HOLLÓSI JÁNOS[1]

Kapszula típusú mesterséges neurális hálózat architektúra kialakítása és hatékonysági vizsgálata az önvezető járművek környezetészlelés területén

Az elmúlt évek során az önvezető járművek környezet érzékelésének területén a konvolúciós neurális hálózatok bizonyultak az egyik leghatékonyabb eszköztárként. Ma már több elterjedt és magas hatásfokú hálózati architektúra létezik, melyek osztályozási és szegmentálási feladatokra egyaránt alkalmasak. Ugyanakkor fontos megjegyezni, hogy bizonyos körülmények között a neurális hálózatok hatékonysága akár nagy mértékben visszaeshet. Ide sorolható az a tény, hogy a neurális hálózatok nem képesek térben realizálni a megtanult mintákat. Ilyen módon előfordulhat, hogy egy objektum adott rálátás mellett magas megbízhatósággal felismerhető, de ha az objektum változtatja a helyzetét és így más betekintésből kell felismernie a hálózatnak, a hatékonyság akár lecsökkenhet. Ez különösen fontos az önvezető járművek területén, hiszen a jármű környezete dinamikusan változik, ahol a különféle környezeti résztvevőket, mint járművek, járókelők, más-más rálátás mellett kell detektálnia. Többek között az ilyen jellegű problémák kiküszöbölése érdekében alkották meg az úgynevezett kapszula hálózatok elméletét, ahol a hálózatok alapvető építőeleme nem a neuron, hanem az úgynevezett kapszula. A kapszulák neuronok csoportjából épülnek fel és az információt skalár értékek helyett vektorok formájában tárolják. A kutatásom elsődleges célja, hogy a kapszula hálózatok alkalmazhatóságának lehetőségét vizsgáljam az önvezető járművek területén felmerülő problémák elkerülése érdekében. Másodsorban a kapszula hálózatok hatékonysági vizsgálatával foglalkozom, ahol a kapszulák közötti útválasztási algoritmus hatékonysági elemzését végzem. A kutatás során három elterjedt neurális hálózat architektúra kialakítást vettem alapul, amiket kombináltam az általam kialakított kapszula hálózati elemmel. Megvizsgáltam, hogy a kapszula réteg alkalmazása milyen mértékben növeli a hálózatok hatékonyságát olyan adatkészletek esetén, ahol a fent leírt probléma jelen van. Ugyanezen hálózatok esetén vizsgáltam az útválasztási algoritmust, ahol különböző iterációs paraméterrel végeztem a hálózatok tanítását. Ennek eredményeként láthatóvá vált a paraméter pontos megválasztásának szükségszerűsége.

[1] SZE Multidiszciplináris Műszaki Tudományi Doktori Iskola, PhD hallgató. Témavezető: Dr. Ballagi Áron, tanszékvezető, egyetemi docens, SZE Gépészmérnöki, Informatikai és Villamosmérnöki Kar, Automatizálási Tanszék.

1. BEVEZETÉS

A kapszula hálózatok megalkotása Geoffrey Hinton és csapata[2] nevéhez köthető, aki több más, ma már elterjedt és alapvető elmélet kidolgozója a mesterséges intelligencia és neurális hálózatok területén. A kapszula hálózatokat a klasszikus neurális hálózatok alapján alkották, ugyanakkor igyekeztek olyan új módszert létrehozni, amely ellenállóbb a neurális hálózatoknál. A neurális hálózatok építőköve a neuron, a kapszula hálózatok alapvető egysége az úgynevezett kapszula. A fő különbség a neuron és a kapszula között, hogy a neuron skalár értékekkel dolgozik, a kapszula pedig tetszőleges dimenziójú vektorokat használ. A végrehajtható műveletek ennek megfelelően módosulnak a következők szerint.

Legyen *i* egy kapszula és *j* egy *i*-nél magasabb szintű kapszula. Ekkor a súlyozott bemenet a következő módon adható meg

$$\hat{u}_{j|i} = \boldsymbol{W}_{ij} u \qquad (1)$$

ahol $\boldsymbol{W}_{ij}$ a súlymátrix és u_i a bemeneti vektor az *i*-edik kapszula esetén.

