Elektronski Fakultet

Univerzitet u Nišu

KAN – Kolmogorov-Arnold Networks

- Tehnički izveštaj -

Predmet:
Duboko učenje

Student:

Dimitrije Petrović (1682)

Mentor:

Prof. dr Aleksandar Milosavljević

Niš, avgust 2024. Godine

Sadržaj

1. Uvod

Kolmogorov-Arnold mreža (Kolmogorov-Arnold Network, skraćeno KAN) [1] [1] je mreža koja se baziraja na Kolmogorov-Arnold reprezentacionoj teoremi[2] . Ova teorema tvrdi da se bilo koja kontinuirana funkcija više promenljivih može konstruisati sa konačnim brojem funkcija s dve promenljive. Nasuprot MLP, gde imamo fiksne aktivacione funkcione i učimo težine, kod KAN su funkcije te koje se uče. Svaka funkcija je sa jednom promenljivom i parametrizovana pomoću B-splajna.

Cilj projekta je proučiti KAN mreže, upoznati se sa postojećim rešenjima, uporediti rezultate sa MLP. Prateći projekat predstavlja proof-of-concept implementaciju KAN mreže, kao i obučavanje Convolutional-KAN[3] mreže na CIFAR10 skupu podataka.

2. Kolmogorov-Arnold reprezentaciona teorema

Kolmogorov-Arnold reprezentaciona teorema tvrdi da svaka funkcija f, $f: [0,1]^n \rightarrow \mathrm{R}$ može biti predstavljana kao konačna kompozicija neprekidnih funkcija sa jednom promenljivom [2] .

$$f(\mathbf{x}) = f(x_1, \ldots, x_n) = \sum_{q=0}^{2n} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right)$$

$$\phi_{q,p}: [0,1] \rightarrow \mathbb{R} \text{ and } \Phi_q: \mathbb{R} \rightarrow \mathbb{R}$$

Kolmogorov Arnold Reprezentaciona Teorema

Problem sa ovim funkcijama je to što ne moraju da budu glatke, takođe mogu da budu fraktali. Ovi problemi, i činjenica da su bile razmatrane potencijalne implementacije neuronskih mreža zasnovanim na Kolmogorov-Arnold reprezentacionoj teoremi sa samo dva sloja su doveli do toga da ova teorema bude smatrana za praktično neupotrebljivu za neuronske mreže[4] .

Noviji radovi su primenili ovu formulu na mreže sa većim brojem slojeva, takođe se pozivaju na to da iako možemo dobiti funkcije koje praktično nije moguće naučiti, da se ovo u stvarnosti retko dešava, takođe, sa većim brojem slojeva, možemo bolje predstaviti neke funkcije koje ne bi bile glatke sa dvoslojnom arhitekturom.

Ova teorema će biti pokazana na f(x,y)=xy
Možemo definisati funkcije na sledeći način:
ϕ1,1(x) = x
ϕ1,2(y) = y
ϕ2,1(x) = x^2
ϕ2,2(y) = y^2

Φ1(z) = $\frac{1}{2}z^2$
Φ2(z) = $-\frac{1}{2}z$
Φ3(z) = 0
Φ4(z) = 0

f(x,y) = Φ1(ϕ1,1(x) + ϕ1,2(y)) + Φ2(ϕ2,1(x) + ϕ2,2(y))

$\Phi 1(\phi 1,1(x) + \phi 1,2(y)) = \Phi 1(x + y) = \frac{1}{2}(x + y)^2$

$\Phi 2(\phi 2,1(x) + \phi 2,2(y)) = \Phi 2(x^2 + y^2) = -\frac{1}{2}(x^2 + y^2)$

$f(x,y) = \frac{1}{2}(x + y)^2 - \frac{1}{2}(x^2 + y^2) = \frac{1}{2}x^2 + xy + \frac{1}{2}y^2 - \frac{1}{2}x^2 - \frac{1}{2}y^2 = xy$

