

Miért ellenkezik a mélytanulás a számítási alapelvekkel?

Why deep learning is against computing principles?

VÉGH János

MTA Doktora, egyetemi tanár

[Miskolci Egyetem, Vegh.Janos@gmail.com](mailto:Miskolci.Egyetem,Vegh.Janos@gmail.com)

Abstract

Computers have always aimed to achieve even faster execution, literally at any cost. Computer processors operating on the basis of computational theory have long since reached the maximum speed physically possible, so-called accelerators (such as graphics accelerator cards) partially operate in violation of the theoretical foundations, with very low efficiency. Deep learning has also been in the spotlight in connection with unsustainable energy consumption, artificial intelligence (and Nobel Prizes).

Keywords: deep learning; accelerating computing; efficiency of computers; principles of computing; sustainable development;

Kivonat – Kivonat cím

A számítógépek mindig a még gyorsabb végrehajtás elérését vették célba, szó szerint bármi áron. A számításelmélet alapján működő számítógép processzorok már rég elérték a fizikailag lehetséges maximális sebességet, az ún. gyorsítók (például a grafikus gyorsító kártyák) részben már az elméleti alapokat megsértve működnek, nagyon alacsony hatásfokkal. A mélyrétegű tanulás a fenntarthatatlan energia fogyasztással, a mesterséges intelligenciával (és a Nobel-díjakkal) kapcsolatban is reflektorfénybe került.

Kulcsszavak: mélytanulás; számítás gyorsítás; a számítás hatékonysága; számítási alapelvek; fenntartható fejlődés;

1. BEVEZETÉS

Az elmúlt években a mesterséges intelligencia (MI) harmadik hulláma nemcsak életünk számos vonatkozását forgatta fel [1]. Amint a második hullámot lezáró jelentés [2] már összefoglalta, ilyen néven egységes technológiáról nem beszélhetünk. Különböző fejlettségi szinteken levő technológiákat bujtatnak el egy ilyen közös név alá, nagyon sok esetben óriási gazdasági érdekektől vezérelve. Már maga a név is félrevezető (R. P. Feynman már a '70-es években, a második hullám előtt javasolta a „fejlett algoritmusok” kifejezés használatát). A két évtizeddel később bekövetkezett harmadik hullám technológiai háttére sokkal magasabb szintre emelte a megtevesztés minőségét (az ún. Turing-teszt lényegében az ember becsapására irányul), órási adatmennyiséget tud háttér ismeretként használni, több nagyságrenddel nagyobb és olcsóbb számítógépeket használhat, csatasorba állította a hálózati keresésre kifejlesztett technológiákat, óriási mértékben fejlődött agyműködés tanulmányokat tud hasznosítani, és nem utolsósorban rengeteget fejlődött az adatok kezelése, amit főként a marketing-célú adatkombinálás igénye táplál. A mögöttük álló óriáscégek tőkeereje és profitigénye, meg az ebből következő gátlástalansága és felelőtlensége óriási.

A mesterséges intelligencia, néhány hasznos hatása és sokféle kártékonyasága mellett, súlyos kérdéseket vetett fel a tudományos teljesítmény fogalmával kapcsolatosan is (mennyire releváns a lényegében csupán drága számítógépes erőforrások használatán alapuló eredmények elérése), újra definiálta a tudományos integritásra vonatkozó elképzeléseinket (évtizedekkel korábbi tudományos alkotások sajátként feltüntetése [3]). Felveti a kérdést, hogy ha nyilvánvaló plágiumért Nobel-díjat lehet kapni, akkor miért nem lehet ugyanazzal diplomát szerezni. Vagy fordítva. A mélyrétegű tanulás pedig azt kérdőjelezi meg, hogy a biológiailag nyilvánvalóan nem releváns módszer összefér-e a számítástudomány alapelveivel; elterjedt használata pedig a fenntartható fejlődéssel.