Az úgynevezett összekapcsolási együttható számítása a softmax függvény segítségével történik a következők szerint:

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})} \qquad (2)$$

ahol b_{ij} annak a valószínűsége, hogy az *i*-edik kapszula a *j*-edik kapszula irányába továbbítja a kimenetét. Az összegzett bemenet a *j*-edik kapszula számára az alábbiak szerint adható meg:

$$s_j = \sum_i c_{ij} \hat{u}_{j|i} \qquad (3)$$

Kapszula hálózatok esetén az előrejelzés számszerűsítésére a kimeneti vektorok hosszát alkalmazzák. Ehhez az is szükséges, hogy a kapott vektorok hossza a [0,1] intervallumba kerüljön. Ehhez az elmélet kidolgozói bevezették az úgynevezett squash függvényt, amit a következő módon számíthatunk:

$$v_j = \frac{\|s_j\|^2}{1+\|s_j\|^2} \frac{s_j}{\|s_j\|} \qquad (4)$$

[2] Sabour, Sara – Frosst, Nicholas – Hinton, Geoffrey: Dynamic routing between capsules, in Guyon, Isabelle – Luxburg, Ulrike von – Bengio, Samy– Wallach, Hanna – Fergus, Rob – Vishwanathan, S. V. N. – Garnett, Roman (eds.): *Advances in Neural Information Processing Systems*, 2017., Hinton, Geoffrey – Sabour, Sara – Frosst, Nicholas: *Matrix capsules with em routing*, 2018, 6th International Conference on Learning Representations, Vancouver, Canada.

Az egymásra épülő kapszulák közötti információáramlását a súlyértékekkel lehet szabályozni. Ezeknek a számítása egy iteratív módon működő algoritmus segítségével történik. Ez az úgynevezett dinamikus útválasztás, ami a következő algoritmussal adható meg.

Metódus Routing $(\hat{u}_{j|i},\ r,\ l)$:
minden *l*-beli *i* kapszula és minden *(i+1)*-beli *j* kapszula esetén: $b_{ij} \leftarrow 0$
ciklus *r* iteráción keresztül

minden *l*-beli *i* kapszula esetén: $c_i \leftarrow \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})}$

minden *(l+1)*-beli *j* kapszula esetén: $s_j \leftarrow \sum_i c_{ij}\hat{u}_{j|i}$

minden *(l+1)*-beli *j* kapszula esetén: $v_j \leftarrow \frac{\|s_j\|^2}{1+\|s_j\|^2}\frac{s_j}{\|s_j\|}$

minden *l*-beli *i* kapszula és minden *(l+1)*-beli *j* kapszula esetén:

$$b_{ij} \leftarrow b_{ij} + \hat{u}_{j|i}v_j$$

ki: v_j

2. KIALAKÍTOTT HÁLÓZATOK

A kutatás során három alapjaiban eltérő hálózati architektúrát alakítottam ki, amiknek a felhasználásával vizsgáltam a kapszula hálózatokat. Létrehoztam egy kapszula hálózati egységet, amit mindhárom hálózati architektúrába integráltam. Így összehasonlítható, hogy a különféle architektúrák hatékonysága hogyan változik a kapszula réteg jelenlétével. Továbbá a kapszula réteg útválasztási algoritmusának paraméterezésével vizsgálható, hogy a hatékonyságot milyen módon befolyásolja az algoritmus működése.

2.1. Kapszula alapú hálózati egység

A kialakított kapszula egység két rétegből tevődik össze. Az első rétegben 256 darab konvolúciós szűrő található. Ezen a ponton a hálózat még neurális hálózatként működik, így bármilyen neurális hálózathoz csatolható. A réteg kimeneti oldalán azonban már a kapszula hálózatoknak megfelelő vektorok találhatóak. Ezt kapja bemenetként az egység következő komponense, ami egy 10 kapszulát tartalmazó réteg. A kapszulák száma azért 10, mert a mérések során olyan osztályozási feladatokat hajtok végre, ahol a bemeneti adatok 10 különböző osztályba sorolhatóak. Így minden osztálycímkéhez egy-egy kapszula tartozik. A kialakított egységben a kapszulák 16 dimenziós vektor formájában továbbítják az információt.

