

Hasonló a bonyodalom a részgráf esetében is. Egy gráf részgráfját úgy kapjuk meg, hogy a gráfból éleket, esetleg szögpontokat és a velük szomszédos éleket töröljük. A román szakirodalom különbséget tesz a kétféle törlés között, első esetben graf parțial, második esetben subgraf a megfelelő kifejezés. Magyar és angol szövegben, ha csak éleket hagyunk el, akkor azt külön kihangsúlyozzuk.

Ha egy részgráf a gráf minden pontját tartalmazza és egyben fa, akkor faváznak nevezzük. Románul ez arbore parțial vagy arbore de acoperire (sajnos, az előbbi terjedt el jobban). Angolul ez spanning tree.

Egyéb fogalmak:

<i>magyar</i>	<i>román</i>	<i>angol</i>
él	muchie	edge
fa	arbore	tree
fok (szögponté)	grad	degree
be-fok	grad interior	indegree
ki-fok	grad exterior	outdegree
folyam	flux	flux
irányított él	arc	arc
kiegészítő gráf	graf complementar	complement (of a graph)
komponens	component	component
közlekedési hálózat	rețea de transport	network
kritikus út	drum critic	critical path
liget	pădure	forest
mélységi keresés	căutare în adâncime	depth-first search
mohó algoritmus	algoritmul greedy	greedy algorithm
összefüggő gráf	graf conex	connected graph
páros gráf	graf bipartit	bipartite graph
párosítás	cuplaj	matching
síkgráf	graf planar	planar graph
szélességi keresés	căuarea în lășime	breadth-first search
szögpont, csúcs	vârf, nod	vertex, node
teljes gráf	graf complet	complete graph

Kása Zoltán

Tud-e olvasni a számítógép?

1. A mesterséges intelligencia fogalma

A számítógépes világ kezdetén a számítógépeknek az volt a szerepük, hogy megszabadítsák az embert a fáradtságos számításoktól, és a bonyodalmas számításokat gyorsabban elvégezzék. Idővel azonban belátták, hogy a számítógépek sokkal többre is képesek, mint bonyolult számítások elvégzése. Mindez azonban az embertől függött. Ha a programozó kellőképpen programozta a számítógépet, az „készségesen” ellátta a rábízott feladatokat, de semmi többet. Azon kezdtek filozófálgatni a kutatók, hogy nem lehetne-e megtanítani a számítógépet „gondolkodni”, és így születik meg a *mesterséges intelligencia* fogalma a század közepén. Maga a *mesterséges intelligencia* (artificial intelligence, AI) elnevezés McCarthytól származik, aki az 1956-os darthmouth-i konferencián használja, amely az első ilyen témájú tudományos összejövetel volt. Russel négyféle rendszer létrehozását fogalmazza meg a mesterséges intelligencia (MI) céljaként:

- az emberhez hasonlóan gondolkodó rendszerek,
- az emberhez hasonlóan cselekvő rendszerek,
- racionálisan gondolkodó rendszerek,
- racionálisan cselekvő rendszerek,

A fentebb említett célok két nagy csoportba sorolhatók, az első csoportba tartoznak a gondolkodással és következtetéssel összefüggő folyamatok, míg a másik csoportban lévő célok az emberi racionalizmust célozzák. Míg az első csoportba tartozó rendszereket emberi teljesítménnyel mérik, a másik csoport esetén egy olyan absztrakt fogalomhoz kell viszonyítanunk, amit racionalizmusnak nevezünk.

1. értelmezés. *Egy rendszert racionálisnak nevezünk, ha mindig a helyes utat választja, és ennek függvényében dönt.*

Mint ahogy az kiderül a fenti értelmezésből is, nagyon nehéz egy pontos megfogalmazást adni arra, hogy mit takar a *mesterséges intelligencia* fogalma. Az alábbiakban megadunk néhány értelmezést, és az olvasóra bízunk, hogy döntse el, hogy mit jelent számára a *mesterséges intelligencia*.

„Az MI izgalmas erőfeszítés a számítógépek gondolkodóvá tételére, értelemmel bíró gépek létrehozására a szó szoros értelmében.” (Haugenland, 1985)

„Az MI az emberi gondolkodáshoz asszociált tevékenységek, mint a döntéshozatal, problémamegoldás, tanulás automatizálása, vizsgálata.” (Bellman, 1978)

„Az MI a mentális képességek tanulmányozása számítógépes modellek segítségével.” (Charmick, 1989)

„Az MI a számítástudomány azon ága, mely az intelligens viselkedés automatizálásával foglalkozik” (Luger, 1993)

„Az MI annak tanulmányozása, hogyan lehet számítógéppel olyan dolgokat tenni, melyeket jelenleg az emberek jobban tudnak” (Rick, 1991)

Ahogy az az előbb felsorolt „értelmezésekből” is kiderül, az MI olyan tudomány, mely megpróbálja az embert helyettesíteni, de nem mint létező egyént, hanem a gondolkodásán, cselekvésén keresztül.