2. TANULÁS ÉS INTELLIGENCIA

A tanulás általános biológiai megfogalmazása[4]: „használjuk úgy a jelenleg rendelkezésre álló információt, hogy azzal javítsuk a rendszer jövőbeli működését”. Az intelligencia végső soron a tanulásra való képességgel áll szoros kapcsolatban. Lényeges momentum azonban, hogy nem csak egyszerű mennyiségi tanulásként, inkrementálisan, a korábban már látott viselkedés minták valamelyikét (esetleg azok valamilyen kombinációját) használjuk, hanem egy *új és váratlan helyzetre is jól reagáljunk* (nem-inkrementális, ugrás-szerű tanulás). A[5] meghatározás szerint *a kizárólag a korábban látott viselkedés mintákon alapuló rendszer nem intelligens: nem szerez új képességet, hanem változatos módokon próbálkozik*. Ennek alapján meg lehet különböztetni betanulást (azaz viselkedés minták elsajátítását) ami esetleg több megtanult viselkedés minta valamilyen módon való kombinálását is jelentheti, de a viselkedés csak a már látott mintákra korlátozódik; és megtanulást, ami az addigi képességekhez hozzáadja az új helyzetben szükséges viselkedés képességét. A „megsejtés” az a nehezen megfogalmazható *minőségi többlet* tevékenység, amelyet aztán algoritmizálható módon be kell bizonyítani.

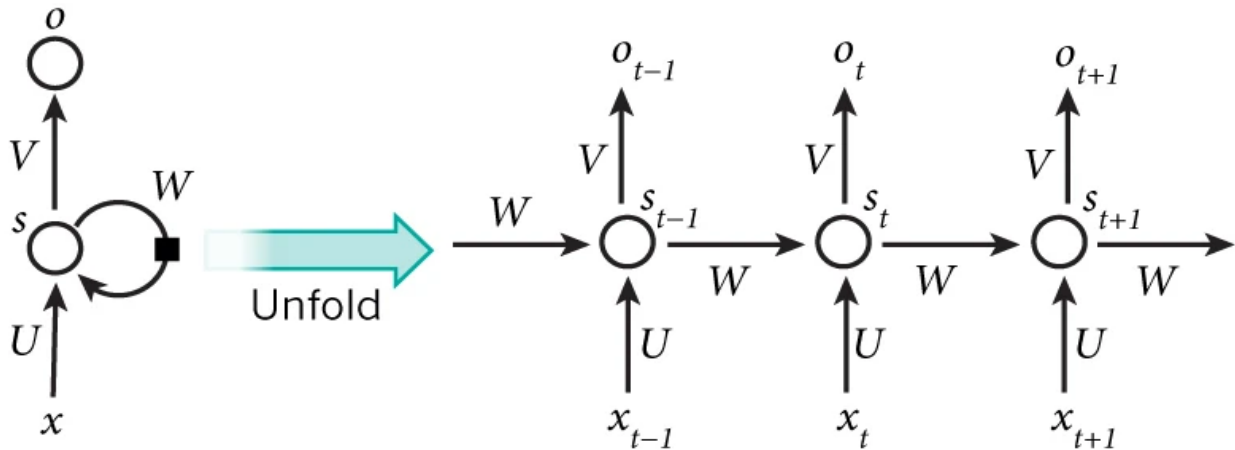
A gépi tanulás lényegében ezt a többletet igyekszik előállítani, nem véve tudomást arról, hogy alapvető működésbeli eltérések vannak az agy és a számítógép program között. Többek között, az egyik adat-vezérelt, analóg, párhuzamos, a másik utasítás-vezérelt, digitális, szekvenciális, az egyik információ ábrázolása időben és térben elosztott, analóg időtartam, a másik koncentrált, digitális feszültség amplitudó. Az agy hibatűrő és javító képességeit, az asszociációt, rehabilitációt, tanulást maga a „konstrukció” tudja, a számítógép esetében pedig matematikai módszereket igyekszünk kitalálni arra, hogy a gyakran félig sem értett biológiai működés eredményéhez legalább felületesen hasonlókat produkáljunk. Az eredmény felemás. Az időnként látszólag meglepően jó eredményeket produkáló csevegő robotok készséggel el is ismerik, hogy az emberi intelligencia és a mesterséges intelligencia alapvetően különböznek; sőt, bevezetik a „gépi intelligencia” fogalmát, ami a matematikai módszerek hatását azok számítógépes megvalósításának esetlegességeivel kombinálja.