3. Implementacija

Kod KAN mreža imamo rezidualne aktivacione funkcije koje se sastoje od silu funkcije i b-splajna, svaka od ovih funkcija ima svoje težine:

$$\phi(x) = w_b b(x) + w_s \text{spline}(x).$$

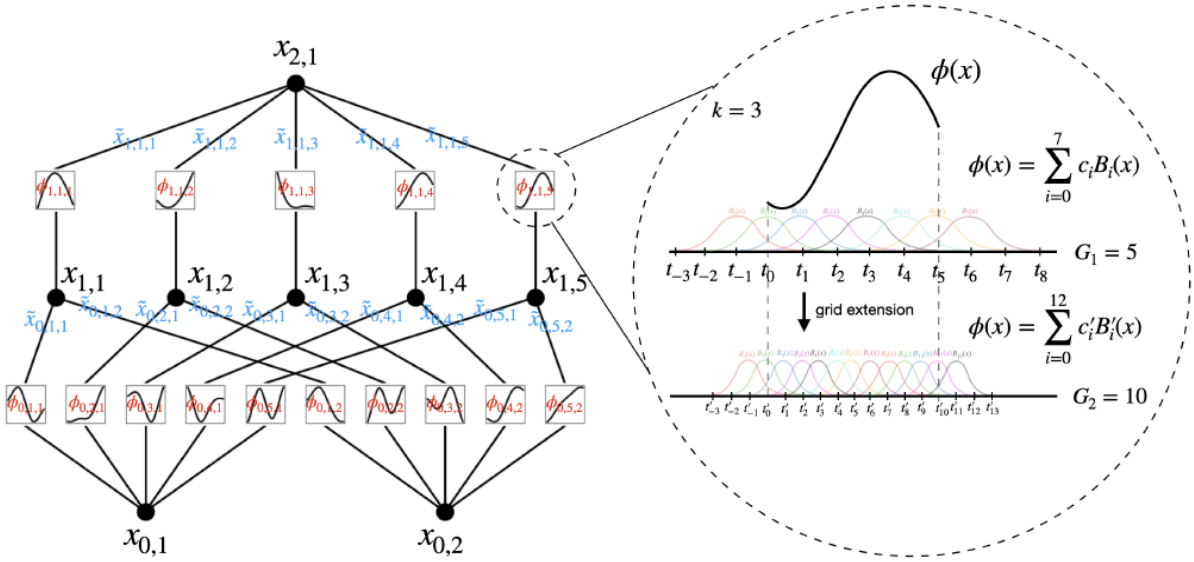
$$b(x) = \text{silu}(x) = x/(1 + e^{-x})$$

$$\text{spline}(x) = \sum_i c_i B_i(x)$$

Na početku inicijalizujemo ws = 1 i spline(x)$\approx$0, za wb koristimo Xavier inicijalizaciju [1] . Pošto je pri treniranju moguć slučaj da su vrednosti aktivacija van fiksiranog regiona nad kojim su splajnovi definisani, potrebno je ažurirati grid splajna - čvorove splajnova dinamički menjamo tokom treniranja modela, na ovaj način nećemo imati pogrešne ili nepredvidljive rezultate.

Za samu mrežu moguć je i fine-tuning, tj. fine-graining, ovo podrazumeva fitovanje novog fino-granulisanog splajna na stari splajn. Grublji grid ima veće intervale i manji broj tačaka, dok finiji grid ima manje intervala i više tačaka, čime omogućavamo precizniju aproksimaciju. Ovo se postiže tako što dodajemo više intervala u grid-u/mreži, zatim se parametri spajna menjaju kako bi se smanjila razlika između naše prethodne aproksimacije na gruboj i na nasoj novoj, finijoj mreži. Ovaj proces se može postići upotrebom najmanjih kvadrata i svako proširenje grid-a se vrši nezavisno na svaki splajn.

Pri širenju grid-a moramo paziti i na broj parametara, zato sto u slučaju da broj podataka jednak broju parametara dolazimo do praga interpolacije, posle čega dolazi do overfitting-a, tj model radi interpolaciju – tačno uči svaku tačku, ovaj problem je takođe prisutan i zbog same upotrebe B-splajna, tačnije zbog polinoma zato što se polinomi koriste za B-splajn.