Rögtön felvetődik egy probléma. Hogy egy ember intelligens-e vagy sem, az társai által megítélhető. Mi történik egy számítógép esetében? Mikor állíthatjuk, hogy egy program intelligens? Erre a kérdésre Alan Turing próbált gyakorlati definícióval szolgálni.

2. értelmezés (Turing-teszt). *Egy program intelligensnek mondható, ha rendelkezik a következő funkciókkal:*

- *természetes nyelvmegértés*, azaz hogy tudjon kommunikálni valamilyen emberi nyelven,
- *megfelelő tudásreprezentáció*, azaz hogy a beszélgetés előtt vagy közben képes legyen az információkat tárolni,
- *automatikus következtetés*, azaz hogy a tárolt információkat később fel tudja használni, vagyis következtetéseket legyen képes levonni,
- *gépi tanulás*, ami azt jelenti, hogy képes alkalmazkodni az újabb körülményekhez, és képes mintákat észrevenni, amelyek segítségével majd képes a továbbiakban extrapolálni.

A Turing-tesztet később kiterjesztették, és így jött létre a kiterjesztett Turing-teszt, amely magába foglalja a következő funkciókat is:

- *számítógépes látás*, a tárgyak észlelése érdekében,
- *robotikai tulajdonságok*, a tárgyak mozgatása érdekében.

Egy fontos tényezőt azonban elfelejtettek bevenni a Turing-teszt funkcióiba, éspedig azt, hogy a fentebb említett funkciókat a rendszer valós időben teljesítse. Sok esetben sikerül elegendet tenni a kiterjesztett Turing-tesztnak, de nem valós időben.

2. Az optikai karakterfelismerés (OCR) fogalma

Mint tudjuk, a számítógép számára nem léteznek betűk, csak képek, amelyeken fekete foltok vannak. Ha el akarunk olvasatni egy szöveget a számítógéppel, akkor a papírról át kell konvertálnunk a szöveget egy olyan formára, amit a számítógép is ismer, vagyis egy adott képformátumba (például BMP, PCX, GIF), majd következik az „olvasás” és ezzel egyidőben a „tanulás” is.

Az optikai karakterfelismerés géppel/ kézzel írt vagy nyomtatott szöveg „számítógépre vitelét” (lásd digitalizálás) teszi lehetővé a kép beolvasásával és megértésével és a számítógép számára érthető kódokká alakításával, hogy majd azokat más programok képesek legyenek felhasználni. (lásd szövegszerkesztők)

Ezt a folyamatot az MI *optikai karakterfelismerés* (optical character recognition, OCR) néven tartja számon.

Az OCR rendszer a következő részfeladatokból áll:

- digitalizálás
- preproceszálás (képporrekción)
- szegmentálás
- tulajdonságok kinyerése
- osztályozás

2.1 Digitalizálás

Ahogy azt fentebb is láttuk, a számítógép nem ismeri a betűket, csak annyit tud, hogy a képen, amely tartalmazza a szöveget, vannak fehér illetve fekete pontok, amelyeket pixeleknek nevezünk.

Magát a szöveget egy lapon kapjuk, amelyet a számítógép számára is érthetővé kell tennünk. Erre szolgálnak a digitalizáló eszközök. Ilyen digitalizáló eszközök a *kézi scanner*, a *síkgyas scanner* illetve a *dob scanner*. Mindhárom eszköz feladata, hogy adott térrészeket vagy 3D (háromdimenziós) objektumokat átalakítsanak. A TV-kamera már térben elhelyezkedő objektumok kezelésére alkalmas.

2.2 Preprocesszálás

Mivel a képeket valamilyen digitalizáló eszköz segítségével nyerjük, általában, nagy a valószínűsége annak, hogy az illető kép bizonyos változásokat szenved. Általában ezek a változások negatívak, azaz az illető kép bizonyos torzításokat szenved (lásd nagyítás, kicsinyítés), vagy bizonyos képpontok elvesznek, és így a kép veszít a minőségéből. Mivel az OCR-rendszerek kezdeti fázisban pixel szinten kezelik a képet, fontos, hogy minél pontosabb („tisztább”) képeket dolgozzunk fel. Ezért a képjavító (képporrekción) algoritmusok igencsak fontos szerepet kapnak, mivel segítségükkel bizonyos lényeges képi információkat kiemelnek (pl. kontúr, élesség). Ezen algoritmusok nem növelik a képi adathalmazt, hanem csak karakterisztikájának dinamikáját emelik ki, ezáltal jobban észrevehetővé, felismerhetővé tesznek egyes képrészleteket. Igen nehéz feladatnak bizonyul kiválasztani, hogy egy rendszer esetén milyen javító algoritmusokat alkalmazzunk, mivel ezen algoritmusok száma nagy és mindegyik algoritmusnak fontos szerepe van.