A természetes és gépi intelligencia és tanulás tehát alapvetően különböznek, lényegében csak a neveik azonosak. A biológiai agy által megoldott feladatokat nem igazán lehet algoritmizálni, éppen a két „hardver” eltérő „számítási” módja miatt. A számítógéppel megvalósítandó, nehezen algoritmizálható feladatok megoldására születtek meg a biológiai neuron hálózatok mintájára a mesterséges neurális hálózatok, amelyeknek lényegi ötlete, hogy hátha a mesterséges neurális hálózatok is képesek önmagukat ráhangolni a feladatokra, és a számítógép programozás nélkül is képessé válik a feladatok megoldására. Ehhez az a matematikai modell ad reményt, hogy a biológiai neuronok sok bemenetű összeadó kapuk, amelyek a bemeneteiket súlyozottan adják össze és a súlyok egy tanításnak nevezett folyamat során módosíthatók. A biológiai rendszerrel való felületes hasonlóság alapján a neuronokat rétegekbe szervezik. Éppen az eltérő működés miatt azonban a tanítási folyamat elképesztően kis hatékonysággal, rendkívül lassan megy végbe, és az eredmény megvalósulása bizonytalan. Amint [6] bemutatja, ilyen módon legfeljebb pár tucat „neuron”-t lehet a számítási elveknek megfelelően működtetni, az újabb MI behemótoknak viszont akár 100 milliárd „neuron”-ja is lehet.

A neuron rétegek közötti kommunikáció átlagos idejének aránya az egyes neuronok működési idejéhez viszonyítva sokkal közelebb áll a biológiai arányhoz, mint az egyes neuronok kommunikációja esetén. A mesterséges neuronok esetén azonban az időzítést a műszaki megvalósítás (főként a közösen használt busz igénybeviteléhez szükséges arbitráció) határozza meg, a biológiai megvalósítás esetén pedig éppen az időzítések tartalmazzák a tanuláshoz szükséges információt.

Az időzítés döntő szerepet játszik a tanulásban. A biológiában ismert, hogy neurális bemenetek súlya dinamikusan változik. A neuronba már beérkezett információ folyamatosan avul, azaz az állapotváltozásról tudósító spike kibocsátási ideje előtt közvetlenül beérkezett spike súlya a legnagyobb (Hebb megfigyelés). Továbbá, a befogadási időablakon kívül érkezett spike-okat a neuron elhanyagolja. Ilyen módon a neuron maga választja ki, hogy melyik neuronoktól fogad egyáltalán bemenő információt és azt milyen súllyal veszi figyelembe. A matematikai modell az időt figyelmen kívül hagyja, ezért teljesen eltérő (biológiaiilag nem releváns) módon próbálja megoldani a tanulást. Az említetteken túlmenően, a matematikai neuron valamennyi bemenetét használja és valamennyi bemenet súlyozott összegéből számítja ki az eredményt. Ami azt is jelenti, hogy az eredmény másféle módon függ a bemenetektől: valamennyi bemenet használatával készül egy eredmény és a súlyvektor javítása (jellemzően gradiens számítással, a természeteshez képest fordított irányú terjedéssel). Az idő információ figyelmen kívül hagyása gyakorlatilag azt jelenti, hogy a neuron a bemenetek