Slika 1. širenje grid-a

4. Dodatne osobine

Kako bi smanjili broj veza izmedju neurona, smanjili složenost i poboljšali generalizaciju koristi se sparsifikacija, za ovo se koristi regularizacija entropije zajedno sa L1 normom, takođe potrebno je definisati novu L1 normu [1] :

$$|\phi|_1 \equiv \frac{1}{N_p} \sum_{s=1}^{N_p} \left| \phi(x^{(s)}) \right| .$$

$$|\mathbf{\Phi}|_1 \equiv \sum_{i=1}^{n_{\text{in}}} \sum_{j=1}^{n_{\text{out}}} |\phi_{i,j}|_1 .$$

$$S(\mathbf{\Phi}) \equiv -\sum_{i=1}^{n_{\text{in}}} \sum_{j=1}^{n_{\text{out}}} \frac{|\phi_{i,j}|_1}{|\mathbf{\Phi}|_1} \log \left(\frac{|\phi_{i,j}|_1}{|\mathbf{\Phi}|_1} \right) .$$

$$\ell_{\text{total}} = \ell_{\text{pred}} + \lambda \left(\mu_1 \sum_{l=0}^{L-1} |\mathbf{\Phi}_l|_1 + \mu_2 \sum_{l=0}^{L-1} S(\mathbf{\Phi}_l) \right) ,$$

4.1. Vizuelizacija

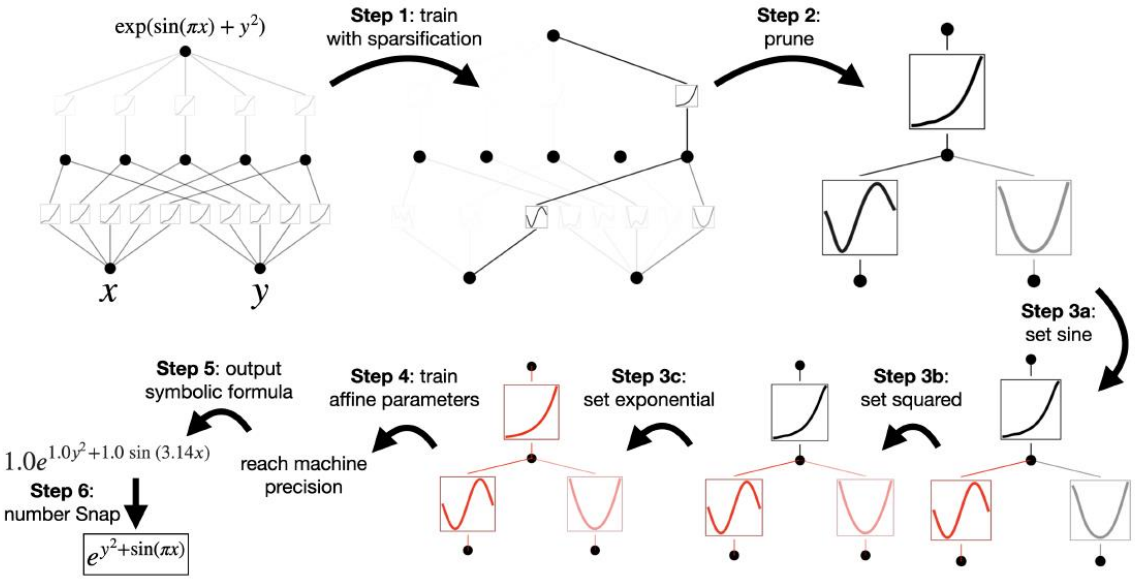
Zbog interpretabilnosti, KAN mreže imaju prikaz funkcija u čvorovima, ovo je takođe i jedan od razloga zašto je originalni autor smatrao da KAN mogu da budu dobra rešenja za pomoć u učenju.

4.2. Odsecanje

Odsecanje se obavlja nakon treniranja sa sparsifikacijom, za svaki čvor definišemo ulazni i izlazni score na osnovu koga radimo odsecanje neurona.

