.06.2023. [Online]. Available: https://arxiv.org/pdf/2211.13991.pdf

[2] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” 2016, accessed on: 27.06.2023. [Online]. Available: https://arxiv.org/pdf/1506.02142.pdf

[3] P. Baldi and P. J. Sadowski, “Understanding dropout,” in Advances in Neural Information Processing Systems, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds., vol. 26. Curran Associates, Inc., 2013, accessed on: 28.06.2023. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2013/file/71f6278d140af599e06ad9bf1ba03cb0-Paper.pdf

[4] C. Corbière, N. THOME, A. Bar-Hen, M. Cord, and P. Pérez, “Addressing failure prediction by learning model confidence,” in Advances in Neural Information Processing Systems, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019, accessed on: 28.06.2023. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/757f843a169cc678064d9530d12a1881-Paper.pdf

[5] K. Lee, H. Lee, K. Lee, and J. Shin, “Training confidence-calibrated classifiers for detecting out-of-distribution samples,” 2018, accessed on: 28.06.2023. [Online]. Available: https://arxiv.org/pdf/1711.09325.pdf

[6] C. C. Aggarwal, Neural Networks and Deep Learning: A Textbook, 1st ed. Springer Publishing Company, Incorporated, 2018, chapter 10.4

[7] F. He, T. Liu, and D. Tao, “Control batch size and learning rate to generalize well: Theoretical and empirical evidence,” in Advances in Neural Information Processing Systems, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019, accessed on: 20.06.2023. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/dc6a70712a252123c40d2adba6a11d84-Paper.pdf