súlyát egyforma mértékben változtatja a „jó” és a „rossz” információt szolgáltató bemenetek esetén, ezért a tanulás igen lassú folyamat. A fordított irányú terjedés biológiailag éppúgy nem releváns, mint annak számítási módja: a biológiai neuronokban a súlyok az időtől függenek, a jel terjedés pedig egyirányú. A fordított irányú terjedéssel megvalósított tanulás jellegzetes matematikai konstrukció: végtelen ciklust indít (l. 1. ábra), amit csak erőszakosan lehet leállítani, és ezért eredménye határozatlan. Csak azért működik valamennyire, mert a technikai kivitel véges számítási ideje korlátozza az ismétlés gyakoriságát.



1. ábra: A fordított irányú terjedéssel megvalósított tanulás [7], (számításelméleti szempontból) soha véget nem érő folyamat.

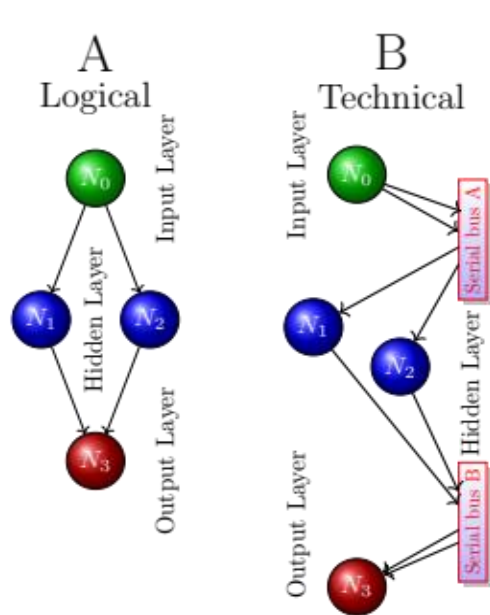
3. A GRAFIKUS GYORSÍTÓ KÁRTYÁK

A grafikus gyorsító kártya (GPU) ugyanazt a műveletet sok adaton végzi el (például, ugyanúgy transzformálja a képernyő valamennyi pontját vagy végez el 'multiply and add' műveletet egy neuron valamennyi bemenetével; látszólag ideális neurális számítások céljára). Egyszer végzi el az utasítás elővételét és dekódolását, aztán egy elegendően széles adatúton sok adattal tud egyidejűleg dolgozni. Megsérti azonban azt a számítási alapelvet, hogy az utasítás végrehajtás megkezdésekor valamennyi bemenő paraméternek rendelkezésre kell állni, és az eredményt csak a művelet befejezése után szabad elszállítani. A tanulás alapvető nehézsége, hogy a betanítás során nem-algoritmikus módon be kell állítani a súlyokat, amiket aztán a célfeladat végrehajtásakor a rendszer használ.