4.3. Simbolifikacija

KAN ima mogućnost da probamo da predstavimo funkciju u simboličkom obliku ako je to moguće (sin, cos, log).

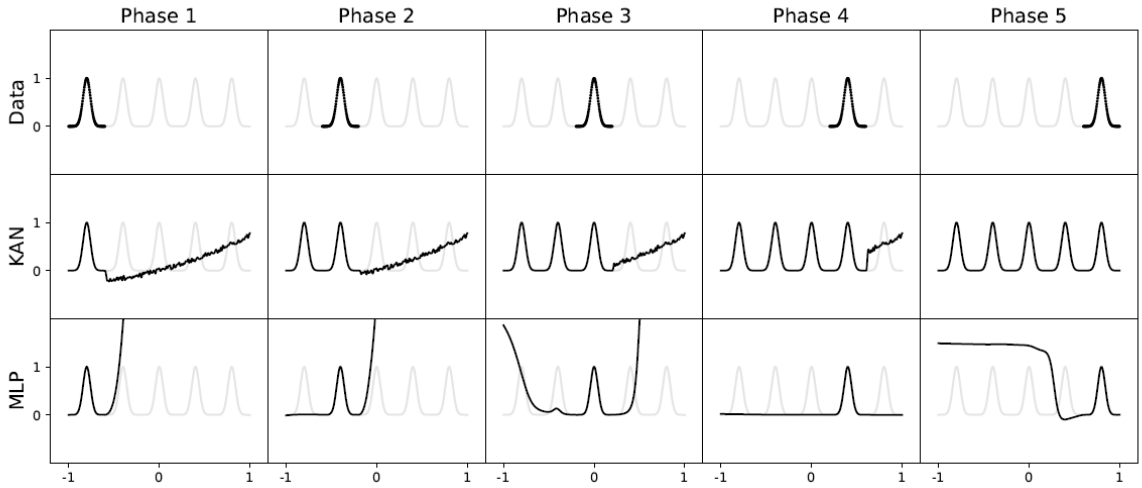


Slika 2. Simbolička regresija za KAN, ovaj primer pokazuje sve potrebne korake pri treniranju i simboličkom predstavljanju formule [1]

4.4. Kontinualno učenje bez katastrofalnog zaboravljanja

U slučaju da mrežu treniramo na jednom problem a zatim na drugom, mreža će ubrzo zaboraviti prvi problem, tj kako da ga izvrši. Zbog upotrebe splajnova KAN ima lokalnu plastičnost i može se izbeći katastrofalno zaboravljanje – promena ulaza će izmeniti samo

nekoliko koeficijenata splajnova, tako da će oni udaljeni biti netaknuti, tj. ako je splajn zapravo nekoliko krivih linija povezanih, samo će se jedna od njih promeniti a ne sve krive koje čine splajn.



Slika 3. KAN i kontinualno učenje, možemo videti da ne dolazi do katastrofalnog zaboravljanja [1]

5. Nedostatci

U radu gde je predstavljena ova mreža, problem je u tome što su svi skupovi podataka toy datasetovi, čak je i to velikodušan naziv za neke pošto su manji/jednostavniji od iris skupa podataka.

Takođe, autor navodi da je vreme za obučavanje bilo deset puta duže nego vreme za obučavanje MLP mreža. Ovde je autor napomenuo da je ovo samo problem optimizacije, što se ispostavilo da je tačno sa EfficientKAN implementacijom, ali zbog upotrebe bsplajnova (kao i u originalnoj implementaciji) i njihove lokalne plastičnosti mogu idalje da overfituju.

Ova mreža je veoma brzo postala veoma popularna zbog tvrdnji da je preciznija od MLP mreža, kao i da prevazilazi kletvu dimenzionalnosti [5] . Velika uzbuđenost za ovaj rad isto polazi iz činjenice da je Maks Tegmark jedan od koautora, kao i zbog toga što je rad lepo napisan, može služiti kao uputstvo za implementaciju.