2.2. Szekvenciális neurális hálózat

Az implementált hálózati architektúrák közül ennek a felépítése a legegyszerűbb. A kialakított hálózat részben a VGG Net[3] hálózati architektúra alapján készült, de annak egy nagy mértékben leegyszerűsített megvalósítása. Ez a hálózati kialakítás 4 fő blokkra osztható, ahol minden blokkon belül szekvenciálisan 1-1 konvolúciós réteg, aktivációs réteg, normalizációs réteg és végül egy úgynevezett dropout réteg figyelhető meg.

2.3. Reziduális neurális hálózat

A hálózat úgynevezett reziduális blokkokból épül fel, amit He és társai[4] vezettek be. A reziduális hálózat olyan blokkokból épül fel, ahol az adott blokk kimenetét a hálózatban található rétegek által végzett számítások eredményéből és a réteg eredeti bemenetének összegzéséből számítjuk. Tehát legyen az ℓ-edik hálózati réteg bemenete x_ℓ és a hálózatban található rétegek által végzett számítást jelölje $\mathcal{F}_\ell(x)$. Ekkor azt mondhatjuk, hogy a reziduális hálózati réteg kimenetén legyen:

$$x_\ell = \mathcal{F}_\ell(x_{\ell-1}) + x_{\ell-1} \quad (5)$$

Minden reziduális blokkban egymás után háromszor 1-1 normalizációs, aktivációs és konvolúciós réteg található. A megvalósított hálózati architektúra 18 reziduális blokkot tartalmaz.

2.4. Sűrűn kapcsolt neurális hálózat

A sűrűn kapcsolt architektúrát Huang és társai[5] dolgozták ki, melynek lényege, hogy a sűrűn kapcsolt rétegen belül található blokkok bemenetként minden olyan blokk kimenetét megkapják, melyek azt megelőzik. Tehát azt mondhatjuk, hogy

$$x_\ell = \mathcal{F}_l([x_0, x_1, x_2, \dots, x_{\ell-1}]) \quad (6)$$

ahol $[x_0, x_1, x_2, \dots, x_{\ell-1}]$ jelentse az 0, 1, 2..., ℓ- 1 rétegek kimenetének konkatenációját. A hálózat 13 darab sűrűn kapcsolt réteget tartalmaz egymás után kapcsolva, ahol minden réteg 3 konvolúciós blokkot tartalmaz aktivációs réteggel. A sűrűn kapcsolt blokkok között 1-1 konvolúciós és összevonó réteg található.

---

[3] Simonyan, Karen – Zisserman, Andrew: *Very Deep Convolutional Networks for Large-Scale Image Recognition*, 2015, 3rd International Conference on Learning Representations, San Diego, USA.
[4] He, Kaiming – Zhang, Xiangyu – Ren, Shaoqing – Sun, Jian: *Deep residual learning for image recognition*, 2016, IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA.
[5] Huang, Gao – Liu, Zhuang – Maaten, Laurens – Weinberger, Kilian Q.: *Densely Connected Convolutional Networks*, 2017, IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, USA.

3. ALKALMAZOTT ADATKÉSZLETEK

A kialakított hálózatok vizsgálatához két elterjedt adatkészlet került alkalmazásra, az MNIST[6] és a CIFAR-10.[7] Az MNIST adatkészlet kézzel írott számjegyekről tartalmaz képeket. Az adatkészlet 28 képpont szélességű és 28 képpont magasságú, szürkeárnyalatos képeket tartalmaz. Összesen 60000 tanító és 10000 teszt mintát foglal magában. A CIFAR-10 adatkészlet 10 egymástól nagy mértékben különböző osztályba sorolható képeket tartalmaz. Minden kép színes, szélességük és magasságuk egyaránt 32 képpont. Az adatkészlet 50000 tanító és 10000 teszt mintából tevődik össze.