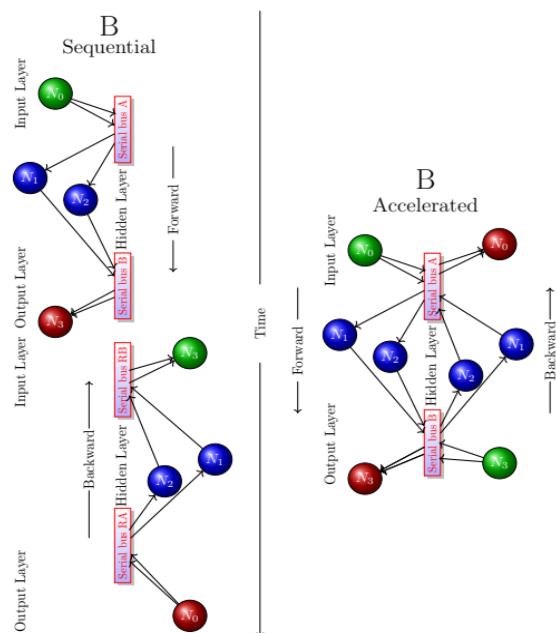
A GPU (és hasonló eszközök) esetén az adatok érkezése/feldolgozása/elszállítása fokozatosan és részben egyidejűleg történik. A művelet lehet akár egyetlen órajel hosszúságú is, amennyiben már valamennyi argumentum rendelkezésre áll. Ekkor azonban az adatátviteli idő igen hosszú (akár argumentumként egy órajel is szükséges lehet). Az egység tárolóit egyesével kell írni és olvasni, még ha az valamilyen rejtett módon történik is. Használhatjuk azonban az egységet olyan módon is, hogy a beérkezett adatok feldolgozását azonnal megkezdjük: a GPU bonyolult stalling/interleaving mechanizmusa biztosítja, hogy valamikor minden adat feldolgozásra kerül. A feldolgozás időablakát ez esetben úgy értelmezzük, hogy az az első GPU számítás kezdetétől az utolsó számítás befejezéséig terjed. Az időablak szélessége futási időben dől el, de tervezéskor egy rögzített értéket kell használni. Ha az érték túl nagy, az jelentősen csökkenti a számítás hatásfokát, ha pedig az aktuálisan szükséges idő hosszabb, mint a tervezett (nagyobb a busz terhelése), akkor a számítások egy része nem fejeződik be a határidőre. Ez a veszély az adatok be- és kivitelére egyaránt fennáll.

Működés közben a GPU egyidejűleg olvassa és írja memóriáját, és azokkal attól függetlenül dolgozik, hogy az a helyes értéket tartalmazza. Ha a busz nem szállít időben, akkor „idegen” adatokkal dolgozik. Ez a hatás nem vehető észre video alkalmazás esetén: egy esetleges hiba a pixelek egymilliomod részén néhány milliszekundum időtartamra látható. Számítási célú alkalmazás esetén (ilyen a mélytanulás is) az eredmények (a szinaptikus súlyok) egy ideig tartósan megmaradnak. Sok GPU adatai függenek egymástól és keverednek egyetlen buszon, az arbitráció miatt kölcsönösen késleltetve egymást. Ezt a hatást elkerülendő, vagy a hiányzó érték nélkül kell folytatni (remélve, hogy a hiányzó érték nem sokkal tér el a helyette használt memória tartalomtól; lényegében egy ismeretlen értékkel kell számolni az eredményt), vagy meg kell várni az összes neurális bemenetet (szinkron számítás) vagy újból el kell végezni a számítást a közben megérkező értékkel.

(ami aztán hasonló hatást vált ki a mélyebb rétegekben). Az utóbbi esetek a szokásos „pipelined” architektúrák esetén a komoly késleltetést okoznak és teljesítmény veszteséget okoznak.



2. ábra: Neuron rétegek közötti adattovábbítás logikai és technikai módja



3. ábra: A fordított irányú szinaptikus súly adjusztálás módszere normál és gyorsított átvitel