Za samu tvrdnju o kletvi dimenzionalnosti je bitno napomenuti da se KAN može predstaviti preko MLP – što bi takođe značilo da i MLP može da prevaziđe ovaj problem [5] , pa me ova tvrdnja veoma optimistična, dobra strana kod kan mreža je manji broj

parametara koji bi možda pomogao prevazilaženju kletve dimenzionalnosti, ali zbog dužeg vremena obučavanja (uzevši u obzir i dodatne optimizacije) kao i druge praktične aspekte ovog problema, ova tvrdnja je neosnovana.

6. Upotreba

KAN mreže bi prvenstveno trebalo da imaju najveći uticaj u nauci, za šta su zamišljene, u teoriji čvorova i u fizici. U samom radu su predstavljeni dobri rezultati u teoriji čvorova, kao i ideja da će KAN mreže biti korišćene kao asistenti budućim fizičarima, tj. studentima doktorskih nauka, kao i generalno u oblastima fizike, simboličke regresije i kontinualnog učenja.

Za računarski vid postoje konvolucione Kolmogorov-Arnold mreže [3][4] , koje će biti tema sledeće sekcije.

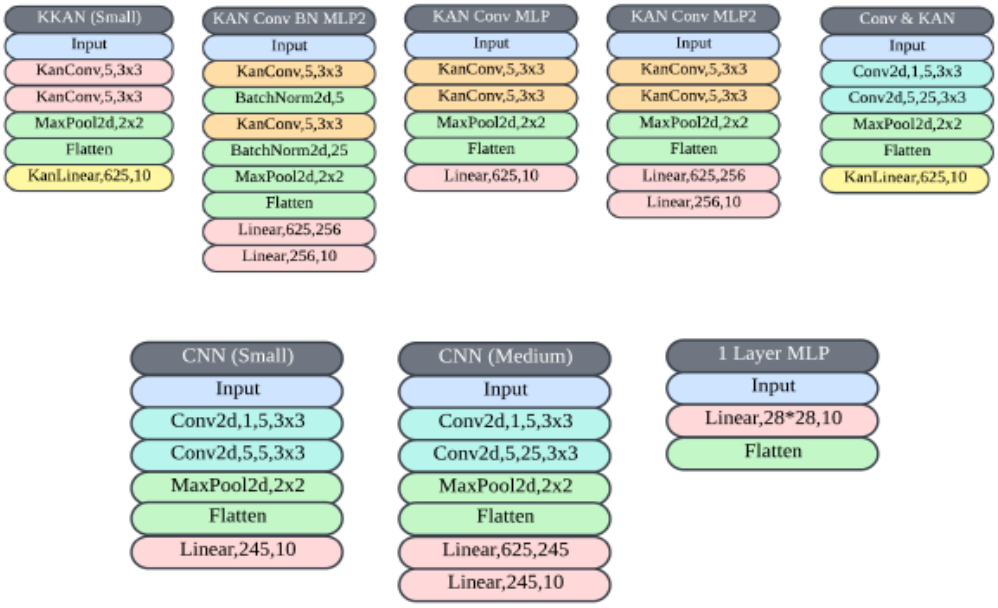
7. Konvolucione Kolmogorov-Arnold Mreže

Kod ovih mreža konvolucioni slojevi su zamenjeni KAN konvolucionim slojevima. Glavna ideja za ovu arhitekturu je smanjenje broja parametra zbog upotrebe B-splajnova. Sam kernel nije sačinjen od težina već od istih funkcija kao KAN, tj. silu i B-splajn, gde svaka od ovih funkcija ima svoju težinu.

$$\text{KAN Kernel} = \begin{bmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{bmatrix}$$

$$\text{Image} * \text{KAN Kernel} = \begin{bmatrix} \phi_{11}(a_{11}) + \phi_{12}(a_{12}) + \phi_{21}(a_{21}) + \phi_{22}(a_{22}) & \cdots & r_{1(p-1)} \\ \phi_{11}(a_{21}) + \phi_{12}(a_{22}) + \phi_{21}(a_{31}) + \phi_{22}(a_{32}) & \cdots & r_{2(p-1)} \\ \vdots & \ddots & \vdots \\ \phi_{11}(a_{m1}) + \phi_{12}(a_{m2}) + \phi_{21}(a_{(m+1)1}) + \phi_{22}(a_{(m+1)2}) & \cdots & r_{m(p-1)} \end{bmatrix}$$