A kutatás célja, hogy olyan körülmények esetén vizsgáljam a hálózatok hatékonyságát, amikor az alkalmazott adatkészletek kialakítása kihasználja a neurális hálózatok azon hiányosságát, hogy a látottakat nem képesek térben realizálni. Ezt olyan módon értem el, hogy az alkalmazott adatkészleteken különféle, véletlenszerű geometriai módosításokat hajtottam végre, mint forgatás és döntés.

4. EREDMÉNYEK

A kialakított hálózatok hatékonyságát két szempont szerint vizsgáltam. Az első vizsgálat esetén a cél annak igazolása volt, hogy a deformált adatkészletek esetén a hálózatok hatékonysága növelhető kapszula típusú réteg integrálásával. Ehhez a három hálózatot két módon valósítottam meg. Mindegyik hálózatot elkészítettem hagyományos neurális hálózatként, valamint kapszula réteggel kiegészített módon is. Az így kialakított 6 hálózatot vizsgáltam a módosított MNIST és CIFAR-10 adatkészletekkel. A tanítási köteg 128 mintából állt és a tanítást 250 iteráción keresztül végeztem. A tanítást csökkentett méretű adatkészlettel is elvégeztem, így az is jól látszik a mérési eredményekből, hogy kevés rendelkezésre álló adat esetén a kapszula hálózat mennyivel képes hatékonyabban működni a klasszikus neurális hálózathoz képest. Az 1. táblázat összegzi a mérési eredményeket, ahol a hálózatok felismerési hatékonysága olvasható le. A tanítást mindkét adatkészlet esetén két adatmérettel végeztem el. Az első esetben a tanítási minták száma 50000, a teszt minták száma 10000. A második esetben a tanítási minták száma 20000, a teszt minták száma 4000. A mérési eredményekből jól látható, hogy az esetek többségében a kutatás során megalkotott kapszula egység integrálása jótékonyan hatott a hálózatok hatékonyságára. A legnagyobb javulást a sűrűn kapcsolt hálózati kialakítás esetén sikerült elérni a deformált adatkészletek esetén.

---

[6] LeCun, Yann – Cortes, Corinna: *handwritten digit database*. http://yann.lecun.com/exdb/mnist, 2018. november 12.

[7] Krizhevsky, Alex – Hinton Geoffrey: *Learning multiple layers of features from tiny images*. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf, 2018. november 12.

1. táblázat: Kialakított hálózatok hatékonysága

	MNIST		CIFAR-10	
	(50*k*,10*k*)	(20*k*,4*k*)	(50*k*,10*k*)	(20*k*,4*k*)
Szekvenciális model	99,18%	98,10%	**90,72%**	**85,32%**
Szekvenciális model kapszula integrációval	**99,19%**	**98,25%**	89,67%	85,02%
Reziduális model	98,94%	97,92%	79,32%	69,15%
Reziduális model kapszula integrációval	**99,18%**	**98,02%**	**79,69%**	**71,23%**
Sűrűn kapcsolt model	89,50%	58,75%	45,27%	36,88%
Sűrűn kapcsolt model kapszula integrációval	**97,74%**	**95,35%**	**64,02%**	**59,98%**

A kapszula hálózatok paraméterezhetőségét a kapszulák közti útválasztási algoritmus tekintetében végeztem. Két paraméter módosításával vizsgáltam a hatékonyság változását a kapszula egységgel integrált három hálózat esetén. Ehhez az útválasztási algoritmus iterációs paraméterét, illetve squash függvény módosításának hatását vizsgáltam, ahol a módosított squash függvény a következő szerint alakult:

$$v_j = \frac{\|s_j\|^2}{\gamma + \|s_j\|^2} \frac{s_j}{\|s_j\|} \qquad (7)$$

ahol γ tetszőlegesen megválasztott paraméter. Hinton és társai megállapítása szerint az útválasztási algoritmust 3 iteráción keresztül célszerű végrehajtani. A mérési eredményeket a 2. és 3. táblázatok foglalják össze, ahol jól látszik, hogy az iterációk számán, valamint a squash függvény paraméterezésén célszerű eltöprengeni. Az optimális paramétereket nagy mértékben befolyásolja az alkalmazott hálózati architektúra, de a tanítási adatkészlet jellege is.