4. A MÉLYTANULÁS ELVE

Együttműködő objektum rendszerek (például mesterséges neuronok hálózata vagy biológiai rendszerek szimulációja) esetén, a párhuzamos és a párhuzamosított szekvenciális működést össze kell hangolni. Az idő hagyományos kezelésének klasszikus (és szokásos) módszere annak feltételezésén alapszik, hogy a szimulált és a számítási idő egymással arányosak. A számítást megadott hosszúságú időszakokban (rács-idő) végzik, és a szakasz végén a számítógépek egymásnak kölcsönösen elküldik eredményeiket vagy pedig a számítást és az eredmények elküldését/feldolgozását folyamatosan végzik, a továbbítást, feldolgozást és időzítést a számítógépes rendszer – más célra készült – erőforrásaira bízva. A számítógép nagyságának növekedésével nyilvánvalóvá vált, hogy a számítógépes rendszer növekedésével ez az arányosság egyre kevésbé teljesül; ennek kiküszöbölésére irányult az „időbélyegzés” használata. Azaz, a küldési módszer lényegében változatlan, de a küldő rendszer az üzenetbe beírja az esemény bekövetkezésének óraidejét. Ez lényegében áttolja a felelősséget az eseményeket feldolgozó egységre, amelyik ugyan tudja keletkezési idő szerinti sorrendben feldolgozni az addig beérkezett eseményeket, de nem oldja meg a még be nem érkezett események feldolgozásának problémáját. Mivel a problémát alapvetően az üzenetek nagy száma okozza, jelentősen javult a helyzet, amikor a biológiai neuronokat szimuláló mesterséges neuronok működését egyetlen üzenet elküldésére korlátozták (Spiking Neural Networks). Ami változtatás hatására az említett problémák csak jelentősen nagyobb számú objektum esetén, de ugyanúgy jelentkeznek. Ennek ára, hogy a neuron állapotváltozójának információ tartalma elvész; a neuron által közvetített idő információ pedig elveszíti valóság tartalmát: az idő túlnyomórészt a technikai működéstől és csak kevésbé a szimulált biológiai működéstől függ. A biológiai neuron a tüzelési idők közötti különbség kb. 200-ad részének megfelelő pontosságú időzítésre képes; továbbá a logikai és az időbélyegben szereplő óraidő közötti kapcsolat esetleges. Az óraidő a biológiai információ átvitelére egyáltalán nem alkalmas; más célú felhasználásának lehetőségét egyedileg, műszaki paraméterektől függően, kell vizsgálni.

A mesterséges neurális hálózatokat rétegekbe szervezik, amelyek egymásnak adatokat állítanak elő. A rétegek természetesen csak akkor tudnak a helyes operandusokkal dolgozni, amikor azokat az előző réteg már előállította. A rétegeknek és az egyes neuronoknak természetesen mindig van valamilyen bemenete, csak az nem feltétlenül az „igazi”. A 2. ábra mutatja, hogy a logikai folyamat (a bal oldalon), amit a matematikai leírás tényleges átviteli útnak tekint, ténylegesen a jobb oldalon mutatott módon, a buszon keresztül valósul meg. Ilyen módon az elvileg párhuzamos (egyidejű) állapotváltozások sohasem egyszerre következnek be (az N_x gömbök függőleges pozíciója az állapotváltozás eltérő időpontjára utal). A szerializált működés miatt a

neuronok bemeneteinek súlyai számítás közben változnak. A 3. ábra mutatja a visszacsatolás terjedésének gyorsított módját. A gyorsított működés során a visszacsatolás hatása már azelőtt eléri a neuronok súlyait, mielőtt valamelyik egység azt kiszámíthatta volna. Ha a bemenet hasonló az előzőhöz, akkor nem jelentős hatása a hiba, csak jelentősen lassítja a tanulást: a használt régi érték kevéssé tér el az újtól. A „katasztrofális felejtés” egyik fő oka, hogy ha a rendszer „újat tanul”, és a két érték jelentősen eltér, akkor gyakorlatilag *kezdőérték nélküli változót* használ: a megtanult régi súly elfelejtődik.

5. A MÉLYRÉTEGŰ TANULÁS HATÁSFOKA

Első hallásra sokkolónak tűnik, hogy „a számítógép energia hatékonyságát tíz nagyságrenddel kellene javítani, hogy az igazi nagy számítási feladatokat meg tudjuk oldani”[8]. Ezen belül a mélyrétegű számítás hatásfokát egyszerűen megbecsülhetjük. Ha egy biológiai hálózat jellemző működési idejét 1 msec nagyságúnak vesszük és valamilyen aktivitást 10 secundum alatt tanul meg, a két idő aránya 10^4 . Ha az elektronikus számítás jellemző idejét 1 nanoszekundum értékűnek vesszük, és egy aktivitást két hét alatt tanul meg, ugyanez az arány 10^{15} (a gigantikus új »nagy nyelvi modell« [9] 118 nap betanulási időt említ). Becslések szerint a processzorok megvalósítási hatásfoka[10] 10^{-3} , az úgynevezett »software stack« hatásfoka [11] szintén 10^{-3} , így a mélyrétegű tanulás hatásfoka 10^{-6} - 10^{-4} között van. A számítási alapelvek figyelmen kívül hagyása katasztrofális hatásfok vesztéshez vezet.