Naredna slika predstavlja arhitekture mreža kao i same performanse nad MNIST i Fashion MNIST Skupovima podataka, možemo videti i da pored manjeg broja parametara kod KAN mreža, vreme za jednu epohu je duže, preciznost je veoma slična.



Model	Accuracy	Precision	Recall	F1 Score	#Params	Minutes per epoch
MNIST Dataset						
KKAN (Small)	98.90%	98.90%	98.89%	98.90%	94.875K	1.8119
Conv & KAN	98.75%	98.75%	98.75%	98.75%	95K	0.4551
KAN Conv & 1 Layer MLP	98.53%	98.54%	98.53%	98.53%	7.4K	1.7867
KAN Conv & 2 Layer MLP	98.58%	98.58%	98.58%	98.58%	164K	1.7779
KAN Conv BN & 2 Layer MLP	98.53%	98.54%	98.53%	98.53%	164K	1.7545
CNN (Medium)	99.12%	99.12%	99.12%	99.12%	157K	0.3110
CNN (Small)	97.59%	97.60%	97.59%	97.59%	2.7K	0.3056
1 Layer MLP	92.34%	92.22%	92.22%	92.22%	7.9K	0.2935

Model	Accuracy	Precision	Recall	F1 Score	#Params	Minutes per epoch
Fashion MNIST Dataset						
KKAN (Small)	89.69%	89.62%	89.69%	89.62%	94.875K	1.8399
Conv & KAN	88.84%	88.76%	88.84%	88.77%	94.95K	0.3347
KAN Conv & 1 Layer MLP	88.57%	88.45%	88.57%	88.47%	7.385K	1.7765
KAN Conv & 2 Layer MLP	89.45%	89.46%	89.45%	89.45%	163.951K	1.7168
KAN Conv BN & 2 Layer MLP	89.08%	89.06%	89.08%	89.01%	163.786K	1.6535
CNN (Medium)	90.14%	90.13%	90.14%	90.13%	157.03K	0.2931
CNN (Small)	87.15%	86.99%	87.15%	87.00%	2.74K	0.2896
1 Layer MLP	84.33%	84.21%	84.33%	84.21%	7.85K	0.2778

Slika 4. Arhitekture za testiranja MNIST I Fashion MNIST skupa podataka[3]

Za CIFAR10 dataset je iskorišćena arhitektura KanConv,5,3x3 - KanConv,5,3x3 - MaxPool2d,2x2 – Flatten – Linear,2700,10. Za obučavanje jedne epohe potrebno je oko 3 sati, rezultati nakon jedne epohe su dati na sledećoj slici.

```
model = CIFAR10KANModel().to(device)
optimizer_Convs_and_KAN = optim.AdamW(model.parameters(), lr=1e-3, weight_decay=1e-4)
scheduler_Convs_and_KAN = optim.lr_scheduler.ExponentialLR(optimizer_Convs_and_KAN, gamma=0.8)
criterion_Convs_and_KAN = nn.CrossEntropyLoss()

all_train_loss_Convs_and_KAN, all_test_loss_Convs_and_KAN, all_test_accuracy_Convs_and_KAN, all_test_precision_Convs_and_KAN, all_test_recall_Convs_and_
```

```
100%|██████████| 782/782 [3:10:02<00:00, 14.58s/it]
End of Epoch 1: Train Loss: 1.532792, Test Loss: 0.0207, Accuracy: 54.25%
```