2. táblázat: Mérési eredmények különböző paraméterek mellett az MNIST adatkészleten

		Szekvenciális model			Reziduális model			Sűrűn kapcsolt model		
		r								
		1	3	5	1	3	5	1	3	5
γ	0,1	88,2%	84,8%	86,6%	96,8%	97,4%	96,4%	93,8%	94,0%	94,8%
	0,5	86,2%	87,8%	88,4%	98,0%	98,1%	97,6%	94,4%	95,0%	93,8%
	1,0	85,8%	84,6%	85,6%	97,6%	97,4%	97,6%	95,2%	91,2%	94,4%
	1,5	87,8%	84,0%	88,0%	98,0%	97,6%	97,4%	94,0%	95,6%	94,4%
	2,0	86,0%	86,6%	85,0%	97,4%	98,0%	98,2%	93,6%	95,8%	95,0%

3. táblázat: Mérési eredmények különböző paraméterek mellett a CIFAR-10 adatkészleten

		Szekvenciális model			Reziduális model			Sűrűn kapcsolt model		
		r								
		1	3	5	1	3	5	1	3	5
γ	0,1	40,0%	55,2%	42,4%	60,0%	55,0%	57,4%	45,8%	45,2%	47,6%
	0,5	54,8%	43,0%	56,4%	62,6%	62,0%	61,0%	47,2%	46,2%	45,0%
	1,0	59,6%	56,4%	56,4%	65,0%	66,4%	64,4%	48,4%	47,8%	47,6%
	1,5	39,0%	40,6%	40,8%	62,6%	63,4%	65,0%	44,2%	45,8%	45,6%
	2,0	53,4%	55,0%	40,0%	64,6%	64,6%	64,4%	45,0%	48,4%	46,2%

5. KONKLÚZIÓ

A kutatás során a kapszula hálózatok gyakorlati alkalmazhatóságát vizsgáltam. Kialakítottam három elterjedt és nagy hatásfokkal bíró hálózati architektúra, ahol rávilágítottam, hogy deformált adatkészletek esetén ezeknek a hálózatoknak a hatékonysága nagyságrenddel csökkenhet. A hatékonyság javítható kapszula réteg alkalmazásával, amit a mérési eredmények alátámasztanak. Megvizsgáltam a kapszula hálózatok útválasztási algoritmusának paraméterezhetőségét, ahol sikerült megmutatni, hogy az optimális paraméter nagy mértékben függ az alkalmazott architektúra és adatkészlet jellegétől.

FELHASZNÁLT IRODALOM

- He, Kaiming – Zhang, Xiangyu – Ren, Shaoqing – Sun, Jian: *Deep residual learning for image recognition*, 2016, IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA.
- Hinton, Geoffrey – Sabour, Sara – Frosst, Nicholas: *Matrix capsules with em routing*, 2018, 6th International Conference on Learning Representations, Vancouver, Canada.
- Huang, Gao – Liu, Zhuang – Maaten, Laurens – Weinberger, Kilian Q.: *Densely Connected Convolutional Networks*, 2017, IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, USA.
- Krizhevsky, Alex – Hinton, Geoffrey: *Learning multiple layers of features from tiny images*. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf, 2018. november 12.
- LeCun, Yann – Cortes, Corinna: *handwritten digit database*. http://yann.lecun.com/exdb/mnist, 2018. november 12.
- Sabour, Sara – Frosst, Nicholas – Hinton, Geoffrey: Dynamic routing between capsules in Guyon, Isabelle – Luxburg , Ulrike von – Bengio, Samy – Wallach, Hanna – Fergus, Rob – Vishwanathan, S. V. N. – Garnett, Roman (eds.): *Advances in Neural Information Processing Systems*, 2017.
- Simonyan, Karen – Zisserman, Andrew: *Very Deep Convolutional Networks for Large-Scale Image Recognition*, 2015, 3rd International Conference on Learning Representations, San Diego, USA.