6. ÖSSZEFOGLALÁS

A számítógépek története arról szól, hogy hogyan lehet működésüket gyorsabbá, működtetésüket olcsóbbá tenni. Az elméletileg helyes működés megvalósítása a számítógép működését lassúvá teszi. A különféle alapötletű gyorsítók csupán speciális esetekre használhatók, általános célú használatuk helytelen eredményt szolgáltat. A mesterséges intelligenciában használt mélyrétegű tanulási módszer (illetve a szükséges többrétegű neurális hálózatok megvalósítása vektor és tenzor számítási egységekkel) rendkívüli mértékben rontja a számító rendszer energia hatékonyságát; eredménye pedig a klasszikus számításelméleti értelemben helytelen (bizonytalan, közelítő érték a pontos érték helyett), csupán a mesterséges intelligencia számára igényelt valószínűségi értelemben fogadható el. A helyes eredmény valószínűsége a körülményektől függ. A mélytanulási módszerek terjedésével fenntarthatatlan módon romlik a számítások energia hatásfoka és nő a számítások végzéséhez szükséges energia felhasználása.

IRODALMI HIVATKOZÁSOK

- [1] Katonai Nemzetbiztonsági Szolgálat: A mesterséges intelligencia és egyéb felforgató technológiák hatásainak átfogó vizsgálata. Budapest, 2023. https://unipub.lib.uni-corvinus.hu/8690/1/MI_2023_konyv.pdf
- [2] Sir Lighthill James: Artificial Intelligence: A General Survey, https://www.chilton-computing.org.uk/inf/literature/reports/lighthill_report/p001.htm
- [3] Schmidhuber J: <https://people.idsia.ch/~juergen/physics-nobel-2024-plagiarism.html>
- [4] Sterling P, Laughlin S: Principles of Neural Design. The MIT Press, (2017) ISBN 978-0-262-53468-0
- [5] Augusto E and Gambino : Can NMDA Spikes Dictate Computations of Local Networks and Behavior? Front. Mol. Neurosci. 12/238(2019) <https://dx.doi.org/10.3389/fnmol.2019.00238>
- [6] Végh J.: Which scaling rule applies to Artificial Neural Networks. Neural Computing and Applications 32/1(2021) <https://dx.doi.org/10.1007/s00521-021-06456-y>
- [7] LeCun Y., Bengio, Y. Hinton, G: Deep learning. Nature 521/7553(2015) 436-444. <https://dx.doi.org/10.1038/nature14539>
- [8] Conklin, A. A., Kumar, S.: Solving the big computing problems in the twenty-first century. (2023) <https://www.nature.com/articles/s41928-023-00985-1.epdf>
- [9] Luccioni A. S., Viguier, S., Ligozat A-N: Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model. J. Machine Learning Research 24(2023)1-15. <https://www.jmlr.org/papers/volume24/23-0069/23-0069.pdf>
- [10] Hameed, R. et al: Understanding Sources of Inefficiency in General-purpose Chips, Proceedings of the 37th Annual International Symposium on Computer Architecture, Saint-Malo, France (2010) 37-47. 978-1-4503-0053-7 <http://doi.acm.org/10.1145/1815961.1815968>
- [11] Ousterhout John K.: Why Aren't Operating Systems Getting Faster As Fast As Hardware? USENIX Summer Conference (1990) <http://www.stanford.edu/~ouster/cgi-bin/papers/osfaster.pdf>