Slika 5. CIFAR10 Convolutional KAN, jedna epoha

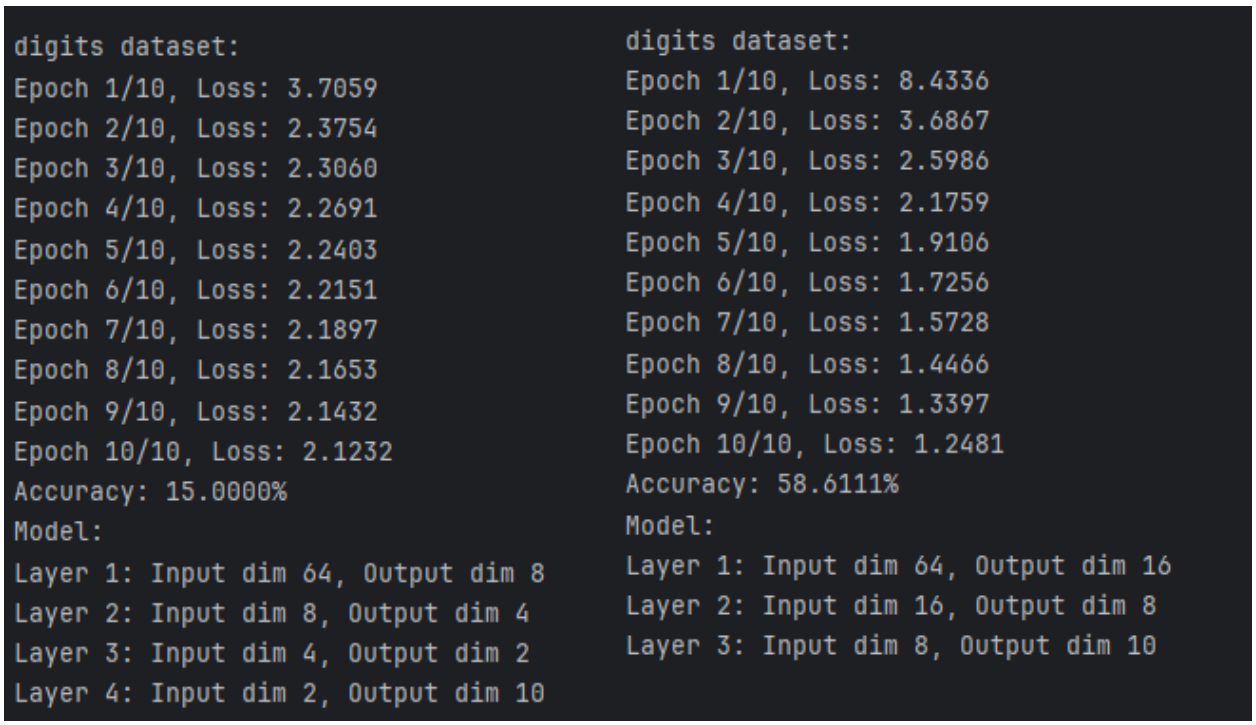
8. Proof of concept implementacija Kolmogorov-Arnold Mreže

Prateći projekat sadrži jednostavnu implementaciju koja koristi silu i B-spline i ima mogućnost za definisanje broja neurona po sloju. Za testiranje su iskorišćeni jednostavni datasetovi – iris, diabetes i digits. Za iris dataset sa mrežom dimenzija 4 (input), 16, 8, 4, 2(output) je moguće dobiti 100% accuracy, ali je najčešće 93.33%. Implementacija se nije pojavila dobro za regresiju, za diabetes dataset je mse približno 18000.

```
iris dataset:
Epoch 1/10, Loss: 1.3723
Epoch 2/10, Loss: 0.8235
Epoch 3/10, Loss: 0.7232
Epoch 4/10, Loss: 0.6763
Epoch 5/10, Loss: 0.6492
Epoch 6/10, Loss: 0.6228
Epoch 7/10, Loss: 0.5981
Epoch 8/10, Loss: 0.5633
Epoch 9/10, Loss: 0.5321
Epoch 10/10, Loss: 0.5001
Accuracy: 93.3333%
Model:
Layer 1: Input dim 4, Output dim 16
Layer 2: Input dim 16, Output dim 8
Layer 3: Input dim 8, Output dim 4
Layer 4: Input dim 4, Output dim 3
```

```
iris dataset:
Epoch 1/10, Loss: 2.2108
Epoch 2/10, Loss: 0.9529
Epoch 3/10, Loss: 0.6325
Epoch 4/10, Loss: 0.5619
Epoch 5/10, Loss: 0.3723
Epoch 6/10, Loss: 0.3030
Epoch 7/10, Loss: 0.2582
Epoch 8/10, Loss: 0.2276
Epoch 9/10, Loss: 0.2058
Epoch 10/10, Loss: 0.1897
Accuracy: 96.6667%
Model:
Layer 1: Input dim 4, Output dim 32
Layer 2: Input dim 32, Output dim 16
Layer 3: Input dim 16, Output dim 8
Layer 4: Input dim 8, Output dim 3
```