A képjavító algoritmusokat az alábbi osztályokba sorolhatjuk:

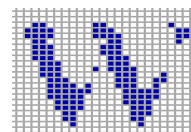
- pixelirányú műveletek: kontraszt megnövelése, a zaj megszüntetése, a kép hisztogramokkal való modellezése,
- térbeli műveletek: zajsűrítés, mediánsűrítés,
- transzformációs műveletek: lineáris sűrítés, homomorfikus sűrítés,
- pseudoszínezés

2.3 Szegmentálás

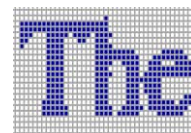
A szegmentálás alapproblémája: az adott kép céltudatos részekre való bontása, vagyis az azonos jelentésű képrészek egy objektumként való tárgyalása. Leegyszerűsítve a problémát, két alapvető dolgot kell megoldani: . (lásd Ábra 1)

- szét kell választani a különböző objektumokhoz tartozó pixeleket (vágás),
- egybe kell sorolni azokat a pixeleket amelyek egy objektum részei.

Mint tudjuk, a betűk összeragadhatnak, és az igazi nehéz feladat



Ábra 1.



Ábra 2.

az, hogy éppen hol kell szétvágni. Ez az OCR egyik alapvető problémája, és ma még sincs teljes mértékben megoldva. (lásd Ábra 2)

2.4 Tulajdonságok kiemelése

Mint tudjuk, minden betűnek bizonyos geometriai tulajdonságai vannak, mint a *görbületek, ívek, lyukak*, illetve *bal és jobb profil* és a *k-ad rendű momentum* értéke, ha a kontúr, mint diszkrét pontokban értelmezett kétváltozós függvényt értjük. Ezen tulajdonságok igen jól meghatározzák a betűket, és ezáltal osztályozni lehet őket.

Fontos azonban, hogy pontosan definiáljuk, hogy mi az, hogy görbület, konvex és konkáv ív, szögpont, lyuk, bal profil, jobb profil, és minél jobban megválasszuk ezen tulajdonságokat, mivel ha túl keveset választunk belőlük, akkor fennáll a veszélye annak, hogy nem tudunk osztályozni adathiány miatt, illetve ha ezen tulajdonságok száma nagy, akkor sok számítást igényelnek, és nagy valószínűséggel az osztályok részben fedni fogják egymást, ami szintén helytelen felismeréshez (osztályozáshoz) vezethet.

2.5 Osztályozás

A képosztályozás feladata hogy a képpontokat, kisszámú (összetartozó) képpont együttesét, illetve szegmentált alakzatokat tulajdonságai alapján felismerje, illetve a megadott „osztályok” valamelyikébe besorolja, s ezzel létrehozza az adott kép magasabb szintű leírását.

Az osztályozás lehet:

- *statikus (döntésméleti) osztályozás*, amely valószínűségszámítási és matematikai statisztikai módszerekkel dolgozik. Az így előállított jellemzők az objektum illetve a textúraelemek között fennálló síkbeli (térbeli) összefüggésekkel nem foglalkoznak.
- *szintaktikus (strukturális) módszerek*, amelyek az objektumok, illetve a textúraelemek közötti síkbeli (térbeli) összefüggéseken alapulnak.

A statikus osztályozás kategóriájába tartoznak a többretegű neuronhálók, amelyek segítségével a rendszer képes tanulni és osztályozni is (lásd *back propagation algoritmus*) illetve a *Nestor Learning System*, amelyet két amerikai tudós szabadalmaztatott.

Egy szintaktikus osztályozási mód, a Borland Delphi 3.0 környezetben lévő ActiveX kontroll, amelynek segítségével, ha már előzőleg felismertük a szöveg nagy részét, szótár segítségével lehetőség nyílik az eddig ismeretlen betűk felismerésére.

3. Hogyan tanul meg a számítógép olvasni?

Mint azt az osztályozásnál láttuk, fontos szerepet játszik a tanítás, mert enélkül nem lehetne szó hatékony felismerésről. A programnak tárolnia az eddig jól osztályozott betűket, és ezeket fel kell tudnia használni a továbbiakban. Problémát jelenthet az, hogy meddig tanítsuk a rendszert, illetve, hogy mekkora legyen ez a tudáshalmaz. Ha betartjuk a fentebb tárgyalt pontokat, akkor hatékony OCR-rendszer megvalósítása válhat lehetővé, és így megtaníthatjuk a számítógépet „olvasni”.

Hatékony OCR rendszert fejlesztett ki a Recognita (www.recognita.hu), illetve az OmniPage (www.caere.com), amelyeknek demováltozatát le is tölthetjük a fenti honlapokról.