Slika 6. Rezultat za iris dataset sa mrežom dimenzija 4(input), 16, 8, 4, 3(output) i mrežom dimenzija 4(input), 32, 16, 8, 3(output)



Slika 7. Rezultat za digits dataset sa mrežom dimenzija 64(input), 8, 4, 2, 10(output) i mrežom dimenzija 64(input), 16, 8, 10(output)

9. Zaključak

Za sada je teško predvideti uticaj ove mreže zbog kontradiktornih rezultata, na malim skupovima podataka su se ove mreže dobro pokazale, ali ovi skupovi podataka nisu uopšte reprezentativni. Od trenutka pojavljivanja ove mreže je izašao veliki broj radova za i protiv ovu mrežu, kao i još jedan rad originalnog autora.

Najveći problem sa ovim mrežama je njihovo dugo treniranje, što se pokazalo i na CIFAR10 testiranju. Interpretabilnost ovih mreža može da bude veoma korisna, ali moguća je i vizualizacija CNN mreža – vizuelizacija KAN mreža je korisna u naučne svrhe, i sa većim domenskim znanjem kao i znanjem “osnovnih” funkcija (sin, cos, log…). Takođe, nismo primorani na B-spline pri korišćenju ovih mreža, moguća su i rešenja sa Furijeovim transformacijama [6].

KAN mreže trenutno ne treba posmatrati kao potencijalnu zamenu za MLP, već kao mreža koja može biti veoma korisna u naučne svrhe, kao i potencijalni korak ka novim arhitekturama koje mogu da zamene MLP u nekim problemima.

10. Literatura

[1] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, “KAN: Kolmogorov–Arnold Networks,” arXiv preprint arXiv:2404.19756v4, 2024.

[2] "Kolmogorov–Arnold representation theorem," *Wikipedia*, The Free Encyclopedia. Available: https://en.wikipedia.org/wiki/Kolmogorov%E2%80%93Arnold_representation_theorem. Accessed: Aug. 26, 2024.

[3] A. D. Bodner, A. S. Tepsich, J. N. Spolski, and S. Pourteau, “CONVOLUTIONAL KOLMOGOROV–ARNOLD NETWORKS,” *arXiv preprint arXiv:2406.13155v1* [cs.CV], June 21, 2024. Available: https://arxiv.org/abs/2406.13155

[4] Federico Girosi and Tomaso Poggio. Representation properties of networks: Kolmogorov’s theorem is irrelevant. Neural Computation, 1(4):465‑469, 1989.

[5] V. Dhiman, "KAN: Kolmogorov–Arnold Networks: A Review," Vikas Dhiman's Personal Website. Available: https://vikasdhiman.info/reviews/KAN_a_review.html. Accessed: Aug. 26, 2024.

[6] J. Xu, S. Yang, Z. Chen, W. Wang, E. C.-H. Ngai, J. Li, and X. Hu, "FourierKAN-GCF: Fourier Kolmogorov-Arnold Network - An Effective and Efficient Feature Transformation for Graph Collaborative Filtering," *arXiv preprint arXiv:2406.01034v2* [cs.IR], June 4, 2024. Available: https://arxiv.org/abs/2406.01034