Irodalom:

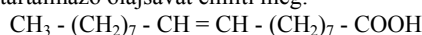
- 1] Berke József, Hegedűs Gy. Csaba, Kelemen Dezső, Szabó József, *Digitális képfeldolgozás és alkalmazásai*, Pannon Agrártudományi Egyetem, Georgikon Mezőgazdaságtudományi Kar, Szaktanácsadási, Továbbképzési és Informatikai Központ, Keszthely, PICTRON Kft., Budapest, 1996
- 2] David Vernon, *Neural Networks and Computer Vision*, Department of Computer Science, Trinity College, Dublin, Ireland, TDC 1991
- 3] D. Dumitrescu, *Modele conexiunite în inteligența artificială*, (note de curs), UBB Cluj-Napoca, 1995
- 4] Raúl Rojas, *Neural Networks. A Systematic Introduction*, Springer Verlag Berlin Heidelberg, 1996

- 5] Marco Cantú, *Delphi 3 mesteri szinten* (II kötet), Kiskapu Kiadó, Budapest, 1998
 6] Futó Iván, *Mesterséges intelligencia*, Aula Könyvkiadó, Budapest, 1999.
 7] Earl Gose, Richard Johnsonbaugh, Steve Jost, *Pattern Recognition and Image Analysis*, Prentice Hall PTR, 1996

Vajda Szilárd
 egyetemi hallgató

A telítetlen zsírsavakról

A középiskolás kémiaanyag az élettani jelentőségű anyagok fejezetében nagyon röviden, csak az egy kettőskötést tartalmazó olajsavat említi meg:



Ezt a magasabb rendű állatok (emlősök) szervezete sztearinsavból elő tudja állítani. Magasabb telítetlenségű fokú zsírsavakat (két, három, négy kettőskötést tartalmazók) nem képes szintetizálni, mivel az az enzim, amely katalizálja a folyamatot csak 2 hidrogén atomot tud leszakítani.

A több kettőskötést tartalmazó zsírsavaknál bebizonyosodott, hogy a magasabb rendű állatok (emlősök) számára kis mennyiségben nélkülözhetetlenek, akárcsak a vitaminok. Hiányuk hiánybetegséget okoz. Ezért F-vitaminnak nevezték el azt a három zsírsav-együttest (linolsav, linolénsav, arachidonsav), melyeket esszenciális zsírsavaknak is nevezünk, s melyeket csak táplálék formájában tud felvenni a szervezet.

1. táblázat

Növényi olajokban előfordulnak:

képlet	$\text{CH}_3 - (\text{CH}_2)_7 - \text{CH} = \text{CH} - (\text{CH}_2)_7 - \text{COOH}$
rövidített jel.	$\text{C}_{18:1}$
szisztematikus név	Oktadecénsav
triviális neve	Olajsav
képlet	$\text{CH}_3 - (\text{CH}_2)_4 - \text{CH} = \text{CH} - \text{CH}_2 - \text{CH} = \text{CH} - (\text{CH}_2)_7 - \text{COOH}$
rövidített jel.	$\text{C}_{18:2}$
szisztematikus név	6,9-Oktadeka-diénsav
triviális neve	Linolénsav
képlet	$\text{CH}_3 - \text{CH}_2 - \text{CH} = \text{CH} - \text{CH}_2 - \text{CH} = \text{CH} - \text{CH}_2 - \text{CH} = \text{CH} - (\text{CH}_2)_7 - \text{COOH}$
rövidített jel.	$\text{C}_{18:3}$
szisztematikus név	9,12,15-Oktadeka-triénsav
triviális neve	α -linolénsav
képlet	$\text{CH}_3 - (\text{CH}_2)_4 - \text{CH} = \text{CH} - \text{CH}_2 - \text{CH} = \text{CH} - \text{CH}_2 - \text{CH} = \text{CH} - (\text{CH}_2)_4 - \text{COOH}$
rövidített jel.	$\text{C}_{18:3}$
szisztematikus név	6,9,12-Oktadeka-triénsav
triviális neve	γ -linolénsav
képlet	$\text{CH}_3 - (\text{CH}_2)_4 - (\text{CH} = \text{CH} - \text{CH}_2)_4 - (\text{CH}_2)_2 - \text{COOH}$
rövidített jel.	$\text{C}_{20:4}$
szisztematikus név	5,8,11,14-Ejkoza-tetraénsav
triviális neve	Arachidonsav

Ezek a zsírsavak trigliceridek formájában vannak jelen a táplálék zsiradékában. Ezek a molekulák nem képesek felszívódni a bélfalon át. A szervezetben a lipáz enzim glicerint és zsírsavakra bontja. A majdnem csak egyharmad méretű zsírsavak már felszívódnak, s a véráramban levő monogliceridek hatására újra észtereződnek.